

**ΑΝΩΤΑΤΟ ΤΕΧΝΟΛΟΓΙΚΟ ΕΚΠΑΙΔΕΥΤΙΚΟ ΙΔΡΥΜΑ
ΜΕΣΟΛΟΓΓΙΟΥ**

ΣΧΟΛΗ ΔΙΟΙΚΗΣΗΣ & ΟΙΚΟΝΟΜΙΑΣ

**ΤΜΗΜΑ ΕΦΑΡΜΟΓΩΝ ΤΗΣ ΠΛΗΡΟΦΟΡΙΚΗΣ ΣΤΗ
ΔΙΟΙΚΗΣΗ & ΤΗΝ ΟΙΚΟΝΟΜΙΑ**

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

**ΜΟΝΤΕΛΟΠΟΙΗΣΗ ΕΠΙΧΕΙΡΗΣΕΩΝ
(ΠΑΡΟΥΣΙΑΣΗ ΤΩΝ ΜΕΘΟΔΩΝ UML ΚΑΙ IDEF)**



**ΕΙΣ ΗΓΗΤΗΣ: ΑΛΕΞΑΝΔΡΗΣ ΓΕΩΡΓΙΟΣ,
ΣΠΟΥΔΑΣΤΗΣ: ΧΑΛΔΟΥΠΗΣ ΓΕΩΡΓΙΟΣ
ΕΞΑΜΗΝΟ: ΠΤΥΧΙΟ 3
ΑΜ: 7745**

ΜΕΣΟΛΟΓΓΙ – ΜΑΙΟΣ 2005



ΕΙΣΑΓΩΓΗ

Το αντικείμενο αυτής της πτυχιακής εργασίας είναι να μελετήσει τις δυνατότητες που παρέχουν δυο βασικές μέθοδοι σχεδιασμού μοντέλων επιχειρήσεων, η UML (Unified Modeling Language) και η IDEF (Integrated DEFinition). Στην εργασία παρουσιάζονται οι βασικές αρχές μοντελοποίησης εφαρμογών πληροφορικής με τις δύο αυτές μεθόδους καθώς και οι δυνατότητες επέκτασης των μεθόδων και σε άλλους τομείς μεταξύ των οποίων είναι και η μοντελοποίηση επιχειρήσεων. Στο τέλος της εργασίας γίνεται προσπάθεια σύγκρισης των δύο μεθόδων

Βέβαια μπορεί να θεωρηθεί από κάποιους «άνιση» η σύγκριση καθώς η IDEF έχει τριάντα χρόνια εξέλιξης ενώ η UML αναπτύχθηκε προς τα μέσα της δεκαετίας του 90. Η εργασία άπαντα σε όλα τα ερωτήματα και τις απορίες για το αν είναι πράγματι «άνιση» η σύγκριση καθώς παρουσιάζονται τα θετικά και τα αρνητικά της κάθε μεθόδου στα πέντε κεφαλαία που ακολουθούν.

Τέλος θα ήθελα να ευχαριστήσω όλους όσους βοήθησαν για τη πραγματοποίηση αυτής της εργασίας και ειδικά τον εισηγητή της εργασίας, Γεώργιο Αλεξανδρή για τη πολύτιμη βοήθεια που μου πρόσφερε.

ΠΕΡΙΕΧΟΜΕΝΑ

ΚΕΦΑΛΑΙΟ 1 - ΜΕΘΟΔΟΛΟΓΙΕΣ ΑΝΑΠΤΥΞΗΣ ΕΦΑΡΜΟΓΩΝ

1.1 Η ΔΟΜΗΜΕΝΗ ΠΡΟΣΕΓΓΙΣΗ.....	5
1.2 Η ΑΝΤΙΚΕΙΜΕΝΟΣΤΡΑΦΗΣ ΠΡΟΣΕΓΓΙΣΗ.....	8
1.3 ΕΝΘΥΛΑΚΩΣΗ.....	9
1.4 ΑΝΤΙΚΕΙΜΕΝΑ.....	10
1.5 ΟΡΟΛΟΓΙΑ.....	11
1.6 Η ΠΡΟΣΑΝΑΤΟΛΙΣΜΕΝΗ ΠΡΟΣ ΤΟ ΑΝΤΙΚΕΙΜΕΝΟ ΣΤΡΑΤΗΓΙΚΗ...11	

ΚΕΦΑΛΑΙΟ 2 - ΠΑΡΟΥΣΙΑΣΗ ΤΗΣ ΜΕΘΟΔΟΥ ΜΟΝΤΕΛΟΠΟΙΗΣΗΣ UML

ΤΙ ΕΙΝΑΙ Η UML.....	13
ΜΙΑ ΕΠΙΣΚΟΠΗΣΗ ΤΗΣ UML.....	13
2.1 ΔΙΑΓΡΑΜΜΑΤΑ ΠΕΡΙΠΤΩΣΕΩΝ ΧΡΗΣΗΣ(USE CASE DIAGRAMS).....	16
2.1.1 ΔΡΑΣΤΕΣ (ACTORS).....	16
2.1.2 Ο ΣΚΟΠΟΣ ΤΩΝ ΠΕΡΙΠΤΩΣΕΩΝ ΧΡΗΣΗΣ.....	18
2.1.3 ΠΕΡΙΓΡΑΦΕΣ ΠΕΡΙΠΤΩΣΕΩΝ ΧΡΗΣΗΣ.....	20
2.1.4 ΕΥΡΕΣΗ ΤΩΝ ΠΕΡΙΠΤΩΣΕΩΝ ΧΡΗΣΗΣ.....	20
2.2 ΤΟ ΔΙΑΓΡΑΜΜΑ ΚΑΤΗΓΟΡΙΑΣ(CLASS DIAGRAMS).....	22
2.2.1 ΕΝΝΟΙΟΛΟΓΙΚΗ ΔΙΑΜΟΡΦΩΣΗ.....	22
2.2.2 ΤΟ ΕΝΝΟΙΟΛΟΓΙΚΟ ΠΡΟΤΥΠΟ ΣΤΗ UML.....	25
2.2.3 ΕΥΡΕΣΗ ΤΩΝ ΙΔΙΟΤΗΤΩΝ.....	26
2.2.4 ΟΔΗΓΙΕΣ ΓΙΑ ΤΗΝ ΕΥΡΕΣΗ ΤΩΝ ΙΔΙΟΤΗΤΩΝ.....	27
2.3 ΔΙΑΓΡΑΜΜΑΤΑ ΣΥΝΕΡΓΑΣΙΑΣ(COLLABORATION DIAGRAMS).....	31
2.3.1 ΣΥΝΤΑΞΗ ΣΥΝΕΡΓΑΣΙΑΣ:ΤΑ ΒΑΣΙΚΑ.....	31
2.3.2 ΕΠΑΝΑΛΗΨΗ (LOOPING).....	33
2.3.3 ΔΗΜΙΟΥΡΓΙΑ ΤΩΝ ΝΕΩΝ ΑΝΤΙΚΕΙΜΕΝΩΝ.....	34
2.3.4 ΠΑΡΑΔΕΙΓΜΑ.....	35
2.3.5 ΜΕΡΙΚΕΣ ΟΔΗΓΙΕΣ ΓΙΑ ΤΑ ΔΙΑΓΡΑΜΜΑΤΑ ΣΥΝΕΡΓΑΣΙΑΣ.....	41
2.4 ΔΙΑΓΡΑΜΜΑΤΑ ΑΚΟΛΟΥΘΙΑΣ(SEQUENCE DIAGRAMS).....	42
2.5 ΛΙΓΑ ΛΟΓΙΑ ΓΙΑ ΤΑ ΥΠΟΛΟΙΠΑ ΔΙΑΓΡΑΜΜΑΤΑ ΤΗΣ ΜΕΘΟΔΟΥ UML.....	45

ΚΕΦΑΛΑΙΟ 3 - ΠΑΡΟΥΣΙΑΣΗ ΤΗΣ ΜΕΘΟΔΟΥ ΜΟΝΤΕΛΟΠΟΙΗΣΗΣ IDEF

ΕΙΣΑΓΩΓΙΚΑ ΣΤΟΙΧΕΙΑ ΓΙΑ ΤΑ IDEF.....	50
3.1 ΓΕΝΙΚΑ ΓΙΑ ΤΟ IDEF0.....	52
3.1.1 IDEF0 ΕΝΝΟΙΕΣ.....	52
3.2 ΓΕΝΙΚΑ ΓΙΑ ΤΟ IDEF1.....	57
3.2.1 IDEF1 ΟΙ ΣΤΟΙΧΕΙΩΔΕΙΣ ΑΡΧΕΣ.....	58
3.2.2 ΕΝΝΟΙΕΣ ΤΟΥ IDEF1.....	59
3.2.3 ΟΙ ΑΝΤΟΧΕΣ ΤΟΥ IDEF1.....	61
3.3 ΓΕΝΙΚΑ ΓΙΑ ΤΟ IDEF1X.....	62
3.3.1 ΕΝΝΟΙΕΣ ΤΟΥ IDEF1X.....	63
3.3.2 Η ΣΥΝΤΑΚΤΙΚΗ ΣΗΜΑΣΙΑ ΤΟΥ IDEF1X ΚΑΙ ΟΙ ΣΧΕΣΕΙΣ ΣΥΝΔΕΣΗΣ.....	64
3.3.2.1 ΣΧΕΣΕΙΣ ΚΑΤΗΓΟΡΙΟΠΟΙΗΣΗΣ.....	65
3.3.2.2 ΙΔΙΟΤΗΤΕΣ.....	65
3.3.2.3 ΚΛΕΙΔΙΑ.....	66
3.3.2.4 ΞΕΝΑ ΚΛΕΙΔΙΑ.....	67
3.3.3 ΑΝΤΟΧΗ ΤΟΥ IDEF1X.....	67
3.4 ΓΕΝΙΚΑ ΓΙΑ ΤΟ IDEF3.....	68
3.4.1 ΤΡΟΠΟΙ ΠΕΡΙΓΡΑΦΗΣ ΤΟΥ IDEF3.....	70
3.5 ΓΕΝΙΚΑ ΓΙΑ ΤΟ IDEF4.....	74
3.5.1 ΟΙ ΕΝΝΟΙΕΣ ΤΟΥ IDEF4.....	75
3.5.2 ΤΟ ΥΠΟ-ΠΡΟΤΥΠΟ ΚΑΤΗΓΟΡΙΑΣ ΤΟΥ IDEF4.....	78
3.5.2.1 ΔΙΑΓΡΑΜΜΑΤΑ ΚΛΗΡΟΝΟΜΙΑΣ ΤΟΥ IDEF4.....	78
3.5.2.2 ΔΙΑΓΡΑΜΜΑΤΑ ΑΜΑΣΟΤΗΤΑΣ ΤΟΥ IDEF4.....	79
3.5.2.3 ΔΙΑΓΡΑΜΜΑΤΑ ΠΡΩΤΟΚΟΛΛΟΥ IDEF4.....	80
3.5.3 ΤΟ ΥΠΟ-ΠΡΟΤΥΠΟ ΤΗΣ ΜΕΘΟΔΟΥ IDEF4.....	81
3.5.3.1 ΔΙΑΓΡΑΜΜΑΤΑ ΤΑΞΙΝΟΜΗΣΗΣ ΜΕΘΟΔΟΥ.....	81
3.5.3.2 ΔΙΑΓΡΑΜΜΑΤΑ ΠΕΛΑΤΩΝ.....	83
3.5.3.3 ΦΥΛΛΑ ΔΕΔΟΜΕΝΩΝ ΣΥΜΒΑΣΕΩΝ.....	83
3.5.4 ΣΥΜΠΕΡΑΣΜΑ.....	83
3.6 ΓΕΝΙΚΑ ΓΙΑ ΤΟ IDEF5.....	82
3.6.1 ΒΑΣΙΚΕΣ ΑΡΧΕΣ ΤΗΣ ΟΝΤΟΛΟΓΙΚΗΣ ΑΝΑΛΥΣΗΣ.....	83
3.6.2 ΟΙ ΕΝΝΟΙΕΣ ΤΟΥ IDEF5.....	85
3.6.3 ΓΛΩΣΣΕΣ ΟΝΤΟΛΟΓΙΑΣ ΤΟΥ IDEF5.....	85
3.6.4 ΣΧΗΜΑΤΙΚΟΙ ΤΥΠΟΙ IDEF5.....	87
3.6.5 ΣΥΜΠΕΡΑΣΜΑ.....	92

ΚΕΦΑΛΑΙΟ 4 - ΣΥΓΚΡΙΣΗ ΤΩΝ ΔΥΟ ΜΕΘΟΔΩΝ ΜΟΝΤΕΛΟΠΟΙΗΣΗΣ

4.1 ΣΥΓΚΡΙΣΗ ΤΩΝ ΜΟΝΤΕΛΩΝ UML / IDEF.....94

4.2 ΤΕΛΙΚΟ ΣΥΜΠΕΡΑΣΜΑ.....96

ΒΙΒΛΙΟΓΡΑΦΙΑ.....99

Κεφάλαιο 1

ΜΕΘΟΔΟΛΟΓΙΕΣ ΑΝΑΠΤΥΞΗΣ ΕΦΑΡΜΟΓΩΝ

Σε αυτό το κεφάλαιο θα εξετάσουμε την μεθοδολογία ανάπτυξης εφαρμογών που βασίζεται στα αντικείμενα, γιατί όπως θα δούμε στη συνέχεια τόσο η γλώσσα UML όσο και η γλώσσα IDEF έχουν ως σκοπό να υποστηρίξουν μεθοδολογίες που βασίζονται στα αντικείμενα.

1.1 Η Δομημένη προσέγγιση

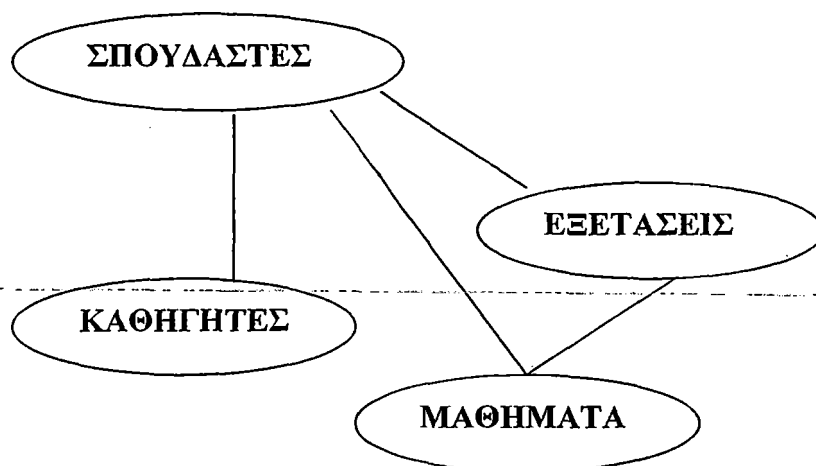
Καταρχήν, εξετάζουμε πώς τα συστήματα λογισμικού σχεδιάζονται χρησιμοποιώντας τη δομημένη (μερικές φορές αποκαλούμενη λειτουργική) προσέγγιση.

Στο δομημένο προγραμματισμό, η γενική μέθοδος ήταν να εξετάσει το πρόβλημα και έπειτα να σχεδιάσει μια συλλογή **λειτουργιών** που να μπορούν να εκτελεσθούν ώστε να επιτευχθούν οι απαραίτητοι στόχοι. Εάν και αυτές οι λειτουργίες είναι πάρα πολύ σύνθετες, κατόπιν οι λειτουργίες χωρίζονται έως ότου να είναι αρκετά απλές για να τις χειριστεί και να τις καταλάβει κάποιος. Αυτό είναι μια διαδικασία γνωστή σαν **λειτουργική αποσύνθεση**.

Οι περισσότερες λειτουργίες θα απαιτήσουν κάποιου είδους στοιχεία για να μπορέσουν να λειτουργήσουν. Τα στοιχεία αυτά, σε επίπεδο λειτουργικού συστήματος, κρατούνται συνήθως σε κάποια βάση δεδομένων (ή ενδεχομένως κρατούνται στη μνήμη). Για απλό παράδειγμα, εξετάζουμε ένα σύστημα διαχείρισης πανεπιστημίου. Αυτό το σύστημα κρατά λεπτομέρειες για κάθε σπουδαστή και καθηγητή στο πανεπιστήμιο. Επιπλέον, το σύστημα αποθηκεύει πληροφορίες για τις διαθέσιμες σειρές μαθημάτων στο πανεπιστήμιο και κατευθύνσεις που ο σπουδαστής ακολουθεί μετά από κάποιες σειρές μαθημάτων. Ένα πιθανό λειτουργικό σχέδιο θα ήταν να γραφτούν οι ακόλουθες λειτουργίες:

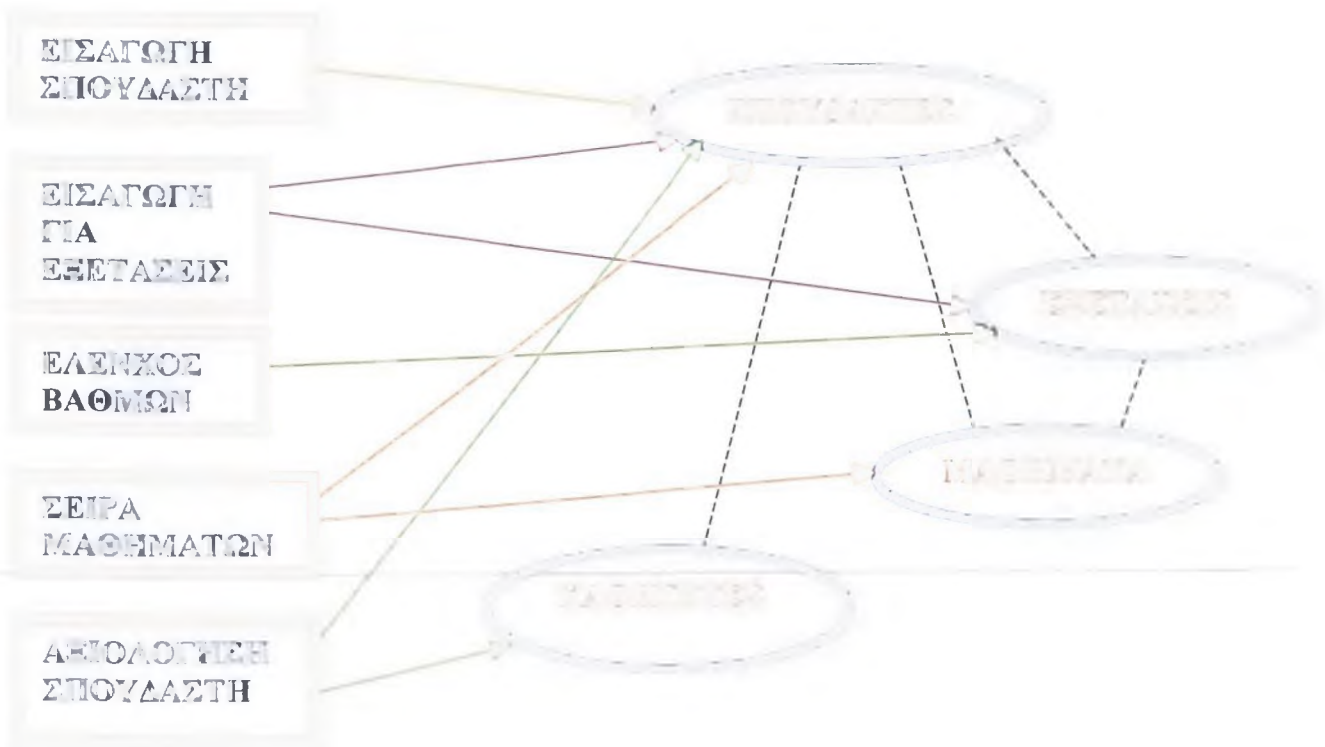
- ΕΙΣΑΓΩΓΗ ΣΠΟΥΔΑΣΤΗ
- ΕΙΣΑΓΩΓΗ ΓΙΑ ΕΞΕΤΑΣΕΙΣ
- ΕΛΕΓΧΟΣ ΒΑΘΜΩΝ
- ΣΕΙΡΑ ΜΑΘΗΜΑΤΩΝ
- ΑΞΙΟΛΟΓΗΣΗ ΣΠΟΥΔΑΣΤΗ

Θα χρειαζόμαστε επίσης ένα σύστημα αποθήκευσης στοιχείων για να υποστηρίξουμε αυτές τις λειτουργίες. Πρέπει να κρατήσουμε πληροφορίες για τους σπουδαστές, τους καθηγητές, τις εξετάσεις και τις σειρές μαθημάτων. Έτσι θα μπορούσαμε να σχεδιάσουμε μία βάση δεδομένων για να φυλαχτούν αυτά τα στοιχεία ως εξής:



Τώρα, οι λειτουργίες που καθορίσαμε νωρίτερα πρόκειται σαφώς να εξαρτηθούν από αυτό το σύνολο στοιχείων. Παραδείγματος χάριν, η «ΕΙΣΑΓΩΓΗ ΣΠΟΥΔΑΣΤΗ» λειτουργία θα πρέπει να τροποποιήσει το περιεχόμενο του στοιχείου "Σπουδαστές". Η λειτουργία «ΣΕΙΡΑ ΜΑΘΗΜΑΤΩΝ» θα πρέπει να έχει πρόσβαση στα στοιχεία σπουδαστών (να παρέχει τις λεπτομέρειες του σπουδαστή που απαιτεί το πιστοποιητικό), και στη λειτουργία θα χρειαστεί επίσης να έχει πρόσβαση και στα στοιχεία εξετάσεων.

Το ακόλουθο διάγραμμα είναι ένα σκίτσο όλων των λειτουργιών, μαζί με τα στοιχεία, και τα βέλη δείχνουν τα σημεία που υπάρχει εξάρτηση μεταξύ των λειτουργιών και των στοιχείων:



Το πρόβλημα με αυτήν την προσέγγιση είναι ότι εάν το πρόβλημα που αντιμετωπίζουμε γίνεται στην πορεία περισσότερο σύνθετο, το σύστημα γίνεται όλο και πιο δύσκολο να συντηρηθεί. Στο παράδειγμα ανωτέρω, τι θα συνέβαινε εάν μια απαίτηση αλλάξει και τότε με ποιο τρόπο αντιμετωπίζεται η αλλαγή στο στοιχείο του σπουδαστή;

Για παράδειγμα, στο παραπάνω παράδειγμα, αν παρουσιασθεί η απαίτηση να γίνουν αλλαγές στο στοιχείο «Σπουδαστές», τότε είναι αρκετά δύσκολο να αντιμετωπιστεί αυτή η αλλαγή. Πιο συγκεκριμένα, φανταστείτε ότι το σύστημά μας ότι «τρέχει» τέλεια, αλλά ότι πραγματοποιεί την αποθήκευση της ημερομηνίας γέννησης του σπουδαστή με δύο ψηφία, πράγμα που τελικά αποδεικνύεται ότι ήταν μια κακή ιδέα. Η προφανής λύση είναι να αλλαχτεί ο τομέας "ημερομηνίας γέννησης" στον πίνακα σπουδαστών, από έναν διψήφιο αριθμό έτους σε ένα τετραψήφιο αριθμό έτους.

Το σοβαρό πρόβλημα με αυτήν την αλλαγή είναι ότι θα μας προκαλέσει απροσδόκητα αποτελέσματα και γι' αυτό θα πρέπει οι αλλαγές να γίνουν πολύ προσεκτικά. Τα στοιχεία των εξετάσεων,

τα στοιχεία των καθηγητών και τα στοιχεία της σειράς των μαθημάτων εξαρτώνται (κατά κάποιον τρόπο) από τα στοιχεία σπουδαστών και θα πρέπει να προσέξουμε έτσι ώστε να μην δημιουργήσουμε τυχόν προβλήματα σε καμία λειτουργία με κάποια απλή αλλαγή που τυχόν κάνουμε.

Παραδείγματος χάριν, η «ΕΙΣΑΓΩΓΗ ΣΠΟΥΔΑΣΤΗ» δεν θα μπορεί πια να λειτουργεί σωστά, δεδομένου ότι θα αναμένει για "την ημερομηνία γέννησης" δυο ψηφία έτους παρά τέσσερα.

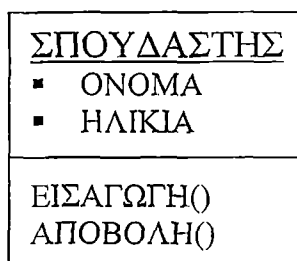
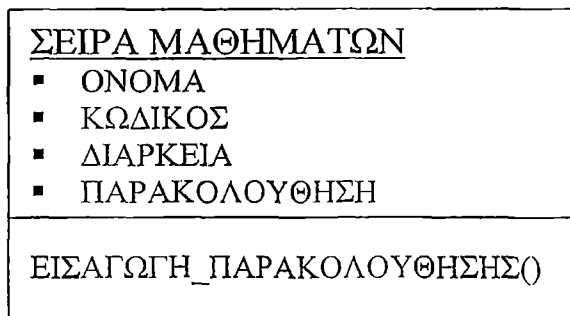
Έτσι έχουμε έναν μεγάλο βαθμό πιθανών προβλημάτων που μπορεί να εμφανιστούν στη πορεία. Αυτό που είναι πολύ, πολύ χειρότερο είναι ότι στον κώδικα του προγράμματός μας, δεν μπορούμε εύκολα να εντοπίσουμε που υπάρχουν τέτοιου είδους εξαρτήσεις.

Πόσες φορές έχει αλλαχθεί μια γραμμή κώδικα(και να φαίνεται τελείως αθώα η αλλαγή), χωρίς να έχει πραγματοποιηθεί ακούσιο σπάσιμο σε μια προφανώς ανεξάρτητη λειτουργία; Το δαπανηρό πρόβλημα του έτους 2000 προκλήθηκε από ακριβώς από μία φαινομενικά αθώα και απλή αλλαγή.

1.2 Η αντικειμενοστραφής (Object Orientated) προσέγγιση

Η προσέγγιση που είναι προσανατολισμένη προς το αντικείμενο προσπαθεί να ελαττώσει τον αντίκτυπο αυτού του προβλήματος απλά με το να συνδυάσει τα σχετικά στοιχεία και λειτουργίες στην ίδια ενότητα. Εξετάζοντας το παραπάνω σχήμα, είναι προφανές ότι τα στοιχεία και οι λειτουργίες συσχετίζονται. Για το παράδειγμα, οι λειτουργίες «ΕΙΣΑΓΩΓΗ_ΣΠΟΥΔΑΣΤΗ» και «ΑΞΙΟΛΟΓΗΣΗ_ΣΠΟΥΔΑΣΤΗ» συσχετίζονται σαφώς πιο πολύ με τα στοιχεία των σπουδαστών.

Το ακόλουθο σχήμα παρουσιάζει πλήρη ομαδοποίηση των σχετικών στοιχείων και των λειτουργιών με την μορφή ενοτήτων:



<u>ΚΑΘΗΓΗΤΗΣ</u>
<ul style="list-style-type: none"> ▪ ΟΝΟΜΑ ▪ ΑΜΟΙΒΗ
ΔΙΑΝΟΜΗ_ΜΑΘΗΜΑΤΩΝ()

<u>ΕΞΕΤΑΣΕΙΣ</u>
<ul style="list-style-type: none"> ▪ ΗΜΕΡΟΜΗΝΙΑ ▪ ΑΙΘΟΥΣΑ
ΕΙΣΑΓΩΓΗ_ΠΑΡΑΚΟΛΟΥΘΗΣΗΣ() ΕΛΕΓΧΟΣ_ΒΑΘΜΩΝ() ΣΕΙΡΑ_ΜΑΘΗΜΑΤΩΝ()

Μερικά σημαντικά σημεία για αυτό το νέο μοντέλο συστήματος προγραμματισμού είναι:

1. Περισσότερες από μια περιπτώσεις μιας ενότητας μπορούν να υπάρχουν όταν τρέχει το πρόγραμμα.

Στο σύστημα πανεπιστημίου, θα υπήρχε μια περίπτωση "σπουδαστή" για κάθε σπουδαστή όπου αυτός να ανήκει στο πανεπιστήμιο. Κάθε περίπτωση θα είχε τις τιμές της για τα στοιχεία του σπουδαστή(π.χ. προφανώς κάθε ένας σπουδαστής θα είχε ένα διαφορετικό όνομα).

2. Οι ενότητες μπορούν "να επικοινωνήσουν" με άλλες ενότητες με την κλήση των λειτουργιών τους.

Για παράδειγμα, όταν καλείται η λειτουργία "ΕΙΣΑΓΩΓΗ_ΣΠΟΥΔΑΣΤΗ" δημιουργείται μια νέα περίπτωση από την ενότητα σπουδαστών. Στην συνέχεια ο νέος σπουδαστής μπορεί να καλέσει την λειτουργία "ΕΙΣΑΓΩΓΗ_ΠΑΡΑΚΟΛΟΥΘΗΣΗΣ" της ενότητας «Εξετάσεις».

1.3 Ενθυλάκωση

Ουσιαστικά μόνο η περίπτωση που είναι κύρια ενός στοιχείου επιτρέπεται για να τροποποιήσει ή να διαβάσει αυτό το στοιχείο. Έτσι παραδείγματος χάριν, μια περίπτωση της ενότητας των καθηγητών δεν μπορεί να ενημερώσει ή να διαβάσει το στοιχείο της "ηλικίας" μιας περίπτωσης της ενότητας σπουδαστών.

Αυτή η συμπεριφορά καλείται **ενθυλάκωση**, και επιτρέπει στη δομή του συστήματος να είναι μακριά και με περισσότερες δυνατότητες ώστε να αποφεύγεται μια κατάσταση όπως περιγράφηκε προηγουμένως, όπου μια μικρή αλλαγή σε ένα

μέλος στοιχείων μπορεί να οδηγήσει σε άλλες αλλαγές οι οποίες θα έχουν δυσάρεστα αποτελέσματα στη σωστή λειτουργία του συστήματος .

Με την ενθουλάκωση, ο προγραμματιστής (π.χ.) της ενότητας «ΣΠΟΥΔΑΣΤΩΝ» μπορεί ακίνδυνα να κάνει αλλαγές στα στοιχεία της ενότητας, με τη διαβεβαίωση ότι δε θα πάθει καμία αλλαγή μια άλλη ενότητα που δεν είναι εξαρτώμενη από εκείνο το στοιχείο. Βέβαια ο προγραμματιστής θα πρέπει να ενημερώσει καλά και τις λειτουργίες που ανήκουν μέσα στην ενότητα. Δημιουργείται όμως έτσι η αίσθηση ότι κάθε ενότητα είναι απομονωμένη από τις υπόλοιπες ενότητες του συστήματος.

1.4 Αντικείμενα

Σε όλο αυτό το κεφάλαιο, έχω αναφερθεί σε αυτές τις συλλογές των σχετικών στοιχείων και λειτουργιών ως "ενότητες". Εντούτοις, εάν εξετάζαμε τα χαρακτηριστικά αυτών των ενότητων, θα μπορούσαμε να δούμε μερικές ομοιότητες με τα αντικείμενα του πραγματικού κόσμου.

Τα αντικείμενα στον πραγματικό κόσμο μπορούν να χαρακτηριστούν από δύο πράγματα: κάθε πραγματικό αντικείμενο έχει **στοιχεία** και **συμπεριφορά**. Παραδείγματος χάριν, μια τηλεόραση είναι ένα αντικείμενο συντονισμένο σε ένα ιδιαίτερο κανάλι, το ποσοστό ανίχνευσης καθορίζεται από μια συγκεκριμένη τιμή, η αντίθεση και η φωτεινότητα του έχει μία συγκεκριμένη τιμή κλπ. Η τηλεόραση μπορεί να ανάψει και από μακριά, το κανάλια μπορούν να αλλάξουν επίσης από μακριά. Μπορούμε να αντιπροσωπεύσουμε αυτές τις πληροφορίες με τον ίδιο τρόπο όπως και τις "ενότητες" των εφαρμογών λογισμικού: Υπό κάποια έννοια, ο πραγματικός κόσμος "αντικείμενα" μπορεί να διαμορφωθεί με παρόμοιο τρόπο με τις ενότητες λογισμικού που συζητήσαμε νωρίτερα. Για αυτόν τον λόγο, καλούμε τις ενότητες αυτές **αντικείμενα**, και ως εκ τούτου έχουμε τον όρο «**Προγραμματισμός προσανατολισμένος στα αντικείμενα**».

Με τα συστήματα λογισμικού λύνουμε πραγματικά προβλήματα (όπως τον τρόπο λειτουργίας ενός συστήματος εγγραφής σε ένα πανεπιστήμιο, ενός συστήματος διαχείρισης αποθηκών εμπορευμάτων ή ενός οπλικού συστήματος). Έτσι μπορούμε να προσδιορίσουμε τα αντικείμενα που υπάρχουν στο πραγματικό πρόβλημα και να τα μετατρέψουμε εύκολα σε αντικείμενα λογισμικού.

Με άλλα λόγια, ο αντικειμενοστραφής προγραμματισμός προέρχεται από την «μεταφορά» του πραγματικού κόσμου μέσα στις εφαρμογές λογισμικού.

1.5 Ορολογία

Τα στοιχεία για ένα αντικείμενο καλούνται γενικά **ιδιότητες** του αντικειμένου. Οι διαφορετικές συμπεριφορές ενός αντικειμένου καλούνται **μέθοδοι** του αντικειμένου. Οι μέθοδοι είναι άμεσα ανάλογες με τις λειτουργίες ή τις διαδικασίες στον προγραμματισμό των γλωσσών.

Ο άλλος μεγάλος όρος της επαγγελματικής γλώσσας είναι η **κατηγορία**. Μια κατηγορία είναι απλά ένα πρότυπο για ένα αντικείμενο. Μια κατηγορία περιγράφει ποιες ιδιότητες και μέθοδοι θα υπάρξουν για όλες τις περιπτώσεις της κατηγορίας.

Παραδείγματος χάριν στο σύστημα του πανεπιστημίου που περιγράψαμε σε αυτό το κεφάλαιο, καλέσαμε μια κατηγορία **σπουδαστή**. Οι ιδιότητες της **κατηγορίας σπουδαστών** ήταν όνομα, ηλικία, κ.λπ.... Οι μέθοδοι ήταν να προσθέτουν (**ΕΙΣΑΓΩΓΗ()**) και αποβάλετε (**ΑΠΟΒΟΛΗ()**). Στον κώδικά μας, θα πρέπει μόνο να καθορίσουμε αυτήν την κατηγορία μια φορά. Μόλις ο κώδικας είναι έτοιμος για εκτέλεση μπορούμε να δημιουργήσουμε τις περιπτώσεις της κατηγορίας - δηλ., μπορούμε να δημιουργήσουμε τα αντικείμενα της κατηγορίας. Κάθε ένα από αυτά τα αντικείμενα θα αντιπροσωπεύει έναν σπουδαστή, και κάθε ένα θα έχει το σύνολο τιμών του από τα στοιχεία.

1.6 Η προσανατολισμένη προς το αντικείμενο στρατηγική

Αν και αυτό το κεφάλαιο έχει αγγίξει εν συντομία τα οφέλη του προσανατολισμού αντικειμένου, έχουμε αφήσει πολλές ερωτήσεις αναπάντητες. Πώς προσδιορίζουμε πότε χρειάζεται να σχεδιάσουμε αντικείμενα για ένα σύστημα; Τι θα έπρεπε οι μέθοδοι και οι ιδιότητες να είναι; Πόσο μεγάλη θα έπρεπε μια κατηγορία να είναι; Η UML μπορεί να απαντήσει σε όλες αυτές τις ερωτήσεις πλήρως.

Μια σημαντική αδυναμία του προσανατολισμού αντικειμένου στο παρελθόν ήταν ότι ενώ ο αντικειμενοστραφής προγραμματισμός είναι ένα ισχυρό εργαλείο για την εργασία στην κατηγορία αντικειμένου, το επίπεδο του αντικειμενοστραφούς προγραμματισμού ήταν φτωχό στην έκφραση της συμπεριφοράς για ένα ολόκληρο σύστημα. Η εξέταση στις κατηγορίες είναι όλη πολύ καλή, αλλά οι κατηγορίες είναι πολύ "χαμηλού επιπέδου" οντότητες και δεν περιγράφουν πραγματικά τι το σύστημα **συνολικά** μπορεί να κάνει. Η χρησιμοποίηση των κατηγοριών μόνο, θα ήταν μάλλον όπως με τη προσπάθεια να γίνει κατανοητό πώς μια εργασία υπολογιστών εξετάζεται σε μια μητρική κάρτα αποτελούμενη από λυχνίες κρυστάλλου!

Η σύγχρονη προσέγγιση, που υποστηρίζεται έντονα από τη UML είναι **να αφήσουν** όλοι τα αντικείμενα και τις κατηγορίες στα αρχικά στάδια ενός προγράμματος, και να επικεντρώνονται αντ' αυτού σε αυτά που το σύστημα πρέπει να είναι σε θέση να κάνει. Κατόπιν, καθώς το πρόγραμμα προχωράει, οι κατηγορίες χτίζονται **βαθμιαία** πραγματοποιώντας την απαραίτητη λειτουργία των συστημάτων.

ΚΕΦΑΛΑΙΟ 2

ΠΑΡΟΥΣΙΑΣΗ ΤΗΣ ΜΕΘΟΔΟΥ ΜΟΝΤΕΛΟΠΟΙΗΣΗΣ UML

Τι είναι η UML;

Η *ενοποιημένη γλώσσα διαμόρφωσης*, ή *UML*, είναι μια γραφική γλώσσα διαμόρφωσης, αυτή παρέχει σε μας μια σύνταξη για την περιγραφή των σημαντικότερων στοιχείων (αποκαλούμενων *χειροποίητα αντικείμενα UML*) των συστημάτων λογισμικού. Σε αυτήν την σειρά θα ερευνήσουμε τις κύριες πτυχές της UML, και για το πώς περιγράφεται η UML και το πώς μπορεί να εφαρμοστεί στα αναπτυξιακά έργα λογισμικού.

Μέσω στον πυρήνα της, η UML προσανατολίζεται προς το ανάπτυξης λογισμικού αντικειμένου.

Σε αυτό το κεφάλαιο, θα εξετάσουμε την προέλευση της UML, και θα συζητήσουμε την ανάγκη για μια κοινή γλώσσα στη βιομηχανία λογισμικού. Κατόπιν θα αρχίσουμε να εξετάζουμε το πώς να εκμεταλλευτείτε τη UML σε ένα πρόγραμμα λογισμικού.

Μια κοινή γλώσσα

Μέχρι σήμερα, η τεχνολογία λογισμικού έχει στερηθεί μια τέτοια δυνατότητα. Μεταξύ 1989 και 1994, (α) περίοδος καλούμενη ως οι "πόλεμοι μεθόδου", περισσότερες από 50 γλώσσες διαμόρφωσης λογισμικού ήταν σε κοινή χρήση, κάθε μια από αυτές φέρνουν και τις δικές τους δυνατότητες.

Κάθε γλώσσα περιλάμβανε ιδιαίτερη σύνταξη ενώ συγχρόνως, κάθε γλώσσα είχε στοιχεία που άντεξαν στις ομοιότητες με τις άλλες γλώσσες. Επιπρόσθετη σύγχυση υπήρχε στο ότι καμία γλώσσα δεν ήταν πλήρης, υπό την έννοια ότι πολύ λίγοι επαγγελματίες λογισμικού βρήκαν την πλήρη ικανοποίηση από μια ενιαία γλώσσα!

Στη μέση τις δεκαετία του '90, τρεις μέθοδοι προέκυψαν ως ισχυρότερες.

Κάθε μέθοδος είχε ιδιαίτερες δυνατότητες:

- **To Booch** ήταν άριστο για το σχέδιο και την εφαρμογή. Το Grady Booch είχε λειτουργήσει εκτενώς με την ADA γλώσσα.
- **OMT** (τεχνική διαμόρφωση αντικειμένου) ήταν καλύτερο για την ανάλυση και *data-intensive* σε συστήματα πληροφοριών.
- **OOSE** (προσανατολισμένη προς τη τεχνολογία λογισμικού αντικειμένου) χαρακτήρισε ένα πρότυπο γνωστό ως **χρήση**

ΠΕΡΙΠΤΩΣΕΩΝ.

Οι περιπτώσεις χρήσης είναι μια ισχυρή τεχνική για τη συμπεριφορά ολόκληρου του συστήματος. Το 1994, ο jim Rumbaugh, ο δημιουργός του OMT, ζάλισε τον κόσμο του λογισμικού όταν έφυγε από την General Electric και ενώθηκε με το Grady Booch με το Rational Corp. Ο στόχος της συνεργασίας ήταν να συγχωνεύσει τις ιδέες τους σε μια ενιαία, ενοποιημένη μέθοδο (ο τίτλος εργασίας για τη μέθοδο ήταν πράγματι η "ενοποιημένη μέθοδος").

Μέχρι το 1995, ο δημιουργός OOSE, Ivar Jacobson, είχε ενώσει επίσης το Rational, και τις ιδέες του (ιδιαίτερα στη έννοια των "περιπτώσεων χρήσης") ώστε να τροφοδοτήσει τη νέα ενοποιημένη μέθοδο - τώρα λέγεται λοιπόν "ενοποιημένη γλώσσα διαμόρφωσης"- UML.

Η ομάδα Rumbaugh, Booch και Jacobson είναι ευρέως γνωστή ως "Three Amigos".

Παρά τους αρχικούς επιχειρηματικούς πολέμους, η νέα μέθοδος άρχισε να βρίσκει την εύνοια στη βιομηχανία λογισμικού και μια κοινοπραξία για τη UML συγκροτήθηκε. Οι βαρέων βαρών εταιρίες ήταν μέρος της κοινοπραξίας, συμπεριλαμβανομένου των Hewlett-Packard, Microsoft και Oracle.

Η UML υιοθετήθηκε από το OMG 2 το 1997, και από τότε το OMG ήταν ο κύριος υποστηρικτής της γλώσσας. Επομένως, η UML είναι αποτελεσματικά μια δημόσια, μη ειδικευμένη γλώσσα.

Μια επισκόπηση της UML

Προτού να αρχίσουμε να εξετάζουμε τη θεωρία της UML, πρόκειται να πάρουμε μια πολύ συνοπτική γεύση μερικών από τις σημαντικότερες έννοιες της UML.

Το πρώτο πράγμα που παρατηρείται για τη UML είναι ότι υπάρχουν πολλά διαφορετικά διαγράμματα (πρότυπα) για να συνηθίσουμε. Ο λόγος για αυτό είναι ότι είναι δυνατό να εξετάσει ένα σύστημα από πολλές διαφορετικές απόψεις. Μια ανάπτυξη λογισμικού θα έχει πολλούς συμμετέχοντες που θα παίζουν έναν ρόλο – παραδείγματος χάριν:

- Αναλυτές
- Σχεδιαστές
- Κωδικοποιητές
- Ελεγκτές
- QA
- Πελάτες
- Τεχνικοί συντάκτες

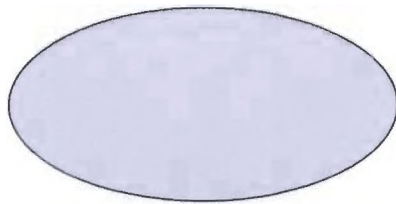
Όλοι αυτοί οι άνθρωποι ενδιαφέρονται για τις διαφορετικές πτυχές του συστήματος, και κάθε μια από αυτές απαιτεί ένα διαφορετικό επίπεδο λεπτομέρειας. Παραδείγματος χάριν, ένας κωδικοποιητής πρέπει να καταλάβει το σχέδιο από το σύστημα και να είναι σε θέση να μετατρέψει το σχέδιο σε έναν χαμηλό κώδικα επιπέδων. Σε αντίθεση, ο τεχνικός συγγραφέας ενδιαφέρεται για τη συμπεριφορά του συστήματος συνολικά, και τις ανάγκες που έχει ώστε να καταλάβει πώς το προϊόν λειτουργεί. Η UML προσπαθεί να παρέχει μια γλώσσα έτσι που όλοι οι συμμετοχοί να μπορούν να ωφεληθούν από τουλάχιστον ένα διάγραμμα UML.

Παρακάτω θα ακολουθήσει μια γρήγορη ματιά σε μερικά από τα σημαντικότερα διαγράμματα.

2.1 Διάγραμμα περιπτώσεων χρήσης (Use Case Diagram)

Ένα πολύ ισχυρό εργαλείο της UML είναι η περίπτωση χρήσης (Use Case). Μια περίπτωση χρήσης είναι απλά μια περιγραφή του συνόλου αλληλεπιδράσεων μεταξύ ενός χρήστη και του συστήματος. Με την ενίσχυση της συλλογής μιας χρήσης περιπτώσεων, μπορούμε να περιγράψουμε ολόκληρο το σύστημα που προγραμματίζουμε να δημιουργήσουμε, με έναν πολύ σαφή και συνοπτικό τρόπο. Οι περιπτώσεις χρήσης περιγράφονται συνήθως χρησιμοποιώντας τους συνδυασμούς ρήματος και ουσιαστικού – παραδείγματος χάριν, " η μισθοδοτική κατάσταση των αναπροσαρμογών ".

Παραδείγματος χάριν, εάν γράφαμε για ένα πυραύλιο σύστημα ελέγχου, χαρακτηριστικές περιπτώσεις χρήσης για το σύστημα αυτό θα να είναι "εκτόξευση πυραύλων", ή "θέματα αμυντικών μέτρων". Μαζί με το όνομα της περίπτωσης χρήσης, θα παράσχουμε μια πλήρη περιγραφή αλληλεπιδράσεων που θα εμφανιστούν μεταξύ του χρήστη και του συστήματος. Αυτές οι περιγραφές θα γίνονταν γενικά αρκετά περίπλοκες, αλλά η UML παρέχει μια απλή σημείωση για να αντιπροσωπεύσει μια περίπτωση χρήσης, ως εξής:

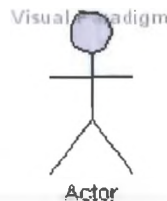


σημείωση περίπτωσης χρήσης

2.1.1 Δράστες (Actor)

Μια περίπτωση χρήσης δεν μπορεί να αρχίσει τις ενέργειες από μόνη της. Ένας **δράστης** είναι κάποιος που **μπορεί** να αρχίσει μια περίπτωση χρήσης.

Παραδείγματος χάριν, εάν αναπτύσσαμε ένα τραπεζικό σύστημα, και έχουμε μια περίπτωση χρήσης αποκαλούμενη "να αποσύρει τα χρήματα", κατόπιν θα προσδιορίζαμε ότι απαιτούμε από **τους πελάτες** να είναι σε θέση να αποσύρουν τα χρήματα, και ως εκ τούτου ένας **πελάτης** θα γινόταν ένας από τους δράστες μας. Πάλι, η σημείωση για έναν δράστη είναι απλή:

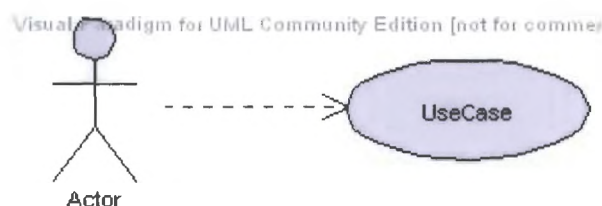


σημείωση της UML για έναν δράστη

Πηγαίνοντας περαιτέρω, οι δράστες μπορούν να είναι περισσότεροι από έναν ακριβώς άνθρωπο. Ένας δράστης μπορεί να είναι τίποτα εξωτερικό στο σύστημα που να αρχίζει μια περίπτωση χρήσης, όπως ένα άλλο συγκρότημα ηλεκτρονικών υπολογιστών. Ένας δράστης θα μπορούσε επίσης ενδεχομένως να είναι μια πιο αφηρημένη έννοια όπως **ο χρόνος**, ή μια συγκεκριμένη **ημερομηνία**. Για παράδειγμα, μπορούμε να καλέσουμε μια περίπτωση χρήσης για "την εκκαθάριση παλαιάς διαταγής". Σε μια επεξεργασία διαταγής το σύστημα, και ο αρχικός δράστης θα μπορούσαν να είναι η "τελευταία εργάσιμη ημέρα".

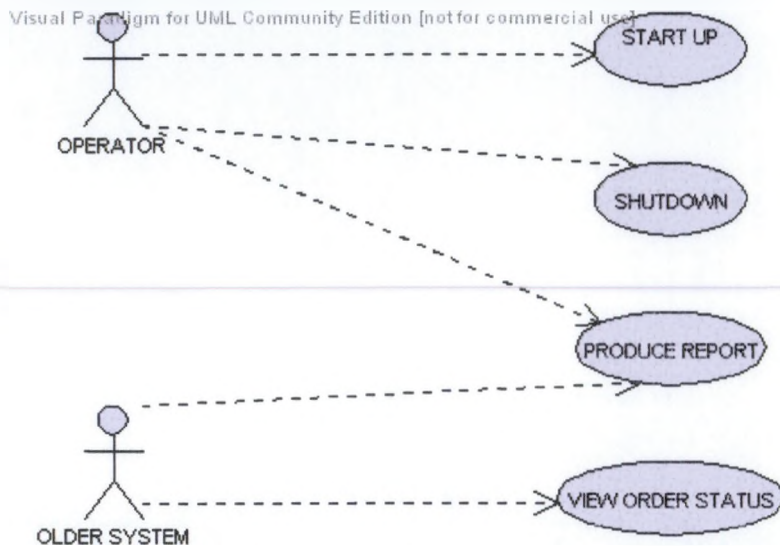
Όπως έχουμε σημειώσει, οι δράστες συσχετίζονται με τις περιπτώσεις χρήσης, υπό την έννοια ότι είναι δράστης αυτός που θα αρχίσει μια ιδιαίτερη περίπτωση χρήσης.

Μπορούμε να αντιπροσωπεύσουμε αυτό σε ένα διάγραμμα περίπτωσης χρήσης κάνοντας σύνδεση του δράστη με την περίπτωση χρήσης:



η σχέση ενός δράστη σε μια περίπτωση χρήσης

Σαφώς, για τα περισσότερα συστήματα, ένας ενιαίος δράστης μπορεί να αλληλεπιδράσει με πολλές περιπτώσεις χρήσης, και το ότι η περίπτωση χρήσης μπορεί να αρχίσει από πολλούς διαφορετικούς δράστες. Αυτό οδηγεί στην πλήρη περίπτωση διαγράμματος χρήσης, ένα παράδειγμα είναι το ακόλουθο:



ένα πλήρες σύστημα που περιγράφεται χρησιμοποιώντας τους δράστες και τις περιπτώσεις χρήσης

2.1.2 Ο σκοπός των περιπτώσεων χρήσης

Λαμβάνοντας υπόψη τον απλό ορισμό της "περίπτωσης χρήσης" και "του δράστη", μαζί με την απλή απεικόνιση των περιπτώσεων χρήσης μέσω του προτύπου UML, θα μπορούσαμε να συγχωρηθούμε για τη σκέψη ότι οι περιπτώσεις εκείνης της χρήσης είναι απλές – σχεδόν πάρα πολλές είναι απλές για να ανησυχούμε. **Λάθος**. Οι περιπτώσεις χρήσης είναι πάρα πολύ ισχυρές.

- Οι περιπτώσεις χρήσης καθορίζουν το πεδίο του συστήματος. Μας επιτρέπουν να απεικονίσουμε το μέγεθος και το πεδίο ολόκληρης της ανάπτυξης.

- Οι περιπτώσεις χρήσης είναι πολύ παρόμοιες με τις απαιτήσεις, αλλά ενώ οι απαιτήσεις τείνουν να είναι ασαφής, και γραπτάς διφορούμενες η σφιχτότερη δομή των περιπτώσεων χρήσης τείνουν πολύ στο να τις αντικαταστήσουν
- Το "σύνολο" των περιπτώσεων χρήσης είναι ολόκληρο το σύστημα. Αυτό σημαίνει ότι καμία περίπτωση χρήσης δε καλύπτεται εφόσον αυτή είναι έξω από το όριο του συστήματος που αναπτύσσουμε. Έτσι το διάγραμμα είναι πλήρης, χωρίς να παρουσιάζει κενά.
- Επιτρέπει την επικοινωνία μεταξύ του πελάτη και των υπεύθυνων που έχουν αναλάβει την ανάπτυξη (το διάγραμμα είναι τόσο απλό, που ο καθένας μπορεί να το καταλάβει)
- Οι περιπτώσεις χρήσης καθοδηγούν τις ομάδες ανάπτυξης μέσω της διαδικασίας ανάπτυξης – εμείς θα δούμε ότι οι περιπτώσεις χρήσης είναι η "σπονδυλική στήλη" της ανάπτυξής μας.
- Θα δούμε ότι οι περιπτώσεις χρήσης παρέχουν μια μέθοδο για το έργο ανάπτυξης, που μας επιτρέπει να υπολογίσουμε πόσο καιρό θα χρειαστούμε για την ανάπτυξη του έργου.
- Οι περιπτώσεις χρήσης παρέχουν τη βάση για τις δομές των συστημάτων
- Τέλος, οι περιπτώσεις χρήσης βοηθούν για τη δημιουργία των οδηγών χρηστών.

Συχνά υποστηρίζεται ότι οι περιπτώσεις χρήσης είναι απλά μια έκφραση των απαιτήσεων του συστήματος. Καθένας που κάνει αυτήν την αξίωση χάνει σαφώς το νόημα των περιπτώσεων χρήσης.

2.1.3 Περιγραφές περιπτώσεων χρήσης

Κάθε περίπτωση χρήσης περιέχει ένα πλήρες σύνολο λεπτομερειών για τις αλληλεπιδράσεις και τα σενάρια που περιλαμβάνουν. Η UML δεν διευκρινίζει τη δομή και το περιεχόμενο ενός εγγράφου για να γίνει αυτό θα πρέπει να βρίσκεται σαν μεμονωμένο πρόγραμμα μιας επιχείρησης για να μπορέσει να διευκρινιστεί .

Θα χρησιμοποιήσουμε τον ακόλουθο πρότυπο:

- **Περίπτωση χρήσης:** *Όνομα περίπτωσης χρήσης*
- **Σύντομη περιγραφή:** *Μια συνοπτική περιγραφή της περίπτωσης χρήσης*
- **Προϋποθέσεις:** *Μια περιγραφή των όρων που πρέπει να ικανοποιηθούν πριν από μια περίπτωση χρήσης*
- **Μετά-όροι :** *Μια περιγραφή που έχει συμβεί στο τέλος της περίπτωσης χρήσης*
- **Κύρια ροή:** *Ένας κατάλογος των αλληλεπιδράσεων των συστημάτων που πραγματοποιούνται κάτω από πολύ κοινά σενάρια. Παραδείγματος χάριν, για "αποσύρετε τα χρήματα", ή "εισάγει την κάρτα"*
- **Εναλλάξτε τη ροή :** *Μια περιγραφή των πιθανών εναλλακτικών αλληλεπιδράσεων.*
- **Ροή εξαίρεσης :** *Μια περιγραφή των πιθανών σεναρίων όπου απροσδόκητα γεγονότα έχουν τη δυνατότητα να πραγματοποιηθούν*

πρότυπο για μια περιγραφή περίπτωσης χρήσης

2.1.4 Εύρεση των περιπτώσεων χρήσης

Μια προσέγγιση στην εύρεση των περιπτώσεων χρήσης είναι μέσω των συνεντεύξεων με τους δυνητικούς χρήστες του συστήματος. Αυτό είναι ένας δύσκολος στόχος, δεδομένου ότι δύο άνθρωποι είναι πιθανό να δώσουν δύο εντελώς διαφορετικές απόψεις σχετικά με αυτό που το σύστημα πρέπει να κάνει (ακόμα κι αν λειτουργούν για την ίδια επιχείρηση)!

Βεβαίως, οι περισσότερες εξελίξεις θα περιλαμβάνουν ως κάποιο βαθμό την άμεση επικοινωνία έναν προς έναν χρήστη. Εντούτοις, λαμβάνοντας υπόψη τη δυσκολία για μια συνεπή άποψη αυτών που το σύστημα θα πρέπει να κάνει, πλησιάζουμε μια άλλη προσέγγιση η οποία γίνεται δημοφιλέστερη και είναι το "εργαστήριο".

Κοινές απαιτήσεις που προγραμματίζουν τα εργαστήρια (JRP)

Η προσέγγιση εργαστηρίων παίρνει μαζί μια ομάδα ανθρώπων ενδιαφερόμενη για την ανάπτυξη του συστήματος (οι *συμμέτοχοι*). Ο καθένας στην ομάδα καλείται για να δώσει την άποψη του για το τι χρειάζεται να κάνει το σύστημα. Το κλειδί για την επιτυχία αυτών των εργαστηρίων είναι ο « βοηθός ». Οδηγούν την ομάδα εξασφαλίζοντας ότι η συζήτηση θα επικεντρωθεί στην εύρεση τρόπου κάλυψης των απαιτήσεων , και ότι όλοι οι συμμετοχοί θα πρέπει να ενθαρρύνονται για να πουν τις απόψεις τους. Οι καλοί βοηθοί είναι ανεκτίμητοι. Ένας γραμματέας θα είναι επίσης παρών, ο οποίος θα εξασφαλίσει ότι όλα είναι τεκμηριωμένα. Ο γραμματέας θα επεξεργαστεί το έγγραφο, αλλά μια καλύτερη μέθοδος είναι να συνδεθεί ένα εργαλείο ΠΕΡΙΠΤΩΣΕΩΝ ή το εργαλείο σχεδίων με το οποίο ο προβολέας να απεικονίζει τα διαγράμματα "ζωντανά".

Η απλότητα του διαγράμματος περίπτωσης χρήσης είναι κρίσιμη εδώ, όλοι οι συμμετοχοί, πρέπει να είναι σε θέση να καταλάβουν την έννοια του διαγράμματος με ευκολία.

Μια απλή μέθοδος στο εργαστήριο είναι:

- 1) Καταιγισμός ιδεών από όλους τους πιθανούς δράστες πρώτα
 - 2) Έπειτα, καταιγισμός ιδεών για όλες τις πιθανές περιπτώσεις χρήσης
 - 3) Μόλις το "brainstorming" ολοκληρωθεί από την ομάδα, αξιολογείται κάθε περίπτωση χρήσης και κατευθείαν παράγεται μια απλή περιγραφική παράγραφος
 - 4) Σύλληψη και καθορισμός προτύπου
- Τα βήματα 1) και 2) μπορούν να αντιστραφούν εάν επιδιώκεται.

Κάποιες καλές συμβουλές για το εργαστήριο:

- Μην απασχοληθείτε πάρα πολύ με το να προσπαθείτε να βρείτε σε κάθε μια περίπτωση χρήση και δράστη. Είναι φυσικό ότι μερικές από τις περισσότερες περιπτώσεις χρήσης θα προκύψουν αργά στη διαδικασία.
- Εάν δεν μπορείτε να δικαιολογήσετε την περίπτωση χρήσης στο βήμα 3), ίσως και να μην είναι μια περίπτωση χρήσης. Αισθανθείτε ελεύθεροι αφαιρέστε οποιοσδήποτε περιπτώσεις χρήσης αισθάνεστε ότι είναι ανακριβής ή περιττές.

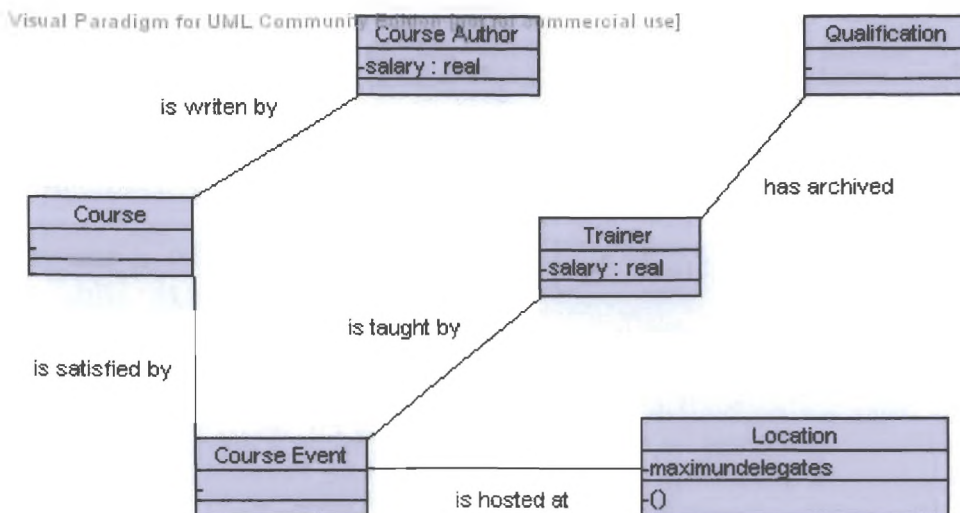
2.2 Το διάγραμμα κατηγορίας (Class Diagram)

Τα διαγράμματα κατηγορίας είναι μια ουσιαστική πτυχή οποιασδήποτε προσανατολισμένης μεθόδου σχεδιασμού προς το αντικείμενο, έτσι δεν είναι έκπληξη το ότι η UML παρέχει σε μας την κατάλληλη σύνταξη. Θα δούμε ότι μπορούμε να χρησιμοποιήσουμε το διάγραμμα κατηγορίας στο στάδιο ανάλυσης καθώς επίσης και στο σχεδιασμό – θα χρησιμοποιήσουμε τη σύνταξη διαγραμμάτων κατηγορίας για να απεικονίσουμε σε ένα σχέδιο τις σημαντικότερες έννοιες ώστε ο πελάτης μας να τις καταλαβαίνει (και θα το καλέσουμε αυτό *εννοιολογικό πρότυπο*).

Μαζί με τις περιπτώσεις χρήσης, το εννοιολογικό πρότυπο είναι μια ισχυρή τεχνική για την ανάλυση απαιτήσεων

2.2.1 Εννοιολογική διαμόρφωση

Η εννοιολογική διαμόρφωση (μερικές φορές αποκαλείται και *περιοχή διαμόρφωσης*) είναι η δραστηριότητα της εύρεσης των εξωτερικών εννοιών που είναι σημαντικές για το σύστημά μας. Αυτή η διαδικασία μας βοηθά για να καταλάβουμε το πρόβλημα περαιτέρω, και να αναπτύξουμε μια καλύτερη μορφή της επιχείρησης του πελάτη μας. Για άλλη μια φορά, η UML δεν μας λέει πώς ή πότε πρέπει να γίνει η διαμόρφωση περιοχών, αλλά αυτό που μας παρέχει είναι η σύνταξη για να εκφραστεί το πρότυπο. Το πρότυπο που πρόκειται να χρησιμοποιήσουμε είναι το **διάγραμμα κατηγορίας (Class Diagram)**.



το διάγραμμα κατηγορίας UML

Η ανάπτυξη ενός διαγράμματος κατηγορίας είναι βασική σε οποιαδήποτε προσανατολισμένη προς το αντικείμενο διαδικασία σχεδιασμού. Το διάγραμμα κατηγορίας θα παρέχει ουσιαστικά τη δομή του ενδεχόμενου κώδικα που παράγουμε.

Σε αυτή τη φάση, εντούτοις, δεν ενδιαφερόμαστε ακόμα για το σχέδιο συστημάτων (είμαστε ακόμα στην ανάλυση), έτσι το διάγραμμα κατηγορίας που παράγουμε σε αυτή τη φάση θα είναι αρκετά περιγραμματικό και δεν περιέχει αποφάσεις σχεδίου.

Παράγουμε ουσιαστικά ένα **διάγραμμα κατηγορίας ανάλυσης**. Πολλοί επαγγελματίες προτιμούν να διαχωρίζουν εντελώς το διάγραμμα κατηγορίας ανάλυσης από το διάγραμμα κατηγορίας σχεδιασμού και να αποκαλούν το διάγραμμα ανάλυσης ως **εννοιολογικό πρότυπο**.

Στο εννοιολογικό πρότυπο, στοχεύουμε να συλλάβουμε όλες τις έννοιες ή ιδέες που ο πελάτης γνωρίζει. Παραδείγματος χάριν, μερικά καλά παραδείγματα των εννοιών θα ήταν:

- **Lift** : ένα σύστημα ελέγχου ανελκυστήρων
- **Order**: ένα σύστημα εγχώριων αγορών
- **Footballer** : η μεταφορά σε ένα σύστημα ενός ποδοσφαιρικού παιχνιδιού

- **Trainer** : σε ένα σύστημα διαχείρισης αποθεμάτων για ένα κατάστημα παπουτσιών
- **Room** : σε ένα σύστημα κράτησης δωματίων

Μερικά πολύ κακά παραδείγματα των εννοιών είναι:

- **Order Purge Daemon**: η διαδικασία που διαγράφει τακτικά τις παλαιές διαταγές από το σύστημα
- **Event Trigger** : η ειδική διαδικασία που περιμένει 5 λεπτά και λέει έπειτα στο σύστημα να « ξυπνήσει » και κάνει κάτι
- **Customer Details Form** : το παράθυρο που ζητά τις λεπτομέρειες ενός νέου πελάτη μέσα ένα σύστημα αγορών
- **Db Archive Table** : ο πίνακας βάσεων δεδομένων που κρατά έναν κατάλογο όλων των παλαιών διαταγών

Αυτές είναι κακές έννοιες, επειδή στρέφουν το σχεδιασμό προς τη λύση και όχι προς το πρόβλημα.

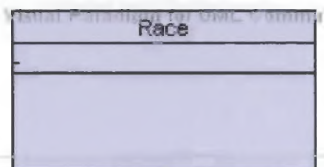
Στο παράδειγμα Db Archive Table, αναγκαζόμαστε ήδη να στραφούμε στη λύση μιας σχεσιακή βάση δεδομένων. Τι και εάν βγάζει αργότερα ότι είναι αποδοτικότερο, φτηνότερο και τελείως αποδεκτό να χρησιμοποιηθεί ένα απλό αρχείο κειμένων; Η καλύτερη εμπειροτεχνική μέθοδος είναι εδώ:

Εάν ο πελάτης δεν καταλαβαίνει την έννοια, δεν είναι πιθανώς μια έννοια!

Οι σχεδιαστές μισούν το εννοιολογικό βήμα που – δεν μπορούν να περιμένουν μέχρι τη φάση σχεδιασμού. Εμείς θα δούμε, εντούτοις, ότι το εννοιολογικό πρότυπο θα μετασχηματιστεί αργά σε ένα πλήρες σχέδιο διαγράμματος κατηγορίας καθώς κινούμαστε μέσω της φάσης κατασκευής.

2.2.2 Το εννοιολογικό πρότυπο στη UML

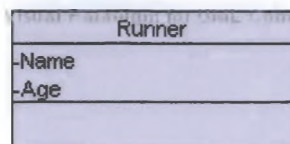
Τώρα που έχουμε δει πώς να ανακαλύψουμε τις έννοιες, πρέπει να δείξουμε πώς να κατανοούμε τις έννοιες στη UML. Θα χρησιμοποιήσουμε τις πτυχές πυρήνων του **διαγράμματος κατηγορίας**. Αντιπροσωπεύουμε την έννοιά μας σε ένα απλό κιβώτιο, με τον τίτλο της έννοιας στην κορυφή του κιβωτίου.



η έννοια "Race" που συλλαμβάνεται σε UML (για ένα σύστημα αγώνων αλόγων)

Η παρατήρηση βρίσκετε μέσα στο μεγάλο κιβώτιο και υπάρχουν ακόμη δύο μικρότερα, κενά κιβώτια.

Το κιβώτιο στη μέση θα χρησιμοποιηθεί σύντομα, για να συλλάβει τις **ιδιότητες** της έννοιας όπου είναι και το σημαντικότερο αυτή τη στιγμή. Το κάτω κιβώτιο χρησιμοποιείται για να κατανοήσει τη **συμπεριφορά** της έννοιας, με άλλα λόγια τι μπορεί η έννοια πραγματικά να κάνει. Η απόφαση σχετικά με τη συμπεριφορά της έννοια είναι ένα περίπλοκο βήμα, και αναβάλλουμε αυτό το στάδιο έως ότου είμαστε στο στάδιο της κατασκευής. Έτσι εμείς δε χρειάζεται να ανησυχούμε για τη συμπεριφορά του προς το παρόν.



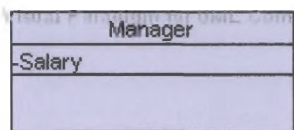
η UML συλλαμβάνει τις ιδιότητες και τη συμπεριφορά μιας έννοιας

Στο παράδειγμα ανωτέρω, έχουμε αποφασίσει ότι κάθε δρομέας θα έχει δύο ιδιότητες – "Όνομα" και "ηλικία". Αφήνουμε το κενό κατώτατης περιοχής μέχρι αργότερα, όταν αποφασίσουμε τι ένας "δρομέας" είναι σε θέση να κάνει .

2.2.3 Εύρεση των ιδιοτήτων

Πρέπει να καθορίσουμε ποιες είναι οι ιδιότητες κάθε έννοιας, πάλι ο καλύτερος τρόπος για να επιτευχθεί αυτό είναι με μια σύσκεψη (brainstorming) με τους συμμετόχους. Παραδείγματος χάριν, αφήνουμε να πούμε ότι εμείς εργαζόμαστε σε ένα σύστημα διαχείρισης προσωπικού. Εμείς έχουμε προσδιορίσει ότι μια έννοια θα ήταν ο "διευθυντής".

Μια πρόταση για μια ιδιότητα θα είναι ο "μισθός", ως εξής:



έννοια διευθυντών, με την ιδιότητα "μισθός"

Αυτό φαίνεται λεπτό, αλλά κάποιος πιθανώς να υποστηρίξει ότι "ο μισθός" είναι επίσης μια έννοια. Έτσι θα πρέπει να το προαγάγουμε από μια ιδιότητα σε μια έννοια.

Έχουν υπάρξει πολλοί σύνοδοι διαμόρφωσης ώστε να μπορέσουν να συμπεριληφθούν όλα τα επιχειρήματα σε ένα ικανοποιητικό και «ελαστικό» μοντέλο ώστε να μην υπάρχουν ανησυχίες για το αν χρειαστεί τροποποίηση σε μια έννοια ή ιδιότητα. Έτσι ακολουθούμε τον εξής κανόνα: *αν υπάρχει αμφιβολία για μια ιδιότητα, τότε φτιάξτε μια άλλη έννοια.*



διευθυντής και μισθός, δύο χωριστές έννοιες

2.2.4 Οδηγίες για την εύρεση των ιδιοτήτων

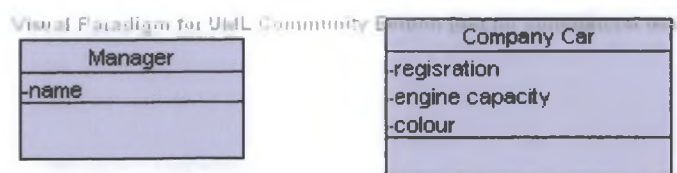
Οι ακόλουθες εμπειροτεχνικές μέθοδοι μπορούν να είναι χρήσιμες κατά την απόφαση μεταξύ των ιδιοτήτων και των εννοιών, αλλά προσέχοντας και τις ανωτέρω συμβουλές δεν θα ανησυχούμε και πάρα πολύ για τη διάκρισή τους.

(«Αν υπάρχει αμφιβολία, το κάνετε μια έννοια!»)

- Οι ενιαίες εκτιμημένες σειρές ή οι αριθμοί είναι συνήθως ιδιότητες
- Εάν μια ιδιότητα μιας έννοιας δεν μπορεί να κάνει τίποτα, π.χ. για την έννοια διευθυντών, το "όνομα" σαφώς ηχεί όπως μια ιδιότητα. Τα "αυτοκίνητα επιχείρησης" όμως ηχούν όπως μια έννοια, επειδή πρέπει να αποθηκεύσουμε τις πληροφορίες για κάθε αυτοκίνητο όπως αριθμό μητρώου και χρώμα.

Ενώσεις

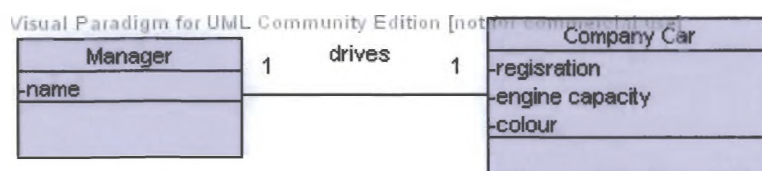
Το επόμενο βήμα είναι να αποφασιστεί πώς οι έννοιές μας θα συσχετίζονται. Σε οποιοδήποτε non-trivial σύστημα, τουλάχιστον μερικές από τις έννοιες πρόκειται να έχουν την κάποια εννοιολογική σχέση με άλλες έννοιες. Παραδείγματος χάριν, πίσω στο σύστημα διαχείρισης προσωπικού μας, λαμβάνοντας υπόψη και τις δύο έννοιες:



έννοιες αυτοκινήτων και διευθυντών της επιχείρησης

Αυτές οι έννοιες συσχετίζονται, επειδή στην επιχείρηση αναπτύσσουμε ένα σύστημα για κάθε διευθυντή που οδηγεί ένα αυτοκίνητο της επιχείρησης.

Μπορούμε να εκφράσουμε αυτήν την σχέση στη UML με το να συνδέσουμε τις δύο έννοιες με μια ενιαία γραμμή (αποκαλούμενη **ένωση**), ως εξής:



"διευθυντής" και "αυτοκίνητο επιχείρησης" σχετίζονται από την ένωση

Δύο σημαντικά πράγματα που σημειώνουμε για αυτήν την ένωση. Καταρχήν, η ένωση έχει το περιγραφικό της όνομα (σε αυτήν την περίπτωση "drives"). Αφετέρου, υπάρχουν αριθμοί σε κάθε τέλος της ένωσης. Αυτοί οι αριθμοί περιγράφουν τον **αριθμό στοιχείων συνόλου** της ένωσης, και μας λένε πόσες περιπτώσεις κάθε έννοιας επιτρέπονται.

Στο προηγούμενο παράδειγμα, λέγαμε ότι "κάθε διευθυντής έχει 1 αυτοκίνητο επιχείρησης" και (πηγαίνοντας από το αριστερό δικαίωμα)"κάθε αυτοκίνητο επιχείρησης οδηγείται από 1 διευθυντή".



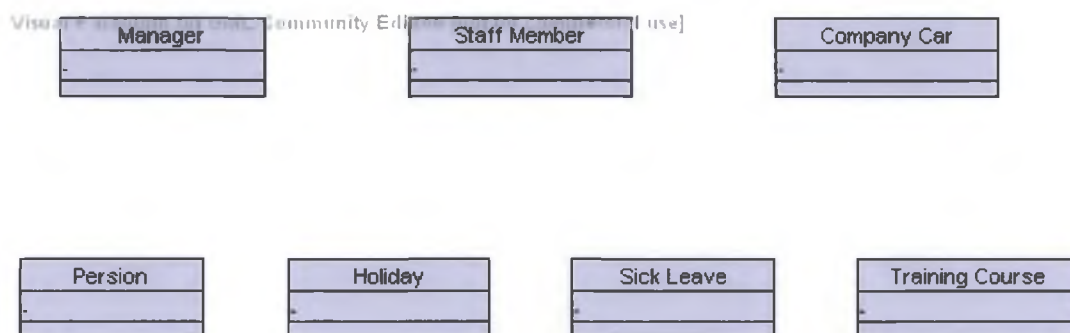
ένα άλλο παράδειγμα ένωσης

Στο ανωτέρω παράδειγμα, βλέπουμε ότι "κάθε διευθυντής διαχειρίζεται 1 ή περισσότερα μέλη προσωπικού" και (πηγαίνοντας στον άλλο τρόπο), "κάθε μέλος προσωπικού ρυθμίζεται από 1 διευθυντή". Κάθε ένωση πρέπει να εργάζεται κατ' αυτό το τρόπο. Κατά την απόφαση μας για το τι όνομα θα δώσουμε στην ένωση, θα πρέπει να αποφεύγουμε τα αδύνατα ονόματα όπως "έχει" ή "είναι συνδεδεμένος", στη χρησιμοποίηση τέτοιας αδύνατης γλώσσας θα μπορούσε εύκολα να κρύψει τα

προβλήματα ή τα λάθη, αυτό ειδάλλως θα είχε αποκαλυφθεί εάν το όνομα ένωσης ήταν σημαντικότερο.

Οικοδόμηση του πλήρους προτύπου

Τέλος, εξετάζουμε ένα μεθοδικό σύστημα για τις ενώσεις μεταξύ των εννοιών. Υποθέστε ότι έχουμε ολοκληρώσει τη σύνοδο καταιγισμού ιδεών (brainstorming) και έχουμε αποκαλύψει αρκετές έννοιες για το σύστημα διαχείρισης προσωπικού. Το σύνολο των εννοιών φαίνεται στο παρακάτω σχήμα (έχουν παραλειφθεί οι ιδιότητες για καλύτερη σαφήνεια).



σύνολο εννοιών για τη διαχείριση προσωπικού

Ο καλύτερος τρόπος να προχωρήσει είναι "να καθοριστεί" μια έννοια, να ειπωθεί "ο διευθυντής" και να θεωρήσει κάθε άλλη έννοια στη συνέχεια. Ρωτηθήκατε αν αυτές οι δύο έννοιες είναι σχετικές; και σε αυτή την περίπτωση, όμως θα πρέπει να αποφασιστεί το όνομα της ένωσης, και ο αριθμός των στοιχείων του συνόλου...

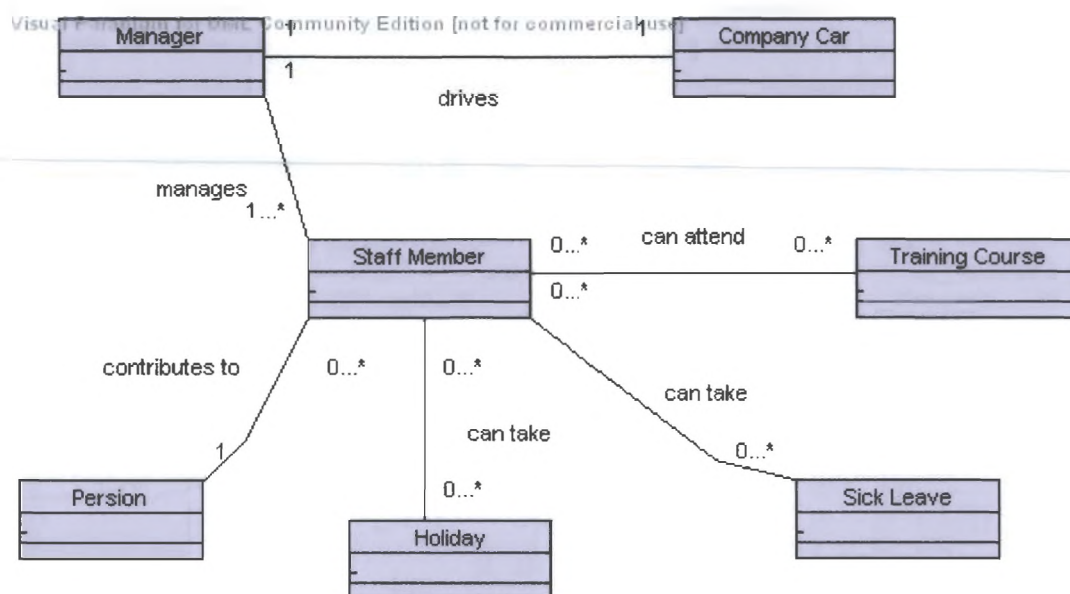
« Έχει σχέση ο **διευθυντής** με το **προσωπικό**; Ναι γιατί κάθε διευθυντής μπορεί να διαχειριστεί περισσότερα από 1 μέλη του προσωπικού.

Διευθυντής και **Αυτοκίνητο Επιχείρησης**; Ναι γιατί κάθε διευθυντής μπορεί να οδηγήσει περισσότερα από 1 αυτοκίνητα που ανήκουν στην επιχείρηση.

Διευθυντής και **σύνταξη**; Ναι γιατί κάθε διευθυντής δικαιούται μια σύνταξη. »

Ένα συνηθισμένο λάθος που γίνεται σε αυτή τη φάση είναι η απόφαση για το αν δύο έννοιες συσχετίζονται, τραβώντας μια γραμμή στο διάγραμμα, αφήνοντας το όνομα της ένωσης για αργότερα.

Αυτό θα μας φέρει πρόσθετη δουλειά καθώς θα το διαπιστώσουμε μόλις έχουμε τελειώσει με τη προσθήκη γραμμών και δεν θα έχουν καμία ιδέα για το τι θα σημαίνουν και θα πρέπει να τα αρχίσουμε ξανά!



το απλό εννοιολογικό πρότυπο, που ολοκληρώνεται

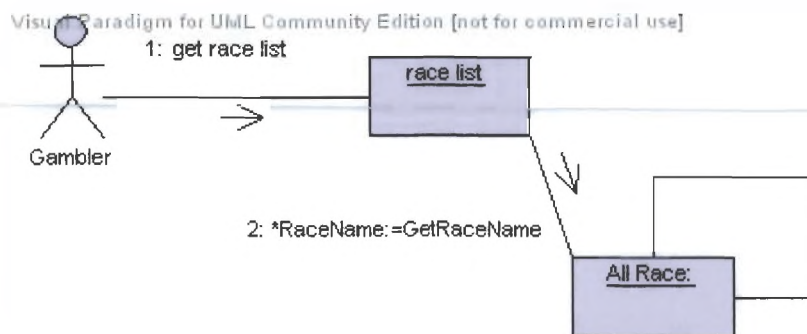
Κατά την οικοδόμηση του προτύπου, είναι σημαντικό να αναφερθεί ότι οι ενώσεις είναι λιγότερο σημαντικές από τις ιδιότητες. Οποιοσδήποτε ελλείπουσες ενώσεις θα συμπληρωθούν εύκολα κατά τη διάρκεια του σχεδιασμού, αλλά αυτό που είναι οδυνηρότερο είναι να υπάρχουν ελλείπουσες ιδιότητες.

Επιπλέον, σύγχυση μπορεί να φέρει και ένα σύνθετο διάγραμμα. Έτσι μια καλή τεχνική μέθοδος είναι να επικεντρωθούμε στις έννοιες και στις ιδιότητες, και να προσπαθήσουμε να καθορίσουμε τις προφανείς ενώσεις.

Στο τέλος της διαμόρφωσης, το διάγραμμα θα πρέπει να κάνει το πελάτη να αισθάνεται οικείος μαζί του και να το κατανοεί απόλυτα.

2.3 Διαγράμματα συνεργασίας (Collaboration Diagrams)

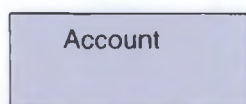
Σε αυτό το τμήμα, θα εξετάσουμε τη σύνταξη του διαγράμματος συνεργασίας της UML. Ένα διάγραμμα συνεργασίας μας επιτρέπει να παρουσιάσουμε τις αλληλεπιδράσεις μεταξύ των αντικειμένων κατά τη διάρκεια του χρόνου. Εδώ είναι ένα παράδειγμα ενός ολοκληρωμένου διαγράμματος συνεργασίας:



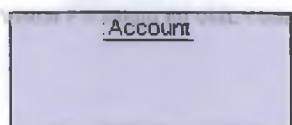
διάγραμμα συνεργασίας

2.3.1 Σύνταξη συνεργασίας: Τα βασικά

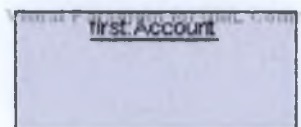
Μια κατηγορία στο διάγραμμα συνεργασίας είναι ως εξής:



Μια περίπτωση μιας κατηγορίας (με άλλα λόγια, ένα αντικείμενο) είναι ως εξής:



Μερικές φορές, θα το βρούμε χρήσιμο να ονομάσουμε μια περίπτωση μιας κατηγορίας. Στον ακόλουθο παράδειγμα, παίρνω ένα αντικείμενο από την κατηγορία λογαριασμού, και θέλω να το καλέσω "first":



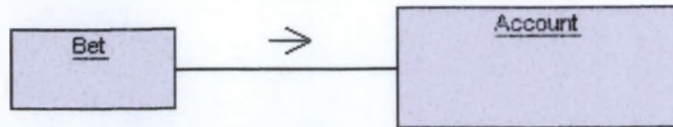
Εάν θέλουμε ένα αντικείμενο να επικοινωνήσει με ένα άλλο αντικείμενο, τότε συνδέουμε τα δύο αντικείμενα, με μια γραμμή. Στο ακόλουθο παράδειγμα, θέλω το πρώτο αντικείμενο "στοίχημα"(bet) να επικοινωνήσει με ένα αντικείμενο "λογαριασμού" (account):



Μόλις έχουμε συνδέσει ένα αντικείμενο για να μπορεί να επικοινωνήσει με ένα άλλο, τότε μπορούμε να στείλουμε ένα ονομασμένο μήνυμα από το ένα αντικείμενο στο άλλο. Εδώ, το αντικείμενο "στοίχημα" στέλνει ένα μήνυμα στο αντικείμενο "λογαριασμού", που το λέει χρέωση:



Εάν θέλουμε να περάσουμε παραμέτρους με το μήνυμα, τότε μπορούμε να τις περιλάβουμε τις παραμέτρους μέσα σε παρενθέσεις ως εξής. Παραδείγματος χάριν ο τύπος δεδομένου της παραμέτρου (σε αυτήν την περίπτωση, θα χρησιμοποιήσουμε μια κατηγορία αποκαλούμενη "Money") μπορεί να παρουσιαστεί, προαιρετικά.

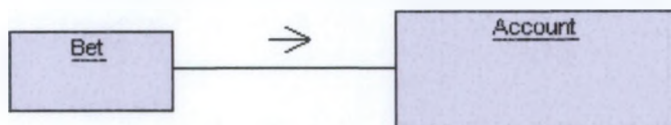


1: debitAccount(stake:Money)

Ένα μήνυμα μπορεί να επιστρέψει μια αξία (ανάλογη σε μια κλήση λειτουργίας στο στάδιο προγραμματισμού). Η ακόλουθη σύνταξη συστήνεται στα πρότυπα UML εάν στοχεύετε για το ουδέτερο γλωσσικό σχεδιασμό. Εντούτοις, εάν έχουμε μια συγκεκριμένη γλώσσα στο μυαλό μας, μπορούμε να προσαρμόσουμε τη σύνταξη της ώστε να ταιριάζει με την γλώσσα της προτίμησής μας. Παραδείγματος χάριν:

`return:=message (parameter : parameterType): returnType`

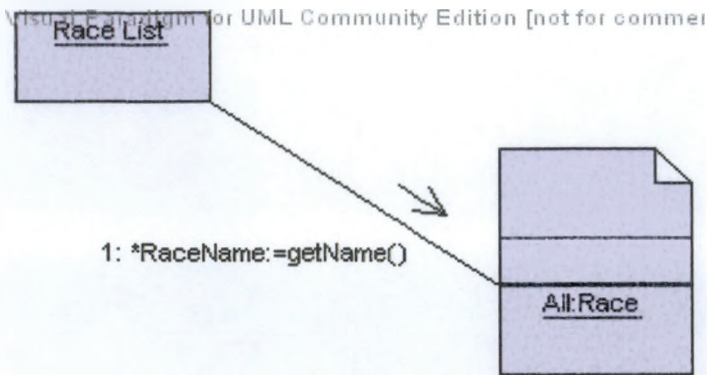
Στο ακόλουθο παράδειγμα, το αντικείμενο του στοιχήματος πρέπει να ξέρει την ισορροπία ενός ιδιαίτερου λογαριασμού. Το μήνυμα "getBalance" στέλνεται, και το αντικείμενο λογαριασμού επιστρέφει έναν ακέραιο αριθμό:



1: balance:=getBalance():Integer

2.3.2 Επανάληψη (Looping)

Εάν πρέπει να εισαγάγουμε έναν βρόχο σε ένα διάγραμμα συνεργασίας, χρησιμοποιούμε την ακόλουθη σύνταξη. Σε αυτό το παράδειγμα, ένα αντικείμενο από την κατηγορία "λίστα αγώνων" πρέπει να συγκεντρωθεί. Για να το κάνει αυτό, ζητά από κάθε μέλος της κατηγορίας "αγώνων" να επιστρέψει το όνομά του.



Ο αστερίσκος δείχνει ότι το μήνυμα πρόκειται να επαναληφθεί. Παρά τη διευκρίνιση για το μεμονωμένο όνομα αντικειμένου, έχουμε χρησιμοποιήσει το όνομα "όλοι" (All) που δείχνουν ότι θα πηγαίνουν επαναλαμβανόμενα σε όλα τα αντικείμενα. Τέλος, έχουμε χρησιμοποιήσει τη σημείωση της UML για τη πρώτη συλλογή των αντικειμένων με τη "συσσώρευση" των κιβωτίων αντικειμένου.

2.3.3 Δημιουργία των νέων αντικειμένων

Μερικές φορές, ένα αντικείμενο θα θελήσει να δημιουργήσει μια νέα περίπτωση ενός άλλου αντικειμένου. Η μέθοδος για αυτό ποικίλλει μεταξύ των γλωσσών, έτσι η UML τυποποιεί τη δημιουργία μέσω της ακόλουθης σύνταξης:



Η σύνταξη είναι μάλλον παράξενη, πραγματικά στέλνετε ένα μήνυμα αποκαλούμενο "Create" από ένα αντικείμενο που δεν υπάρχει ακόμα!

Αρίθμηση μηνυμάτων

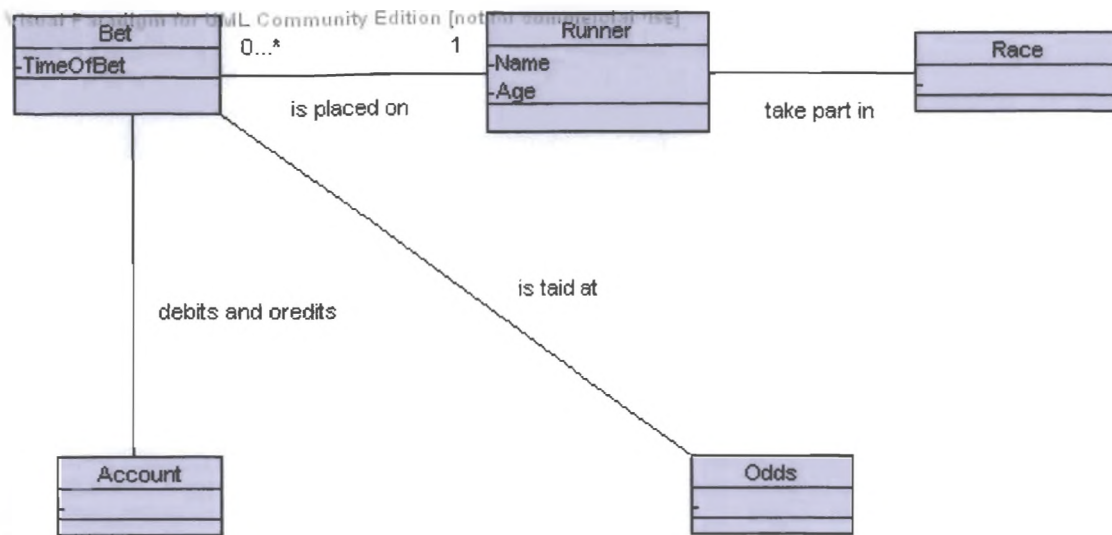
Οι σημειώσεις σε όλα τα μηνύματα που έχουμε περιλαμβάνουν παράλληλά τους ένα μυστήριο "1". Αυτό δείχνει τη διαταγή στην οποία το μήνυμα εκτελείται προκειμένου να ικανοποιήσει την περίπτωση χρήσης. Δεδομένου ότι προσθέτουμε περισσότερα μηνύματα, διαδοχικά αυξάνετε και ο αριθμός των μηνυμάτων. Στην πραγματικότητα, στέλνετε ένα μήνυμα κατηγορίας, που στις περισσότερες γλώσσες προγραμματισμού καλείτε επίσης και ως κατασκευαστής.

2.3.4 Παράδειγμα

Αφήνουμε όλη τη θεωρία στην άκρη προς το παρόν και ας δούμε πως αυτά λειτουργούν στη πράξη με το παράδειγμα του στοιχήματος, δημιουργώντας ένα διάγραμμα συνεργασίας (collaboration diagram) και χρησιμοποιώντας ένα διάγραμμα περίπτωσης χρήσης (class diagram). Αυτό το παράδειγμα είναι κάθε άλλο παρά ιδανικό, και αφήνει πολλές ερωτήσεις αναπάντητες. Εντούτοις, το παράδειγμα επεξηγεί πώς χτίζονται τα διαγράμματα συνεργασίας.

Για να χτίσουμε αυτό το διάγραμμα, χρειαζόμαστε μερικά αντικείμενα. Από πού όμως θα πάρουμε τα αντικείμενα; Σίγουρα, θα πρέπει να εφεύρουμε μερικά νέα αντικείμενα καθώς πηγαίνουμε εμπρός, αλλά τα πιο πολλά από τα υποψήφια αντικείμενα πρέπει να προέλθουν άμεσα από τον παλαιό φίλο μας, το εννοιολογικό πρότυπο που χτίσαμε στη φάση επεξεργασίας.

Εδώ παρουσιάζετε το εννοιολογικό πρότυπο για το σύστημα στοιχήματος:



σύστημα στοιχήματος, εννοιολογικό πρότυπο

Όπου υπάρχουν ενώσεις, όπως "is placed on", θα τις χρησιμοποιήσουμε πιθανώς αυτές τις ενώσεις για να περάσουμε τα μηνύματα στο διάγραμμα συνεργασίας. Θα μπορούσαμε ενδεχομένως να αποφασίσουμε ότι πρέπει (παραδείγματος χάριν) να περάσουμε ένα μήνυμα μεταξύ του "Account" και του "Race". Αυτό είναι τελείως έγκυρο, αλλά ως ένωση δεν αποκαλύφθηκε στο εννοιολογικό στάδιο, εμείς πιθανόν να έχουμε τη δυνατότητα να σπάμε μερικές από τις απαιτήσεις του πελάτη. Εάν αυτό συμβαίνει, τότε ελέγχουμε το πελάτη!

Με τη περιγραφή της περίπτωσης χρήσης και το εννοιολογικό πρότυπο στο μυαλό μας, ξεκινάμε το χτίσιμο του διαγράμματος συνεργασίας για "τη θέση του στοιχήματος".

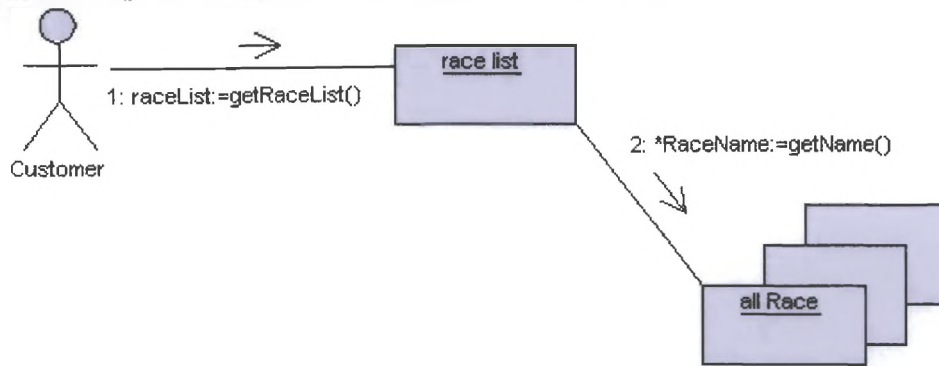
1. Καταρχήν, αρχίζουμε με τον αρχικό δράστη, το πελάτη. Το σύμβολο δραστή δεν αποτελεί μόνο μέρος του διαγράμματος συνεργασίας στη UML, αλλά αυτό είναι εξαιρετικά χρήσιμο και θα το περιλάβουμε στο διάγραμμα οπωσδήποτε.



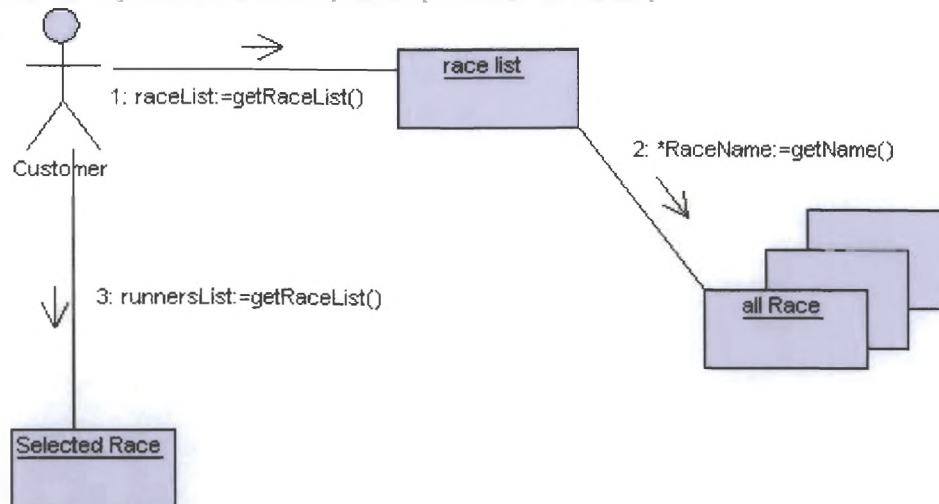
2. Τώρα, όταν ο πελάτης διαλέξει την επιλογή "θέση στοιχήματος", μια επιλογή των φυλών του παρουσιάζεται. Έτσι θα χρειαστούμε ένα αντικείμενο που να περιέχει έναν πλήρη κατάλογο των φυλών για τη σημερινή μέρα, έτσι δημιουργούμε ένα αντικείμενο αποκαλούμενο " Race List"(λίστα αγώνων) . Αυτό είναι ένα αντικείμενο που δεν αντιπροσωπεύθηκε στον εννοιολογικό πρότυπο. Αυτό καλείται *κατηγορία σχεδίου(design class)*.



3. Έτσι, ο δράστης στέλνει ένα μήνυμα στο νέο αντικείμενο "Race List" αποκαλούμενο "getRaceList". Τώρα, η επόμενη εργασία είναι να συγκεντρωθεί ο κατάλογος Race List. Αυτό γίνεται με μια δομή επανάληψης (looping) για κάθε αντικείμενο race ακολουθούμενο από μια ερώτηση για το τι όνομά έχει. Το αντικείμενο race έχει ληφθεί από το εννοιολογικό πρότυπο.

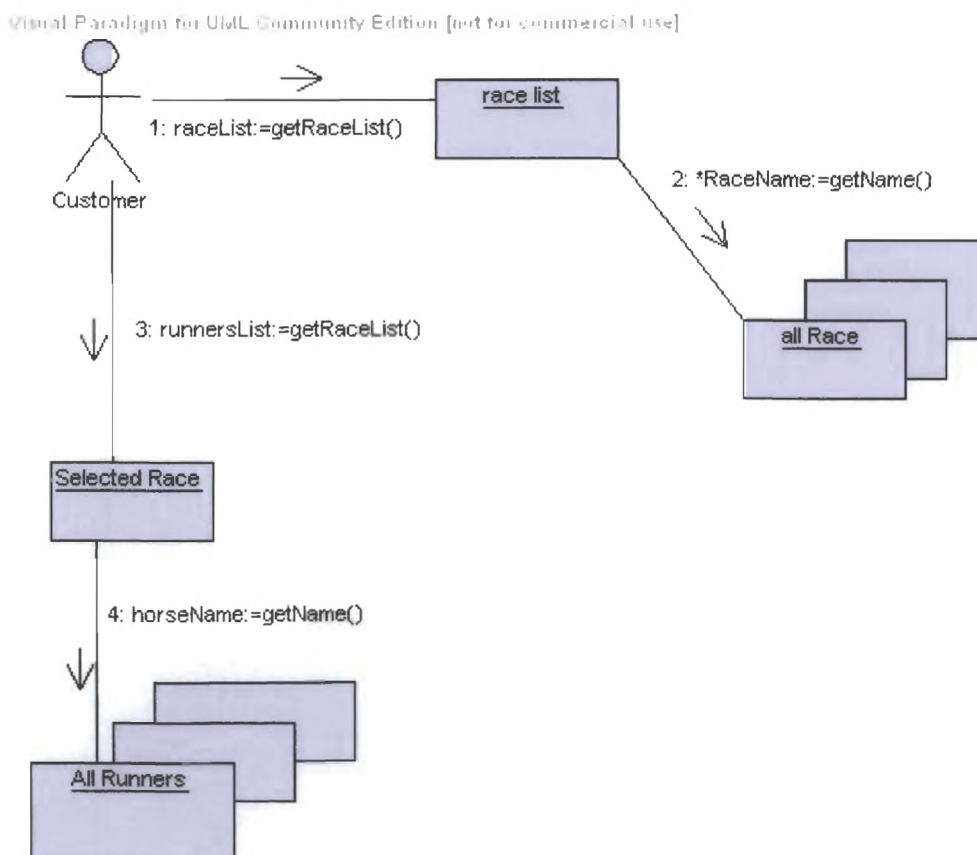


4. Έπειτα, υποθέτουμε ότι ο κατάλογος race list επιστρέφεται πίσω στον πελάτη. Τώρα ο πελάτης έχει τη δυνατότητα να επιλέξει ένα race από το κατάλογο race list. Μπορούμε τώρα να υποθέσουμε ότι ένας αγώνας(race) έχει επιλεγεί. Χρειάζεται τώρα να πάρουμε έναν κατάλογο για τους δρομείς (runners) για κάθε επιλεγμένο αγώνα(race), και να αποφασίσει για ποιόν αγώνα θέλει να κάνει κράτηση και για ποιο δρομέα από τη λίστα των δρομέων.



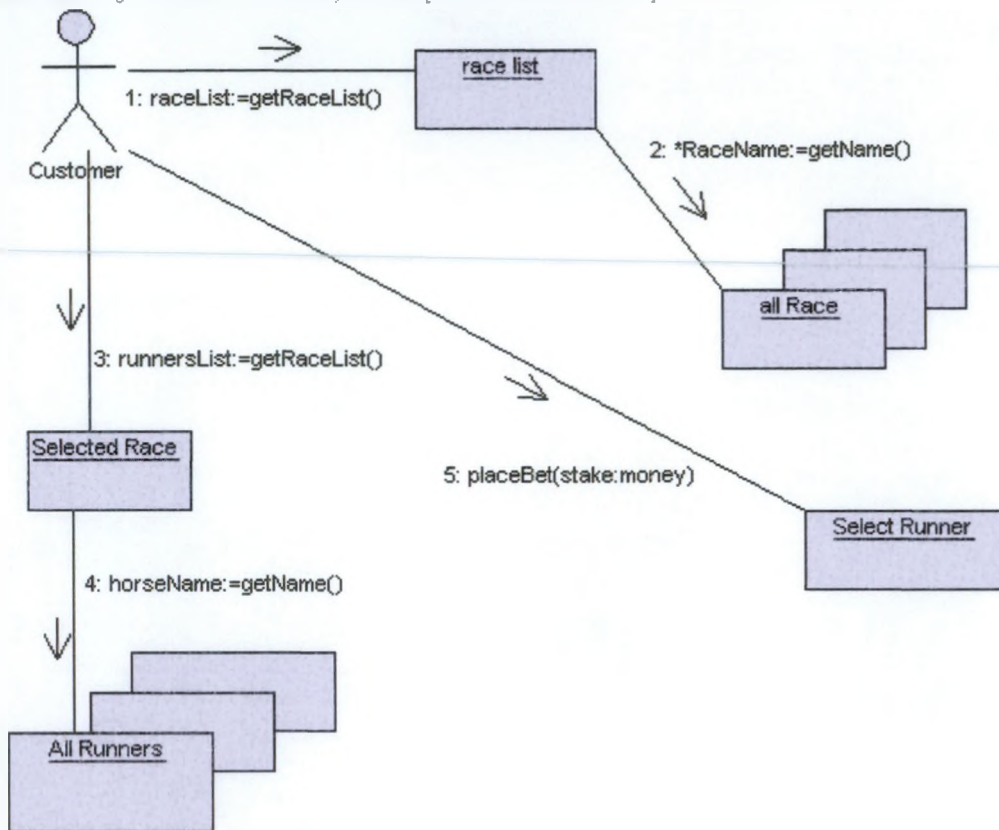
Έτσι, το μήνυμα 3 στέλνεται στον επιλεγμένο αγώνα, και ρωτά ποιο δρομέα θέλει για εκείνο τον αγώνα.

5. Πώς το αντικείμενο αγώνων ξέρει ποιοι δρομείς παίρνουν μέρος σε εκείνο τον αγώνα; Για άλλη μια φορά, θα το κάνουμε αυτό χρησιμοποιώντας έναν βρόχο και θα πάρουμε το αντικείμενο αγώνων για να το συγκεντρώσουμε σε έναν κατάλογο δρομέων. Το πώς αυτό επιτυγχάνεται πραγματικά στο πραγματικό σύστημα δεν είναι τετριμμένο. Σαφώς, πηγαίνουμε να αποθηκεύσουμε αυτούς τους δρομείς σε μια κάποια είδους βάση δεδομένων, έτσι μέρος από τη διαδικασία φυσικού σχεδιασμού πρόκειται να κατασκευάσει ένα μηχανισμό για την ενημέρωση των αρχείων των αλόγων από τη βάση δεδομένων. Για τώρα, εντούτοις, είναι αρκετό να πει ότι το επιλεγμένο αντικείμενο αγώνων είναι αρμόδιο για τη συγκέντρωση ενός καταλόγου δρομέων για αυτόν τον αγώνα.



6. Ο κατάλογος δρομέων επιστρέφεται τώρα πίσω στον πελάτη. Τώρα ο πελάτης θα πρέπει να επιλέξει ένα δρομέα και να στοιχηματίσει τα χρήματά του. Τώρα που ξέρουμε το δρομέα και το στοίχημα, μπορούμε να στείλουμε ένα μήνυμα στον επιλεγμένο δρομέα και να τον ενημερώνει ότι ένα στοίχημα έχει τοποθετηθεί επάνω του.

Visual Paradigm for UML Community Edition [not for commercial use]



Με την οικοδόμηση αυτού του διαγράμματος συνεργασίας, δεν έχουμε χαράξει ακριβώς πως θα μοιάζει ο κώδικας. Τα ακόλουθα ζητήματα εκκρεμούν:

1. Δεν έχουμε αναφέρει τίποτα στο διάγραμμα για το πώς γίνονται οι εισαγωγές των στοιχείων των χρηστών στο σύστημα και πώς τα στοιχεία (όπως ο κατάλογος δρομέων) εμφανίζονται στην οθόνη. Όλο αυτό φαίνεται να συμβαίνει από κάποιο μαγικό "εσωτερικό" δράστης.

2. Πώς το "Race" αντικείμενο βρίσκει ποιοι δρομείς παίρνουν μέρος σε εκείνο τον αγώνα;
Σαφώς, υπάρχει κάποια λειτουργία των βάσεων δεδομένων (ή ακόμα και των δικτύων) που λειτουργεί εδώ.

2.3.5 Μερικές οδηγίες για τα διαγράμματα συνεργασίας

1. Κρατήστε το διάγραμμα απλό!!! Ο καλύτερος κανόνας για να ισχύει και για το διάγραμμα συνεργασίας (και για τα άλλα διαγράμματα UML) είναι να τα κρατήσουν όσο το δυνατόν απλούστερα. Εάν η συνεργασία(collaboration) για μια περίπτωση χρήσης(use case) είναι περίπλοκη, τότε το χωρίζουμε. Ίσως να παράγετε ένα χωριστό διάγραμμα για κάθε αλληλεπίδραση των χρηστών και των συστημάτων.

2. Μην προσπαθήσετε να συλλάβετε κάθε σενάριο. Κάθε περίπτωση χρήσης περιλαμβάνει διάφορα διαφορετικά σενάρια (τη κύρια ροή, διάφορες εναλλακτικές λύσεις και διάφορες εξαιρέσεις). Συνήθως, οι εναλλακτικές λύσεις είναι αρκετά τετριμμένες και όχι τόσο πραγματικής αξίας. Ένα κοινό λάθος είναι να φορτώνουμε κάθε σενάριο σε ένα διάγραμμα, με αποτέλεσμα να κάνει το διάγραμμα σύνθετο και δύσκολο για να κωδικοποιηθεί.

3. Αποφύγετε να δημιουργείτε κατηγορίες των οποίων το όνομα περιέχει "ελεγκτής, χειριστής, διευθυντής ή οδηγός". Να είστε προσεκτικοί εάν βρίσκετε κάποιο αντικείμενο με ένα τέτοιο όνομα. Γιατί;

Αυτές οι κατηγορίες τείνουν να σας προτείνουν ένα σχέδιο το οποίο δεν είναι προσανατολισμένο στο αντικείμενο. Παραδείγματος χάριν, στη περίπτωση χρήσης "θέση στοιχήματος", θα μπορούσαμε να δημιουργήσουμε μια κατηγορία αποκαλούμενη "BetHandler" που να εξετάζει όλη τη λειτουργία του στοιχήματος. Αλλά αυτό θα ήταν μια λύση προσανατολισμένη σε μια πράξη παρά σε ένα αντικείμενο.

4. Αποφύγετε τις κύριες κατηγορίες (God Classes).

Εάν καταλήγετε με ένα ενιαίο αντικείμενο που κάνει ένα μέρος της εργασίας και δεν συνεργάζεται πολύ με τα άλλα αντικείμενα, τότε πιθανώς έχετε φτιάξει μια λύση προσανατολισμένη στη πράξη. Οι καλές λύσεις προσανατολισμένου αντικειμένου αποτελούνται από μικρά αντικείμενα που δεν κάνουν πάρα πολλή δουλειά από μόνα τους, αλλά συνεργάζονται με άλλα αντικείμενα για να επιτύχουν το στόχο τους.

2.4 Διάγραμμα ακολουθίας (Sequence Diagram)

Η παραγωγή των περιγραφών περίπτωσης χρήσης είναι δύσκολη. Πολλοί άνθρωποι δύσκολα βρίσκουν τη διάκριση μεταξύ της ανάλυσης και του σχεδιασμού, συχνά οι περιγραφές των περιπτώσεων χρήσης πηγαίνουν στα 'σκουπίδια' κατά τις αποφάσεις του σχεδιασμού.

Εδώ είναι ένα παράδειγμα από την περίπτωση χρήσης του στοιχήματος θέσεων:

"Ο χρήστης επιλέγει τον αγώνα που θέλει να στοιχηματίσει. Το σύστημα ρωτά τη βάση δεδομένων για τον αγώνα και συντάσσει μια λίστα δρομέων για το συγκεκριμένο αγώνα. "

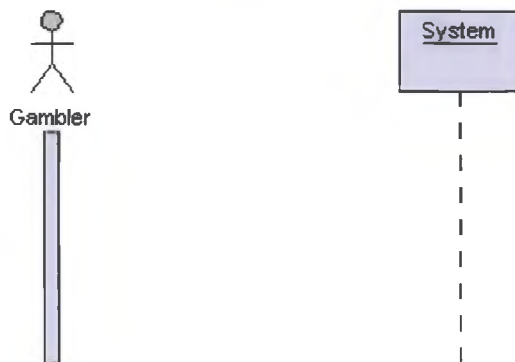
Αυτό είναι μια απλή περιγραφή περίπτωσης χρήσης. Κατά την οικοδόμηση των περιπτώσεων χρήσης, πρέπει να μεταχειριστούμε το σύστημα ως "μαύρο κιβώτιο", το οποίο μπορεί δέχεται τα αιτήματα από τους δράστες και να επιστρέφει τα αποτελέσματα στο δράστη. Δεν μας ενδιαφέρει (ακόμα) με ποιο τρόπο το μαύρο κιβώτιο εκπληρώνει εκείνο το αίτημα. Έτσι προτείνουμε τη χρήση ενός **διαγράμματος ακολουθίας (Sequence Diagram)** της UML.

Ένα διάγραμμα ακολουθίας είναι χρήσιμο σε πολλές διαφορετικές καταστάσεις, ειδικά στο στάδιο σχεδιασμού. Εντούτοις, το διάγραμμα μπορεί να χρησιμοποιηθεί και στην ανάλυση για να μας βοηθήσει με την ανάλυση των "μαύρων κιβωτίων" στο σύστημα. Εδώ θα δούμε πώς το διάγραμμα λειτουργεί:

1

Τοποθετούμε το δράστη αριστερά του διαγράμματος, και στα δεξιά του διαγράμματος, παρουσιάζουμε το σύστημα ολόκληρο σαν ένα ενιαίο κιβώτιο. Έπειτα, προσθέτουμε τις κατακόρυφες όπου είναι "υποδείξεις ως προς το χρόνο". Οι γραμμές αντιπροσωπεύουν τη μετάβαση στο χρόνο από πάνω προς τα κάτω.

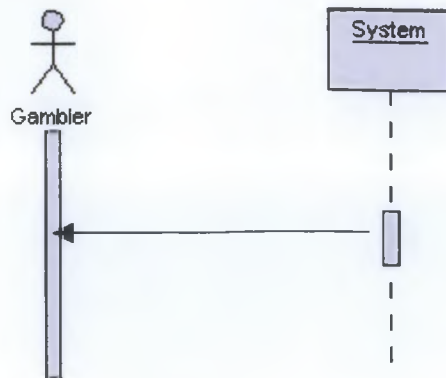
Visual Paradigm for UML Community Visual Paradigm for UML Community



2

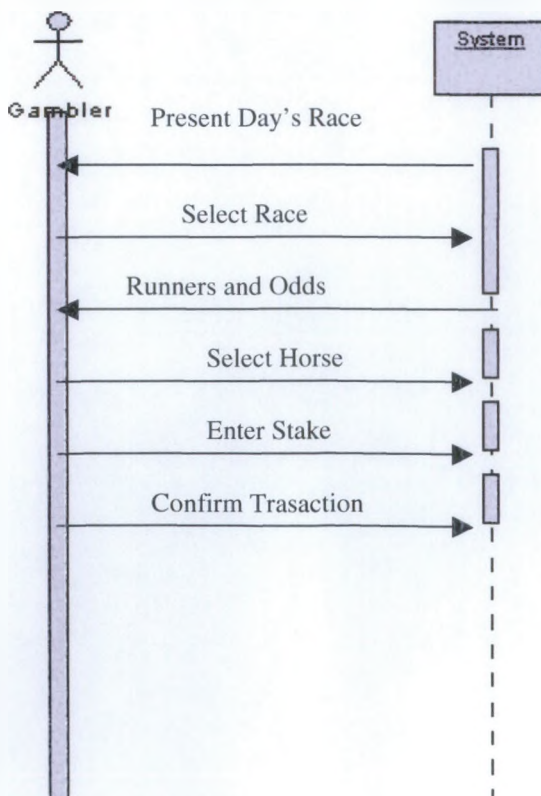
Αλληλεπιδράσεις μεταξύ του χρήστη και του συστήματος αντιπροσωπεύονται ως μια γραμμή με βέλος μεταξύ του συστήματος και του δράστη. Μια περιγραφή αλληλεπίδρασης γράφεται παράλληλα με το βέλος.





4

Συνεχίζουμε τη προσθήκη των αλληλεπιδράσεων προς τα κάτω όπου δηλώνετε ο χρόνος. Τα μακριά κιβώτια κάτω δείχνουν το χρόνο όταν το σύστημα και ο δράστης είναι "ενεργά". Αυτή η σημείωση είναι σημαντικότερη όταν εμείς θα τοποθετήσουμε τα διαγράμματα ακολουθίας μέσα στο σχεδιασμό, για τώρα, αυτό δεν μας ενδιαφέρει πραγματικά.



Μόλις το διάγραμμα ακολουθίας συστημάτων είναι πλήρες, είναι αρκετά απλό να γραφτεί η περιγραφή της κύριας ροής για την περίπτωση χρήσης. Δεν υπάρχει καμία ανάγκη να σύρουμε για κάθε διάγραμμα μια ενιαία ροή εναλλαγής και εξαίρεσης, αν και θα ήταν σημαντικό για τις πολύ περίπλοκες ή ενδιαφέρουσες εναλλακτικές λύσεις.

Τέλος το διάγραμμα ακολουθίας συσχετίζεται, στην πραγματικότητα, άμεσα με το διάγραμμα συνεργασίας και επιδεικνύει τις ίδιες πληροφορίες, αλλά σε μια ελαφρώς διαφορετική μορφή.

2.5 Λίγα λόγια για τα υπόλοιπα διαγράμματα της μεθόδου UML

1. Διάγραμμα κρατικής μετάβασης (State diagram)

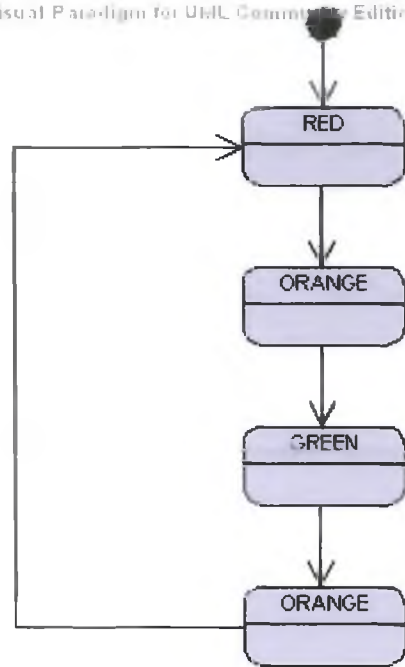
Μερικά αντικείμενα μπορούν, σε οποιοδήποτε ιδιαίτερο χρόνο, να είναι σε ένα ορισμένο *κράτος*. Παραδείγματος χάριν, σε μια οδική κυκλοφορία το φανάρι μπορεί να είναι σε οποιοδήποτε από τα ακόλουθα κράτη: **κόκκινο, πορτοκαλί και πράσινο**.

Μερικές φορές, η ακολουθία μεταβάσεων μεταξύ των κρατών μπορεί να είναι αρκετά σύνθετη, στο επάνω παράδειγμα, δεν θα θέλαμε να είμαστε σε θέση να πάμε άμεσα από το "πράσινο" στο "Κόκκινο" γιατί θα προκαλούσαμε τα ατυχήματα.

Αν και το φως κυκλοφορίας μπορεί να φανεί ως ένα τετριμμένο παράδειγμα, σκεφτείτε τα σοβαρά και ενοχλητικά προβλήματα που μπορούν να εμφανιστούν στο λογισμικό αν υπάρξει αυθαιρεσία μεταξύ των κρατών.

Παίρνουμε μια περίπτωση όπου ένας λογαριασμός ρεύματος στέλνεται σε έναν πελάτη που πέθανε τέσσερα έτη πριν, το οποίο μπορεί να συμβεί πραγματικά επειδή ένας προγραμματιστής κάπου δεν έχει φτιάξει με προσοχή τις κρατικές μεταβάσεις.

Όπως οι κρατικές μεταβάσεις μπορούν να είναι αρκετά σύνθετες, για αυτό η UML παρέχει μια σύνταξη για να μας επιτρέψει τη διαμόρφωση τους.



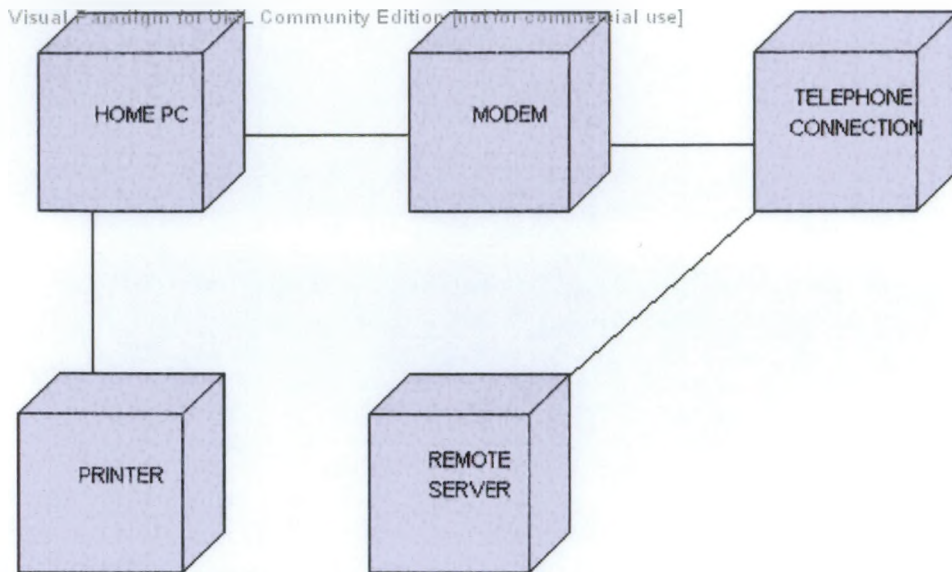
ΠΑΡΑΔΕΙΓΜΑ STATE DIAGRAM

2. Διάγραμμα συσκευασίας (Package diagram)

Οποιοδήποτε σημαντικό σύστημα πρέπει να κατανεμηθεί σε μικρότερα και ευκολότερα στη κατανόηση "κομμάτια". Το διάγραμμα συσκευασίας της UML επιτρέπει σε μας να το διαμορφώσουμε αυτό με έναν απλό και αποτελεσματικό τρόπο. Τέλος το διάγραμμα συσκευασίας χρίζεται ως το καλύτερο εργαλείο για την "αρχιτεκτονική συστημάτων".

3. Συστατικά διαγράμματα (Component diagrams)

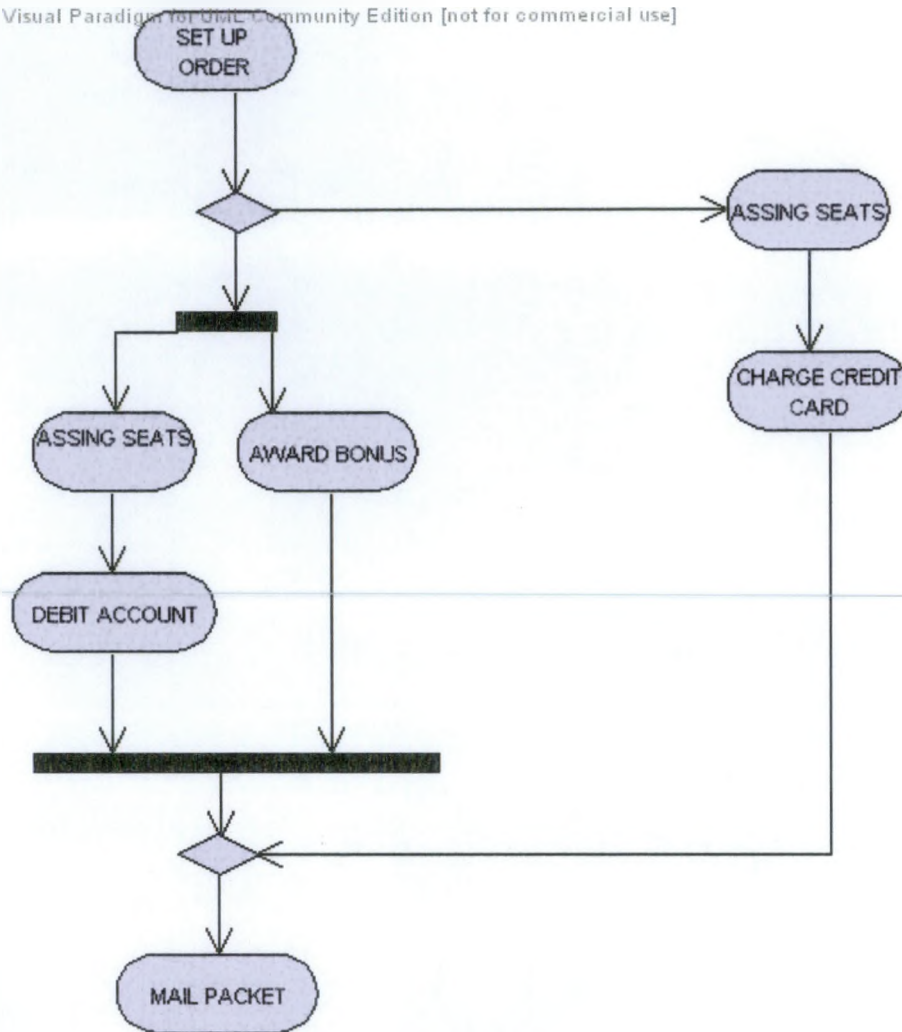
Το συστατικό διάγραμμα είναι παρόμοιο με το διάγραμμα συσκευασίας - μας επιτρέπει να δούμε πώς το σύστημά μας είναι χωρισμένο, και ποιες είναι οι εξαρτήσεις μεταξύ κάθε ενότητας.



ΠΑΡΑΔΕΙΓΜΑ DEPLOYMENT DIAGRAM

5. Διάγραμμα δραστηριότητας (Activity diagram)

Ένα διάγραμμα δραστηριότητας μοντελοποιεί τη ροή της εργασίας, αναπαριστώντας τις διάφορες καταστάσεις εκτέλεσης ενός λογισμικού. Παρουσιάζει τη ροή ελέγχου μεταξύ των δραστηριοτήτων του ίδιου αντικειμένου ή πολλών αντικειμένων. Μια κατάσταση δραστηριότητας συμβολίζεται ως ένα ορθογώνιο με καμπύλες γωνίες με μια περιγραφή της συγκεκριμένης δραστηριότητας. Η μετάβαση κατά τη συμπλήρωση μιας δραστηριότητας συμβολίζεται ακμή, οι διακλαδώσεις συμβολίζονται είτε με συνθήκες φρουρούς επί των μεταβάσεων είτε με κόμβους απόφασης (ρόμβους) με πολλαπλές ακμές. Τέλος μια ένωση συμβολίζει συνένωση πολλών εισερχόμενων μεταβάσεων σε μια εξερχόμενη, ενώ μια διχάλα την ανάλυση μιας εισερχόμενης σε πολλές παράλληλες ταυτόχρονες.



ΠΑΡΑΔΕΙΓΜΑ ACTIVITY DIAGRAM

ΚΕΦΑΛΑΙΟ 3

ΠΑΡΟΥΣΙΑΣΗ ΤΗΣ ΜΕΘΟΔΟΥ ΜΟΝΤΕΛΟΠΟΙΗΣΗΣ IDEF

Εισαγωγικά στοιχεία για τα IDEF

Η επιχειρηματική ανάλυση και ο σχεδιασμός μεθοδολογίας έχουν την ικανότητα να γεφυρώνουν το κενό μεταξύ των απαιτήσεων των συστημάτων πληροφόρησης και των συστημάτων εξέλιξης λογισμικού. Παραδοσιακά όμως οι επιχειρηματικοί αναλυτές έχουν δυσκολίες στο να συλλάβουν τα « as-is » μοντέλα επιχειρήσεων με σκοπό να τα κατανοήσουμε και να τα βελτιώσουμε. Ευτυχώς η άφιξη των εργαλείων σχεδίασης που υποστηρίζουν την εφαρμογή των μεθόδων επιχειρηματικών μοντέλων έχει αλλάξει την κατάσταση.

Αυτό το κεφάλαιο θα παρουσιάσει εν συντομία την ομάδα γλωσσών IDEF, με έμφαση στο IDEF0, IDEF1, IDEF1X, IDEF3, IDEF4 και IDEF5. Οι σημαντικοί κανόνες της συγκεκριμένης γλώσσας πρέπει να περιγραφούν με σκοπό να βοηθήσουν τον αναγνώστη να κατανοήσει τη συγκεκριμένη γλώσσα και τις εφαρμογές της.

Ιστορικά στοιχεία

Η IDEF (ICAM DEFINITION γίνεται INTEGRATED DEFINITION) μεθοδολογία αρχικά προοριζόταν για χρήση σε συστήματα μηχανολογίας. Τη δεκαετία του 1970 το σύστημα ανάλυσης και σχεδιασμού που κυριαρχούσε είχε ανάγκη από υποστηρικτικές μεθόδους σχεδιασμού. Η ICAM (Integrated Computer Aided Manufacturing) της πολεμικής αεροπορίας των Ηνωμένων Πολιτειών, κάλυψε αυτήν την ανάγκη με το να δημιουργήσει αυτή τη σειρά μεθόδων γνωστή ως IDEF.

Αρχικά η IDEF σειρά περιείχε μια λειτουργία μεθόδων σχεδιασμού ονομαζόμενη ως IDEF0 και μια ιδέα μεθόδου σχεδιασμού ονομαζόμενη IDEF1 και ένα μοντέλο προσομοίωσης σχεδιασμού το IDEF2. Απο τότε βελτιώσεις προστέθηκαν στο IDEF1 οι οποίες έδωσαν τη «γέννηση» του IDEF1X. Επίσης εργάστηκαν σε μια διαδικασία μεθόδου σχεδίασης γνωστή ως IDEF3 που ξεκίνησε στη δεκαετία του 1980. Το IDEF3 έχει στη πραγματικότητα αναπτυχθεί πολύ πριν από το IDEF2 επειδή μπορεί να χρησιμοποιηθεί για να προσδιορίζει προκαταρκτικά μοντέλα προσομοίωσης. Επιπλέον το IDEF3 έχει ένα συστατικό αντικείμενου βάσης που μπορεί να χρησιμοποιηθεί για να σχεδιάσει πως τα αντικείμενα αλλάζουν κατά τη διάρκεια της διαδικασίας.

Τα IDEF4 και IDEF5 ακολούθησαν τη δεκαετία του 1990. Το IDEF4 είναι ένα αντικειμενοστραφή λογισμικό σχεδιασμού που ενσωματώνει τις απαιτήσεις οι οποίες καθορίζονται από άλλες μεθόδους. Επίσης το IDEF4 επιτρέπει να συλλάβει και να διαχειριστεί τις αρχές σχεδιασμού των αντικειμενοστραφή αντικειμένων. Το IDEF5 είναι μια μέθοδος απαίτησης γνώσης και μηχανολογικού σχεδιασμού που στοχεύει να υποστηρίξει επιχειρηματικές οντολογίες (δηλαδή θεωρίες που υποστηρίζουν τα μετά-μοντέλα).

Ο πλήρης κατάλογος των μεθόδων IDEF κυμαίνεται από IDEF0 έως IDEF14. Στις μέρες μας οι μέθοδοι που χρησιμοποιούνται περισσότερο είναι IDEF0, IDEF1, IDEF1X, IDEF3 και IDEF4. Πρόσφατες εξελίξεις στοχεύουν ώστε αποτελεσματικά να βελτιώσουν και να ολοκληρώσουν αυτές τις μεθόδους έτσι ώστε εύκολα να γίνεται η ανταλλαγή πληροφοριών μεταξύ τους.

3.1 Γενικά για το IDEF0

Το IDEF0 είναι μια μέθοδος με σκοπό να διαμορφώσει τις αποφάσεις, τις ενέργειες, και τις δραστηριότητες μιας οργάνωσης ή ενός συστήματος. Το IDEF0 προήλθε από μια καθιερωμένη γραφική γλώσσα, της δομημένης ανάλυσης και την τεχνική σχεδίου (SADT). Η Ηνωμένη Πολεμική Αεροπορία ανάθεσε στους υπεύθυνους για την ανάπτυξη SADT να αναπτύξουν μια μέθοδο διαμόρφωσης λειτουργίας για τη λειτουργική προοπτική ενός συστήματος.

Τα αποτελεσματικά πρότυπα IDEF0 βοηθούν να οργανώσουν την ανάλυση ενός συστήματος και να προωθήσουν την καλή επικοινωνία μεταξύ του αναλυτή και του πελάτη. Το IDEF0 είναι χρήσιμο στην καθιέρωση του πεδίου μιας ανάλυσης, ειδικά για μια λειτουργική ανάλυση. Σαν εργαλείο επικοινωνίας, το IDEF0 ενισχύει την ειδική συμμετοχή περιοχών και τη λήψη αποφάσεων συναίνεσης μέσω των απλουστευμένων γραφικών συσκευών. Σαν εργαλείο ανάλυσης, το IDEF0 βοηθά το κατασκευαστή του μοντέλου στο προσδιορισμό ποιου λειτουργίες εκτελούνται, αυτό που απαιτείται για να εκτελέσει εκείνες τις λειτουργίες, τι το τρέχον σύστημα διορθώνει, και τι το τρέχον σύστημα βλάπτει. Κατά συνέπεια, τα πρότυπα IDEF0 δημιουργούνται συχνά ως ένα από τα πρώτα μέρη μιας προσπάθειας ανάπτυξης συστημάτων.

Τον Δεκέμβριο του 1993, το εργαστήριο συγκροτημάτων ηλεκτρονικών υπολογιστών Το εθνικό ίδρυμα προτύπων και η τεχνολογία (NIST) απελευθέρωσαν το IDEF0 ως πρότυπο για τη διαμόρφωση συναρτήσεων στο FIPS (δημοσίευση 183).

3.1.1 IDEF0 έννοιες

Η μέθοδος IDEF0 έχει τις βασικές έννοιες που καλύπτουν κάθε μια από τις ανάγκες που αναφέρθηκαν προηγουμένως. Οι βασικές έννοιες του IDEF0 περιλαμβάνουν τα ακόλουθα.

1. Κελί που διαμορφώνει τη γραφική αντιπροσώπευση

Η γραφική παράσταση "κιβωτίων και βελών" ενός διαγράμματος IDEF0 παρουσιάζει τη λειτουργία ως ένα κιβώτιο και τις διεπαφές προς ή από τη λειτουργία ως βέλη που εισάγονται ή που αφήνονται από το κουτί.

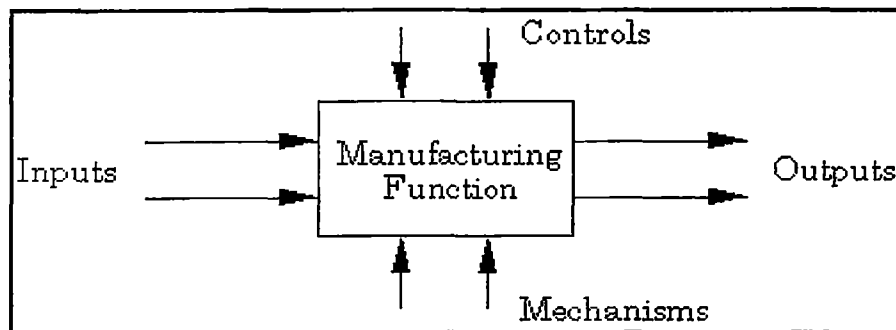
Για να εκφράσουν τις λειτουργίες, τα κιβώτια λειτουργούν ταυτόχρονα με άλλα κιβώτια, με τα βέλη διεπαφών τα οποία είναι "περιορισμένα" όταν προκαλούνται ελεγχόμενες διαδικασίες.

2. Επικοινωνία

Στο IDEF0 οι έννοιες που έχουν σκοπό να ενισχύσουν την επικοινωνία περιλαμβάνουν τα εξής:

- Διαγράμματα βασισμένα στην απλή γραφική παράσταση κιβωτίων και βελών.

Αγγλικές ετικέτες κειμένων για να περιγράψει τα κιβώτια και τα βέλη και το γλωσσάριο και το κείμενο για να καθορίσει τις ακριβείς έννοιες των στοιχείων των διαγραμμάτων.



IDEF0 ΛΕΙΤΟΥΡΓΙΑ ΚΙΒΩΤΙΩΝ ΚΑΙ ΒΕΛΩΝ ΔΙΕΠΑΦΗΣ

- Η βαθμιαία έκθεση της λεπτομέρειας που χαρακτηρίζει μια ιεραρχική δομή, με τις σημαντικότερες λειτουργίες στην κορυφή και με τα διαδοχικά επίπεδα υπολειτουργιών που αποκαλύπτουν το καλά-οριακό ξεμπλοκάρισμα της κάθε λεπτομέρειας.

- Ένα "διάγραμμα κόμβων" που παρέχει έναν γρήγορο δείκτη για την εντόπιση των λεπτομερειών μέσα στην ιεραρχική δομή των διαγραμμάτων.

3. Ακαμψία και ακρίβεια

Οι κανόνες IDEF0 απαιτούν τις ικανότητες ακαμψίας και την ακρίβεια για να ικανοποιήσει τις ανάγκες χωρίς υπερβολικά να περιορίσει τον αναλυτή. Οι IDEF0 κανόνες περιλαμβάνουν τα εξής:

- Έλεγχος των λεπτομερειών που ανακοινώνονται σε κάθε επίπεδο (τρία έως έξι κιβώτια λειτουργίας σε κάθε επίπεδο μιας αποσύνθεσης).
- Οριακό πλαίσιο (καμία παράλειψη ή πρόσθετη λεπτομέρεια έξω από το όριο του πεδίου).
- Συνδετικότητα διαγραμμάτων διεπαφών (αριθμοί κόμβων, αριθμοί κιβωτίων, C-αριθμοί, έκφραση αναφοράς και λεπτομέρειας).
- Συνδετικότητα δομών δεδομένων (κώδικες ICOM και η χρήση των παρενθέσεων).
- Μοναδικές ετικέτες και τίτλοι (κανένα αναπαραγόμενο όνομα).
- Κανόνες σύνταξης για τη γραφική παράσταση (κιβώτια και βέλη).
- Περιορισμός κλάδων βελών στοιχείων (ετικέτες για τον περιορισμό της ροής στοιχείων στους κλάδους).
- Εισαγωγή εναντίον του χωρισμού ελέγχου (ένας κανόνας για το ρόλο των στοιχείων).
- Απαιτήσεις ετικετών στοιχείων βελών (ελάχιστοι κανόνες μαρκαρίσματος).
- Ελάχιστος έλεγχος της λειτουργίας (όλες οι λειτουργίες απαιτούν τουλάχιστον έναν έλεγχο).
- Σκοπός και άποψη (όλα τα πρότυπα έχουν μια δήλωση σκοπού και άποψης).

4. Μεθοδολογία

Οι βαθμιαίες διαδικασίες παρέχονται για τη διαμόρφωση, την αναθεώρηση, και τους στόχους ολοκλήρωσης.

5. Οργάνωση εναντίον της λειτουργίας

Ο χωρισμός της οργάνωσης από τη λειτουργία είναι συμπεριλαμβανόμενος στο σκοπό του προτύπου και πραγματοποιήσιμος από τις επιλεχθείσες λειτουργίες και τα ονόματα των διεπαφών κατά τη διάρκεια της πρότυπης ανάπτυξης.

6. Ακολουθία και ανεξαρτησία συγχρονισμού

Η εφαρμογή της μεθόδου IDEF0 οδηγεί σε μια οργανωμένη αντιπροσώπευση των δραστηριοτήτων των σημαντικών σχέσεων μεταξύ αυτών των δραστηριοτήτων σε μια μόνιμη μόδα. Το IDEF0 δεν υποστηρίζει την προδιαγραφή μιας συνταγής ή μιας διαδικασίας. Τέτοιες λεπτομερείς περιγραφές της συγκεκριμένης λογικής ή του συγχρονισμού που συνδέεται με τις δραστηριότητες απαιτούν τη περιγραφική διαδικασία, το IDEF3 το οποίο συλλαμβάνει τη μέθοδο.

7. Δυνάμεις και αδυναμίες IDEF0

Η αρχική δύναμη IDEF0 είναι ότι η μέθοδος έχει αποδειχθεί αποτελεσματική με τις λεπτομέρειες των δραστηριοτήτων των συστημάτων για τη διαμόρφωση της λειτουργίας. Οι δραστηριότητες μπορούν να περιγραφούν από τις εισαγωγές, τα αποτελέσματα, τους ελέγχους, και τους μηχανισμούς τους (ICOMs).

Επιπλέον, η περιγραφή των δραστηριοτήτων ενός συστήματος μπορεί να καθοριστεί εύκολα αριθμώντας ολόένα και περισσότερες έως ότου το πρότυπο να είναι τόσο περιγραφικό ανάλογα με τις ανάγκες και το στόχο για τη λήψη πιο προσιτών αποφάσεων.

Στην πραγματικότητα, ένα από τα παρατηρηθέντα προβλήματα με τα πρότυπα IDEF0 είναι ότι είναι συχνά τόσο συνοπτικές που είναι κατανοητές μόνο εάν ο αναγνώστης είναι εμπειρογνώμονας περιοχών ή έχει συμμετέχει στην πρότυπη ανάπτυξη. Η ιεραρχική φύση IDEF0 διευκολύνει τη δυνατότητα να κατασκευαστούν (τα πραγματικά) πρότυπα που έχουν από επάνω προς τα κάτω μια

αντιπροσώπευση και μια ερμηνεία, αλλά που είναι βασισμένα σε μια από κάτω προς τα επάνω διαδικασία ανάλυσης. Αρχίζοντας με τα ακατέργαστα στοιχεία (γενικά αποτελέσματα συνέντευξης με τους εμπειρογνώμονες περιοχών), ο κατασκευαστής του μοντέλου αρχίζει με τις δραστηριότητες που είναι πολύ σχετικές ή λειτουργικά παρόμοιες. Μέσω αυτής της διαδικασίας ομαδοποίησης, προκύπτει η ιεραρχία. Εάν η λειτουργική αρχιτεκτονική μιας επιχείρησης σχεδιάζεται (συχνά αναφερόμενη ως διαμόρφωση), η από επάνω προς τα κάτω κατασκευή είναι συνήθως πιο κατάλληλη.

Αρχίζοντας με την κορυφαία δραστηριότητα, η διαμορφωμένη λειτουργία της επιχείρησης μπορεί να περιγραφεί μέσω μιας λογικής αποσύνθεσης. Η διαδικασία μπορεί να συνεχιστεί κατ' επανάληψη μέχρι το επιθυμητό επίπεδο λεπτομέρειας. Όταν μια υπάρχουσα επιχείρηση αναλύεται και διαμορφώνεται (συχνά αναφερόμενη ως πραγματική διαμόρφωση), παρατηρείτε ότι οι δραστηριότητες μπορούν να περιγραφούν και να συνδυαστούν έπειτα σε μια δραστηριότητα πιο υψηλού επιπέδου. Αυτή η διαδικασία συνεχίζεται επίσης έως ότου έχει περιγραφεί η δραστηριότητα πιο υψηλού επιπέδου. Ένα πρόβλημα με το IDEF0 είναι η τάση των προτύπων IDEF0 που ερμηνεύονται σαν να αντιπροσωπεύουν μια ακολουθία δραστηριοτήτων. Ενώ το IDEF0 δεν προορίζεται να χρησιμοποιηθεί για τη διαμόρφωση των ακολουθιών μιας δραστηριότητας, είναι εύκολο το να γίνει αυτό.

Οι δραστηριότητες μπορούν να τοποθετηθούν σε μια αριστερή προς δεξιά ακολουθία και να συνδεθούν με τις ροές. Είναι φυσικό να διαταχθούν οι δραστηριότητες από τα αριστερά προς τα δεξιά επειδή εάν μια δραστηριότητα που έχει την έννοια αποτελέσματος χρησιμοποιείται ως εισαγωγή από μια άλλη δραστηριότητα, ενώνοντας τα κιβώτια δραστηριότητας και τις έννοιες κάνεις τις συνδέσεις σαφέστερες. Κατά συνέπεια, χωρίς πρόθεση, η αλληλουχία δραστηριότητας μπορεί να συναρμοστεί στο πρότυπο IDEF0. Σε περιπτώσεις όπου οι ακολουθίες δραστηριότητας δεν συμπεριλαμβάνονται στο πρότυπο, οι αναγνώστες του προτύπου μπορούν να μπουν στον πειρασμό και να προσθέσουν μια τέτοια ερμηνεία. Αυτή η ανώμαλη κατάσταση θα μπορούσε να θεωρηθεί αδυναμία του IDEF0. Εντούτοις, το σωστό θα οδηγούσε στη δωροδοκία των βασικών αρχών στις οποίες το IDEF0 είναι βασισμένο και ως εκ τούτου θα έχανε τα αποδεδειγμένα οφέλη της μεθόδου.

Η αφαίρεση μακριά από το συγχρονισμό, την αλληλουχία, και τη λογική της απόφασης επιτρέπει τη συνοπτικότητα σε ένα πρότυπο IDEF0. Εντούτοις, τέτοια αφαίρεση συμβάλλει επίσης στις δυσκολίες κατανόησης μεταξύ των αναγνωστών έξω από την περιοχή. Αυτό το ιδιαίτερο πρόβλημα απευθύνεται προς τη μέθοδο IDEF3.

Το KBSI έχει αναπτύξει ένα αυτοματοποιημένο εργαλείο διαμόρφωσης λειτουργίας το **AIO WIN**, για να υποστηρίξει τη μέθοδο IDEF0.

3.2 Γενικά για το IDEF1

Το IDEF1 σχεδιάστηκε ως μέθοδος και για την ανάλυση και για την επικοινωνία στην καθιέρωση των απαιτήσεων. Το IDEF1 χρησιμοποιείται γενικά

- 1) για να προσδιορίζει ποιες πληροφορίες ρυθμίζονται αυτήν την περίοδο στην οργάνωση,
- 2) για να καθορίζει τα όποια προβλήματα που προσδιορίζονται κατά τη διάρκεια των αναγκών της ανάλυσης που προκαλείται από την έλλειψη της διαχείρισης των σωστών πληροφοριών, και
- 3) διευκρινίζει ποιες πληροφορίες θα ρυθμιστούν σε μια TO-BE εφαρμογή.

Το IDEF1 συλλαμβάνει τις πληροφορίες που υπάρχουν για τα αντικείμενα στο πλαίσιο μιας επιχείρησης. Η προοπτική IDEF1 ενός συστήματος πληροφοριών περιλαμβάνει όχι μόνο τα αυτοματοποιημένα τμήματα συστημάτων, αλλά και τα μη αυτοματοποιημένα αντικείμενα όπως οι άνθρωποι, ντουλάπια αρχειοθέτησης, τηλέφωνα, κ.λπ. Το IDEF1 σχεδιάστηκε ως μέθοδος για τις οργανώσεις που αναλύουν και που εκφράζουν σαφώς τις ανάγκες των πηγών πληροφοριών τους και τις διοικητικές ανάγκες, παρά μια μέθοδο σχεδίου βάσεων δεδομένων.

Το IDEF1 είναι μια μέθοδος ανάλυσης που χρησιμοποιείται για να προσδιορίσει τα εξής:

- Πληροφορίες που έχουν συλλεχθεί, που αποθηκεύονται και διοικούνται από την επιχείρηση.
- Κανόνες που κυβερνούν τη διαχείριση των πληροφοριών.
- Λογικές σχέσεις μέσα στην επιχείρηση που απεικονίζεται στις πληροφορίες.
- Προβλήματα ως αποτέλεσμα της έλλειψης καλής διαχείρισης πληροφοριών.

Τα αποτελέσματα της ανάλυσης πληροφοριών μπορούν να χρησιμοποιηθούν από τους στρατηγικούς και τακτικούς αρμόδιους για το σχεδιασμό μέσα στην επιχείρηση εκμεταλεύοντας τα προτερήματα πληροφοριών τους για να επιτύχουν στα ανταγωνιστικά πλεονεκτήματα.

Τα σχέδιά τους μπορούν να περιλάβουν το σχέδιο και την εφαρμογή των αυτοματοποιημένων συστημάτων που μπορούν να εκμεταλλευθούν αποτελεσματικότερα τις διαθέσιμες πληροφορίες στην επιχείρηση. Τα IDEF1 πρότυπα παρέχουν τη βάση για εκείνες τις αποφάσεις σχεδίου, που εφοδιάζουν τους διευθυντές με τη διορατικότητα και τη γνώση που απαιτούνται για να καθιερώσει την καλή πολιτική διαχείρισης των πληροφοριών.

3.2.1 IDEF1 οι στοιχειώδεις αρχές

Το IDEF1 χρησιμοποιεί τις απλές γραφικές συμβάσεις για να εκφράσει ένα ισχυρό σύνολο κανόνων που βοηθούν το σχεδιαστή να διακρίνει μεταξύ

- 1) των πραγματικών αντικειμένων,
- 2) φυσικές ή αφηρημένες ενώσεις που διατηρούνται μεταξύ των πραγματικών αντικειμένων,
- 3) πληροφορίες διοικούμενες για ένα πραγματικό αντικείμενο,

4) της δομής δεδομένων που χρησιμοποιείται για να αντιπροσωπεύσει εκείνες τις πληροφορίες για την απόκτηση, την εφαρμογή, και τη διαχείριση εκείνων των πληροφοριών.

Το IDEF1 παρέχει ένα σύνολο κανόνων και διαδικασιών για την ανάπτυξη των προτύπων πληροφοριών. Ένας από τους στόχους του IDEF1 είναι να παρασχεθεί μια δομημένη και πειθαρχημένη διαδικασία για τις διοικούμενες πληροφορίες από μια οργάνωση. Αυτός ο στόχος ολοκληρώνεται με την εξελικτική διαδικασία που καθορίζεται στη μέθοδο και από τα μετρήσιμα αποτελέσματα και τα συγκεκριμένα προϊόντα που απαιτούνται από τη μέθοδο.

Το IDEF1 επιβάλλει ένα μοντέλο διαμόρφωσης που αποβάλλει την ημιτέλεια, την ανακρίβεια, τις ασυνέπειες, και τις ανακρίβειες που βρίσκονται στη διαδικασία διαμόρφωσης.

3.2.2 Έννοιες του IDEF1

Υπάρχουν δύο σημαντικές σφαίρες για τους σχεδιαστές για να εξετάσουν τον καθορισμό των αναγκών σε πληροφορίες. Η πρώτη σφαίρα είναι ο πραγματικός κόσμος όπως γίνεται αντιληπτός από τους ανθρώπους σε μια οργάνωση. Αποτελείται από τα φυσικά και εννοιολογικά αντικείμενα (π.χ., άνθρωποι, θέσεις, πράγματα, ιδέες, κ.λπ.), τις ιδιότητες εκείνων των αντικειμένων, και τις σχέσεις που συνδέουν εκείνα τα αντικείμενα. Η δεύτερη σφαίρα είναι η σφαίρα των πληροφοριών. Περιλαμβάνει τις εικόνες πληροφοριών εκείνων των αντικειμένων που υπάρχουν στη πραγματικότητα.

Μια εικόνα πληροφοριών δεν είναι το πραγματικό αντικείμενο, αλλά οι πληροφορίες οι οποίες συλλέχθηκαν, αποθηκεύθηκαν και διαχειρίστηκαν από τα πραγματικά αντικείμενα. Το IDEF1 έχει ως σκοπό να βοηθήσει στην ανακάλυψη, την οργάνωση και την τεκμηρίωση αυτής της εικόνας πληροφοριών και είναι έτσι περιορισμένο στη σφαίρα πληροφοριών.

Μια οντότητα IDEF1 αντιπροσωπεύει τις πληροφορίες που διατηρούνται σε μια συγκεκριμένη οργάνωση για τα φυσικά ή εννοιολογικά αντικείμενα. Μια κατηγορία οντοτήτων IDEF1 αναφέρεται σε μια συλλογή των οντοτήτων ή την κατηγορία πληροφοριών που κρατούνται για τα αντικείμενα στον πραγματικό

κόσμο. Υπάρχουν δύο βασικές έννοιες που διακρίνουν τις οντότητες:

- **Είναι επίμονοι.** Η οργάνωση χρησιμοποιεί τους πόρους που παρατηρούν, που κωδικοποιούν, που καταγράφουν, που οργανώνουν, και που αποθηκεύουν την ύπαρξη των μεμονωμένων οντοτήτων.
- **Μπορούν να είναι.** Μπορούν να προσδιοριστούν μεμονωμένα από άλλες οντότητες.

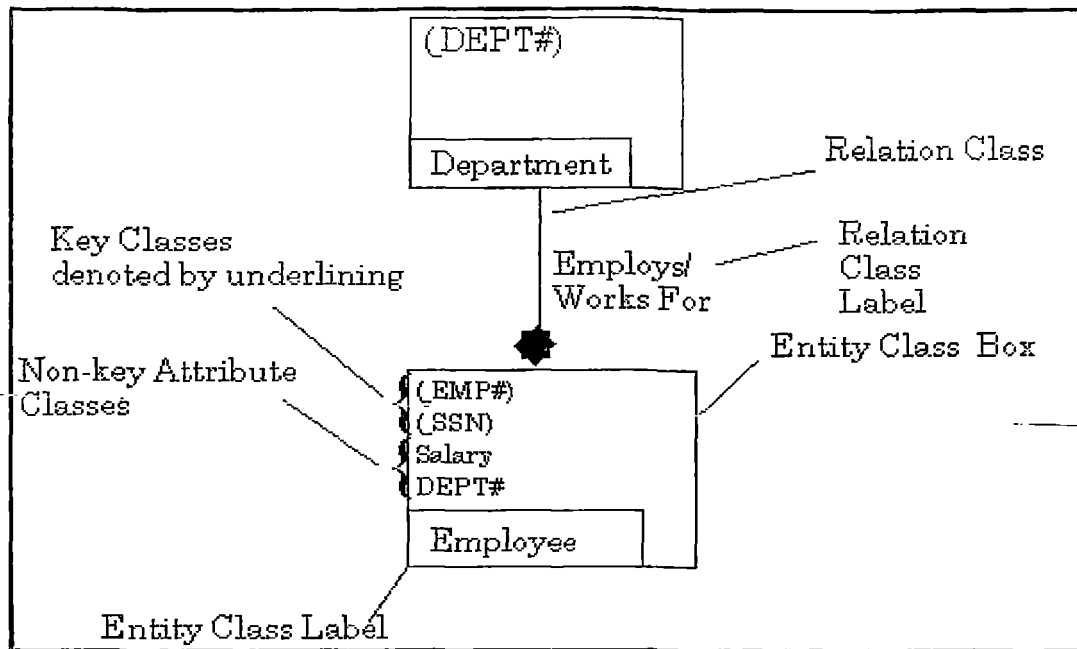
Οι οντότητες συνδέουν τις χαρακτηριστικές ιδιότητες μεταξύ τους. Οι ιδιότητες καταγράφουν τις τιμές των ιδιοτήτων των πραγματικών αντικειμένων. Η κατηγορία ιδιοτήτων αναφέρεται στο σύνολο ζευγαριών ιδιότητας-αξίας που διαμορφώνονται με την ομαδοποίηση του ονόματος των ιδιοτήτων και των τιμών εκείνης της ιδιότητας για τα μεμονωμένα μέλη κατηγορίας οντοτήτων (οντότητες). Μια συλλογή μιας ή περισσότερων κατηγοριών ιδιοτήτων που διακρίνει ένα μέλος μιας κατηγορίας οντοτήτων από άλλο καλείται βασική κατηγορία.

Μια σχέση σε IDEF1 είναι ένωση μεταξύ δύο μεμονωμένων εικόνων πληροφοριών. Η ύπαρξη μιας σχέσης ανακαλύπτεται ή ελέγχεται με τη σημείωση ότι οι κατηγορίες ιδιοτήτων μιας κατηγορίας οντοτήτων περιέχουν τις κατηγορίες ιδιοτήτων της βασικής κατηγορίας του παραπεμφθέντος μέλους της κατηγορίας οντοτήτων. Μια κατηγορία σχέσης μπορεί να θεωρηθεί ως πρότυπο για τις ενώσεις που υπάρχουν μεταξύ των κατηγοριών οντοτήτων.

Ένα παράδειγμα μιας σχέσης σε IDEF1 είναι η ετικέτα "εργασίες για" στη σύνδεση μεταξύ της οντότητας πληροφοριών αποκαλούμενης "υπάλληλο" και της οντότητας πληροφοριών αποκαλούμενης "τμήμα." Εάν καμία πληροφορία δεν κρατείται για μια ένωση μεταξύ δύο ή περισσότερων αντικειμένων στον πραγματικό κόσμο, τότε, καμία σχέση δεν υπάρχει από άποψη IDEF1. Οι κατηγορίες σχέσης αντιπροσωπεύονται από τις συνδέσεις μεταξύ των κιβωτίων κατηγορίας οντοτήτων σε ένα διάγραμμα IDEF1.

Τα διαμάντια στο τέλος των συνδέσεων και τα μισά διαμάντια στη μέση των συνδέσεων κωδικοποιούν τις πρόσθετες πληροφορίες για την κατηγορία σχέσης (δηλ., αριθμός στοιχείων συνόλου και

εξάρτηση). Το σχήμα επεξηγεί κατωτέρω τον τρόπο με τον οποίο τα διαγράμματα IDEF1 σχεδιάζονται.



ΔΙΑΓΡΑΜΜΑ IDEF1

3.2.3 Οι αντοχές του IDEF1

Το IDEF1 είναι μια αποτελεσματική μέθοδος για τις ενημερωτικές απαιτήσεις μιας επιχείρησης. Η άσκηση διαμόρφωσης IDEF1 αποτελεί τη βάση για το σχέδιο βάσεων δεδομένων, δίνει έναν καθορισμό της δομής πληροφοριών, και παρέχει μια δήλωση απαιτήσεων που απεικονίζει τις ανάγκες βασικών πληροφοριών. Το IDEF1 χρησιμοποιεί μια πειθαρχημένη, δομημένη τεχνική για να αποκαλυφθούν οι πληροφορίες και οι επιχειρησιακοί κανόνες που χρησιμοποιούνται από μια οργάνωση.

Αυτό δίνει την αναγκαία ακαμψία στη μέθοδο για τη σύνθετη πρόκληση της διαμόρφωσης των πληροφοριών της οργάνωσης. Το IDEF1 απαιτεί την ενεργό συμμετοχή των χρηστών πληροφοριών. Αυτό χρησιμεύει στο να διαμορφώσει ακριβώς την οργάνωση με τον καταναγκασμό των χρηστών για να σκεφτεί για το πώς και που οι πληροφορίες χρησιμοποιούνται και ρυθμίζονται.

Τέλος, τα πρότυπα πληροφοριών είναι χρήσιμα καθ' όλη τη διάρκεια του κύκλου της ζωής της επιχείρησης. Το KBSI έχει αναπτύξει ένα αυτοματοποιημένο εργαλείο διαμόρφωσης πληροφοριών και στοιχείων, το *SmartER*, για να υποστηρίξουν τις μεθόδους IDEF1 και IDEF1X.

3.3 Γενικά για το IDEF1X

Το IDEF1X είναι μια μέθοδος για τις σχεσιακές βάσεις δεδομένων με μία σύνταξη με σκοπό να υποστηρίξει τα σημασιολογικά κατασκευάσματα που είναι απαραίτητα στην ανάπτυξη ενός εννοιολογικού σχήματος.

Ένα εννοιολογικό σχήμα είναι ένας ενιαίος ενσωματωμένος καθορισμός του επιχειρηματικού στοιχείου που είναι αμερόληπτο προς οποιασδήποτε ενιαίας εφαρμογής και ανεξάρτητο της πρόσβασης και της φυσικής αποθήκευσής του. Επειδή είναι μια μέθοδος σχεδίου, το IDEF1X δεν είναι ιδιαίτερα σχεδιασμένο για να χρησιμεύσει ως ένα πραγματικό εργαλείο ανάλυσης, αν και χρησιμοποιείται συχνά με εκείνη την ιδιότητα ως εναλλακτική λύση του IDEF1. Το IDEF1X είναι το πιο χρησιμοποιούμενο εργαλείο για το λογικό σχεδιασμό βάσεων δεδομένων αφότου είναι γνωστές οι ανάγκες σε πληροφορίες και η απόφαση να εφαρμοστεί μια σχεσιακή βάση δεδομένων έχει ληφθεί. Ως εκ τούτου, η προοπτική συστημάτων IDEF1X στρέφεται στα πραγματικά στοιχεία σε μια σχεσιακή βάση δεδομένων. Εάν το σύστημα στόχων δεν είναι ένα συγγενικό σύστημα, παραδείγματος χάριν, ένα αντικειμενοστρεφές σύστημα, το IDEF1X δεν είναι η καλύτερη μέθοδος.

Υπάρχουν διάφοροι λόγοι για τους οποίους το IDEF1X δεν είναι καλοσχεδιασμένο για τις μη-συγγενικές εφαρμογές συστημάτων. Το IDEF1X απαιτεί, παραδείγματος χάριν, ότι ο σχεδιαστής υποδεικνύει ένα βασικό κλειδί κατηγορία για να διακρίνει μια οντότητα από μια άλλη, ενώ τα αντικειμενοστρεφή συστήματα δεν απαιτούν κάτι τέτοιο. Περαιτέρω, σε εκείνες τις καταστάσεις όπου περισσότερες από μια ιδιότητα ή σύνολα ιδιοτήτων θα χρειάζεται να εξυπηρετηθούν εξίσου καλά για όλες τις οντότητες IDEF1X, ο σχεδιαστής πρέπει να υποδείξει το ένα ως αρχικό κλειδί και να

απαριθμήσει όλες τις άλλες ως εναλλάσσομαι κλειδιά. Επίσης η αναφορά στο ξένο κλειδί κρίνεται απαραίτητη.

Τα προκύπτοντα λογικά πρότυπα σχεδίου IDEF1X προορίζονται να χρησιμοποιηθούν από τους προγραμματιστές που παίρνουν το σχεδιάγραμμα για το λογικό σχέδιο βάσεων δεδομένων και εφαρμόζουν εκείνο το σχέδιο.

Εντούτοις, η γλώσσα διαμόρφωσης IDEF1X είναι αρκετά παρόμοια με την IDEF1 σε εκείνα τα πρότυπα που παράγονται από τις ανάγκες σε πληροφορίες IDEF1 μπορεί να αναθεωρηθεί και να γίνει κατανοητή από τους τελευταίους χρήστες του προτεινόμενου συστήματος.

Τον Δεκέμβριο του 1993, το εργαστήριο συγκροτημάτων ηλεκτρονικών υπολογιστών **Το εθνικό ίδρυμα προτύπων και η τεχνολογία (NIST)** απελευθέρωσαν τα IDEF1X ως πρότυπα για το σχεδιασμό δεδομένων, **FIPS δημοσίευση 184**.

3.3.1 Έννοιες του IDEF1X

Αν και η ορολογία μεταξύ IDEF1 και IDEF1X είναι πολύ παρόμοια, υπάρχουν θεμελιώδεις διαφορές στα θεωρητικά θεμέλια και τις έννοιες των δύο μεθόδων. Μια οντότητα σε IDEF1X αναφέρεται ως μια συλλογή στοιχείων ή ένα σύνολο παρόμοιων περιπτώσεων που μπορούν να διακριθούν χωριστά μεταξύ τους. Τα μεμονωμένα μέλη του συνόλου καλούνται περιπτώσεις οντοτήτων. Κατά συνέπεια, ένα κιβώτιο σε IDEF1X αντιπροσωπεύει ένα σύνολο στοιχείων δεδομένων στην πραγματική σφαίρα. Μια ιδιότητα είναι μια αξία αυλακώσεων που συνδέεται με κάθε μεμονωμένο μέλος του συνόλου. Στη σχέση που υπάρχει μεταξύ των μεμονωμένων μελών αυτών των συνόλων δίνεται ένα όνομα. Σε αυτήν την περίπτωση, αυτή η σχέση καθιερώνει έναν αναφερόμενο περιορισμό ακεραιότητας.

Ένα ισχυρό χαρακτηριστικό γνώρισμα του IDEF1X είναι η υποστήριξή του για τη διαμόρφωση των λογικών τύπων των δεδομένων μέσω της χρήσης ενός κατασκευάσματος δομών ή γενικής / ειδικής ταξινόμησης. Αυτό το κατασκεύασμα είναι μια προσπάθεια να επιστρωθούν τα πρότυπα των φυσικών ειδών πραγμάτων που το στοιχείο αντιπροσωπεύει ενώ τα παράθυρα, ή

οι οντότητες, προσπαθούν να διαμορφώσουν τους τύπους των πραγμάτων στοιχείων.

Αυτές οι σχέσεις κατηγοριοποίησης αντιπροσωπεύουν τα αμοιβαία αποκλειστικά υποσύνολα μιας γενικής οντότητας ή ενός συνόλου.

Τα υποσύνολα των υπερθεμάτων δεν μπορούν να έχουν κοινές περιπτώσεις. Παραδείγματος χάριν, ένα γενικό ΠΡΟΣΩΠΟ οντοτήτων έχει δύο υποσύνολα που αντιπροσωπεύουν όλες τις πλήρη κατηγορίες, δηλαδή, το ΑΡΣΕΝΙΚΟ και το ΘΗΛΥΚΟ. Καμία περίπτωση του ΑΡΣΕΝΙΚΟΥ υποσυνόλου δεν μπορεί να είναι μια περίπτωση του ΘΗΛΥΚΟΥ υποσυνόλου, και αντίστροφα. Η μοναδική ιδιότητα προσδιοριστικών για κάθε υποσύνολο είναι η ίδια ιδιότητα με αυτήν για μια γενική περίπτωση οντοτήτων.

3.3.2 Η συντακτική σημασία του IDEF1X

Σε IDEF1X, οι οντότητες είναι είτε προσδιοριστικές-ανεξάρτητες είτε προσδιοριστικές-εξαρτώμενες. Οι περιπτώσεις των προσδιοριστικών-ανεξάρτητων οντοτήτων μπορούν να υπάρξουν χωρίς οποιαδήποτε άλλη περίπτωση οντοτήτων, ενώ οι περιπτώσεις των προσδιοριστικών-εξαρτώμενων οντοτήτων είναι χωρίς νόημα (εξ ορισμού) χωρίς μια άλλη σχετική περίπτωση οντοτήτων. Η εξάρτηση και η ανεξαρτησία είναι συγκεκριμένες για ένα πρότυπο.

3.3.3 Σχέσεις σύνδεσης

Οι σχέσεις σύνδεσης (στερεές ή ορμούμενες γραμμές με τους γεμισμένους κύκλους και σε μια ή και τις δύο άκρες) δείχνουν πώς οι οντότητες (σύνολα περιπτώσεων στοιχείων) αφορούν η μια την άλλη. Οι σχέσεις σύνδεσης είναι πάντα μεταξύ ακριβώς δύο οντοτήτων. Η αρχή της σχέσης σύνδεσης στην ανεξάρτητη οντότητα γονέων και το τελείωμα στην οντότητα εξαρτώμενων παιδιών ονομάζονται με μια φράση, περιγραφή της σχέσης. Κάθε σχέση σύνδεσης έχει έναν σχετικό αριθμό στοιχείων συνόλου. Ο αριθμός στοιχείων συνόλου διευκρινίζει τον αριθμό περιπτώσεων της εξαρτώμενης οντότητας που συσχετίζονται με μια περίπτωση της ανεξάρτητης οντότητας.

3.3.3.1 Σχέσεις κατηγοριοποίησης

Οι σχέσεις κατηγοριοποίησης επιτρέπουν στο σχεδιαστή να καθορίσει την κατηγορία μιας οντότητας. Μια οντότητα μπορεί να ανήκει μόνο σε μια κατηγορία. Παραδείγματος χάριν, θα μπορούσε να υπάρξει ένα ΑΥΤΟΚΙΝΗΤΟ οντοτήτων που είναι η γενική οντότητα σε μια κατηγορία που παρουσιάζει διαφορετικούς τύπους αυτοκινήτων. Κάθε οντότητα κατηγορίας πρέπει να έχει το ίδιο αρχικό κλειδί με το ΑΥΤΟΚΙΝΗΤΟ. Επίσης, πρέπει να υπάρξει ένας τρόπος μεταξύ των οντοτήτων κατηγορίας. Οι οντότητες κατηγορίας διακρίνονται από μια ιδιότητα διευκρινιστικών που πρέπει να έχει μια διαφορετική αξία για κάθε οντότητα κατηγορίας.

3.3.3.2 Ιδιότητες

Οι ιδιότητες είναι οι ιδιότητες που χρησιμοποιούνται για να περιγράψουν μια οντότητα. Τα ονόματα των ιδιοτήτων είναι μοναδικά σε όλο το πρότυπο IDEF1X, και η έννοια των ονομάτων πρέπει να είναι συνεπής. Παραδείγματος χάριν, οι ιδιότητες "χρώμα" θα μπορούσαν να έχουν διάφορες πιθανές χρήσεις για το χρώμα τρίχας, χρώμα δέρματος, ή ένα χρώμα σε ένα ουράνιο τόξο. Κάθε χρήση έχει μια σειρά σημαντικών τιμών, και έτσι, η οντότητα πρέπει να ονομαστεί ευδιάκριτα. Κάθε ιδιότητα είναι κύρια από ακριβώς μια οντότητα.

Η ιδιότητα «αριθμός κοινωνικής ασφάλισης», παραδείγματος χάριν, θα μπορούσε να χρησιμοποιηθεί σε ισχύ πολλές φορές σε ένα πρότυπο, αλλά θα ήταν κύριο από μόνο μια οντότητα (π.χ., ΠΡΟΣΩΠΟ). Άλλα περιστατικά των ιδιοτήτων αριθμού κοινωνικής ασφάλισης θα κληρονομούσαν στις σχέσεις. Κάθε ιδιότητα πρέπει να έχει μια αξία (κανόνας-μηδενικός κανόνας), και καμία ιδιότητα δεν μπορεί να έχει τις πολλαπλάσιες τιμές (κανόνας καμίας-επανάληψης).

Οι κανόνες επιβάλλουν τη δημιουργία των κατάλληλων προτύπων. Σε μια κατάσταση όπου φαίνεται ότι ένας κανόνας δεν μπορεί να κρατήσει, το πρότυπο είναι πιθανόν λανθασμένο.

3.3.3.3 Κλειδιά

Ένα κλειδί είναι ομάδα ιδιοτήτων που προσδιορίζουν μεμονωμένα μια περίπτωση οντοτήτων. Υπάρχουν αρχικά και εναλλάσσομαι κλειδιά. Κάθε οντότητα επιδεικνύει ακριβώς ένα αρχικό κλειδί επάνω από την οριζόντια γραμμή στο κιβώτιο οντοτήτων. Οι οντότητες μπορούν να έχουν τα εναλλάσσομαι κλειδιά που επίσης μεμονωμένα προσδιορίζουν την οντότητα, αλλά δεν χρησιμοποιούνται για την περιγραφή των σχέσεων με άλλες οντότητες.

Σε μια σχέση σύνδεσης, το αρχικό κλειδί του γονέα μεταναστεύει στο παιδί. Εάν η σχέση είναι μια σχέση κατηγορίας, το αρχικό κλειδί του παιδιού είναι το ίδιο με το γενικό. Εάν η σχέση είναι μια πρόσδιοριστική σχέση, το αρχικό κλειδί του παιδιού πρέπει να περιέχει τις ιδιότητες που κληρονομούνται από το γονέα. Εκτός από το γεγονός ότι ένα κλειδί πρέπει μεμονωμένα να προσδιορίζει μια οντότητα, όλες οι ιδιότητες στο κλειδί πρέπει να συμβάλουν στο μοναδικό προσδιορισμό (μικρός-βασικός κανόνας). Κατά συνέπεια, όταν η απόφαση εάν μια κληρονομημένη ιδιότητα πρέπει να είναι μέρος ενός κλειδιού, ένα ζήτημα είναι εάν εκείνη η ιδιότητα είναι απαραίτητη για το μοναδικό προσδιορισμό. Δεν είναι ικανοποιητικό ότι συνέβαλε στο μοναδικό προσδιορισμό του γονέα.

Υπάρχουν επίσης δύο κανόνες εξάρτησης: Ο κανόνας πλήρης - λειτουργικής-εξάρτησης δηλώνει ότι εάν το αρχικό κλειδί αποτελείται από πολλαπλές ιδιότητες, όλες οι χωρίς - κλειδί ιδιότητες πρέπει να εξαρτηθούν λειτουργικά από ολόκληρο το αρχικό κλειδί. Ο κανόνας καμίας -μεταβατικής-εξάρτησης δηλώνει ότι κάθε ιδιότητα χωρίς -κλειδί πρέπει να εξαρτηθεί λειτουργικά μόνο από τις βασικές ιδιότητες.

3.3.3.4 Ξένα κλειδιά

Τα ξένα κλειδιά δεν είναι πραγματικά κλειδιά καθόλου, αλλά ιδιότητες που κληρονομούνται από τα αρχικά κλειδιά άλλων οντοτήτων. Τα ξένα κλειδιά ονομάζονται (FK) για να δείξουν ότι δεν είναι κύρια από εκείνη την οντότητα.

Τα ξένα κλειδιά είναι σημαντικά επειδή παρουσιάζουν τις σχέσεις μεταξύ των οντοτήτων. Επειδή οι οντότητες περιγράφονται από τις ιδιότητές τους, εάν μια οντότητα αποτελείται από τις ιδιότητες που κληρονομούνται από άλλες οντότητες, εκείνη η οντότητα είναι παρόμοια με εκείνες τις οντότητες.

3.3.4 Αντοχή του IDEF1X

Το IDEF1X είναι ένα ισχυρό εργαλείο για το σχεδιασμό δεδομένων ακόμα κι αν υπάρχουν πολυάριθμες άλλες μέθοδοι διαμόρφωσης δεδομένων συμπεριλαμβανομένου του ER και ENALIM. Μια δύναμη IDEF1X βρίσκεται στις ρίζες της. Λόγω της ακριβούς τυποποίησης των προγραμμάτων DoD, το IDEF1X θα δραπετεύσει πιθανώς έχοντας πολυάριθμες παραλλαγές που έχουν εμποδίσει τη χρήση του ER. Έχοντας πρότυπα και εμμένοντας ότι είναι κρίσιμο για τη μεταφορά της γνώσης μεταξύ των οργανώσεων.

Μια αδυναμία σχεδόν όλων των μεθόδων συμπεριλαμβανομένου και του IDEF1X είναι ότι ο σχεδιαστής πρέπει να έχει εμπειρία προκειμένου να δημιουργηθούν καλά πρότυπα. Η διαμόρφωση δεν είναι μια διαισθητική διαδικασία, και πολλά χρονικά πρότυπα θα απορριφθούν λόγω μιας φτωχής έναρξης.

Όσο απλούστερη είναι η μέθοδος τόσο καλύτερα χρησιμοποιείται αλλά η μέθοδος πρέπει ακόμα να έχει την απαραίτητη εκφραστική δύναμη.

Ένα καλό παράδειγμα μιας ισχυρής έννοιας που δε μπορεί να χρησιμοποιηθεί σωστά είναι η σχέση κατηγορίας. Εκτιμώντας ότι υπάρχουν περιπτώσεις όταν οι κατηγορίες είναι απαραίτητες, υπάρχουν άλλες περιπτώσεις όταν χρησιμοποιούνται για να δημιουργήσουν τις χωρίς νόημα οντότητες. Πολλοί άπειροι

σχεδιαστές του IDEF1X τείνουν να περιέλθουν στην παγίδα της χρησιμοποίησης των χαρακτηριστικών γνωρισμάτων κατηγοριοποίησης IDEF1X για να διαμορφώσουν τις φυσικές ταξινομήσεις σε αντιδιαστολή με τις ταξινομήσεις των στοιχείων δεδομένων που προορίζονται.

Λόγω των τμημάτων κατηγοριοποίησης της μεθόδου IDEF1X, πολλοί εμπειρογνώμονες περιοχών έχουν περιέλθει στην παγίδα της χρησιμοποίησης της μεθόδου για τον καθορισμό έννοιας και ορολογίας. Δυστυχώς, οι εκτιμήσεις διαμόρφωσης στοιχείων που χτίζονται στους κανόνες IDEF1X δεν το επιτρέπουν για να λειτουργήσουν επαρκώς για αυτόν το λόγο. Το αποτέλεσμα είναι ότι ένα μεγάλο μέρος των πληροφοριών που συγκεντρώνονται δεν μπορεί να εκφραστεί ή εκφράζεται λανθασμένα. Παραδείγματος χάριν, το IDEF1X θα πρέπει να είναι ικανό για γεγονότα όπως "μια δήλωση της εργασίας είναι ένα έγγραφο και είναι μια νομική σύμβαση" και "ένα τετράγωνο είναι ένα πολύγωνο με τέσσερις ίσες πλευρές" δείχνει ότι η γλώσσα καθορίζει επαρκώς τη λειτουργία της έννοιας και της ορολογίας.

Το KBSI έχει αναπτύξει ένα αυτοματοποιημένο εργαλείο διαμόρφωσης πληροφοριών και στοιχείων, **SmartER**, για να υποστηρίξουν τις μεθόδους IDEF1 και IDEF1X.

3.4 Γενικά για το IDEF3

Η περιγραφική διαδικασία της μεθόδου IDEF3 παρέχει έναν μηχανισμό για τη συλλογή και την αρχειοθέτηση των διαδικασιών. Το IDEF3 συλλαμβάνει τις σχέσεις προτεραιότητας και αιτιότητας μεταξύ των καταστάσεων και των γεγονότων σε μια φυσική μορφή των έμπειρων περιοχών με την παροχή μιας δομημένης μεθόδου για τη γνώση για το πώς ένα σύστημα, μια διαδικασία, ή μια εργασία οργανώνεται.

Οι περιγραφές του IDEF3 μπορούν :

- Να καταγράψουν τα ακατέργαστα στοιχεία ως αποτέλεσμα των συνεντεύξεων διερεύνησης στις δραστηριότητες ανάλυσης συστημάτων.
- Να καθορίσουν τον αντίκτυπο της πηγής πληροφοριών μιας οργάνωσης στα σημαντικότερα σενάρια λειτουργίας μιας επιχείρησης.
- Να τεκμηριώσουν τις διαδικασίες απόφασης που έχουν επιπτώσεις στις καταστάσεις και τον κύκλο της ζωής των κρίσιμων κοινών στοιχείων, ιδιαίτερα στη κατασκευή, εφαρμογή, και στο καθορισμό των στοιχείων των προϊόντων συντήρησης.
- Να διαχειριστούν τη διαμόρφωση στοιχείων και το καθορισμό ελέγχου αλλαγής της πολιτικής.
- Να κάνουν την ανάλυση σχεδίου συστημάτων και ανταλλαγής σχεδίου.
- Να παρέχουν την πρότυπη παραγωγή προσομοίωσης.

Το IDEF3 συλλαμβάνει τις συμπεριφοριστικές πτυχές ενός υπάρχοντος ή προτεινόμενου συστήματος. Η συλλήφθαισα γνώση μιας διαδικασίας είναι δομημένη μέσα στο πλαίσιο ενός σεναρίου, που κάνει το IDEF3 μια δισαιθητική συσκευή αποκτήσεων γνώσης για ένα σύστημα. Το IDEF3 συλλαμβάνει όλες τις χρονικές πληροφορίες, συμπεριλαμβανομένων των σχέσεων προτεραιότητας και αιτιότητας που συνδέονται με τις επιχειρηματικές διαδικασίες.

Οι προκύπτουσες περιγραφές IDEF3 παρέχουν μια δομημένη βάση γνώσεων για την αναλυτική κατασκευή και τα πρότυπα σχεδιασμού. Αντίθετα από τις γλώσσες προσομοίωσης (π.χ., SIMAN, SLAM, GPSS, WITNESS) που χτίζουν τα προβλεπτικά μαθηματικά μοντέλα, το IDEF3 χτίζει τις δομημένες περιγραφές.

Αυτές οι περιγραφές συλλαμβάνουν τις πληροφορίες για αυτό που ένα σύστημα κάνει πραγματικά ή θα κάνει και θα προβλέψει επίσης για την οργάνωση και την έκφραση των διαφορετικών απόψεων των χρηστών του συστήματος.

3.4.1 Τρόποι περιγραφής του IDEF3

Υπάρχουν δύο τρόποι περιγραφής IDEF3, η ροή διαδικασίας και το δίκτυο κρατικής μετάβασης αντικειμένου. Μια περιγραφή ροής διαδικασίας συλλαμβάνει τη γνώση "πώς τα πράγματα λειτουργούν" σε μια οργάνωση, π.χ., περιγράφει αυτό που συμβαίνει σε ένα μέρος καθώς διατρέχει μια ακολουθία διαδικασιών κατασκευής.

Η περιγραφή δικτύων κρατικής μετάβασης αντικειμένου συνοψίζει τις επιτρεπόμενες μεταβάσεις που ένα αντικείμενο μπορεί να υποβληθεί όλο σε μια ιδιαίτερη διαδικασία.

Η περιγραφή ροής διαδικασίας και η περιγραφή κρατικής μετάβασης αντικειμένου περιέχουν τις μονάδες των πληροφοριών που αποτελούν την περιγραφή των συστημάτων. Αυτές οι πρότυπες οντότητες, καθώς καλούνται, διαμορφώνουν τις βασικές μονάδες μιας περιγραφής IDEF3. Τα προκύπτοντα διαγράμματα και το κείμενο περιλαμβάνουν το τι καλείται μια "περιγραφή" σε αντιδιαστολή με την εστίαση αυτών που παράγεται με τις άλλες μεθόδους IDEF των οποίων το προϊόν είναι ένα "πρότυπο." Μια περιγραφή ροής διαδικασίας IDEF3 συλλαμβάνει μια περιγραφή μιας διαδικασίας και το δίκτυο των σχέσεων που υπάρχει μεταξύ των διαδικασιών μέσα στο πλαίσιο του γενικού σεναρίου στο οποίο εμφανίζονται. Η πρόθεση αυτής της περιγραφής πρόκειται να επιδείξει πώς τα πράγματα λειτουργούν σε μια ιδιαίτερη οργάνωση όταν αντιμετωπίζεται ως μέρος μιας ιδιαίτερης λύσης προβλήματος ή επαναλαμβανόμενης κατάστασης.

Η ανάπτυξη μιας περιγραφής ροής διαδικασίας IDEF3 αποτελείται από την έκφραση των γεγονότων, που συλλέγονται από τους εμπειρογνώμονες περιοχών, από την άποψη πέντε βασικών περιγραφικών δομικών μονάδων.

Το ακόλουθο παράδειγμα επεξηγεί πώς οι δομικές μονάδες της μεθόδου IDEF3 μπορούν να περιγράψουν ένα σενάριο που βρίσκεται χαρακτηριστικά σε ένα περιβάλλον κατασκευής. Η κατάσταση που περιγράφεται είναι μια διαδικασία ζωγραφικής και επιθεώρησης που συνδέεται με την εφαρμογή του αρχικού χρώματος σε ένα μέρος που θα γίνει ένα στοιχείο ενός υποσυγκροτήματος για ένα κομμάτι του βαριού εξοπλισμού κατασκευής.

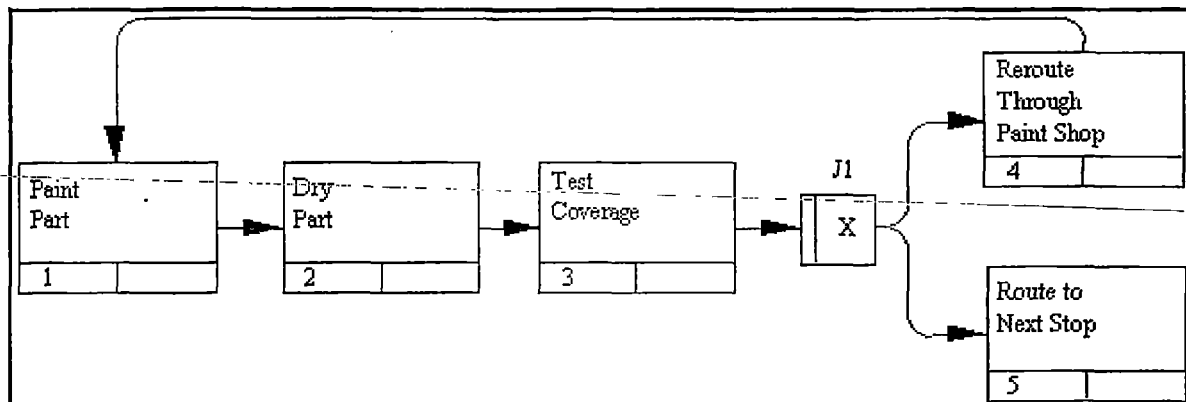
Η περιγραφή παραδείγματος IDEF3 που παρουσιάζεται στο σχήμα 1 είναι η γραφική αντιπροσώπευση του σεναρίου που λέγεται από έναν επόπτη καταστημάτων χρωμάτων όταν καλείται να περιγράψει: "Τι ακολουθεί σ' ένα κατάστημα στο αρχικό του στάδιο;"

Η ιστορία που ακολουθεί είναι ένα καλό παράδειγμα:

Τα μέρη εισάγονται στο κατάστημα έτοιμα για την αρχική επίστρωση που πρέπει να εφαρμοστεί. Εφαρμόζουμε μια πολύ βαριά επίστρωση του χρώματος σε μια πολύ υψηλή θερμοκρασία. Το χρώμα πρέπει να ξεραθεί καθώς ψήνεται στο φούρνο, ακολουθεί μετά μια δοκιμή κάλυψης με χρώματα σε κάθε μέρος. Εάν η δοκιμή αποκαλύψει ότι δεν ήταν αρκετή η επικάλυψη με τα χρώματα που ψεκάστηκαν στη κάθε επιφάνεια του κάθε μέρους, τότε το μέρος επανακαθοδηγείται μέσω του καταστήματος χρωμάτων, πάλι. Εάν το κάθε μέρος περάσει την επιθεώρηση, οδηγείται στην επόμενη στάση (σημείωση διαδικασίας: οι δραστηριότητες που περιγράφηκαν στο σενάριο προσδιορίζονται σαφώς και εμφανίζονται ως επονομαζόμενα κιβώτια στο σχήμα 1, και ότι τα επονομαζόμενα κιβώτια μπορούν να περιγράψουν τις δραστηριότητες, τις διαδικασίες, τα γεγονότα, κ.λπ.... Ο όρος IDEF3 για τα στοιχεία που αντιπροσωπεύονται από τα κιβώτια είναι μια μονάδα της συμπεριφοράς (Unit Of Behavior). Τα βέλη (συνδέσεις) συνδέουν τα κιβώτια (δραστηριότητες) και καθορίζουν τις λογικές ροές. Τα μικρότερα κιβώτια καθορίζουν τις συνδέσεις που παρέχουν έναν μηχανισμό για τη λογική στις ροές.

Αντιπροσωπεύονται αλλά όχι άμεσα στο σχήμα 1 το τμήμα, η αποσύνθεση και η επεξεργασία στο IDEF3. Κάθε μια μονάδα συμπεριφοράς (UOB) μπορεί να έχει συνδεθεί με την ή και τις δύο "περιγραφές από την άποψη μιας άλλης UOB" και μια "περιγραφή από την άποψη ενός συνόλου συμμετεχόντων αντικειμένων και σχέσεών τους". Αναφερόμαστε στα πρώτα ως αποσυνθέσεις ενός UOB και στα τελευταία ως επεξεργασία ενός UOB. Διαισθητικά, μια αποσύνθεση είναι ένα πιο στενό βλέμμα σε κάποιο δεδομένο UOB μέσα σε ένα μεγαλύτερο διάγραμμα. Μια αποσύνθεση είναι ένα διάγραμμα, και μπορεί να είναι μια αποσύνθεση κάποιου UOB στο διάγραμμα κορυφαίου επιπέδου ή μπορεί να είναι η αποσύνθεση ενός UOB σε μια αποσύνθεση. Ακριβέστερα, μια αποσύνθεση ενός δεδομένου UOB είναι μια

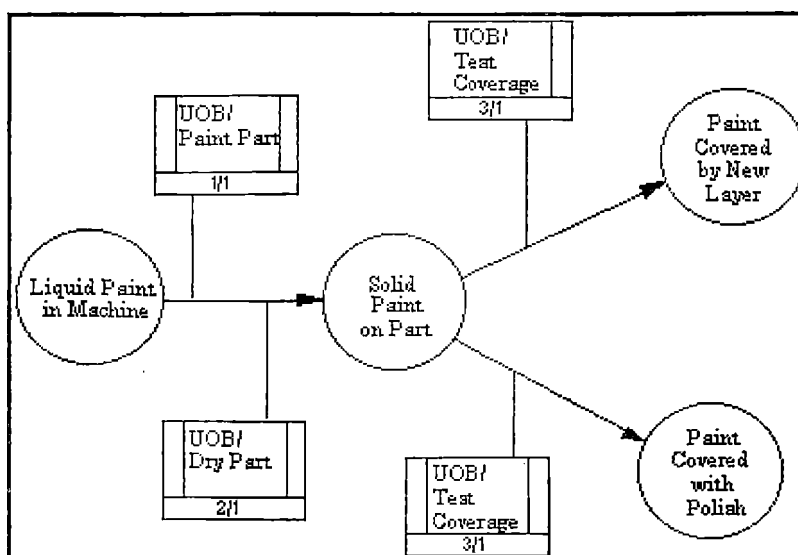
λεπτότερη κοκκιώδης αντιπροσώπευση IDEF3 εκείνου του UOB. Οι πολλαπλάσιες απόψεις (αποσυνθέσεις) επιτρέπονται σε IDEF3 πρώτιστα επειδή προορίζεται να χρησιμοποιηθεί δεδομένου ότι μια περιγραφή συλλαμβάνει τη μέθοδο. Η εμπειρία με τις σχετικές μεθόδους διαμόρφωσης έχει καταδείξει την ανάγκη να συλληφθούν οι διαφορετικές απόψεις της ίδιας δραστηριότητας. Το IDEF3 παρέχει αυτήν την ικανότητα με την άδεια της πολλαπλάσιας αποσύνθεσης του ίδιου UOB.



ΣΧΗΜΑ 1. ΠΑΡΑΔΕΙΓΜΑ ΔΙΑΓΡΑΜΜΑΤΟΣ IDEF3

Μια επεξεργασία είναι ένα στοιχείο της περιγραφής IDEF3 που συλλαμβάνει τα αντικείμενα που συμμετέχουν σε μια ιδιαίτερη δραστηριότητα και τα γεγονότα και τους περιορισμούς που καθορίζονται σε αυτά τα αντικείμενα και στις περιπτώσεις εκείνης της δραστηριότητας. Κάθε στοιχείο μιας περιγραφής IDEF3 μπορεί να έχει μια επεξεργασία. Είναι στην επεξεργασία ότι οι απαιτήσεις των πόρων των συστημάτων θα πρέπει να συλληφθούν.

Τα διαγράμματα δικτύων κρατικής μετάβασης αντικειμένου (Object State Transition Network) συλλαμβάνουν τις αντικείμενο-κεντροθετημένες απόψεις των διαδικασιών που κόβουν απέναντι τα διαγράμματα διαδικασίας και συνοψίζουν τις επιτρεπόμενες μεταβάσεις. Το σχήμα 2 δείχνει ένα διάγραμμα δικτύων κρατικής μετάβασης αντικειμένου (OSTN).



ΣΧΗΜΑ 2. ΠΑΡΑΔΕΙΓΜΑ ΔΙΑΓΡΑΜΜΑΤΟΣ IDEF3 OSTN

Τα τόξα μετάβασης κρατών και κράτους αντικειμένου είναι τα βασικά στοιχεία ενός διαγράμματος OSTN. Στα διαγράμματα OSTN, τα κράτη αντικειμένου αντιπροσωπεύονται από τους κύκλους και τα τόξα κρατικής μετάβασης αντιπροσωπεύονται από τις γραμμές που συνδέουν τους κύκλους.

Ένα κράτος αντικειμένου καθορίζεται από την άποψη των γεγονότων και των περιορισμών που πρέπει να ισχύουν για τη συνεχή ύπαρξη του αντικειμένου σε εκείνο το κράτος και να χαρακτηρίζεται από τους όρους εισόδων και εξόδων. Οι όροι εισόδων διευκρινίζουν τις απαιτήσεις που πρέπει να καλυφθούν προτού να μπορέσει ένα αντικείμενο μεταβεί σε ένα κράτος. Οι όροι εξόδων χαρακτηρίζουν τους όρους κάτω από τους οποίους ένα αντικείμενο μπορεί μεταβεί από ένα κράτος. Οι περιορισμοί διευκρινίζονται από έναν απλό κατάλογο ζευγαριών ιδιοκτησίας / αξίας ή από μια δήλωση περιορισμού. Οι τιμές των ιδιοτήτων πρέπει να ταιριάζουν με τις διευκρινισμένες τιμές για τις απαιτήσεις που καλύπτονται.

Τα τόξα κρατικής μετάβασης αντιπροσωπεύουν τις επιτρεπόμενες μεταβάσεις μεταξύ των κρατών αντικειμένου εστίασης. Είναι συχνά κατάλληλο να δοθεί έμφαση στη συμμετοχή μιας διαδικασίας σε μια κρατική μετάβαση.

Η σημασία ενός τέτοιου περιορισμού διαδικασίας μεταξύ δύο κρατών αντικειμένου μπορεί να αντιπροσωπευθεί στο IDEF3 με την ένωση μιας αναφοράς UOB με τόξο μετάβασης μεταξύ των δύο κρατών αντικειμένου.

Η KBSI έχει αναπτύξει δύο αυτοματοποιημένα εργαλεία διαμόρφωσης διαδικασίας, *ProSim* και *ProCap*, για να υποστηρίξει τη μέθοδο IDEF3.

3.5 Γενικά για το IDEF4

Η διαισθητική φύση του αντικειμενοστρεφούς προγραμματισμού τον καθιστά ευκολότερο στο να παραγάγει κώδικα. Δυστυχώς, η ευκολία με την οποία το λογισμικό παράγεται επίσης το καθιστά ευκολότερο στο να δημιουργήσει το λογισμικό ενός φτωχού σχεδίου, με συνέπεια τα συστήματα να στερούνται την ικανότητα επαναχρησιμοποίησης, επανασχεδιασμού, και την ικανότητα της συντηρησιμότητας. Η μέθοδος IDEF4 έχει ως σκοπό να βοηθήσει στη σωστή εφαρμογή αυτής της τεχνολογίας.

Με πάνω από τριάντα αντικειμενοστρεφής μεθόδους σχεδιασμού σήμερα, γιατί πρέπει εμείς να επιλέξουμε τη μέθοδο IDEF4; Το IDEF4 βλέπει το αντικειμενοστρεφής σχέδιο ως τμήμα ενός μεγαλύτερου πλαισίου ανάπτυξης συστημάτων, παρά μια αντικειμενοστρεφή μέθοδο ανάλυσης και σχεδιασμού που είναι διφορούμενη. Το IDEF4 τονίζει την αντικειμενοστρεφή διαδικασία σχεδιασμού πέρα από τη γραφική σύνταξη, χρησιμοποιώντας τη γραφική σύνταξη και τα διαγράμματα ως ενισχύσεις στην εστίαση και επικοινωνία με τα σημαντικά ζητήματα σχεδιασμού. Το IDEF4 είναι σημαντικά διαφορετικό από άλλες μεθόδους σχεδιασμού αντικειμένου, πρώτιστα στην υποστήριξή του στρατηγικών "λιγότερης υποχρέωσης" και στην υποστήριξή του για την αξιολόγηση του αντίκτυπου σχεδίου της αλληλεπίδρασης μεταξύ της κληρονομικής κατηγορίας, της σύνθεσης αντικειμένου, της λειτουργικής αποσύνθεσης, και του πολυμορφισμού.

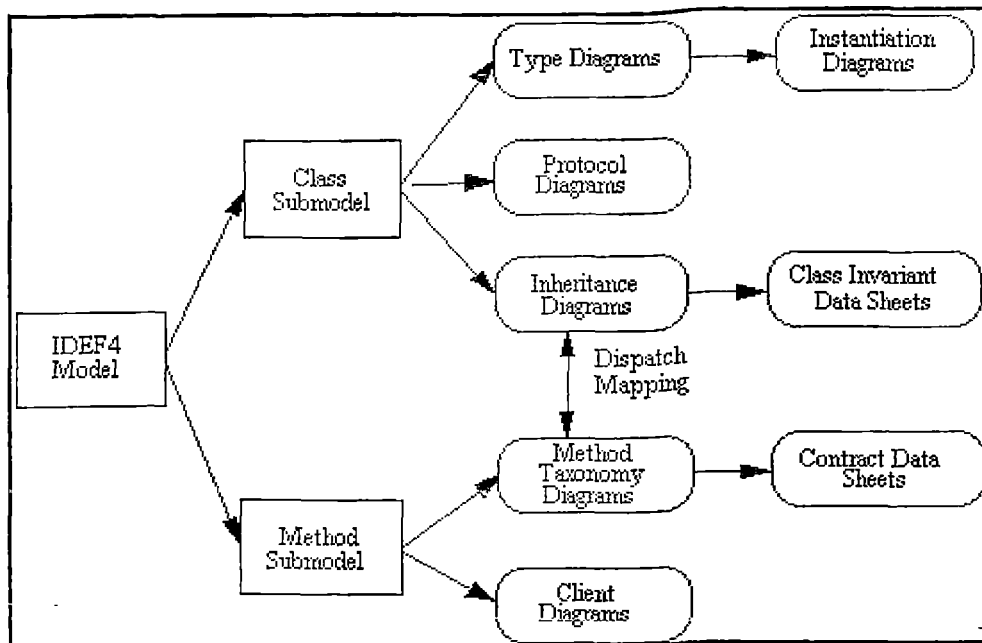
3.5.1 Οι έννοιες του IDEF4

Το IDEF4 διαιρεί την αντικειμενοστρεφή δραστηριότητα σχεδιασμού σε ιδιαίτερα, εύχρηστα χοντρά κομμάτια. Κάθε υπό-δραστηριότητα υποστηρίζεται από μια γραφική σύνταξη που δίνει έμφαση στις αποφάσεις σχεδιασμού που πρέπει να ληφθούν και ο αντίκτυπός τους σε άλλες προοπτικές του σχεδιασμού.

Ούτε ένα διάγραμμα δεν παρουσιάζει όλες τις πληροφορίες που περιλαμβάνονται στο πρότυπο σχεδίου IDEF4, περιοριστικός κατά συνέπεια και επιτρέποντας τη γρήγορη επιθεώρηση των επιθυμητών πληροφοριών. Η προσεκτικά σχεδιασμένη επικάλυψη μεταξύ των τύπων διαγραμμάτων χρησιμεύει στο να εξασφαλίσει συμβατότητα μεταξύ των διαφορετικών υπό-προτύπων.

Η μέθοδος IDEF4 επιτρέπει στο σχεδιαστή να κάνει εύκολα τις ανταλλαγές μεταξύ της σύνθεσης κατηγορίας, της κληρονομιάς κατηγορίας, της λειτουργικής αποσύνθεσης, και του πολυμορφισμού σε ένα σχέδιο. Το IDEF4 είναι περισσότερο από μια γραφική σύνταξη -- η γραφική σύνταξη παρέχει ένα κατάλληλο πλαίσιο για ένα εξελισσόμενο αντικειμενοστρεφές σχέδιο που διευκρινίζεται τελικά στα αμετάβλητα φύλλα δεδομένων κατηγορίας και με τις καθορισμένες συμβάσεις μεθόδου.

Το σχήμα 3 παρουσιάζει τη βασική οργάνωση ενός προτύπου IDEF4. Εννοιολογικά, ένα πρότυπο IDEF4 αποτελείται από δύο υπό-πρότυπα, το υπό-πρότυπο κατηγορίας και το υπό-πρότυπο μεθόδου. Τα δύο υπό-πρότυπα συνδέονται μέσω μιας χαρτογράφησης αποστολών. Αυτές οι δύο δομές συλλαμβάνουν όλες τις πληροφορίες που αντιπροσωπεύονται σε ένα πρότυπο σχεδιασμού. Λόγω του μεγέθους των υπό-προτύπων κατηγορίας και μεθόδου, ο σχεδιαστής αντικειμένου IDEF4 δεν βλέπει ποτέ αυτές τις δομές στην ολότητά τους. Αντ' αυτού, ο σχεδιαστής χρησιμοποιεί τη συλλογή των μικρότερων διαγραμμάτων και των φύλλων δεδομένων που συλλαμβάνουν αποτελεσματικά τις πληροφορίες που αντιπροσωπεύονται στα υπό-πρότυπα κατηγορίας και μεθόδου.



ΣΧΗΜΑ 3. ΟΡΓΑΝΩΣΗ ΕΝΟΣ ΜΟΝΤΕΛΟΥ IDEF4

Το υπό-πρότυπο κατηγορίας αποτελείται από τους ακόλουθους τύπους διαγραμμάτων:

- 1) Διαγράμματα κληρονομιάς που διευκρινίζουν τις σχέσεις κληρονομιάς κατηγορίας
- 2) Ταξινομήστε τα διαγράμματα που διευκρινίζουν τη σύνθεση κατηγορίας
- 3) Διαγράμματα πρωτοκόλλου που διευκρινίζουν τα πρωτόκολλα επίκλησης μεθόδου και
- 4) Διαγράμματα αμεσότητας που περιγράφουν την αμεσότητα του αντικειμένου και τα σενάρια που βοηθούν το σχεδιαστή στην επικύρωση του σχεδίου.

Το υπό-πρότυπο μεθόδου αποτελείται από τους ακόλουθους δύο τύπους διαγραμμάτων:

1) Διαγράμματα ταξινόμησης μεθόδου που ταξινομούν τους τύπους μεθόδου από την ομοιότητα της συμπεριφοράς τους και

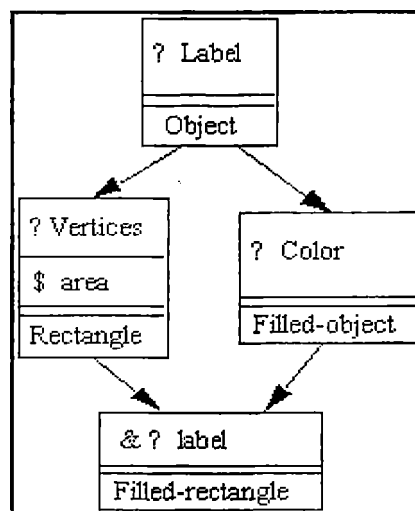
2) Διαγράμματα πελατών που επεξηγούν τους πελάτες και τους προμηθευτές των μεθόδων, για να διευκρινίσει τη λειτουργική αποσύνθεση.

3.5.2 Το υπό-πρότυπο κατηγορίας του IDEF 4

Το υπό-πρότυπο κατηγορίας αποτελείται από τα διαγράμματα κληρονομιάς, τα διαγράμματα τύπων, τα διαγράμματα αμεσότητας, τα διαγράμματα πρωτοκόλλου, και τα κατηγορία-αμετάβλητα φύλλα δεδομένων. Το υπό-πρότυπο κατηγορίας παρουσιάζει την κληρονομιά κατηγορίας και τη δομή σύνθεσης κατηγορίας.

3.5.2.1 Διαγράμματα κληρονομιάς IDEF 4

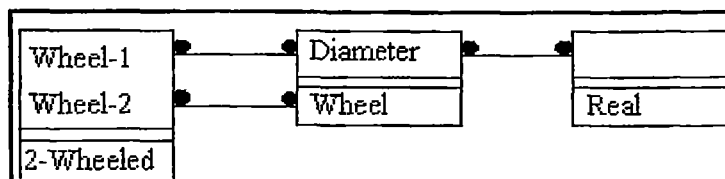
Τα διαγράμματα κληρονομιάς διευκρινίζουν τις σχέσεις κληρονομιάς μεταξύ των κατηγοριών. Παραδείγματος χάριν, το σχήμα 4 παρουσιάζει το ορθογώνιο κατηγορίας κληρονομώντας τη δομή και τη συμπεριφορά άμεσα από το ορθογώνιο και το αντικείμενο των κατηγοριών έμμεσα από το αντικείμενο κατηγορίας.



ΣΧΗΜΑ 4. ΔΙΑΓΡΑΜΜΑ ΚΛΗΡΟΝΟΜΙΑΣ

3.5.2.2 Διαγράμματα αμεσότητας του IDEF 4

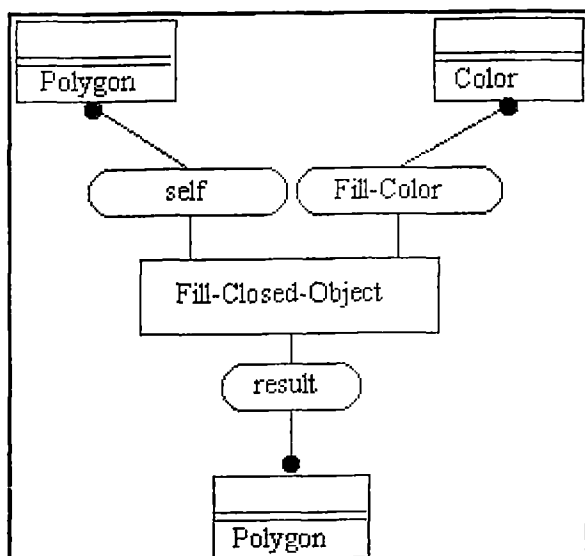
Τα διαγράμματα αμεσότητας συνδέονται με τα διαγράμματα τύπων στο υπό-πρότυπο κατηγορίας. Τα διαγράμματα αμεσότητας περιγράφουν τις προσδοκώμενες καταστάσεις των σύνθετων συνδέσεων μεταξύ των άμεσων αντικειμένων που χρησιμοποιούνται για την επικύρωση του σχεδίου.



ΣΧΗΜΑ 5. ΔΙΑΓΡΑΜΜΑ ΚΑΤΑΤΑΞΗΣ

3.5.2.3 Διαγράμματα πρωτοκόλλου IDEF 4

Τα διαγράμματα πρωτοκόλλου διευκρινίζουν τους τύπους επιχειρήματος κατηγορίας για την επίκληση της μεθόδου. Το σχήμα 6 επεξηγεί ένα διάγραμμα πρωτοκόλλου για το γεμάτο-κλειστό-αντικείμενο: Ζευγάρι στερεότυπης-κατηγορίας πολυγώνων. Από το διάγραμμα, ο αναγνώστης μπορεί να πει ότι το γεμάτο -κλειστό-αντικείμενο θα δεχτεί μια περίπτωση του πολυγώνου κατηγορίας ως αρχικό (μόνο) επιχειρήματά του και μια περίπτωση του χρώματος κατηγορίας ως δευτεροβάθμιο επιχειρήμα, και θα επιστρέψει μια περίπτωση ενός πολυγώνου.



ΣΧΗΜΑ 6.ΔΙΑΓΡΑΜΜΑ ΠΡΩΤΟΚΟΛΛΟΥ

3.5.3 Το υπό-πρότυπο της μεθόδου IDEF4

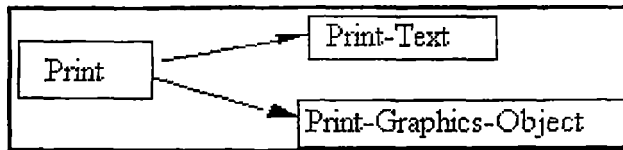
Το υπό-πρότυπο μεθόδου αποτελείται από τα διαγράμματα ταξινόμησης μεθόδου, τα διαγράμματα πελατών, και τα φύλλα στοιχείων.

3.5.3.1 Διαγράμματα ταξινόμησης μεθόδου

Τα διαγράμματα ταξινόμησης μεθόδου ταξινομούν τους τύπους μεθόδου από την ομοιότητα συμπεριφοράς. Ένα διάγραμμα ταξινόμησης μεθόδου ταξινομεί έναν συγκεκριμένο τύπο συμπεριφοράς του συστήματος σύμφωνα με τους περιορισμούς που τοποθετούνται στα σύνολα μεθόδου που αντιπροσωπεύονται στην ταξινόμηση.

Τα βέλη δείχνουν τους πρόσθετους περιορισμούς που τοποθετούνται στα σύνολα μεθόδου. Το σχήμα 6 παρουσιάζει ένα διάγραμμα ταξινόμησης μεθόδου τυπωμένων υλών. Τα σύνολα μεθόδου στην ταξινόμηση ομαδοποιούνται σύμφωνα με τις πρόσθετες συμβάσεις που τοποθετούνται στις μεθόδους σε κάθε σύνολο. Στο παράδειγμα, το πρώτο σύνολο μεθόδου, τυπωμένης ύλης, έχει μια σύμβαση που δηλώνει ότι το αντικείμενο πρέπει να είναι εκτυπώσιμο. Η μέθοδος ύλη-κειμένων είναι η καθορισμένη

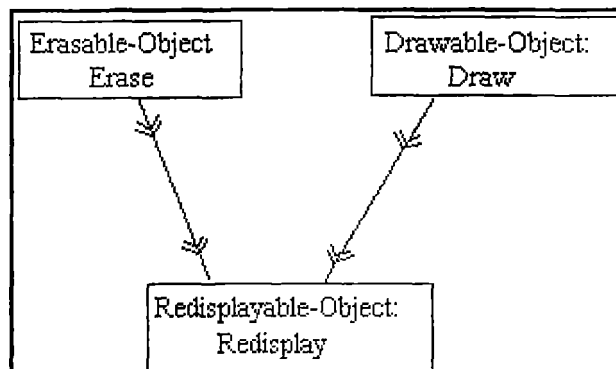
σύμβαση που θα είχε τους περιορισμούς όπως, "το αντικείμενο που τυπώνεται πρέπει να είναι κείμενο."



ΣΧΗΜΑ 7. ΔΙΑΓΡΑΜΜΑ ΤΑΞΙΝΟΜΗΣΗΣ ΜΕΘΟΔΟΥ

3.5.3.2 Διαγράμματα πελατών

Τα διαγράμματα πελατών επεξηγούν στους πελάτες και τους προμηθευτές των ζευγαριών στερεότυπης κατηγορίας. Τα βέλη διπλής διεύθυνσης δείχνουν από τη ρουτίνα που καλείται στην καλούμενη ρουτίνα. Το σχήμα 8 παρουσιάζει διάγραμμα πελατών όπου η Redisplay ρουτίνα που συνδέεται με το Redisplay αντικείμενο κατηγορίας καλεί την Erase ρουτίνα της κατηγορίας Erase αντικειμένου και τη Draw ρουτίνα στην κατηγορία Draw αντικειμένου.



ΣΧΗΜΑ 8. ΔΙΑΓΡΑΜΜΑ ΠΕΛΑΤΩΝ

3.5.3.3 Φύλλα δεδομένων

Επειδή το IDEF4 δεν είναι μόνο μια γραφική γλώσσα, οι πρόσθετες πληροφορίες για τα διαγράμματα κληρονομιάς, τα διαγράμματα ταξινόμησης μεθόδου και τα διαγράμματα τύπων διατηρούνται στα λεπτομερή φύλλα δεδομένων.

Αμετάβλητα φύλλα δεδομένων κατηγορίας

Τα αμετάβλητα φύλλα δεδομένων κατηγορίας συνδέονται με τα διαγράμματα κληρονομιάς και διευκρινίζουν τους περιορισμούς που ισχύουν για κάθε περίπτωση μιας ιδιαίτερης κατηγορίας αντικειμένων. Υπάρχει μια κατηγορία αμετάβλητου φύλλου δεδομένων για κάθε κατηγορία.

Ο περιορισμός, "κάθε τρίγωνο έχει τρεις πλευρές," είναι ένας αμετάβλητος περιορισμός στο τρίγωνο κατηγορίας.

Φύλλα δεδομένων συμβάσεων

Τα φύλλα δεδομένων συμβάσεων συνδέονται με τα σύνολα μεθόδου στα διαγράμματα ταξινόμησης μεθόδου και διευκρινίζουν τις συμβάσεις που οι εφαρμοσμένες μέθοδοι σε ένα σύνολο μεθόδου πρέπει να ικανοποιήσουν. Υπάρχει ένα φύλλο δεδομένων συμβάσεων για κάθε σύνολο μεθόδου.

3.5.4 Συμπεράσματα

Η μέθοδος IDEF4, που αναπτύσσεται κάτω από την εγγυοδοσία του εργαστηρίου Armstrong Πολεμικής Αεροπορίας των Η.Π.Α., έχει ως σκοπό να βοηθήσει στη σωστή εφαρμογή της αντικειμενοστρεφούς τεχνολογίας. Το IDEF4 αναπτύχθηκε από τους επαγγελματικούς αντικειμενοστρεφείς σχεδιαστές και τους προγραμματιστές. Ο σημαντικότερος λόγος για τη μέθοδο IDEF4 είναι ότι βλέπει το αντικειμενοστρεφές σχέδιο ως τμήμα ενός μεγαλύτερου πλαισίου ανάπτυξης συστημάτων, παρά μια αντικειμενοστρεφή μέθοδο ανάλυσης και σχεδιασμού που είναι όλα σε κάθεμία. Τονίζει την αντικειμενοστρεφή διαδικασία σχεδίου

πέρα από τη γραφική σύνταξη, χρησιμοποιώντας τη γραφική σύνταξη και τα διαγράμματα ως ενισχύσεις για να εστιάσει πάνω στα σημαντικά ζητήματα του σχεδιασμού.

Η KBSI έχει αναπτύξει ένα αυτοματοποιημένο αντικειμενοστρεφές εργαλείο σχεδίου, SmartClass, για να υποστηρίξει τη μέθοδο IDEF4.

3.6 Γενικά για το IDEF5

Ιστορικά, οι οντολογίες προέκυψαν από τον κλάδο της φιλοσοφίας γνωστής ως μεταφυσική, η οποία εξετάζει τη φύση της πραγματικότητας -- αυτό που δηλαδή υπάρχει. Ο παραδοσιακός στόχος της οντολογικής έρευνας, ειδικότερα, είναι να διαιρεθεί ο κόσμος "στις ενώσεις του," για να ανακαλύψει εκείνες τις θεμελιώδεις κατηγορίες ή τα είδη που καθορίζουν τα αντικείμενα του κόσμου. Έτσι η φυσική επιστήμη παρέχει ένα άριστο παράδειγμα της οντολογικής έρευνας. Παραδείγματος χάριν, ένας στόχος της μικρο-ατομικής φυσικής είναι να αναπτυχθεί μια ταξινόμηση των πιο βασικών ειδών αντικειμένων που υπάρχουν μέσα στο φυσικό κόσμο (π.χ., πρωτόνια, ηλεκτρόνια, νετρόνια). Ομοίως, οι βιολογικές επιστήμες επιδιώκουν να ταξινομήσουν και να περιγράψουν τα διάφορα είδη οργανισμών διαβίωσης που αποικούν στον πλανήτη.

Οι φυσικοί και αφηρημένοι κόσμοι της καθαρής επιστήμης, εντούτοις, δεν εξαντλούν τις εφαρμόσιμες περιοχές της οντολογίας. Υπάρχουν απέραντα, ανθρωπίνως σχεδιασμένα και ανθρωπίνως-κατασκευασμένα συστήματα όπως οι εγκαταστάσεις κατασκευής, οι επιχειρήσεις, οι στρατιωτικές βάσεις, και τα πανεπιστήμια στις οποίες η οντολογική έρευνα είναι εξίσου σχετική και εξίσου σημαντική. Σε αυτά τα ανθρωπίνως-δημιουργημένα συστήματα, η οντολογική έρευνα παρακινείται πρώτιστα από την ανάγκη να καταλάβει, να σχεδιάσει, να κατασκευάσει και να ρυθμίσει τέτοια συστήματα αποτελεσματικά. Συνεπώς, είναι χρήσιμο να προσαρμοστούν οι παραδοσιακές τεχνικές οντολογικής έρευνας στις φυσικές επιστήμες σε αυτές τις περιοχές επίσης.

Η μέθοδος IDEF5 παρέχει μια θεωρητικά και εμπειρικά βάσιμη μέθοδο με σκοπό συγκεκριμένα να βοηθήσει στη δημιουργία, την τροποποίηση, και τη διατήρηση των οντολογιών. Οι τυποποιημένες διαδικασίες, η δυνατότητα να αντιπροσωπευθούν οι πληροφορίες οντολογίας σε μια διαισθητική και φυσική μορφή,

και τα υψηλότερα ποιοτικά αποτελέσματα που επιτρέπονται μέσω της εφαρμογής IDEF5 χρησιμεύουν επίσης στο να μειωθεί το κόστος αυτών των δραστηριοτήτων.

3.6.1 Βασικές αρχές της οντολογικής ανάλυσης

Η οντολογική ανάλυση ολοκληρώνεται με την εξέταση του λεξιλογίου που χρησιμοποιείται για να συζητήσει τα χαρακτηριστικά αντικείμενα και τις διαδικασίες που συνθέτουν την περιοχή, που αναπτύσσει τους αυστηρούς ορισμούς των βασικών όρων σε εκείνο το λεξιλόγιο και που χαρακτηρίζει τις λογικές συνδέσεις μεταξύ εκείνων των όρων. Το προϊόν αυτής της ανάλυσης, μια οντολογία, είναι ένα λεξιλόγιο περιοχών πλήρες με ένα σύνολο ακριβών ορισμών, ή αξιώματα, τα οποία περιορίζουν τις έννοιες των όρων για να επιτρέψουν αρκετά τη συνεπή ερμηνεία των στοιχείων που χρησιμοποιούν εκείνο το λεξιλόγιο.

Μια οντολογία περιλαμβάνει έναν κατάλογο των χρησιμοποιούμενων όρων σε μια περιοχή, των κανόνων που κυβερνούν πώς εκείνοι οι όροι μπορούν να συνδυαστούν για να κάνουν τις έγκυρες δηλώσεις για τις καταστάσεις σε εκείνη την περιοχή, και των *εγκεκριμένων συμπερασμάτων* που μπορεί να γίνουν όταν χρησιμοποιούνται τέτοιες δηλώσεις σε εκείνη την περιοχή. Σε κάθε περιοχή, υπάρχουν φαινόμενα που οι άνθρωποι σε εκείνη την περιοχή κάνουν διακρίσεις όπως (τα εννοιολογικά ή φυσικά) αντικείμενα, τις ενώσεις, και καταστάσεις. Μέσω των διάφορων γλωσσικών μηχανισμών, συνδέουμε τους καθορισμένες περιγραφές (π.χ., ονόματα, ονοματικές φράσεις, κ.λπ...) σε εκείνα τα φαινόμενα. Στα πλαίσια της οντολογίας, μια *σχέση* είναι ένας καθορισμένος περιγραφέας που αναφέρεται σε μια ένωση, στον πραγματικό κόσμο ένας όρος είναι ένας καθορισμένος περιγραφέας που αναφέρεται σε ένα αντικείμενο ή μια κατάσταση-όπως υπάρχει στον πραγματικό κόσμο.

Στην κατασκευή μιας οντολογίας, προσπαθούμε να καταχωρήσουμε τις περιγραφές (όπως ένα λεξικό στοιχείων) και να δημιουργήσουμε ένα πρότυπο της περιοχής, εάν περιγράφεται με εκείνες τις περιγραφές. Κατά συνέπεια, στην οικοδόμηση μιας οντολογίας, πρέπει να εκτελέσετε τρεις στόχους:

1) καταχωρήστε τους όρους

2) συλλάβετε τους περιορισμούς που κυβερνούν, για το πώς εκείνοι οι όροι μπορούν να χρησιμοποιηθούν για να κάνουν τις περιγραφικές δηλώσεις για την περιοχή και

3) κατασκευή ενός προτύπου που, όταν παρέχεται μια συγκεκριμένη περιγραφική δήλωση, να μπορεί να παραγάγει τις "κατάλληλες" πρόσθετες περιγραφικές δηλώσεις.

Οι κατάλληλες περιγραφικές δηλώσεις έκφρασης σημαίνουν δύο πράγματα. Κατ' αρχάς, επειδή υπάρχει γενικά ένας μεγάλος αριθμός πιθανών δηλώσεων που θα μπορούσαν να παραχθούν, το πρότυπο παράγει μόνο το υποσύνολο που είναι "χρήσιμο" στο πλαίσιο. Δεύτερον, οι περιγραφικές δηλώσεις που παράγονται αντιπροσωπεύουν τα γεγονότα ή τις πεποιθήσεις που κατέχει χαρακτηριστικά ένας ευφυής πράκτορας στην περιοχή που είχε λάβει τις ίδιες πληροφορίες. Στο πρότυπο λέγεται έπειτα να ενσωματώσει τα εγκεκριμένα συμπεράσματα στην περιοχή. Λέγεται επίσης να "χαρακτηρίσει" τη συμπεριφορά των αντικειμένων και των ενώσεων στην περιοχή. Κατά συνέπεια, μια οντολογία είναι παρόμοια με ένα στοιχείο-λεξικού αλλά περιλαμβάνει και μια γραμματική και ένα πρότυπο της συμπεριφοράς της περιοχής.

3.6.2 Οι έννοιες του IDEF5

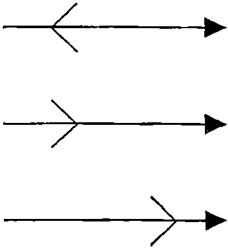
Η διαδικασία ανάπτυξης οντολογίας IDEF5 αποτελείται από τις ακόλουθες πέντε δραστηριότητες.

- **Οργάνωση και Ορισμός Αρμοδιοτήτων.** Η οργανωτική δραστηριότητα ο ορισμός των αρμοδιοτήτων καθιερώνουν το σκοπό, την άποψη, και το πλαίσιο για το αναπτυξιακό έργο οντολογίας, και ορίζει τους ρόλους στα μέλη των ομάδων.
- **Συλλογή δεδομένων.** Κατά τη διάρκεια της συλλογής δεδομένων, το ακατέργαστο στοιχείο που απαιτείται για την ανάπτυξη οντολογίας αποκτιέται.
- **Ανάλυση στοιχείων.** Η ανάλυση στοιχείων περιλαμβάνει την ανάλυση των στοιχείων για να διευκολύνει την εξαγωγή οντολογίας.
- **Αρχική ανάπτυξη οντολογίας.** Η initial δραστηριότητα ανάπτυξης οντολογίας αναπτύσσει μια προκαταρκτική οντολογία από τα στοιχεία που συγκεντρώνονται.
- **Καθορισμός και επικύρωση οντολογίας.** Η οντολογία καθορίζεται και επικυρώνεται για να ολοκληρωθεί η διαδικασία ανάπτυξης.

3.6.3 Γλώσσες οντολογίας του IDEF5

Η υποστήριξη της διαδικασία ανάπτυξης οντολογίας είναι οι γλώσσες οντολογίας IDEF5'S. Υπάρχουν δύο τέτοιες γλώσσες: η **σχηματική γλώσσα IDEF5** και η **γλώσσα επεξεργασίας IDEF5**.

Η σχηματική γλώσσα είναι μια γραφική γλώσσα, που προσαρμόζεται συγκεκριμένα για να επιτρέψει στους εμπειρογνώμονες περιοχών να εκφράσουν τις πιο κοινές μορφές οντολογικών πληροφοριών (σχήμα 9). Αυτό επιτρέπει στους μέσους χρήστες και να εισάγουν τις βασικές πληροφορίες που απαιτούνται για μια οντολογία πρώτων-περικοπών και για να αυξήσει ή να αναθεωρήσει τις υπάρχουσες οντολογίες με τις νέες πληροφορίες. Η άλλη γλώσσα είναι η γλώσσα επεξεργασίας IDEF5, μια δομημένη κειμενική γλώσσα που επιτρέπει το λεπτομερή χαρακτηρισμό των στοιχείων στην οντολογία.

<p>ΕΙΔΗ ΣΥΜΒΟΛΩΝ, ΑΤΟΜΙΚΑ ΣΥΜΒΟΛΑ, ΑΝΑΦΟΡΕΣ</p>	<p>ΣΥΜΒΟΛΑ ΣΧΕΣΕΩΝ, ΚΥΡΙΑ ΣΥΜΒΟΛΑ ΜΕΤΑΔΟΣΗΣ</p>	<p>ΣΥΜΒΟΛΑ ΔΙΑΔΙΚΑΣΙΑΣ, ΣΥΜΒΟΛΑ ΣΥΝΔΕΣΕΩΝ ΕΝΩΣΕΙΣ</p>				
<p><u>ΕΙΔΗ ΣΥΜΒΟΛΩΝ</u></p> <div data-bbox="192 750 469 924" style="border: 1px solid black; border-radius: 50%; padding: 10px; text-align: center;"> ΕΙΔΟΣ ΕΤΙΚΕΤΑΣ </div> <p>ΑΤΟΜΙΚΟ ΣΥΜΒΟΛΟ</p> <div data-bbox="215 1017 469 1192" style="border: 1px solid black; border-radius: 50%; padding: 10px; text-align: center;"> ΑΤΟΜΙΚΟ ΣΥΜΒΟΛΟ ● </div> <p>ΑΝΑΦΟΡΑ</p> <table border="1" data-bbox="170 1280 469 1450" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 15%; text-align: center;">ID</td> <td style="text-align: center;">ΟΝΟΜΑ</td> </tr> <tr> <td colspan="2" style="text-align: center;">ΕΤΙΚΕΤΑ ΑΝΑΦΟΡΑΣ</td> </tr> </table>	ID	ΟΝΟΜΑ	ΕΤΙΚΕΤΑ ΑΝΑΦΟΡΑΣ		<p><u>N—ΠΕΡΙΟΧΗΣ ΠΡΩΤΗΣ ΕΝΤΟΛΗΣ</u></p> <p><u>ΣΥΜΒΟΛΟ ΣΧΕΣΗΣ</u></p> <div data-bbox="586 838 833 936" style="border: 1px solid black; border-radius: 10px; padding: 5px; text-align: center;"> ΕΤΙΚΕΤΑ ΣΧΕΣΗΣ </div> <p><u>ΔΕΥΤΕΡΗ ΠΕΡΙΟΧΗ ΠΡΩΤΗ ΕΝΤΟΛΗ</u></p> <p style="text-align: center;">ΕΤΙΚΕΤΑ ΣΧΕΣΗΣ →</p> <p><u>ΔΕΥΤΕΡΗ ΠΕΡΙΟΧΗ ΔΕΥΤΕΡΗ ΕΝΤΟΛΗ</u></p> <p style="text-align: center;">← ΕΤΙΚΕΤΑ ΣΧΕΣΗΣ</p> <p><u>ΚΥΡΙΑ ΣΥΜΒΟΛΑ ΜΕΤΑΔΟΣΗΣ</u></p> <p>ΒΕΛΟΣ ΑΔΥΝΑΜΗΣ ΜΕΤΑΔΟΣΗΣ</p> <p style="text-align: center;">→ ○</p> <p>ΒΕΛΟΣ ΙΣΧΥΡΗΣ ΜΕΤΑΔΟΣΗΣ</p> <p style="text-align: center;">→ ○ →</p> <p>ΣΤΙΓΜΙΑΙΟΣ ΣΗΜΕΙΩΤΗΣ ΜΕΤΑΔΟΣΗΣ</p> <p style="text-align: center;">Δ</p>	<p><u>ΣΥΜΒΟΛΟ ΔΙΑΔΙΚΑΣΙΑΣ</u></p> <div data-bbox="976 750 1248 873" style="border: 1px solid black; padding: 5px;"> ΕΤΙΚΕΤΑ ΔΙΑΔΙΚΑΣΙΑΣ </div> <p><u>ΣΥΜΒΟΛΑ ΣΥΝΔΕΣΕΩΝ</u></p> <div style="text-align: center;">  </div> <p><u>ΕΝΩΣΕΙΣ</u></p> <div style="display: flex; justify-content: space-around; align-items: center;"> <div style="border: 1px solid black; border-radius: 50%; padding: 10px; text-align: center;">X</div> <div style="border: 1px solid black; border-radius: 50%; padding: 10px; text-align: center;">0</div> </div>
ID	ΟΝΟΜΑ					
ΕΤΙΚΕΤΑ ΑΝΑΦΟΡΑΣ						

ΣΧΗΜΑ 9.ΒΑΣΙΚΑ ΣΥΜΒΟΛΑ ΤΗΣ ΓΛΩΣΣΑΣ ΕΠΕΞΕΡΓΑΣΙΑΣ ΤΟΥ IDEF5

Οι διάφοροι τύποι διαγραμμάτων, ή οι σχηματικές αναπαραστάσεις, μπορούν να κατασκευαστούν στη σχηματική γλώσσα IDEF5. Ο σκοπός αυτών των σχηματικών αναπαραστάσεων, όπως οποιασδήποτε αντιπροσώπευσης, είναι να αντιπροσωπευθούν οι πληροφορίες οπτικά. Κατά συνέπεια, οι σημασιολογικοί κανόνες πρέπει να παρασχεθούν για την ερμηνεία κάθε πιθανής σχηματικής αναπαράστασης. Αυτοί οι κανόνες παρέχονται με την περιγραφή των κανόνων για τα πιο βασικά κατασκευάσματα της γλώσσας, κατόπιν εφαρμόζονται κατ' επανάληψη στα πιο σύνθετα κατασκευάσματα. Εντούτοις, ο χαρακτήρας της σημασιολογίας για τη σχηματική γλώσσα διαφέρει από το χαρακτήρα της σημασιολογίας για άλλες γραφικές γλώσσες. Συγκεκριμένα, σε κάθε βασική σχηματική αναπαράσταση παρέχεται μόνο μια σημασιολογία προεπιλογής που μπορεί να αγνοηθεί στη γλώσσα επεξεργασίας. Ο λόγος για αυτό είναι ότι ο κύριος σκοπός της σχηματικής γλώσσας είναι να χρησιμεύσει ως μια ενίσχυση για την κατασκευή των οντολογιών. Εκείνος ο στόχος μειώνεται στη γλώσσα επεξεργασίας.

Η σχηματική γλώσσα είναι, εντούτοις, χρήσιμη για τις πρώτες οντολογίες στις οποίες η κεντρική ανησυχία είναι να καταγραφούν, με έναν τραχύ τρόπο, τα βασικά στοιχεία που υπάρχουν σε μια περιοχή, τις χαρακτηριστικές ιδιότητές τους και τις εμφανείς σχέσεις που μπορούν να ληφθούν μεταξύ των αντικειμένων εκείνων των ειδών και μεταξύ των ίδιων ειδών. Συνεπώς, τα βασικά κατασκευάσματα της σχηματικής γλώσσας έχουν ως σκοπό συγκεκριμένα να συλλάβουν αυτό το είδος πληροφοριών.

3.6.4 Σχηματικοί τύποι IDEF5

Ορισμένες σχέσεις υπερισχύουν όταν εκφράζουν οι άνθρωποι τη γνώση τους για μια περιοχή λόγω της ιδιαίτερης σημασίας τους, αυτές οι σχέσεις συμπεριλαμβάνονται ρητά στη γλώσσα IDEF5. Υπάρχουν τέσσερις αρχικοί σχηματικοί τύποι που προέρχονται από τη βασική σχηματική γλώσσα IDEF5 που μπορεί να χρησιμοποιηθεί για να συλλάβει τις πληροφορίες οντολογίας άμεσα σε μια μορφή που είναι διαισθητική στον εμπειρογνώμονα περιοχών.

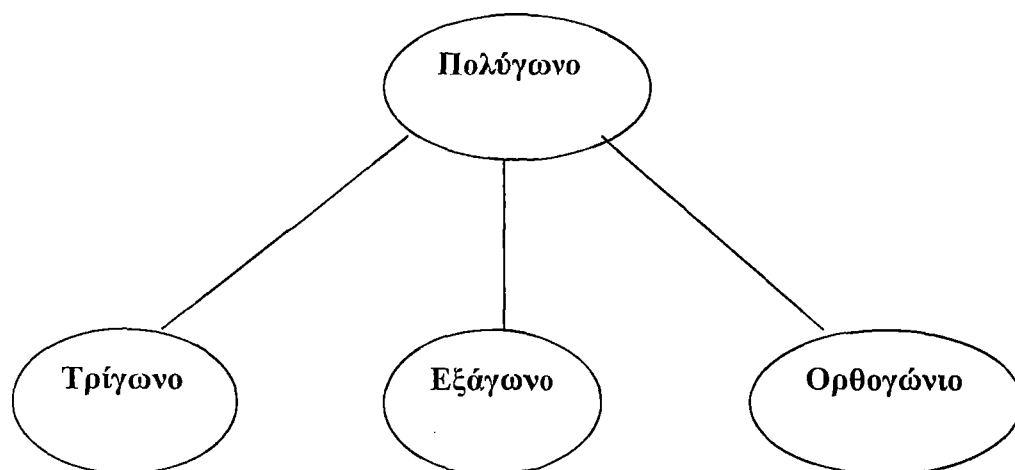
1. Σχηματικές αναπαραστάσεις ταξινόμησης

Οι σχηματικές αναπαραστάσεις ταξινόμησης παρέχουν τους μηχανισμούς για τους ανθρώπους ώστε να οργανώσουν τη γνώση τους σε λογικές κατηγορίες ταξινόμησης. Από την ιδιαίτερη αξία είναι δύο τύποι ταξινομήσεων: η *συμπεριλειπτική περιγραφή* και η *φυσική ταξινόμηση ειδών*.

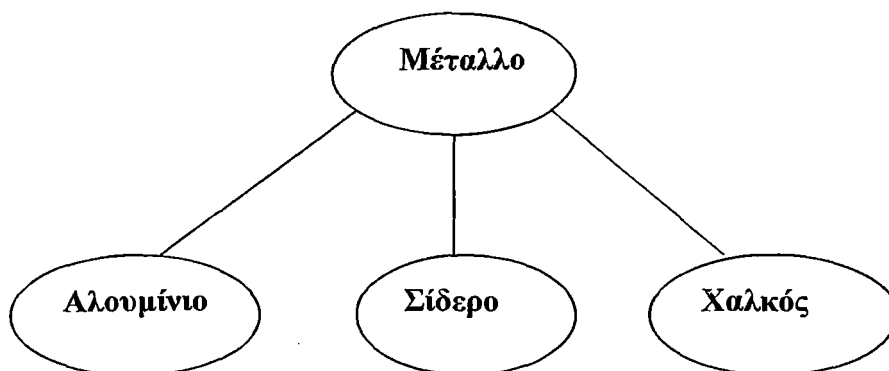
Στη συμπεριλειπτική περιγραφή, οι ιδιότητες καθορισμού του "κορυφαίου επιπέδου" των ειδών K περιλαμβάνονται στη ταξινόμηση, καθώς επίσης και εκείνοι όλων των υπό-ειδών της, αποτελούν τους αυστηρούς απαραίτητους και ικανοποιητικούς όρους για την ιδιότητα μέλους σε εκείνα τα είδη. Επιπλέον, οι ιδιότητες καθορισμού όλων των υπό-ειδών "εντάσσονται" από τις ιδιότητες καθορισμού του K υπό την έννοια ότι οι ιδιότητες καθορισμού κάθε είδους συνεπάγονται στις ιδιότητες καθορισμού του K άρα οι ιδιότητες καθορισμού του K αποτελούν μια γενικότερη έννοια.

Αντιθέτως, η φυσική ταξινόμηση ειδών δεν υποθέτει ότι υπάρχουν αυστηρά ευπροσδιόριστοι, απαραίτητοι και ικανοποιητικοί όροι για την ιδιότητα ενός μέλους στο κορυφαίο επίπεδο ειδών K . Εν τούτοις, υπάρχουν μερικές κρυμμένες ιδιότητες κάτω από δομικές ιδιότητες περιπτώσεων που, ειδικεύονται με τους διάφορους τρόπους, με τους οποίους παράγονται τα υπό-είδη του K . Η διαφορά μεταξύ των δύο τύπων ταξινομήσεων είναι διευκρινισμένη στο παράδειγμα που ακολουθεί.

Συμπεριλειπτική περιγραφή



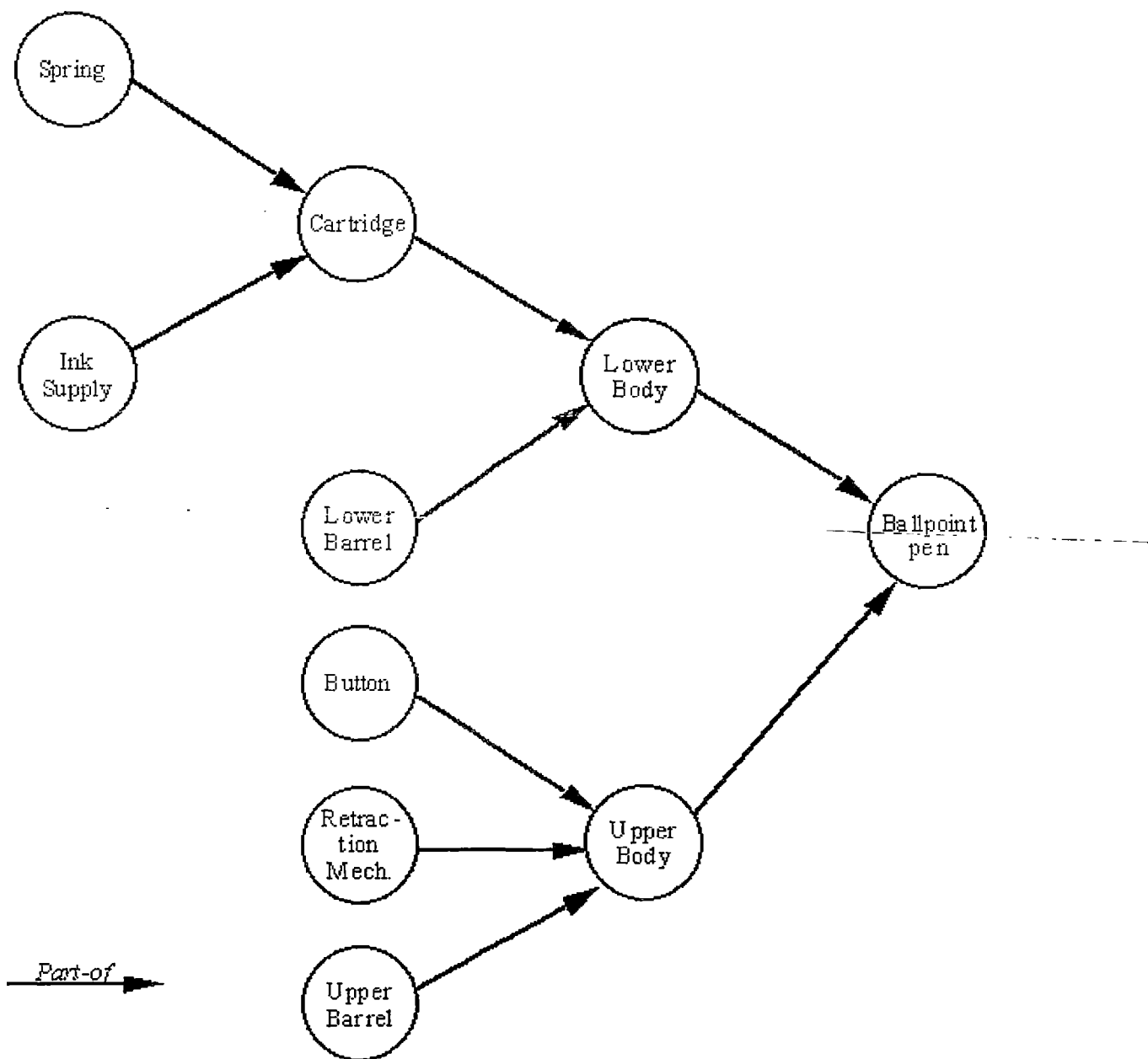
Φυσική ταξινόμηση ειδών



ΣΧΗΜΑ 10. ΔΙΑΦΟΡΕΤΙΚΟΙ ΤΥΠΟΙ ΤΑΞΙΝΟΜΗΣΗΣ

2. Σχηματικές αναπαραστάσεις σύνθεσης

Οι σχηματικές αναπαραστάσεις σύνθεσης χρησιμεύουν ως οι μηχανισμοί που θα αντιπροσωπεύσουν γραφικά μέρος της σχέσης που είναι τόσο κοινή μεταξύ των συστατικών μιας οντολογίας. Ειδικότερα, αυτή η ικανότητα επιτρέπει στους χρήστες να εκφράσουν τα γεγονότα για τη σύνθεση ενός δεδομένου είδους αντικειμένου. Παραδείγματος χάριν, κάποιος να θελήσει να αντιπροσωπεύσει τη συστατική δομή για ένα ορισμένο είδος στυλό, ballpoint.



ΣΧΗΜΑ 11. ΣΧΗΜΑΤΙΚΗ ΠΕΡΙΓΡΑΦΗ ΓΙΑ ΕΝΑ ΕΙΔΟΣ ΣΤΥΛΟ Ballpoint

Όπως η σχηματική αναπαράσταση παρουσιάζει, ένα στυλό ballpoint στην εν λόγω περιοχή έχει και ένα ανώτερο σώμα και ένα χαμηλότερο σώμα, όπου το πρώτο αποτελείται από ένα κουμπί, έναν μηχανισμό απόσυρσης, και ένα ανώτερο βαρέλι ενώ το τελευταίο αποτελείται από ένα χαμηλότερο βαρέλι και μια κασέτα, η οποία αποτελείται στη συνέχεια από ένα ελατήριο και έναν ανεφοδιασμό μελανιού.

3. Σχηματικές αναπαραστάσεις σχέσης

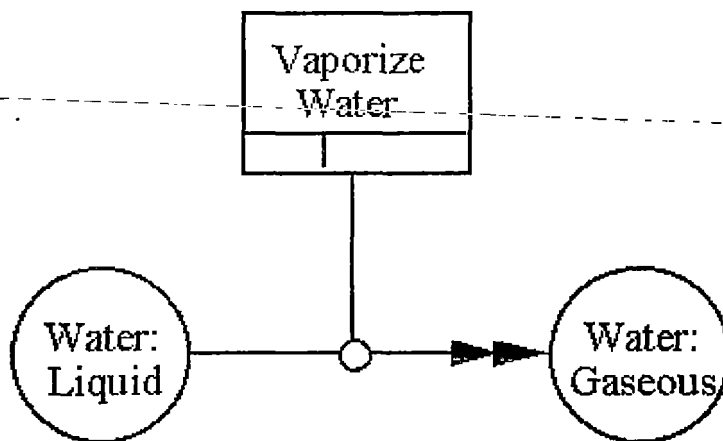
Οι σχηματικές αναπαραστάσεις σχέσης επιτρέπουν στους υπεύθυνους για την ανάπτυξη οντολογίας να απεικονίσουν και να καταλάβουν τις σχέσεις μεταξύ των ειδών σε μια περιοχή, και να μπορούν επίσης να χρησιμοποιηθούν για να συλλάβουν και να επιδείξουν τις σχέσεις μεταξύ των πρώτων-εντολών των σχέσεων. Το κίνητρο για την ανάπτυξη αυτής της ικανότητας είναι ότι οι άνθρωποι περιγράφουν συχνά και ανακαλύπτουν τις νέες έννοιες από την άποψη των υπαρχών εννοιών. Αυτό σημαίνει δημιουργία και καθορισμό των νέων εννοιών που είναι σύμφωνες με τη θεωρία εκμάθησης Ausubel, όπου η εκμάθηση εμφανίζεται συχνά με την ένταξη των νέων πληροφοριών κάτω από τη γενικότητα των περισσότερων συμπεριλαμβανομένων εννοιών. Με βάση αυτήν την υπόθεση, ένας φυσικός τρόπος να περιγραφεί μια νέα (ή ανεπαρκώς κατανοητή) σχέση είναι να συνδεθεί με μια σχέση που ήδη γίνεται καλά κατανοητή και, γενικότερα, για να ταξινομήσει τη θέση της σε ένα "εννοιολογικό διάστημα" άλλων σχέσεων. Η βιβλιοθήκη σχέσης IDEF5 (που περιλαμβάνεται ως παράρτημα στην *οντολογία IDEF5 συλλαμβάνετε στην έκθεση μεθόδου*) παρέχει μια αναφορά βασικών γραμμών για να βοηθήσει τους χρήστες να ανακαλύψουν και να χαρακτηρίσουν τις σχέσεις.

4. Σχηματικά αντικείμενα βάσης

Επειδή δεν υπάρχει κανένα καθαρό τμήμα μεταξύ των πληροφοριών για τα είδη και τη βάση των πληροφοριών για τις διαδικασίες, η σχηματική γλώσσα IDEF5 επιτρέπει στους σχεδιαστές να εκφράζουν με πιο λεπτομερή τρόπο τις αντικείμενο-θετημένες πληροφορίες διαδικασίας (δηλ., πληροφορίες για τα είδη αντικειμένων και των διάφορων βάσεων που μπορούν να είναι σχετικές με ορισμένες διαδικασίες). Τα διαγράμματα που χτίζονται από αυτά τα κατασκευάσματα είναι γνωστά ως αντικείμενο-βασικές σχηματικές αναπαραστάσεις.

Δύο τύποι αλλαγών μπορούν να παρατηρηθούν στα αντικείμενα που υποβάλλονται στις διαδικασίες: αλλαγή σε είδος και αλλαγή σε βάση. Δεν υπάρχει καμία επίσημη διαφορά μεταξύ αυτών των δύο τύπων αλλαγών: τα αντικείμενα ενός δεδομένου είδους **Κ** που είναι σε μια ορισμένη βάση μπορούν απλά να θεωρηθούν ως ένα υπό-είδος του **Κ**. Για επίσημους λόγους, παραδείγματος χάριν, το **ζεστό νερό** μπορεί να θεωρηθεί ως υπό-είδος του **νερού**.

Εντούτοις, είναι χρήσιμο να διακριθούν τα δύο στη σχηματική γλώσσα για να δείξει ρητά το είδος του πράγματος που είναι σε μια ορισμένη βάση. Αυτό γίνεται χρησιμοποιώντας τη σημείωση των άνω και κάτω τελειών (π.χ., **kind:state**). Παραδείγματος χάριν, το ζεστό νερό θα υποδειχθεί από την ετικέτα **water:warm**, το παγωμένο νερό από **water:frozen**. Η σχηματική γλώσσα IDEF5 επιτρέπει στους σχεδιαστές να αντιπροσωπεύσουν οπτικά τις αλλαγές στο είδος ή να δηλώσουν επίσης και τις διαδικασίες ενός αντικειμένου που επιφέρουν τέτοιες αλλαγές.



ΣΧΗΜΑ 12.ΠΑΡΑΔΕΙΓΜΑ ΣΧΗΜΑΤΙΚΗΣ ΜΕΤΑΔΟΣΗΣ ΑΝΤΙΚΕΙΜΕΝΟΥ ΒΑΣΗΣ

3.6.5 Συμπεράσματα

Η φύση οποιασδήποτε περιοχής αποκαλύπτεται μέσω τριών στοιχείων:

- 1) το λεξιλόγιο που χρησιμοποιείται για να συζητήσει τα χαρακτηριστικά και τις διαδικασίες που αποτελούνται στην περιοχή,
- 2) αυστηροί ορισμοί των βασικών όρων σε εκείνο το λεξιλόγιο, και
- 3) χαρακτηρισμός των λογικών συνδέσεων μεταξύ εκείνων των όρων.

Μια οντολογία είναι ένα λεξιλόγιο περιοχών πλήρες με ένα σύνολο ακριβών ορισμών, ή αξιώματα, τα οποία περιορίζουν τις έννοιες των όρων σε εκείνο το λεξιλόγιο για να επιτρέψουν αρκετά τη συνεπή ερμηνεία των στοιχείων. Η μέθοδος IDEF5 παρέχει μια δομημένη τεχνική, από την οποία ένας εμπειρογνώμονας περιοχών μπορεί αποτελεσματικά να αναπτύξει και να διατηρήσει τις χρησιμοποιήσιμες, ακριβείς οντολογίες περιοχών. Η μέθοδος IDEF5 χρησιμοποιείται για να κατασκευάσει τις οντολογίες με τη σύλληψη των ισχυρισμών για τα πραγματικά αντικείμενα, τις ιδιότητές τους, και τις αλληλεξαρτήσεις τους.

Κεφάλαιο 4

ΣΥΓΚΡΙΣΗ ΤΩΝ ΔΥΟ ΜΕΘΟΔΩΝ ΜΟΝΤΕΛΟΠΟΙΗΣΗΣ

4.1 Σύγκριση των μοντέλων UML / IDEF

Σαν γενικό συμπέρασμα οι προσεγγίσεις και της UML και του IDEF μπορούν να χρησιμοποιηθούν για να απεικονίσουν σχεδόν οποιαδήποτε χρήσιμη λειτουργία μιας επιχείρησης.

Η οικογένεια των γλωσσών IDEF έχει περίπου 30 χρόνια εξέλιξης και μερικές κυβερνητικές υπηρεσίες των Ηνωμένων Πολιτειών κρύβονται πίσω από αυτές (Π.χ. το υπουργείο άμυνας, η πολεμική αεροπορία κ.λ.π.). Οι καινούριες εκδόσεις όπως του IDEF4, IDEF5 και πάνω καλύπτουν και τις καινούριες - διαφοροποιημένες απαιτήσεις.

Η UML είναι μια νέα γλώσσα μοντελοποίησης σε σύγκριση με την IDEF και στοχεύει περισσότερο προς την ανάπτυξη λογισμικού. Ο σωστός χρόνος εμφάνισής της και η υποστήριξη των κατασκευαστικών εργαλείων λογισμικού συνετέλεσαν στην μεγάλη άνθησή της. Όπως μπορούμε να δούμε από τα παραδείγματα που παρουσιάστηκαν η UML μπορεί να χρησιμοποιηθεί αποτελεσματικά μόνο όταν συμπληρώνεται από

- Πρότυπα σχεδιασμού που επιτρέπουν τον σχεδιασμό της διάδοσης της γνώσης
- Συγκεκριμένες επεκτάσεις που επιτρέπουν την αποτελεσματική σύλληψη των επιχειρηματικών διαδικασιών.

Είναι αρκετά περίεργο ότι μια γλώσσα που περιέχει εννιά διαφορετικά διαγράμματα χρειάζεται ακόμα να επεκταθεί και ότι έχει ελλείψεις στον τομέα μοντελοποίησης της διάδοσης της γνώσης. Ο μεγάλος αριθμός διαφορετικών διαγραμμάτων (εκ των οποίων δυο μόνο στοχεύουν αποκλειστικά στην ανάπτυξη λογισμικού) προέρχεται από τον τρόπο που η UML δημιουργήθηκε (από ενοποίηση μεθόδων που προϋπήρχαν) και δε φαίνεται να βοηθάει πολύ όταν απομακρυνθούμε από τη περιοχή σχεδιασμού του λογισμικού.

Από τη θετική πλευρά, η UML είναι εύκολο να την μάθει κάποιος και μπορεί να εισάγει τους ενδιαφερόμενους χρήστες σε προχωρημένες αρχές σχεδιασμού όπως είναι π.χ. τα μετα-μοντέλα.

Οι χρήστες της δεν είναι υποχρεωμένοι να χρησιμοποιούν όλα τα είδη των διαγραμμάτων της και είναι ελεύθεροι να σχεδιάσουν τα μοντέλα τους με όποιο τρόπο θέλουν.

Τα μοντέλα του IDEF0 διαχωρίζουν τις λειτουργίες των επιχειρήσεων αλλά δεν υποστηρίζουν την έννοια της «διαδικασίας», ούτε μπορούν να συλλαμβάνουν την απαιτούμενη χρονική σειρά με την οποία συμβαίνουν οι διάφορες δραστηριότητες. Επίσης πρέπει να υπάρξει μια ισορροπία μεταξύ του επιπέδου επιτομής που αντιπροσώπευε στα κιβώτια και το επίπεδο που εκφράζεται στα ICOMs' (βέλη). Επίσης όπως είδαμε και από το σχετικό διάγραμμα, το IDEF0 δεν μπορεί να μοντελοποιήσει τις συνθήκες που είναι υποχρεωτικές για την είσοδο ή την έξοδο από μια διαδικασία. Για αυτό το λόγο το IDEF0 είναι προτιμότερο να χρησιμοποιείται σε συνδυασμό με άλλες μεθόδους IDEF (π.χ. με την IDEF3).

Το IDEF1 δε μπορεί να χρησιμοποιηθεί άμεσα στη φάση της υλοποίησης. Όμως όπως έχουμε δείξει και σε προηγούμενο σχήμα μπορεί να είναι παρά πολύ χρήσιμο στην μοντελοποίηση του τρόπου ροής των πληροφοριών μέσα σε μια επιχείρηση με ένα τρόπο ανεξάρτητο από τους περιορισμούς που θα θέσει η υλοποίηση.

Το IDEF1X είναι ένα καλό εργαλείο για το σχεδιασμό βάσης δεδομένων, αλλά δεν ακολουθεί τους κανόνες ενός καλού γραφικού σχεδιασμού. Τα σύμβολά του δεν αντιστοιχούν με σαφή και ξεκάθαρο τρόπο στις ιδέες τις οποίες υποτίθεται ότι αντιπροσωπεύουν. Για παράδειγμα ένας συμπαγής κύκλος μπορεί να σημαίνει οτιδήποτε (εξαρτάται από την θέση στην οποία θα χρησιμοποιηθεί) και μια ακριβώς ίδια κατάσταση μπορούμε να την συμβολίσουμε με περισσότερα από ένα σετ συμβόλων. Επιπλέον διορθώνοντας ένα λάθος σε ένα IDEF1X διάγραμμα συνήθως ξεκινάει μια σειρά αλλαγών σε όλο το υπόλοιπο διάγραμμα.

Υπερβολικά πολύπλοκα και συνωστισμένα τα διαγράμματα IDEF3 μπορούν να έχουν σαν αποτέλεσμα τον συνδυασμό πολλών λειτουργιών και σεναρίων μέσα σ' ένα διάγραμμα. Το IDEF3 είναι μια μέθοδος περιγραφικής σύλληψης και είναι σχεδιασμένο να είναι ανεκτικό σε ατελείς και ασυνεπείς περιγραφές. Πολύ συχνά αυτές οι ασυνέπειες είναι η ρίζα των προβλημάτων σε έναν οργανισμό και θα πρέπει να εμφανίζονται με σωστό και ξεκάθαρο τρόπο στα διαγράμματα του IDEF3 και όχι να αποκρύπτονται.

Οι γλώσσες IDEF έχουν αναπτυχθεί για να απαντήσουν σε τρεις βασικές απαιτήσεις του ολοκληρωμένου σχεδιασμού μιας επιχείρησης:

- Να συλλάβουν ότι είναι γνωστό για τον πραγματικό κόσμο και τις σχέσεις μεταξύ ανθρώπων, γεγονότων, κλπ.
- Να συλλάβουν υπάρχουσες και μελλοντικές απαιτήσεις σχετικές με την διαχείριση των πληροφοριών.
- Να υποστηρίζουν τον σχεδιασμό συστημάτων που να καλύπτουν τις παραπάνω δύο απαιτήσεις.

Η περιγραφική σύλληψη των διαδικασιών με την IDEF3 και οι μέθοδοι περιγραφής οντοτήτων της IDEF5 καλύπτουν τη πρώτη από αυτές τις ανάγκες.

Η μοντελοποίηση λειτουργιών με την IDEF0 και η μοντελοποίηση της διακίνησης των πληροφοριών με την IDEF1 απαντούν στη δεύτερη ανάγκη.

Τέλος οι μέθοδοι σχεδιασμού βάσεων δεδομένων της IDEF1X, ο σχεδιασμός της δυναμικής των συστημάτων με την IDEF2 και η μέθοδος αντικειμενοστραφούς σχεδιασμού της IDEF4 στοχεύουν στο να καλύψουν την τρίτη ανάγκη.

4.2 Τελικό συμπέρασμα

Δεν μπορεί να υπάρξει οριστικό συμπέρασμα, ούτε μπορεί να υπάρξει μια και μοναδική μέθοδος μοντελοποίησης κατάλληλη για όλα τα συστήματα. Αυτό που μας χρειάζεται δεν είναι να βρούμε την μία και μοναδική μέθοδο μοντελοποίησης, αλλά να μπορούμε να βρούμε πια είναι η καταλληλότερη μέθοδος για το πρόβλημα που έχουμε κάθε φορά να αντιμετωπίσουμε.

Ο επιχειρηματικός σχεδιασμός εμφανίστηκε σαν απάντηση στην ανάγκη να δημιουργήσουμε ένα πρότυπο σχεδιασμού των επιχειρηματικών διαδικασιών έτσι ώστε να επιτυγχάνεται η συντήρηση και η επαναχρησιμοποίηση των επιχειρηματικών γνώσεων. Είναι σημαντικό για μια επιχείρηση να είναι ευέλικτη και ανθεκτική σε μια απώλεια ενός μεγάλου μέρους των στελεχών της (π.χ. συνταξιοδότηση προσωπικού).

Αυτό όμως μπορεί να επιτευχθεί όταν οι επιχειρηματικές γνώσεις έχουν προηγουμένως συλλεχθεί και οργανωθεί με έναν ξεκάθαρο τρόπο.

Στο σημερινό ανταγωνιστικό κόσμο μια επιχείρηση πρέπει συνέχεια να προσπαθεί να βελτιώσει την αποδοτικότητα της (καλύτερα προϊόντα σε χαμηλότερες τιμές είναι μόνο η κορυφή του «παγόβουνου»), γιατί αλλιώς δε θα αντέξει για πάρα πολύ καιρό. Έτσι άλλος ένας σημαντικός παράγοντας για την διάρκεια ζωής μιας επιχείρησης είναι να μπορεί να αντιμετωπίζει σωστά τις αλλαγές. Μακροπρόθεσμες προβλέψεις είναι όλο και δυσκολότερο να γίνουν σε ένα τόσο γρήγορα εναλλασσόμενο περιβάλλον. Σαν αποτέλεσμα μια επιχείρηση δε πρέπει απλώς να προσαρμόζεται στις αλλαγές αλλά και να μπορεί διαρκώς να εξελίσσεται. Κάτω από αυτές τις συνθήκες η αλλαγή είναι μια μόνιμη εσωτερική διαδικασία παρά ένας εξωτερικός παράγοντας στον οποίο η επιχείρηση πρέπει να προσαρμοστεί.

Ο επιχειρηματικός σχεδιασμός έχει τη προέλευση του από την αυτοματοποιημένη βιομηχανία που στόχος της ήταν η αυτοματοποιημένη βιομηχανική παραγωγή με χρήση ηλεκτρονικών υπολογιστών. Όμως φάνηκε ξεκάθαρα ότι οι αρχές με τις οποίες λειτουργεί η αυτοματοποιημένη βιομηχανική παραγωγή μπορούν στην πραγματικότητα να εφαρμοσθούν σε οποιοδήποτε επιχείρηση ή σύστημα επιχείρησης.

Ένας πολύ μεγάλος τομέας του επιχειρηματικού σχεδιασμού έχει ασχοληθεί απλά και μόνο με την ανάπτυξη λογισμικού για τις επιχειρήσεις. Όμως ενώ το λογισμικό είναι μέρος της επιχείρησης και στοχεύει στην υποστήριξη της, πολλές φορές δεν παρουσιάζει τις πραγματικές ανάγκες της. Ένας σημαντικός λόγος για αυτό είναι η σχεδίαση και η ανάπτυξη του λογισμικού να γίνεται ξεχωριστά και μακριά από την επιχείρηση. Στο εξής επικρατούν οι ιδέες για :

- Χρήση των εργαλείων ανάπτυξης λογισμικού για την υποστήριξη της επιχείρησης (με κατάλληλες αλλαγές ή επεκτάσεις)
- Χρήση του επιχειρηματικού μοντέλου σαν βάση για τον καθορισμό των απαιτήσεων του λογισμικού της επιχείρησης.

Σε αυτή την πτυχιακή εργασία έχουμε εξετάσει τις δυο βασικές μεθόδους που προσπαθούν να καλύψουν τις παραπάνω ιδέες. Οι μέθοδοι προσεγγίζουν διαφορετικές ομάδες ανθρώπων και προβλημάτων. Κάθε μέθοδος έχει τα πλεονεκτήματα της και τα μειονεκτήματα της αλλά προωθούν την ίδια τάση: τη προσπάθεια να καλύψουν τις υπόλοιπες μεθόδους που κυριαρχούσαν στο χώρο του σχεδιασμού του λογισμικού των επιχειρήσεων.

Το IDEF προέρχεται από την αυτοματοποιημένη βιομηχανική παραγωγή και στοχεύει να καλύψει την τάση για σχεδιασμό που βασίζεται στα «αντικείμενα» (object oriented), την αναπαράσταση των γνώσεων και την ανάπτυξη λογισμικού.

Η UML είναι ακόμη στο νεανικό της στάδιο, προέρχεται από το χώρο της ανάπτυξης αντικειμενοστραφούς λογισμικού, αλλά γίνονται μεγάλες προσπάθειες να επεκταθεί προς τον σχεδιασμό των διαδικασιών των επιχειρήσεων.

Ποια μέθοδος λοιπόν είναι η πλέον κατάλληλη και επικρατέστερη; Εξαρτάται από την δουλειά την οποία πρόκειται να κάνουμε, το υπόβαθρο που υπάρχει, τον τρόπο που είναι αποθηκευμένες οι υπάρχουσες πληροφορίες κ.λ.π.. Όλα αυτά παίζουν το ρόλο τους για να επιλέγουμε κάθε φορά το καταλληλότερο εργαλείο για την συγκεκριμένη κάθε φορά δουλειά.

BIBΛΙΟΓΡΑΦΙΑ

1. Krutchen, Philippe. (2000). *The Rational Unified Process An Introduction*. Second Edition Addison-Wesley
2. Larman, Craig. (1998). *Applying UML and Patterns An Introduction to Object Oriented Analysis and Design*. Prentice Hall.
3. Murray, Cantor. (1998). *Object-Oriented Project Management with UML*. John Wiley and Sons
4. Novak, J., and Gowin, D. B. (1984). *Learning How to Learn*. Cambridge: Cambridge University Press.
5. Sarris, A. K. (1992). "Needs Analysis and Requirements Document: Integration Toolkit and Methods, Corporate Data Integration Tools." MANTECH Report WL-TR-92-8027. WPAFB, OH.
6. Schmuller, Joseph. (1999). *Teach Yourself UML in 24 Hours*. Sams
7. R.J. Mayer: IDEF Family of Languages Overview and Practical Guidelines for IDEF use – paper on the IDEF web site:
http://www.idef.com/complete_reports
8. R. Panrahan et al: The IDEF Modeling Methodology – paper on the web site: <http://stsc.hill.af.mil/crosstalk/1995/jun/idef.asp>