ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΤΟΜΕΑΣ ΤΕΧΝΟΛΟΓΙΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ

ΕΡΓΑΣΤΗΡΙΟ ΣΥΣΤΗΜΑΤΩΝ ΒΑΣΕΩΝ ΓΝΩΣΕΩΝ ΚΑΙ ΔΕΔΟΜΕΝΩΝ

Μη Παραδοσιακές Τεχνικές Οπτικοποίησης Πολυδιάστατων Δεδομένων

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

του

ΗΛΙΑ Μ. ΜΙΧΑΛΑΡΙΑ

Επιβλέπων : Τιμολέων Σελλής

Καθηγητής Ε.Μ.Π.

Αθήνα, Σεπτέμβριος 2004

ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΤΕΧΝΟΛΟΓΙΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ
ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ

Μη Παραδοσιακές Τεχνικές Οπτικοποίησης Πολυδιάστατων Δεδομένων

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

του

ΗΛΙΑ Μ. ΜΙΧΑΛΑΡΙΑ

Επιβλέπων : Τιμολέων Σελλής
Καθηγητής Ε.Μ.Π.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 1η Σεπτεμβρίου 2004.

……	……..	……
Τιμολέων Σελλής	Ανδρέας-Γεώργιος Σταφυλοπάτης	Νεκτάριος Κοζύρης
Καθηγητής	Καθηγητής	Επίκουρος Καθηγητής

Αθήνα, Σεπτέμβριος 2004

(Υπογραφή)

.....

ΗΛΙΑΣ Μ. ΜΙΧΑΛΑΡΙΑΣ

Διπλωματούχος Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών Ε.Μ.Π.

© 2004 – Με επιφύλαξη παντός δικαιώματος. All rights reserved

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

Οι απόψεις και τα συμπεράσματα που περιέχονται σε αυτό το έγγραφο εκφράζουν τον συγγραφέα και δεν πρέπει να ερμηνευθεί ότι αντιπροσωπεύουν τις επίσημες θέσεις του Εθνικού Μετσόβιου Πολυτεχνείου.

Πρόλογος

Το έγγραφο τούτο τεκμηριώνει τη διπλωματική εργασία που εκπονήθηκε στο Εργαστήριο Συστημάτων Βάσεων και Γνώσεων Δεδομένων (ΕΒΓΔ) του τμήματος Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών του Εθνικού Μετσόβιου Πολυτεχνείου, και είχε ως αντικείμενο την ανάπτυξη μη παραδοσιακών τεχνικών οπτικοποίησης πολυδιάστατων δεδομένων και την σχεδίαση ενός εργαλείου που τις υλοποιεί. Ο σχεδιασμός και η ανάπτυξη της εφαρμογής πραγματοποιήθηκε το εαρινό εξάμηνο του 2004.

Πρώτα από όλα θα ήθελα να ευχαριστήσω τον υπεύθυνο καθηγητή μου κ. Τίμο Σελλή για το γεγονός ότι με ενέπνευσε και παρακίνησε να ασχοληθώ με τον ερευνητικό χώρο των βάσεων δεδομένων, καθώς και για την καλοπροαίρετη βοήθεια του σε πολλά επίπεδα.

Επίσης ένα μεγάλο ευχαριστώ οφείλω στο διδακτορικό φοιτητή και μέλος του ΕΒΓΔ κ.Αντρέα Μανιάτη, με τον οποίο είμασταν σε στενή συνεργασία καθόλη τη διάρκεια της εκπόνησης της διπλωματικής εργασίας τούτης . Η παρούσα δουλειά βασίζεται άλλωστε σε δικό του παλαιότερο ερευνητικό έργο. Οι καλοπροαίρετες παρατηρήσεις του καθόλη τη διάρκεια της υλοποίησης ήταν πραγματικά χρήσιμες. Επίσης θέλω να τον ευχαριστήσω για την ηθική του συμπαράστασή, όταν κάποιες δυσκολίες φαινόντουσαν αρχικά ανυπέρβλητες. Τελικά τα καταφέραμε και το αποτέλεσμα νομίζω ότι δικαιώνει την προσπάθειά μας.

Τέλος θα ήθελα να ευχαριστήσω την οικογένεια μου, το στενό μου περιβάλλον, τους Βασίλη, Ματθαίο, Μιχάλη, Νίκο, και τους κατά καιρούς συναδέλφους μου για την υποστήριξή τους καθόλη τη διάρκεια των σπουδών μου.

Περίληψη

Σκοπός της διπλωματικής εργασίας, είναι ο καθορισμός των απαιτήσεων, ο σχεδιασμός και η ανάπτυξη ενός εργαλείου ,το οποίο θα παρέχει τη δυνατότητα οπτικοποίησης πολυδιάστατων δεδομένων.

Τα τελευταία χρόνια, παρόλη την έντονη ερευνητική δραστηριότητα πάνω σε θέματα σχετικά με την μοντελοποίηση των δεδομένων σε αποθήκες δεδομένων, η οποία και απέφερε σημαντικά αποτελέσματα, δυσανάλογη υπήρξε η έρευνα που αφορά το κομμάτι της οπτικοποίησης των δεδομένων. Επιπλέον τα περισσότερα εμπορικά εργαλεία δεν υιοθετούν ένα αυτόνομο και διακριτό μοντέλο παρουσίασης των δεδομένων αυτών, αλλά αντίθετα τις περισσότερες φορές αυτό αποτελεί μέρος του λογικού ή εννοιολογικού μοντέλου με αποτέλεσμα τον περιορισμό του εύρους της χρηστικότητας και επαναχρησιμοποίησής του.

Στη διπλωματική τούτη υλοποιήθηκε ένα εργαλείο το οποίο οπτικοποιεί πολυδιάστατα δεδομένα χρησιμοποιώντας το CPM(Cube Presentation Model). Το CPM προερχόμενο από την ακαδημαϊκή κοινότητα, αποτελεί μία πρόσφατα διατυπωμένη πρόταση για ένα αυτόνομο και διακριτό μοντέλο παρουσίασης για απεικόνιση OLAP λειτουργιών.

Επιπρόσθετα η ραγδαία ανάπτυξη των κινητών συσκευών (κινητά τηλέφωνα, Smartphones, PDAs, Pocket PCs) με τις συνεχώς αυξανόμενες δυνατότητες αλλά και τους εγγενείς περιορισμούς, όπως για παράδειγμα το περιορισμένο μέγεθος της οθόνης, δημιουργεί νέα δεδομένα στον σχεδιασμό κάθε εφαρμογής. Στο πλαίσιο αυτό υλοποιήθηκε μία τεχνική που βασίζεται στη μεγέθυνση συγκεκριμένων περιοχών που παρουσιάζουν ενδιαφέρον για τον τελικό χρήστη και στην αντίστοιχη σμίκρυνση των υπολοίπων, έτσι ώστε να αντιμετωπιστεί το περιορισμένο μέγεθος της οθόνης. Η τεχνική αυτή βασίστηκε στη γνωστή τεχνική Table Lens.

Το εργαλείο αυτό αποτελεί κομμάτι μίας συνολικότερης προσπάθειας που αφορά την οπτικοποίηση πολυδιάστατων δεδομένων και που μελλοντικά θα επεκταθεί και σε ασύρματες συσκευές.

Λέξεις Κλειδιά: Αποθήκες δεδομένων, OLAP, Οπτικοποίηση πολυδιάστατων δεδομένων, Παρουσιαστκό μοντέλο, Microsoft Analysis Services, Microsoft .NET Framework ,C#.NET, ADOMD.NET, Sourcegrid.

Abstract

The objective of this Diploma Thesis is the defining of the user requirements, the design and implementation of a front end tool, which will provide visualization of multidimensional data facilities.

In the last years, On-Line Analytical Processing (OLAP) and data warehousing have become major research areas in the database community . Although the modeling of data has been extensively dealt with, an equally important issue of the OLAP domain, the visualization of data, has not been adequately investigated.

In addition to that, most commercial tools do not describe a discrete and autonomous presentation model for OLAP. Contrary, in most cases, the presentational model is a part of the logical or conceptual model ,which results in a limitation of its usefulness and portability.

In this thesis , a front end tool was developed , which visualizes multidimensional data using CPM(Cube Presentation Model), a novel and autonomous presentation model for OLAP screens, recently presented by the academia.

Moreover, the rapid growing and development of mobile devices (mobile phones, Smartphones, PDAs, Pocket PCs) with their increasing features but also with their limitations, like for example their small screen size, poses new topics on the design of each application. In this context, a technique was implemented, which is based on focusing and maximizing specific areas, that interest more the end user ,and minimizing the rest ,so that the small screen limitation can be coped with. This technique stems from the Table Lens technique.

This tool is a part of related research work which deals overally with visualization of multidimensional data , and is planned to be extended and deployed in mobile devices in the future.

Keywords: Data Warehouses, OLAP, Visualization of multidimensional data, Presentation model, Microsoft Analysis Services, Microsoft .NET Framework ,C#.NET, ADOMD.NET, Sourcegrid.

Πίνακας περιεχομένων

1 ΕΙΣΑΓΩΓΗ 15

1.1 ΑΝΤΙΚΕΙΜΕΝΟ ΤΗΣ ΔΙΠΛΩΜΑΤΙΚΗΣ 15
1.2 ΟΡΓΑΝΩΣΗ ΤΟΥ ΤΟΜΟΥ 17

2 ΘΕΩΡΗΤΙΚΟ ΥΠΟΒΑΘΡΟ 19

2.1 ΑΛΛΗΛΕΠΙΔΡΑΣΗ ΑΝΘΡΩΠΟΥ ΜΕ ΥΠΟΛΟΓΙΣΤΗ 19
2.1.1 Ορισμός *19*
2.2 ΠΕΔΙΟ ΔΡΑΣΗΣ 20
2.2.1 Ιστορική αναδρομή *20*
2.2.2 Βασικές επιδράσεις *22*
2.3 ΑΛΛΗΛΕΠΙΔΡΑΣΗ ΑΝΘΡΩΠΟΥ ΜΕ ΣΥΣΤΗΜΑΤΑ ΒΑΣΕΩΝ ΔΕΔΟΜΕΝΩΝ 23
2.4 ΑΠΟΘΗΚΕΣ ΔΕΔΟΜΕΝΩΝ , OLAP 24
2.5 ΟΠΤΙΚΟΠΟΙΗΣΗ ΩΣ ΒΑΣΙΚΟ ΚΟΜΜΑΤΙ ΤΟΥ HCI 26
2.6 ΟΠΤΙΚΟΠΟΙΗΣΗ OLAP ΕΦΑΡΜΟΓΩΝ 27
2.7 ΠΑΡΟΥΣΙΑΣΤΙΚΑ ΕΡΓΑΛΕΙΑ 29

3 ΣΧΕΔΙΑΣΗ 33

3.1 ΜΟΝΤΕΛΑ ΠΑΡΟΥΣΙΑΣΗΣ 33
3.2 CPM(CUBE PRESENTATION MODEL) 35
3.2.1 Εισαγωγή *35*
3.2.2 Λογικό στρώμα του CPM *36*
3.2.3 Παρουσιαστικό στρώμα του CPM *39*
3.2.3.1 Βασικές οντότητες 39
3.2.3.2 Ο Πολυδιάστατος χώρος 42
3.2.3.3 Τα περιεχόμενα του πολυδιάστατου χώρου 46
3.2.4 Table Lens *50*
3.2.4.1 Αντιστοίχιση του CPM στο Table Lens 51
3.2.4.2 Επιλογή παραθύρου με το περισσότερο ενδιαφέρον 52
3.2.5 OLAP Lens *52*
3.3 ΑΠΑΙΤΗΣΕΙΣ ΧΡΗΣΤΗ 54

4 ΥΛΟΠΟΙΗΣΗ 59

4.1 ΑΡΧΙΤΕΚΤΟΝΙΚΗ 59
4.1.1 Υλοποιούμενη Αρχιτεκτονική *59*
4.2 ΕΡΓΑΛΕΙΑ 61
4.2.1 Έρευνα και δοκιμή εργαλείων *61*
4.2.2 Analysis Sevices *63*
4.2.3 .NET *65*
4.2.3.1 .NET Framework 65
4.2.3.2 Visual Studio.NET 66
4.2.3.3 C#.NET 67
4.2.4 MDX *69*
4.2.4.1 Γενικά 69
4.2.4.2 Βασικές οντότητες 69
4.2.5 ADOMD.NET *71*

4.2.6 Sourcegrid ... *73*
4.3 ΤΕΧΝΙΚΕΣ ΥΛΟΠΟΙΗΣΗΣ ... 73
4.3.1 Σύνδεση με την πηγή δεδομένων ... *73*
4.3.2 Ανάκτηση δεδομένων ... *74*
4.3.3 Υλοποιούμενη δομή CPM ... *75*
4.3.3.1 Αρχικοποίηση ... 75
4.3.3.2 Δημιουργία multicubes ... 76
4.3.4 Παράμετροι συστήματος ... *80*
4.3.5 Απεικόνιση ... *81*
4.3.6 Επεκτασιμότητα εφαρμογής ... *85*
4.4 ΠΑΡΟΥΣΙΑΣΗ ΚΑΙ ΑΝΑΛΥΣΗ ΥΛΟΠΟΙΗΣΗΣ ... 86
4.4.1 Οθόνη επιλογής πηγής δεδομένων ... *87*
4.4.2 Οθόνη επιλογής αρχικών παραμέτρων ... *87*
4.4.3 Οθόνη επιλογής διαστάσεων ... *88*
4.4.4 Οθόνη επιλογής φίλτρου ... *89*
4.4.5 Οθόνη αλλαγής αξόνων ... *90*
4.4.6 Οθόνη drill within ... *91*
4.4.7 Οθόνη κατώτερου επιπέδου της ιεραρχίας ... *92*
4.4.8 Οθόνη ανώτερου επιπέδου της ιεραρχίας ... *92*
4.4.9 Οθόνη επιλογής συνάρτησης ... *93*
4.5 ΈΛΕΓΧΟΣ ΟΡΘΟΤΗΤΑΣ ... 94
4.6 ΜΕΤΡΗΣΗ ΑΠΟΔΟΣΗΣ ... 96

5 ΣΥΜΠΕΡΑΣΜΑΤΑ-ΜΕΛΛΟΝΤΙΚΗ ΔΟΥΛΕΙΑ ... 101

5.1 ΜΕΛΛΟΝΤΙΚΗ ΔΟΥΛΕΙΑ ... 101
5.1.1 Χαρακτηριστικά κινητών συσκευών που επηρεάζουν τις εφαρμογές χρήστη ... *101*
5.1.2 Αρχιτεκτονική Mobile OLAP ... *103*
5.1.3 MOLAP Lens ... *104*
5.2 ΣΥΜΠΕΡΑΣΜΑΤΑ ΚΑΙ ΣΥΝΟΨΗ ΔΙΠΛΩΜΑΤΙΚΗΣ ... 107

6 ΒΙΒΛΙΟΓΡΑΦΙΑ ... 109

Πίνακας σχημάτων

Σχήμα	Όνομα	σελ.
σχήμα 3.1	Παράδειγμα από το Cube Model [MS98]	34
σχήμα 3.2	Μία δομή ταινίας(tape) με 2 τεμνόμενες ταινίες	35
σχήμα 3.3	Αρχιτεκτονική του CPM	36
σχήμα 3.4	CPM Οντότητες	40
σχήμα 3.5	Αντιστοίχιση CPM οντοτήτων σε 3D και 2D μορφές/απεικονίσεις	41
σχήμα 3.6	Η διδιάστατη τομή(2D-Slice) SL για το παράδειγμα του Σχ.3.1	41
σχήμα 3.7	Τα σημεία των αξόνων Rows και Columns.	46
σχήμα 3.8	Ο πολυδιάστατος χώρος για το παράδειγμά του Σχ.3.2 , επεκταμένο με Sections	49
σχήμα 3.9	Ένα παράδειγμα από Table Lens: (α) ένα focus παράθυρο 2x4 ορισμένο σε ένα χώρο 8x8 σημείων (β) Table Lens παραμόρφωση του άξονα Rows	51
σχήμα 3.10(α)	Συνάρτηση μεταφοράς και συνάρτηση DOI αρχικά	53
σχήμα 3.10(β)	Συνάρτηση μεταφοράς και συνάρτηση DOI μετά τη μεγένθυνση	53
σχήμα 4.1	Απλοποιημένη Αρχιτεκτονική του OLAP Lens	60
σχήμα 4.2	Ένα τυπικό OLAP σύστημα	63
σχήμα 4.3	Η ιεραρχία αντικειμένων στις Analysis Services	64
σχήμα 4.4	Αρχιτεκτονική Analysis Server	64
σχήμα 4.5	Το .NET Framework με μια ματιά	66
σχήμα 4.6	Βασικά συστατικά του .NET κατά το χρόνο μεταγλώτισσης και εκτέλεσης	68
σχήμα 4.7	Ένα κλασσικό σχεσιακό σχήμα	70
σχήμα 4.8	Παράδειγμα ενός κύβου με 3 διαστάσεις και 2 μέτρα	70
σχήμα 4.9	Το object model του ADOMD.NET	72

σχήμα 4.10	Cube_Schema: Δομή αρχικοποίησης	76
σχήμα 4.11	MDX_Query:Κλάση ερωτήσεων	77
σχήμα 4.12	Κλάση MultiCube	77
σχήμα 4.13	Δεικτοδότηση ανάμεσα σε multicubes	80
σχήμα 4.14	Αρχική οθόνη κατάσταση OLAP Lens	82
σχήμα 4.15	Φόρμες που συμμετέχουν στην απεικόνιση	83
σχήμα 4.16	Φόρμα Lens_Form	84
σχήμα 4.17	OLAP Lens software architecture	86
σχήμα 4.18	Διάγραμμα ροής OLAP Lens	86
σχήμα 4.19	Οθόνη σύνδεσης με την πηγή δεδομένων	87
σχήμα 4.20	Οθόνη επιλογής αρχικών παραμέτρων	88
σχήμα 4.21	Οθόνη επιλογής διαστάσεων	89
σχήμα 4.22	Οθόνη επιλογής φίλτρου	89
σχήμα 4.23	Οθόνη αλλαγής αξόνων	90
σχήμα 4.24	Οθόνη drill-within	91
σχήμα 4.25	Οθόνη κατώτερου επιπέδου της ιεραρχίας	92
σχήμα 4.26	Οθόνη ανώτερου επιπέδου της ιεραρχίας	93
σχήμα 4.27	Οθόνη επιλογής συνάρτησης	93
σχήμα 4.28	Στιγμυότυπο OLAP Lens	95
σχήμα 4.29	Στιγμυότυπο MDX Sample Application	95
σχήμα 4.30	TPCD κύβος	96
σχήμα 4.31	FoodMart2000 κύβος	98
σχήμα 5.1	Απλοποιημένη Αρχιτεκτονική ενός MOLAP συστήματος	104
σχήμα 5.2	Αρχιτεκτονική Cube View	104
σχήμα 5.3(α)	MOLAP Lens(α)	106
σχήμα5.3(β)(γ)	MOLAP Lens(β)(γ)	107

1 *Εισαγωγή*

Στο παρόν κεφάλαιο γίνεται μία εισαγωγή στην θέμα της εργασίας. Στην παράγραφο 1.1 αναλύεται το αντικείμενο της παρούσας διπλωματικής εργασίας και στην παράγραφο 1.2 γίνεται μια επισκόπηση της οργάνωσης του τόμου που κρατάτε στα χέρια σας.

1.1 Αντικείμενο της διπλωματικής

Η αύξηση του ρυθμού με τον οποίο μεγαλώνει σήμερα ο όγκος και το είδος της πληροφορίας είναι ραγδαίος, Ο χώρος των βάσεων δεδομένων είναι ίσως ο πιο άμεσα επηρεαζόμενος από την τάση αυτή. Το γεγονός αυτό έχει άμεση επίδραση όχι μόνο στα συστήματα που καλούνται να διαχειριστούν την πληροφορία αυτή αλλά και στον ανθρώπινο παράγοντα.

Για την αντιμετώπιση των νέων αναγκών αναπτύχθηκαν την τελευταία δεκαετία τα συστήματα Αποθηκών Δεδομένων, ακριβώς για τη διαχείριση και αξιοποίηση μεγάλου όγκου δεδομένων. Παράλληλα αναπτύχθηκαν και οι εφαρμογές OLAP(Online Analytical Processing) ένας όρος που χρησιμοποιείται για να περιγραφεί η ανάλυση πολύπλοκων δεδομένων από αποθήκες δεδομένων.

Παρόλη όμως την έντονη ερευνητική δραστηριότητα πάνω σε θέματα σχετικά με την μοντελοποίηση των δεδομένων σε αποθήκες δεδομένων, η οποία και απέφερε σημαντικά

αποτελέσματα, δυσανάλογη υπήρξε η έρευνα που αφορά το κομμάτι της οπτικοποίησης των δεδομένων αυτών. Μία εξήγηση της αδράνειας αυτής μπορεί να υπάρχει στην κλασσική αντίληψη πως ο τελικός χρήστης βρίσκεται μπροστά από μία κονσόλα εκτελώντας κάθε φορά ένα μόνο query και λαμβάνοντας μία μόνο απάντηση. Αυτό γινόταν βέβαια παλαιότερα όμως σήμερα οι ανάγκες είναι διαφορετικές.

Επιπρόσθετα, μέχρι πριν όχι πολλά χρόνια τα άτομα που ερχόντουσαν σε επαφή με συστήματα βάσεων δεδομένων ήταν κατά βάση εξειδικευμένα στελέχη με τουλάχιστον αρκετές τεχνικές γνώσεις. Σήμερα το εύρος των ατόμων που χρησιμοποιούν τέτοια συστήματα είναι εξαιρετικά μεγάλο. Σα φυσική συνέπεια ένας πιθανός χρήστης μπορεί να είναι λιγότερο εξοικειωμένος με τεχνικούς όρους ή μπορεί να μην έχει καμία προοπτική ή διάθεση εκμάθησης νέων πραγμάτων. Ειδικά οι OLAP εφαρμογές απευθύνονται συνήθως σε διευθυντικά στελέχη, τα οποία καλούνται να πάρουν αποφάσεις σύμφωνα με τα δεδομένα μίας αποθήκης δεδομένων, χωρίς όμως οι ίδιοι να έχουν τεχνικές γνώσεις για τα συστήματα αυτά.

Γίνεται σαφές πως η οπτικοποίηση δεδομένων για OLAP εφαρμογές παρουσιάζει επίσης μεγάλο ενδιαφέρον, και αποτελεί έναν ερευνητικό χώρο στον οποίο θα δοθεί ιδιαίτερη προσοχή τα επόμενα χρόνια.

Από έρευνα που έγινε στα πλαίσια της διπλωματικής προέκυψε ότι προς το παρόν τα περισσότερα εμπορικά εργαλεία δεν υιοθετούν ένα αυτόνομο και διακριτό μοντέλο παρουσίασης των δεδομένων. Αντίθετα τις περισσότερες φορές η παρουσίαση αποτελεί μέρος του λογικού ή εννοιολογικού μοντέλου με αποτέλεσμα τον περιορισμό του εύρους της χρηστικότητας και επαναχρησιμοποίησής του.

Σκοπός της διπλωματικής εργασίας, είναι ο καθορισμός των απαιτήσεων, ο σχεδιασμός και η ανάπτυξη ενός desktop εργαλείου, το οποίο θα παρέχει τη δυνατότητα οπτικοποίησης πολυδιάστατων δεδομένων, χρησιμοποιώντας παράλληλα το CPM(Cube Presentation Model), το οποίο αποτελεί μία πρόσφατα διατυπωμένη από την ακαδημαϊκή κοινότητα αυτόνομη πρόταση για ένα μοντέλο παρουσίασης για απεικόνιση OLAP λειτουργιών.

Επιπρόσθετα η ραγδαία ανάπτυξη των κινητών συσκευών (κινητά τηλέφωνα, Smartphones, PDAs, Pocket PCs) με τις συνεχώς αυξανόμενες δυνατότητες αλλά και τους εγγενείς περιορισμούς, όπως για παράδειγμα το περιορισμένο μέγεθος της οθόνης, δημιουργεί νέα δεδομένα στον σχεδιασμό κάθε εφαρμογής. Στο πλαίσιο αυτό υλοποιήθηκε μία τεχνική που βασίζεται στη μεγένθυνση συγκεκριμένων περιοχών που παρουσιάζουν ενδιαφέρον για τον τελικό χρήστη και στην αντίστοιχη σμίκρυνση των υπολοίπων, έτσι ώστε να αντιμετωπιστεί το περιορισμένο μέγεθος της οθόνης. Η τεχνική αυτή βασίζεται στην γνωστή τεχνική Table Lens.

Σε αυτό το πλαίσιο οι απαιτήσεις χρήστη που καθορίστηκαν δεν περιορίστηκαν ώστε να αφορούν μόνο desktop συστήματα, αλλά ενσωματώνουν επιπρόσθετα θέματα που ανακύπτουν σε ασύρματες συσκεύες.

Ο σχεδιασμός του εργαλείου αποτέλεσε ένα πολυ σημαντικό κομμάτι της διπλωματικής τούτης, μιας και η αρχιτεκτονική της εφαρμογής και η εύρεση των εργαλείων που θα έκαναν εφικτή την υλοποίηση, δεν ήταν εκ των προτέρων καθορισμένα. Μάλιστα κατά τη διάρκεια της διαδικασίας αυτής έγινε φανερό πως οι υπάρχουσες προγραμματιστικές πλατφόρμες για ανάπτυξη εφαρμογών σε ασύρματες συσκευές, είναι προς το παρόν μάλλον ανεπαρκείς, αφού οι δυνατότητες που προσφέρουν είναι πολύ περιορισμένες. Για το λόγο αυτό η υλοποίηση έγινε σε desktop περιβάλλον.

Το εργαλείο που προέκυψε και το οποίο ονομάσαμε OLAP Lens, οπτικοποιεί δεδομένα ενός συγκεκριμένου OLAP server , και ικανοποιεί τις σημαντικότερες απαιτήσεις χρήστη μίας τυπικής εφαρμογής οπτικοποίησης OLAP. Η αντικατάσταση του server αυτού με έναν εναλλακτικό είναι σχετικά απλή.Η προσπάθεια αυτή αποτελεί κομμάτι μίας συνολικότερης δουλειάς που αφορά την οπτικοποίηση πολυδιάστατων δεδομένων και που μελλοντικά θα επεκταθεί και σε ασύρματες συσκευές.

1.2 Οργάνωση του τόμου

Ο παρών τόμος αποτελείται από 6 κεφάλαια και αναλύει πλήρως την ανάπτυξη της εργασίας.

Το ***δεύτερο κεφάλαιο*** κάνει μία εισαγωγή στο χώρο της αλληλεπίδρασης ανθρώπου-υπολογιστή(HCI). Το ενδιαφέρον εστιάζεται σε συστήματα βάσεων δεδομένων. Αναδεικνύεται η οπτικοποίηση ως ένα βασικό κομμάτι του HCI. Αναλύονται κάποιες βασικές απαιτήσεις σε εφαρμογές οπτικοποίησης για OLAP.

Το ***τρίτο κεφάλαιο*** ασχολείται με τη σχεδίαση του εργαλείου που θα υλοποιήσουμε. Γίνεται μία παρουσίαση του CPM, ενός αυτόνομου μοντέλου παρουσίασης, που αποτελεί μία πρόσφατη ακαδημαϊκή πρόταση για OLAP εφαρμογές και το οποίο θα χρησιμοποιηθεί στην εφαρμογή μας. Τέλος καθορίζονται οι απαιτήσεις χρήστη του εργαλείου.

Στο ***τέταρτο κεφάλαιο*** γίνεται ανάλυση και περιγραφή της υλοποίησης του εργαλείου. Αρχικά παρουσιάζεται η αρχιτεκτονική του συστήματος. Έτσι προκύπτουν οι ανάγκες των προγραμματιστικών εργαλείων, η επιλογή των οποίων τεκμηριώνεται στη συνέχεια. Έπειτα αναλύονται οι τεχνικές υλοποίησης. Στη συνέχεια απεικονίζονται στιγμυότυπα οθονών του εργαλείου καθώς και κα΄ποα θέματα σχετικά με την απόδοση του.

Το ***πέμπτο κεφάλαιο*** αναφέρεται στα συμπεράσματα που προέκυψαν από όλη τη διαδικασία ανάπτυξης του εργαλείου αυτού καθώς και στα μελλοντικά ερευνητικά θέματα που προκύπτουν.Ιδιαίτερη μνεία γίνεται σε μία μελλοντική επέκταση σε ασύρματες συσκευές.

Στο ***έκτο κεφάλαιο***, τέλος, δίνεται η βιβλιογραφία και γενικότερα οι πηγές από τις οποίες αντλήθηκαν οι απαραίτητες πληροφορίες για τη συγγραφή της διπλωματικής.

Τον τόμο τούτο συνοδεύει ένα CD-ROM όπου περιέχεται ο πηγαίος κώδικας της εφαρμογής, ο εκτελέσιμος κώδικας, ο τόμος σε ηλεκτρονική μορφή και οι διαφάνειες παρουσίασης της διπλωματικής εργασίας.

2 *Θεωρητικό Υπόβαθρο*

Στο παρόν κεφάλαιο, στην παράγραφο 2.1 γίνεται μία πρώτη αναφορά στην αλληλεπίδραση ανθρώπου υπολογιστή (HCI) , στην παράγραφο 2.2 αναφερόμαστε συγκεκριμένα σε συστήματα βάσεων δεδομένων, στην παράγραφο 2.3 γίνεται μία σύντομη εισαγωγή στις αποθήκες δεδομένων και OLAP εφαρμογών , στις παραγράφους 2.4 και 2.5 γίνεται αναφορά στην οπτικοποίηση ενώ συην παράγραφο 2.6 παρουσιάζονται περιληπτικά τα κυριότερα εμπορικά εργαλεία.

2.1 Αλληλεπίδραση ανθρώπου με υπολογιστή

2.1.1 Ορισμός

Η αλληλεπίδραση ανθρώπου με υπολογιστή (human computer interaction) αποτελεί αντικείμενο έρευνας της επιστήμης στο οποίο εμπλέκονται αρκετά διαφορετικοί επιστημονικοί κλάδοι. Παρά την ενασχόληση αρκετών επιστημόνων δεν υπάρχει σαφής ορισμός εξαιτίας της γενικότητάς του αντικειμένου. Ένας από τους πιο συχνά χρησιμοποιούμενους ορισμούς είναι σύμφωνα με το [HBC02] ο ακόλουθος:

Η αλληλεπίδραση ανθρώπου με υπολογιστή είναι η αρχή σύμφωνα με την οποία σχεδιάζουμε, αποτιμούμε και υλοποιούμε ένα υπολογιστικό σύστημα το οποίο αλληλεπιδρά με ανθρώπους, λαμβάνοντας υπ' όψιν όλα τα φαινόμενα που περικλείουν την αλληλεπίδραση αυτή.

Πιο συγκεκριμένα η αλληλεπίδραση ανθρώπου με υπολογιστή ενδιαφέρεται για την αποδοτική συνεργασία ανθρώπων και υπολογιστικών μηχανών, τη δομή της επικοινωνίας μεταξύ των, τις ανθρώπινες ικανότητες ως προς τη χρήση υπολογιστών, τον προγραμματισμό εφαρμογών και των διεπαφών των υπολογιστών με τους ανθρώπους, τη σχεδίαση και υλοποίηση του υλικού των υπολογιστικών μηχανών καθώς και των ίδιων των μηχανών, ως ένα βαθμό.

Γίνεται φανερό ότι η αλληλεπίδραση ανθρώπου με υπολογιστή είναι ένα πεδίο έρευνας στο οποίο εμπλέκονται αρκετοί και διαφορετικοί επιστημονικοί κλάδοι. Από την πλευρά της επιστήμης των υπολογιστών το ενδιαφέρον επικεντρώνεται στην αλληλεπίδραση καθώς και στο υπολογιστικό σύστημα. Ο προγραμματιστής ενδιαφέρεται κυρίως για τη σχεδίαση της εφαρμογής και του περιβάλλοντος χρήστη, λαμβάνοντας όμως υπ' όψιν και το περιεχόμενο της εφαρμογής. Άλλοι τομείς, όπως η ψυχολογία, ασχολούνται με την ανάλυση της ανθρώπινης συμπεριφοράς και της μαθησιακής ικανότητας μέσω της γνωστικής θεωρίας και της εμπειρικής ανάλυσης. Η κοινωνιολογία και η ανθρωπολογία μελετάνε την επίδραση των υπολογιστικών μηχανών στο άνθρωπο και σε κοινωνικές ομάδες.

2.2 Πεδίο δράσης

Θα επιχειρήσουμε μια ανασκόπηση στην ιστορία της αλληλεπίδρασης του ανθρώπου με τους υπολογιστές και τις επιρροές που διαμόρφωσαν το πεδίο αυτό καθώς και μια αναφορά στις τρέχουσες τάσεις, σύμφωνα με την περιγραφή στο [SIG04].

2.2.1 Ιστορική αναδρομή

Η αλληλεπίδραση ανθρώπου και μηχανής είχε ρίζες σε διαφορετικά ερευνητικά πεδία, όπως η γραφική υπολογιστών, τα λειτουργικά συστήματα, η εργονομία, ο βιομηχανικός σχεδιασμός, η βιοτεχνολογία, η επιστήμη κόστους εργασίας, η γνωστική ψυχολογία. Τα γραφικά με υπολογιστές γεννήθηκαν από τη χρήση των οθονών καθοδικής λυχνίας (CRT) και συσκευών ένδειξης (pen devices) τα πρώτα χρόνια της ιστορίας των υπολογιστών. Αυτό οδήγησε στην ανάπτυξη διαφόρων τεχνικών αλληλεπίδρασης ανθρώπου-υπολογιστή. Πολλές τεχνικές χρονολογούνται από την εποχή της διδακτορικής διατριβής του Sutherland που αναφερόταν στο Sketchpad [Suth63] η οποία και έθεσε τις βάσεις για την ανάπτυξη του τομέα των γραφικών υπολογιστών. Η συνεχιζόμενη έρευνα στο τομέα των γραφικών επέτρεψε την ανάπτυξη αλγορίθμων και γραφικών που επέτρεπαν την εμφάνιση και διαχείριση ακόμα πιο ρεαλιστικών αντικειμένων (για παράδειγμα τα μέρη μιας μηχανής ή οι ιατρικές εικόνες τμημάτων του σώματος). Τα γραφικά υπολογιστών σχετίζονται άμεσα με την αλληλεπίδραση ανθρώπου

υπολογιστή ως "αλληλεπιδραστικά γραφικά" (π.χ. διαχείριση μοντέλων σε σχεδιαστικές εφαρμογές).

Άμεσα σχετιζόμενες ήταν και ένα σύνολο προσπαθειών για την επίτευξη της "συμβίωσης ανθρώπου-μηχανής" (man-machine symbiosis, Licklider, 1960) [Lick60], "η επαύξηση της ανθρώπινης διάνοιας" (Engelbart, 1963) και το "Dynabook" (Kay και Goldberg, 1977) [KaGo77]. Μέσα από αυτή τη σειρά των εξελίξεων προήλθε ένα σημαντικό τμήμα της αλληλεπίδρασης ανθρώπου υπολογιστή. Μερικές από αυτές τις εξελίξεις συμπεριλαμβάνουν το ποντίκι, τις ψηφιογραφικές οθόνες (bitmapped displays), τους προσωπικούς υπολογιστές, τα παράθυρα, τη μεταφορά του γραφείου/ χώρου εργασίας (desktop metaphor) και τους κειμενογράφους με δυνατότητα χρήσης συσκευής κατάδειξης (Baecker & Buxton, 1987) [BaBu].

Την ίδια περίοδο η έρευνα στο χώρο των λειτουργικών συστημάτων, είχε ως αποτέλεσμα την ανάπτυξη τεχνικών για τη διασύνδεση μονάδων εισόδου/ εξόδου, για την ρύθμιση του χρόνου απόκρισης των συστημάτων σε αντιστοιχία με τον ανθρώπινο χρόνο, για πολυεπεξεργασία και για την υποστήριξη περιβαλλόντων με παράθυρα και κινούμενων εικόνων. Αυτή η κίνηση έδωσε ώθηση σε συστήματα διαχείρισης διεπαφών (user interface management systems) και σε εργαλειοθήκες διεπαφών (user interface toolkits).

Ο τομέας των ανθρωπίνων παραγόντων (Human Factors), εμφανίζεται με την έρευνα του σχεδιασμού εξοπλισμού και της χρήσης του από ανθρώπους κατά τη διάρκεια του δευτέρου Παγκοσμίου πολέμου (Sanders & McCormick, 1987). Πολλά από τα προβλήματα που αντιμετωπίσθηκαν τότε είχαν να κάνουν με κινητικά – μηχανικά χαρακτηριστικά (όπως το σχεδιασμό των χειριστηρίων και των οργάνων των αεροπλάνων). Το πρόβλημα της λειτουργικότητας των υπολογιστών ήταν μια φυσική επέκταση του τομέα, εκτός από το γεγονός ότι τα νέα προβλήματα που έπρεπε να αντιμετωπισθούν είχαν σοβαρές γνωστικές, επικοινωνιακές και αλληλεπιδραστικές πτυχές που δεν είχαν ανακύψει παλαιότερα, ωθώντας μια επέκταση προς αυτές τις κατευθύνσεις. Η εργονομία είναι συναφής με τον τομέα των Ανθρώπινων Παραγόντων, προήλθε όμως από μελέτες στο χώρο της εργασίας. Και η εργονομία ενδιαφέρεται για κινητικά-μηχανικά χαρακτηριστικά, με περισσότερο όμως βάρος στην ανατομική πλευρά και στο άγχος/ κούραση. Η αλληλεπίδραση ανθρώπου-υπολογιστή ήταν επίσης άμεσου ενδιαφέροντος για την εργονομία, όμως η επέκταση προς γνωστικά θέματα ήταν αναγκαία δημιουργώντας τον τομέα της γνωστικής εργονομίας (cognitive ergonomics) και της γνωστικής τεχνολογίας (cognitive engineering). Λόγω της ιστορίας τους η εργονομική εξέταση των υπολογιστών δίνει έμφαση στη σχέση με το περιβάλλον εργασίας και τα αποτελέσματα των παραγόντων πίεσης/ άγχους, όπως η επαναληπτική εργασία, η θέση εργασίας ή ο σχεδιασμός των οθονών.

Η βιομηχανική τεχνολογία (industrial engineering) ξεπήδησε από τις προσπάθειες αύξησης της βιομηχανικής παραγωγής, ξεκινώντας τα πρώτα χρόνια του τρέχοντος αιώνα. Η αρχική έμφαση ήταν στο σχεδιασμό αποτελεσματικών μεθόδων εργασίας με τα χέρια (π.χ. η μέθοδος τοποθέτησης τούβλων με τα δύο χέρια), ο σχεδιασμός εξειδικευμένων εργαλείων για την αύξηση της παραγωγικότητας και τη μείωση της κόπωσης και σε μικρότερο βαθμό στο σχεδιασμό του κοινωνικού περιβάλλοντος (π.χ. η εφεύρεση του κουτιού παραπόνων). Η διάδραση με τους υπολογιστές έχει άμεση σχέση με το στόχο της βιομηχανικής μηχανολογίας σε σχέση με το πως χρησιμοποιούνται οι υπολογιστές στο ευρύτερο πλαίσιο των μεθόδων εργασίας.

Η γνωστική ψυχολογία εμφανίζεται ως τομέας στο τέλος του 18ου αιώνα, όταν γίνονται οι πρώτες πειραματικές προσπάθειες μελέτης των αισθήσεων. Στα 1950, μια εισροή από ιδέες από τον τομέα της μηχανολογίας επικοινωνίας, της γλωσσολογίας και της μηχανολογίας υπολογιστών οδήγησαν σε μια επιστήμη που βασιζόταν σε πειραματικές αποδείξεις και με κύριο πεδίο έρευνας την επεξεργασία των πληροφοριών από τον άνθρωπο και τις επιδόσεις. Οι ερευνητές της γνωσιακής ψυχολογίας έχουν επικεντρωθεί στην εκμάθηση συστημάτων, στη μεταφορά της γνώσης, τη νοητική αναπαράσταση των συστημάτων από τον άνθρωπο και τις επιδόσεις του ανθρώπου με τέτοια συστήματα.

Τέλος η ανάπτυξη της αγοράς των προσωπικών υπολογιστών συνέδεσε σε μεγάλο βαθμό τις πωλήσεις των υπολογιστών με την ποιότητα των διεπαφών τους. Το αποτέλεσμα ήταν η σταδιακή εξέλιξη μιας τυποποιημένης αρχιτεκτονικής διεπαφών από το επίπεδο της υποστήριξης ποντικιών, από το υλικό μέχρι τα παραθυρικά περιβάλλοντα και επίπεδα διαχείρισης εφαρμογών (application management layers). Μαζί με αυτές τις αλλαγές οι ερευνητές και οι σχεδιαστές έχουν ξεκινήσει την ανάπτυξη τεχνικών παραγωγής προδιαγραφών για διεπαφές και τεχνικών αξιολόγησης για διεπαφές.

2.2.2 Βασικές επιδράσεις

Υπάρχουν μερικές βασικές επιδράσεις στον τομέα της αλληλεπίδρασης ανθρώπου με υπολογιστές. Μέχρι τη δεκαετία του ογδόντα οι σημαντικότερες από αυτές ήταν ο άμεσος χειρισμός γραφικών αντικειμένων, η δημιουργία συσκευών εισόδου με γνώμονα τη χρηστικότητα και την ευκολία του χρήστη και τέλος η δημιουργία εφαρμογών σε υπολογιστές με χρήση πολλαπλών γραφικών παραθύρων. Ο άμεσος χειρισμός γραφικών αντικειμένων είναι συνέπεια της έρευνας στο πεδίο της γραφικής υπολογιστών, η οποία σκόπευε στη διευκόλυνση του χρήστη κατά τη χρησιμοποίηση υπολογιστικών μηχανών μέσω της αλληλεπίδρασης με γραφικά αντικείμενα με γρήγορο και εύκολο τρόπο. Η σχεδίαση και δημιουργία συσκευών εισόδου, όπως για παράδειγμα το ποντίκι στους προσωπικούς και φορητούς υπολογιστές, επηρέασε σημαντικά

τον τρόπο δημιουργίας γραφικού περιβάλλοντος για το χρήστη στους ηλεκτρονικούς υπολογιστές. Τέλος το γραφικό περιβάλλον με χρήση πολλαπλών κα επικαλυπτόμενων παραθύρων επέφερε ριζική αλλαγή στον τρόπο σχεδίασης των εφαρμογών.

Τη δεκαετία του ογδόντα η εξάπλωση των προσωπικών υπολογιστών και η μεγάλη ενασχόληση των ερευνητικών ιδρυμάτων με τα λειτουργικά συστήματα των συσκευών αυτών έδωσε μια νέα ώθηση στη έρευνα της αλληλεπίδρασης ανθρώπου και μηχανής. Τη δεκαετία του ενενήντα η κύρια επιρροή ήταν η εξάπλωση των δικτύων υπολογιστών και ειδικότερα του διαδικτύου, καθώς κα η δημιουργία φορητών συσκευών. Τέλος τα τελευταία χρόνια η δημιουργία και εξάπλωση κινητών συσκευών, όπως τηλεφώνων, υπολογιστών παλάμης, φορητών υπολογιστών καθώς και η ελαχιστοποίηση του μεγέθους και της ενέργειας που καταναλώνουν οι υπολογιστικές μηχανές έστρεψε το ενδιαφέρον του τομέα της αλληλεπίδρασης ανθρώπου μηχανής στη δημιουργία κατάλληλων, εύχρηστων διεπαφών εφαρμογών και εξαρτημάτων χρήσης για τις εν λόγω συσκευές.

2.3 Αλληλεπίδραση ανθρώπου με συστήματα βάσεων δεδομένων

Ο ρυθμός με τον οποίο αυξάνεται σήμερα ο όγκος και το είδος της πληροφορίας έχει ξεπεράσει κάθε προηγούμενο. Ο χώρος των βάσεων δεδομένων είναι ίσως ο πιο άμεσα επηρεαζόμενος από την τάση αυτή. Το γεγονός αυτό έχει άμεση επίδραση όχι μόνο στα συστήματα που καλούνται να διαχειριστούν την πληροφορία αυτή αλλά και στον ανθρώπινο παράγοντα.

Μέχρι πριν όχι αρκετά χρόνια τα άτομα που ερχόντουσαν σε επαφή με συστήματα βάσεων δεδομένων ήταν κατά βάση εξειδικευμένα στελέχη με τουλάχιστον αρκετές τεχνικές γνώσεις. Σήμερα το εύρος των ατόμων που χρησιμοποιούν τέτοια συστήματα είναι εξαιρετικά μεγάλο. Σα φυσική συνέπεια ένας πιθανός χρήστης μπορεί να είναι λιγότερο εξοικειωμένος με τεχνικούς όρους ή μπορεί να μην έχει καμία προοπτική/διάθεση εκμάθησης νέων πραγμάτων.

Προσθετικά στην πολυπλοκότητα του προβλήματος λειτουργούν οι νέες ανάγκες που οδηγούν σε δημιουργία νέων τύπων βάσεων δεδομένων όπως είναι οι χωρικές, οι χρονικές και οι αντικειμενοστρεφείς. Σε αυτούς τους νέους τύπους είναι απαραίτητη η υιοθέτηση νέων τεχνικών βελτιστοποίησης.

Από τεχνικής πλευράς είναι σαφές πως η αλληλεπίδραση χρήστη-βάσης έγκειται σε δύο βασικούς παράγοντες:το μοντέλο δεδομένων(***data model***) και τη γλώσσα επερωτήσεων(***query***

language). Οι σχεδιαστές ΣΔΒΔ εφιστούν εκεί την προσοχή τους με σκοπό τη δημιουργία ενός interface που να μεγιστοποιεί την απόδοση του χρήστη. Για να γίνει αντιληπτό το πόσο σημαντικό είναι αυτό θα πρέπει να αναφέρουμε πως ο χρήστης ασυναίσθητα ταυτοποιεί το σύστημα με το interface που βλέπει στο τερματικό του.

Στην κλασσική θεωρία των βάσεων δεδομένων υπάρχει η εξής ιεραρχία μοντέλων :

- εννοιολογικό(conceptual)(παρέχει έννοιες που βρίσκονται κοντά στον τρόπο με τον οποίο ο χρήστης αντιλαμβάνεται τα δεδομένα).
- λογικό(logical) (παρέχει έναν πιο τυποποιημένο τρόπο περιγραφής των δεδομένων
- φυσικό(physical) (παρέχει έννοιες που περιγράφουν τις λεπτομέρειες του τρόπου αποθήκευσης των δεδομένων στον υπολογιστή)

Η διαδικασία ανάκτησης δεδομένων από ένα πληροφοριακό σύστημα συχνά προϋποθέτει πως ο χρήστης είναι σε θέση να κατανοήσει τον τρόπο λειτουργίας του τόσο σε εννοιολογικό όσο και σε λογικό επίπεδο. Αναπόφευκτα λοιπόν το επίπεδο των χρηστών καθώς και οι ιδιαιτερότητες του καθένα ξεχωριστά επηρεάζουν σημαντικά τη διαδικασία αυτή. Πως όμως θα μπορούσε να καταστεί δυνατή η υλοποίηση συστημάτων τα οποία θα μπορούσαν να χρησιμοποιηθούν από διαφορετικούς τύπους χρηστών?

Προς την κατεύθυνση αυτή κινείται η χρήση τεχνικών μοντελοποίησης που μετατοπίζουν το μοντέλο του interface σε πιο υψηλά(εννοιολογικά) επίπεδα. Από την άλλη εξίσου σημαντική μπορεί να αποδειχτεί η χρήση γραφικών γλωσσών επερωτήσεων(visual query languages). Αναμφισβήτητα όταν αναφερόμαστε σε μεγιστοποίηση της απόδοσης ενός χρήστη κάποιου συστήματος βάσεων δεδομένων και το ρόλο του HCI, εννοούμε κυρίως τη ικανότητά του να κάνει επερωτήσεις. Η ικανότητά του αυτή εν πολλοίς εξαρτάται από:

- ατομικές διαφορές
- τη δομή της βάσης
- την απεικόνιση της βάσης (όψη(view) της βάσης στο εννοιολογικό μοντέλο)
- τα ιδιαίτερα χαρακτηριστικά της εφαρμογής
- την γραφική απεικόνιση/οπτικοποίηση (visualization)των βασικών λειτουργιών.

Στο κομμάτι του visualization θα εστιάσουμε την προσοχή μας αμέσως τώρα.

2.4 Αποθήκες δεδομένων , OLAP

Υπάρχουν δύο βασικοί παράγοντες που έρχονται να αλλάξουν τα δεδομένα στο χώρο των βάσεων δεδομένων. Από τη μία η υπερβολική αύξηση του όγκου των δεδομένων που πρέπει να

διαχειριστεί ένα ΣΔΒΔ (ενώ παλαιότερα αναφερόμασταν σε Mega-bytes τώρα είμαστε αναγκασμένοι να μιλάμε για tera ή ακόμα και για peta-bytes) και από την άλλη η ραγδαία διείσδυση των βάσεων δεδομένων σε επιχειρηματικούς χώρους διαφορετικών αναγκών και απαιτήσεων, οδήγησαν στην αναζήτηση νέων τεχνολογιών που θα μπορούσαν να ανταποκριθούν στις νέες αυτές ανάγκες.

Δεν υπάρχει η παραμικρή αμφιβολία πως η σημαντικότερη από αυτές είναι οι αποθήκες δεδομένων (Data Warehouses). Ο W. H. Inmon, στον οποίον και αποδίδεται αρχικά η χρήση του όρου αυτού χαρακτήρισε μία αποθήκη δεδομένων ως μία συλλογή θεματικών, ενοποιημένων, μη ασταθών, μεταβαλλόμενων με το χρόνο δεδομένων για υποστήριξη διοικητικών αποφάσεων [Inmo96]. Μέσα σε μία αποθήκη διατηρούμε στοιχεία που μεταβάλλονται σταθερά μεν, με σχετικά αργούς ρυθμούς δε. Τυπικά σε μία αποθήκη τα δεδομένα μοντελοποιούνται πολυδιάστατα. Τα πολυδιάστατα μοντέλα επωφελούνται από τις έμφυτες συσχετίσεις των δεδομένων για να τοποθετούν τα δεδομένα σε πολυδιάστατους πίνακες που ονομάζονται κύβοι (cubes). Για δεδομένα που προσφέρονται για μορφοποίηση σε διαστάσεις, η απόδοση των επερωτήσεων σε πολυδιάστατους πίνακες μπορεί να είναι πολύ καλύτερη από ότι στο σχεσιακό μοντέλο δεδομένων. Σε μία παραδοσιακή βάση δεδομένων μπορεί να γίνονται χιλιάδες δοσοληψίες το λεπτό κάτι τέτοιο δε συμβαίνει με τις αποθήκες, για το λόγο αυτό άλλωστε είναι απαραίτητη μία διαφορετική σχεδιαστική προσέγγιση.

Όπως προαναφέραμε οι αποθήκες δεδομένων είναι στενά συνδεδεμένες με τον επιχειρηματικό κόσμο. Οι άνθρωποι που παίρνουν αποφάσεις σε τέτοια περιβάλλοντα έχουν ανάγκη από γρήγορες και άμεσες λειτουργίες πάνω σε τεράστιους όγκους δεδομένων. Οι παραδοσιακές βάσεις δεδομένων υποστηρίζουν άμεση επεξεργασία δοσοληψιών (online transaction processing (OLTP)) όπου όμως έχει γίνει βελτιστοποίηση για την επεξεργασία επερωτήσεων που μπορεί να προσεγγίζουν ένα μικρό μέρος της βάσης. Το κενό αυτό έρχονται να καλύψουν οι εφαρμογές OLAP(Online Analytical Processing). Αυτός ο όρος χρησιμοποιείται για να περιγραφεί η ανάλυση πολύπλοκων δεδομένων από την αποθήκη δεδομένων. Τα εργαλεία OLAP χρησιμοποιούν δυνατότητες κατανεμημένης επεξεργασίας για αναλύσεις που απαιτούν περισσότερη αποθήκευση και δυνατότητες επεξεργασίας από αυτές που μπορεί να υπάρχουν σε ατομικούς υπολογιστές γραφείου.

Πολλά έχουν ειπωθεί και έχουν γραφτεί όσο αναφορά το τι ακριβώς σημαίνει OLAP. Εμείς θα αρκεστούμε σε αυτά μιας και σκοπός μας είναι η αναφορά στο κομμάτι του visualization και των εργαλειών που υπάρχουν για αυτό.

2.5 Οπτικοποίηση ως βασικό κομμάτι του HCI

Στο παρελθόν οι ερευνητικές δραστηριότητες στο χώρο των βάσεων δεδομένων και στο χώρο του HCI ήταν μάλλον ανεξάρτητες. Η ανάγκη όμως για πιο αποτελεσματική απεικόνιση της πληροφορίας έκανε το έδαφος πρόσφορο για σύγκλιση δραστηριοτήτων. Η ανάγκη για νέες και αποδοτικές τεχνικές απεικόνισης της πληροφορίας προέκυψε από τη ραγδαία αύξηση του όγκου δεδομένων που διαχειρίζεται και κατά συνέπεια θα πρέπει να έχει τη δυνατότητα ένα ΣΔΒΔ. Μάλιστα υπάρχουν κάποιες εφαρμογές όπως είναι για παράδειγμα τα συστήματα γεωγραφικής πληροφορίας(GIS) ή όπως οι αποθήκες δεδομένων όπου η αποτελεσματική απεικόνιση των δεδομένων παίζει το σημαντικότερο λόγο.

Εύκολα μπορεί να καταλάβει κανείς γιατί το visualization θεωρείται από τα βασικότερα κομμάτια του HCI. Η ανθρώπινη φύση επιβάλλει τη δημιουργία εφαρμογών που αλληλεπιδρούν πιο άμεσα με τις ανθρώπινες αισθήσεις. Για να γίνουμε όμως πιο συγκεκριμένοι θα πρέπει να αναφέρουμε τα χαρακτηριστικά εκείνα που καθιστούν ένα visual interface αποδοτικό σε ένα σύστημα βάσεων δεδομένων με τις τυπικές λειτουργίες, αν και κάποιος μπορεί να αντιτείνει πως κάτι τέτοιο είναι καθαρά υποκειμενικό.

Συνοπτικά αναφέρουμε τις εξής προδιαγραφές :

- Ευκολία στη δημιουργία/επεξεργασία επερωτήσεων
- εύρεση πληροφορίας χρησιμοποιώντας παράλληλα επερωτήσεις (Querying) και περιήγηση (browsing/exploring)
- δυνατότητα εκμετάλλευσης πληροφορίας που προέρχεται από άλλους χρήστες
- Όσο το δυνατόν πιο κατανοητή απεικόνιση όχι μόνο του περιεχομένου αλλά και της εσωτερικής δομής της πληροφορίας.
- Δυνατότητα παρουσίασης μεγάλου όγκου πληροφοριών, πιθανά και σε περισσότερες από 2 διαστάσεις,
- Χρήση κατάλληλων χρωματισμών

Εξαιτίας του μεγάλου πλήθους εφαρμογών, είναι σαφές πως σε κάθε περίπτωση θα πρέπει να λαμβάνεται ειδική μέριμνα όσο αναφορά τις ιδιαιτερότητες της κάθε εφαρμογής. Αυτός ήταν και ο λόγος που οι πρώτες ερευνητικές προσπάθειες κατέληγαν σε διαφορετικές προσεγγίσεις, καθιστώντας αδύνατη τη δημιουργία ενός βασικού κορμού. Εδώ και μερικά χρόνια αυτό έχει αρχίσει να αλλάζει μιας και έχει γίνει περισσότερο από ποτέ έντονη η ανάγκη ενιαίας απεικόνισης της πληροφορίας, ακόμα και αν αυτή προέρχεται από διαφορετικές πηγές.

Είναι σαφές πως ο κλάδος αυτός , και ιδιαίτερα στην κοινότητα των βάσεων δεδομένων, δεν έχει γίνει αντικείμενο εκτενούς έρευνας. Νέες τεχνολογίες όμως, όπως οι αποθήκες δεδομένων κάνουν την ανάγκη αυτή επιτακτική.

2.6 Οπτικοποίηση OLAP εφαρμογών

Τα τελευταία χρόνια έχει παρατηρηθεί μεγάλη ερευνητική δραστηριότητα στο χώρο των αποθηκών δεδομένων και των OLAP εφαρμογών, με σημαντικά μάλιστα αποτελέσματα. Εντούτοις όπως προαναφέρθηκε, η προσπάθεια αυτή επικεντρώθηκε κυρίως στην μοντελοποίηση των δεδομένων (modeling of data), παραμερίζοντας κατά κάποιο τρόπο τον επίσης σημαντικό τομέα του visualization. Το "κενό" αυτό έχει δημιουργήσει μία σχετική σύγχυση για τους λόγους που θα αναφέρουμε στη συνέχεια.

Οι κατασκευαστές OLAP συστημάτων έχουν δημιουργήσει αρκετά (πολυδιάστατα) μοντέλα δεδομένων πάνω στα οποία έχουν χτιστεί οι εφαρμογές τους . Το κύριο όμως μειονέκτημα πολλών από αυτών των εφαρμογών είναι πως στην ουσία πρόκειται για λογικά και όχι εννοιολογικά μοντέλα, κάτι που δυσκολεύει την ανάλυση τους. Επιπρόσθετα πολύ λίγα από αυτά παρέχουν ένα διακριτό μοντέλο παρουσίασης (presentational model) για OLAP λειτουργίες. Τις περισσότερες φορές αυτό αποτελεί μέρος του λογικού ή εννοιολογικού μοντέλου με αποτέλεσμα τον περιορισμό του εύρους της χρηστικότητας.

Η σύγχυση στην οποία αναφερθήκαμε όσο αφορά το κομμάτι του visualization έγκειται σε αυτό ακριβώς το γεγονός. Η απουσία ενός κοινά αποδεκτού μοντέλου παρουσίασης (presentational model) επιβραδύνει την πρόοδο στον τομέα αυτό. Ανεξάρτητα από αυτό στη συνέχεια θα γίνει προσπάθεια να καταγραφούν οι απαιτήσεις που έχει ο τελικός χρήστης από ένα visualization tool. Σύμφωνα με τις απαιτήσεις αυτές θα αξιολογήσουμε στην επόμενη παράγραφο μερικά από τα πιο διαδεδομένα εμπορικά εργαλεία.

Συνοπτικά λοιπόν από τις εφαρμογές προκύπτει πως σημαντικά χαρακτηριστικά είναι τα εξής :

Pivot Table

Άμεση και εύκολη πρόσβαση στο σχήμα του κύβου. Ο χρήστης θα πρέπει να είναι σε θέση να βλέπει όλες τις διαστάσεις καθώς και την ιεραρχία που διέπει καθεμία από αυτές.

OLAP navigation

Φιλικό προς τον χρήστη interface. Είναι απαραίτητο ο χρήστης να μπορεί με ευκολία να προσθέτει/αφαιρεί(slicing) στο σχήμα τις διάφορες διαστάσεις. Αυτό θα είναι εφικτό μέσω κάποιου panel που θα υπάρχει. Δηλαδή απαιτείται η ύπαρξη κάποιου OLAP navigator.

OLAP operations

Βασικές OLAP operations. Πιο συγκεκριμένα απαιτούνται οι εξής λειτουργίες

- *Αναλυτική Άνοδος (Drill up)*:Κρατώντας την ίδια ιεραρχία θέλουμε μετάβαση σε πιο γενικό (συναθροιστικό) επίπεδο
- *Αναλυτική Κάθοδος (Drill down):* Κρατώντας την ίδια ιεραρχία θέλουμε μετάβαση σε πιο λεπτομερές επίπεδο
- *Οριζόντιος Τεμαχισμός (Slice):* (σχεσιακή) επιλογή
- *Κάθετος Τεμαχισμός (Dice):* (σχεσιακή) προβολή
- *Περιστροφή (Pivot):* αναδιάταξη της 2D προβολής του πολυδιάστατου κύβου στην οθόνη

OLAP patterns

Υποστήριξη λειτουργιών/συναρτήσεων πάνω στα δεδομένα της οθόνης όπως για παράδειγμα :

- max:εύρεση μέγιστης τιμής
- min: εύρεση ελάχιστης τιμής
- average:Εύρεση μέσης τιμής
- multiple colors :Χρησιμοποίηση διαφορετικού χρωματισμού στα κελιά που ανήκουν στα αποτελέσματα των patterns

Διαγράμματα (charts)

Ευρεία χρήση διαγραμμάτων . Ιδιαίτερα σημαντική ως προς το τελικο αισθητικό αποτέλεσμα, καθώς κάνει το περιβάλλον πολύ πιο φιλικό για το χρήστη. Η απεικόνιση των στοιχείων που παρουσιάζονται στην οθόνη, πρέπει να γίνεται σε διάφορους τύπους διαγραμμάτων.

Αναφορές(reports)

Εύκολη μεταφορά αποτελεσμάτων, συμπερασμάτων με σκοπό τη δημιουργία αναφορών. Ιδιαίτερα σημαντική είναι η σύνδεση με σελίδες web.

Offline επεξεργασία

Είναι πολύ σημαντική η δυνατότητα επεξεργασίας δεδομένων ακόμα και όταν δεν υπάρχει σύνδεση με τον OLAP server.

Presentational model

Η ανάπτυξη ενός σαφώς καθορισμένου και αυτόνομου presentational model συγκαταλέγεται στα πολύ θετικά στοιχεία.

Mobility

Αποτελεσματική απεικόνιση σε κάθε είδος ασύρματης συσκευής(Pocket PC,Palm,mobile phones) λαμβάνοντας υπόψη βέβαια τα ιδιαίτερα χαρακτηριστικά αυτών, και ιδιαίτερα το μικρό μέγεθος της οθόνης τους.

2.7 Παρουσιαστικά εργαλεία

Τα τελευταία 5 χρόνια εμφανίστηκε στην αγορά μία πληθώρα εμπορικών εργαλείων που υποστηρίζουν OLAP εφαρμογές. Όπως είναι λογικό, καθένα από αυτά έχει να επιδείξει κάποια ιδιαίτερα χαρακτηριστικά. Σκοπός της παραγράφου τούτης δεν είναι η ανάλυση λεπτών τεχνικών λεπτομερειών αλλά μία συνοπτική απεικόνιση των στρατηγικών που οι κατασκευαστές ακολούθησαν ώστε να ανταποκριθούν στις νέες ανάγκες που περιγράφηκαν προηγουμένως. Στην προσπάθεια βέβαια αυτή , ανασταλτικό ρόλο αποτέλεσε το γεγονός πως τα περισσότερα από αυτά δεν είναι open source οπότε αναπόφευκτα βασιστήκαμε στους ισχυρισμούς των κατασκευαστών.

- *Active Cube:* Πρόκειται για ένα εργαλείο βασισμένο στην ROLAP αρχιτεκτονική. Αποτελείται από δύο κομμάτια, ένα για την επικοινωνία με τα δεδομένα(τον data server δηλαδή), και ένα για την επεξεργασία ώστε η παρουσίαση να γίνεται σε πολυδιάστατη μορφή. Χρησιμοποιείται η γλώσσα MDX της Microsoft. Παρέχεται η δυνατότητα επεξεργασίας ακόμα και σε offline κατάσταση. Επίσης παρέχονται ικανοποιητικές δυνατότητες reporting. Μειονέκτημα η περιορισμένη συμβατότητα με OLAP servers(υποστηρίζει κυρίως προϊόντα της Microsoft). Απουσιάζει η υποστήριξη σε ασύρματες συσκευές.
- *POWER OLAP:* Οι κατασκευαστές του υποστηρίζουν πως παρέχουν την πρώτη υβριδική μορφή OLAP συστήματος, δημιουργώντας μία HOLAP (Hybrid OLAP) αρχιτεκτονική, αξιοποιώντας τα πλεονεκτήματα τόσο του σχεσιακού όσο και του πολυδιάστατου κόσμου. Το PowerOLAP δεν προ-υπολογίζει όλες τις παραγόμενες τιμές στους κύβους. Οι τιμές υπολογίζονται μόνο μετά από απαίτηση του χρήστη, υπάρχει δηλαδή μία δυναμική συμπεριφορά. Η απόδοση διατηρείται σε ικανοποιητικά επίπεδα αποθηκεύοντας δεδομένα στη μνήμη και χρησιμοποιώντας αλγόριθμους βασισμένους

στα κατά κανόνα αραιά δεδομένα των επιχειρηματικών εφαρμογών. Παρέχεται η δυνατότητα επεξεργασίας ακόμα και σε offline κατάσταση. Εξαιρετικές κρίνονται οι δυνατότητες reporting. Απουσιάζει η υποστήριξη σε ασύρματες συσκευές.

- *Hyperion-Brio:* Μέχρι και πριν από λίγο καιρό αποτελούσαν ξεχωριστές υλοποιήσεις με το Hyperion να επιδεικνύει το μόνο ολοκληρωμένο περιβάλλον παρέχοντας αυτόνομο MOLAP ΣΔΒΔ, και το Brio να παρέχει πολλαπλές δυνατότητες visualization και reporting, υποστηρίζοντας παράλληλα broadcasting σε ασύρματες συσκευές. Το φθινόπωρο του 2003 η Hyperion εξαγόρασε την Brio, παρέχοντας ένα πολύ ικανοποιητικό και ολοκληρωμένο πακέτο υπηρεσιών. Η απουσία δυνατότητας drill anywhere, αντιμετωπίστηκε σε τελευταία έκδοση με την πρόσβαση σε σχεσιακά δεδομένα μέσω ενός σχήματος αστέρα([BRIO03]).
- *Microstrategy:* Αποτελεί ένα ολοκληρωμένο ROLAP σύστημα. Η αρχιτεκτονική αυτή επιτρέπει έναν ad-hoc προσδιορισμό των drill ιδιοτήτων, αφού τα συναθροιστικά δεδομένα υπολογίζονται δυναμικά, σε αντίθεση με την MOLAP αρχιτεκτονική όπου κάτι τέτοιο προκαθορίζεται στην αρχή μιας λειτουργίας. ΟΙ δυνατότητες reporting είναι πάρα πολλές. Μεγάλη προσοχή έχει δοθεί στη διασύνδεση με web περιεχόμενο. Υπάρχει η δυνατότητα διασύνδεσης μέσω ενός κοινού browser μιας και πολλά εργαλεία της πλατφόρμας είναι βασισμένα σε XML αρχιτεκτονική. Υποστηρίζεται broadcasting σε ασύρματες συσκευές([MSTR03]).
- *ADO Explorer:* Ο ADO Explorer είναι ένα SQL εργαλείο που όπως υποδηλώνει το όνομά του χρησιμοποιεί αντικείμενα ADO (ActiveX Database Objects). Πέρα από τις ικανοποιητικές επιδόσεις σε θέματα reporting, το πλέον εντυπωσιακό είναι η συμβατότητά του με όλες τις βάσεις. Απουσιάζει η υποστήριξη σε ασύρματες συσκευές.
- *Polaris*: Πρόκειται για μία ακαδημαϊκή πλατφόρμα, που αναπτύχθηκε στο πανεπιστήμιο του Stanford. Βασίστηκε στο γνωστό Pivot Table για να μπορέσει να απεικονίσει αποτελέσματα σχεσιακών επερωτήσεων με χρήση αρκετά ικανοποιητικών οπτικών μεθόδων. Παράγει πλούσιες πινακοειδείς οπτικές απεικονίσεις αντί για πίνακες κειμένου. Το interface θεωρείται ως ένα visual specification που συνίσταται σε :α)παραμετροποίηση πίνακα β)είδος γραφικών σε κάθε pane γ)κωδικοποίηση των δεδομένων ως οπτικές ιδιότητες στοιχείων. Κύριο χαρακτηριστικό του η δυνατότητα κατασκευής visual specifications για οπτική απεικόνιση, και η αυτόματη δημιουργία κάποιων συγκεκριμένων επερωτήσεων. Απουσιάζει η υποστήριξη σε ασύρματες συσκευές.

Υπάρχουν πολλά ακόμα εργαλεία στα οποία θα μπορούσε να γίνει αναφορά. Αυτό που έχει σημασία όμως, και προκύπτει από όσα προαναφέραμε, είναι η σημαντική πρόοδος του συνόλου αυτών. Πράγματι οι δυνατότητες που παρέχονται σε θέματα visualization και reporting είναι μάλλον αρκετές. Αντίθετα παρατηρήθηκε πως το κομμάτι που αφορά τις ασύρματες συσκευές δεν είναι στις άμεσες προτεραιότητες των κατασκευαστών. Η απουσία ενός αυτόνομου μοντέλου παρουσίασης είναι σχεδόν καθολική, αν εξαιρέσουμε την υλοποίηση της *Microstrategy.*

3 *Σχεδίαση*

Το παρόν κεφάλαιο ασχολείται με τη σχεδίαση του εργαλείου μας, Στην παράγραφο 3.1 γίνεται μία γενική αναφορά στην έννοια του μοντέλου παρουσίασης , στην παράγραφο 3.2 γίνεται μία εκτενής παρουσίαση του CPM(Cube Presentation Model) [MVSV03] το οποίο αποτελεί ένα νέο αυτόνομο μοντέλο παρουσίασης, ενώ στην παράγραφο 3.3 γίνεται μία ανάλυση των απαιτήσεων χρήστη της εφαρμογής οπτικοποίησης πολυδιάστατων δεδομένων που θα υλοποιήσουμε .

3.1 Μοντέλα παρουσίασης

Τα τελευταία χρόνια η έρευνα στο χώρο των βάσεων δεδομένων εστίασε την προσοχή της στις αποθήκες δεδομένων καθώς και στις OLAP εφαρμογές. Η δραστηριότητα αυτή όμως δεν ήταν τόσο ισορροπημένη μιας και παρόλο που στον τομέα της μοντελοποίησης των δεδομένων υπήρξε ραγδαία πρόοδος, στον εξίσου σημαντικό τομέα της οπτικοποίησης των δεδομένων η πρόοδος υπήρξε μικρή.

Όσο αφορά τη μοντελοποίηση των δεδομένων για τη δημιουργία μίας αποθήκης δεδομένων, παρόλες τις διαφορετικές προσεγγίσεις που ακολουθήθηκαν από τους κατασκευαστές δεδομένων, εν τούτοις τα μοντέλα που χρησιμοποιήθηκαν μοιράζονται κάποιες κοινές έννοιες, όπως αυτές της διάστασης, της ιεραρχίας, των επιπέδων και του μέτρου.Το γεγονός αυτό όμως δεν έχει οδηγήσει ακόμα στην υιοθέτηση κάποιων κοινά αποδεκτών και προτυποποιημένων μοντέλων.

Το κομμάτι της οπτικοποίησης δεδομένων είναι αναμφίβολα εξίσου σημαντικό. Παρόλα αυτά η έννοια του μοντέλου παρουσίασης πολύ σπάνια υπάρχει αυτόνομα σε κάποια OLAP εφαρμογή.Τα μοντέλα παρουσίασης δεν αποτελούν μέρος της κλασσικής ιεραρχίας εννοιολογικού-λογικού-φυσικού των μοντέλων βάσεων δεδομένων. Λόγω όμως της φύσης των OLAP εφαρμογών, που είναι άμεσα συνδεδεμένες με συστήματα αποφάσεων(DSS-Decision Support Systems), είναι αυτονόητο πως η οπτικοποίηση των δεδομένων είναι ιδιαίτερα σημαντική, πολύ περισσότερο αν αναλογιστεί κανείς πως συχνά οι εφαρμογές αυτές χρησιμοποιούνται από μη τεχνικά εξειδικευμένο προσωπικό.

Η ανάγκη αυτή έγινε αντιληπτή από τη Microsoft η οποία έχει ήδη παρουσιάσει ένα εμπορικό πρότυπο για πολυδιάστατες βάσεις δεδομένων, όπου θέματα παρουσίασης κατέχουν σημαντικό ρόλο [MS98]. Στην προσέγγιση αυτή εξέχων ρόλο παίζει μία δυνατή γλώσσα επερωτήσεων ,η MDX (Multidimensional Expressions), την οποία και θα χρησιμοποιήσουμε στην εφαρμογή μας. Το πρότυπο αυτό όμως έχει κάποια βασικά μειονεκτήματα.Για παράδειγμα το λογικό μοντέλο και το μοντέλο παρουσίασης περιπλέκονται αρκετά, με αποτέλεσμα η γλώσσα αν και αρκετά δυνατή να είναι δύσχρηστη. Στο Σχήμα 3.1 παρατίθεται ένα παράδειγμα από το πρότυπο αυτό.

```
SELECT  CROSSJOIN ({Venk,Netz}, {USA_N.Children, USA_S,Japan}) ON COLUMNS
        {Qtr1.CHILDREN, Qtr2, Qtr3, Qtr4.CHILDREN} ON ROWS
FROM    SalesCube
WHERE   Sales, [1991], Products.ALL)
```

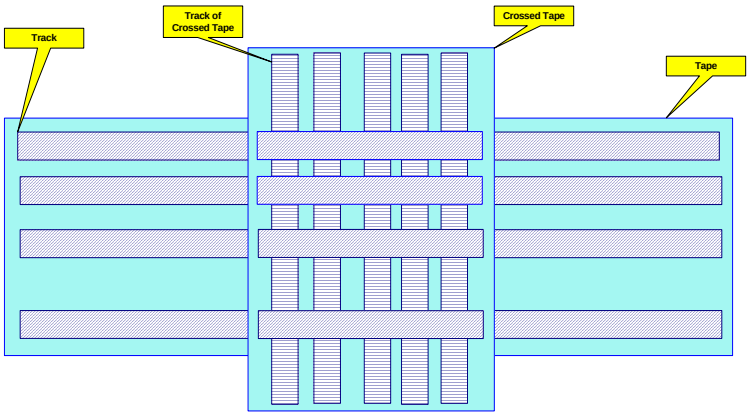
Year = 1991
Product = ALL

			C1		C2	C3	C4		C5	C6
			Venk				Netz			
			USA			Japan	USA			Japan
			USA_N		USA_S		USA_N		USA_S	
			Seattle	Boston			Seattle	Boston		
		Size (City)								
R1	Qtr1	Jan	20	32	62	97	23	40	75	12
		Feb	25	40	74	121	18	32	51	20
		Mar	18	12	36	110	42	48	65	3
R2	Qtr2		56	63	150	253	50	70	280	50
R3	Qtr3		52	65	147	200	53	64	270	50
R4	Qtr4	Oct	25	24	64	98	32	12	64	76
		Nov	28	28	76	102	40	21	83	69
		Dec	23	30	68	150	42	29	99	77

Σχήμα 3.1 -- Παράδειγμα από το Cube Model [MS98]

Η ακαδημαϊκή κοινότητα έχει με τη σειρά της συνεισφέρει στην προσπάθεια αυτή, με το Tape Model [GJJ97]. Το μοντέλο αυτό βασίζεται, όπως γίνεται φανερό και από την ονομασία του, στην έννοια της ταινίας (tape).Παρόλα κάποια ενδιαφέροντα χαρακτηριστικά, το μοντέλο αυτό

παρουσιάζεται μάλλον περιορισμένο ως προς την 'εκφραστικότητά' του σε σχέση με την πρόταση της Microsoft.



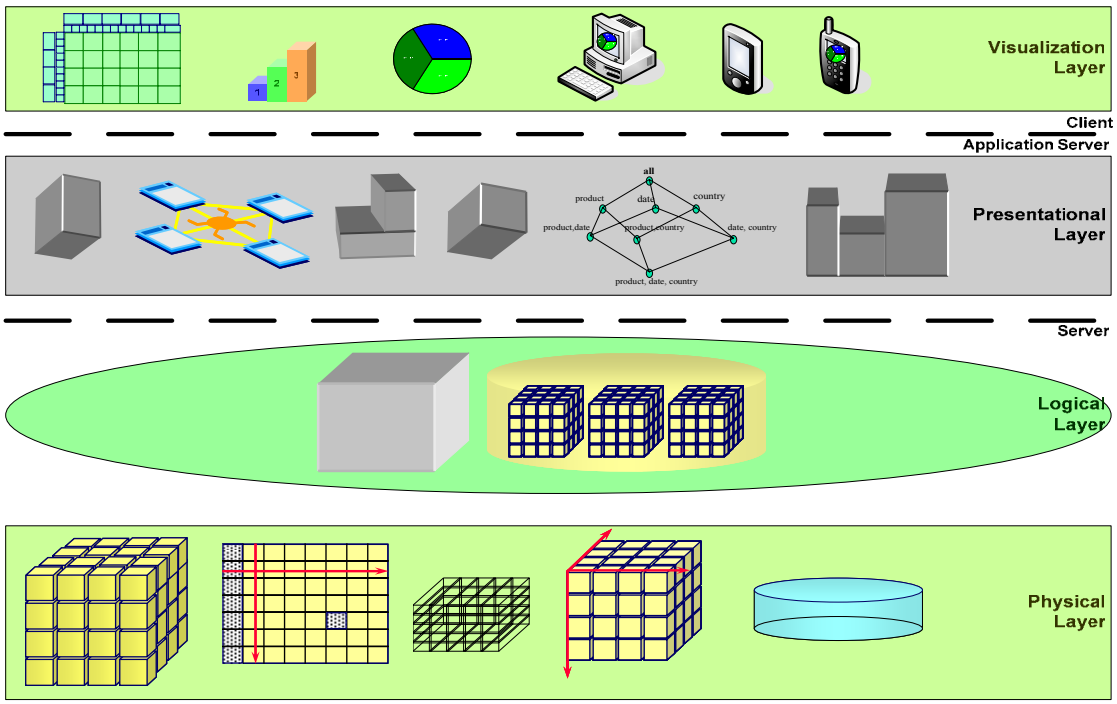
Σχήμα 3.2 -- Μία δομή ταινίας(tape) με 2 τεμνόμενες ταινίες

3.2 CPM(Cube Presentation Model)

3.2.1 Εισαγωγή

Στην παρούσα ενότητα θα γίνει μία αναλυτική παρουσίαση του CPM(Cube Presentation Model) [MVSV03], το οποίο και αποτελεί τη μοναδική ολοκληρωμένη πρόταση ενός διακριτού και αυτόνομου μοντέλου παρουσίασης για απεικόνιση OLAP εφαρμογών σε διδιάστατες απεικονίσεις.

Το κύριο χαρακτηριστικό του CPM είναι ο πλήρης διαχωρισμός της ανάκτησης των δεδομένων, που γίνεται στο λογικό στρώμα του μοντέλου (logical layer), και της απεικόνισης των δεδομένων αυτών, η οποία γίνεται στο παρουσιαστικό στρώμα του μοντέλου (presentational layer). Αυτός ο διαχωρισμός συμβάλλει σημαντικά στην χρηστικότητα του μοντέλου, μιας και με τον τρόπο αυτό είναι δυνατή η αντικατάσταση του ενός από τα δύο στρώματα χωρίς να δημιουργείται πρόβλημα στο άλλο. Η αντικατάσταση για παράδειγμα του λογικού στρώματος με οποιοδήποτε άλλο μοντέλο που να υιοθετεί τις κλασσικές έννοιες του πολυδιάστατου μοντέλου (ιεραρχίες,διαστάσεις,κύβους) μπορεί να υλοποιηθεί χωρίς δυσκολία.



Σχήμα 3.3 -- Αρχιτεκτονική του CPM

3.2.2 Λογικό στρώμα του CPM

Στην παράγραφο τούτη θα γίνει μία σύντομη περιγραφή του λογικού στρώματος του CPM, περισσότερο για λόγο πληρότητας μιας και ο κύριος στόχος της διπλωματικής είναι η υλοποίηση του παρουσιαστικού στρώματος. Το λογικό στρώμα λοιπόν αποτελεί ουσιαστικά μία επέκταση του παλαιότερου λογικού μοντέλου [VaSk00]. Η επέκταση γίνεται με σκοπό τον υπολογισμό πιο πολύπλοκων κύβων. Στο σημείο αυτό θα παραθέσουμε τις βασικότερες αρχές/παραδοχές του [VaSk00] (για λεπτομερέστερη περιγραφή ο αναγνώστης παραπέμπεται στην αντίστοιχη δημοσίευση):

- Μία διάσταση είναι ένα πλέγμα από επίπεδα διαστάσεων $(L,\prec)$, όπου $\prec$ είναι μία μερική σειρά πάνω στα επίπεδα του L.
- Ορίζεται μία οικογένεια μονότονων, διμελών συνεπών *ancestor συναρτήσεων* anc^{L2}_{L1}, έτσι ώστε για κάθε ζευγάρι επιπέδων L_1 και L_2 με $L_1 \prec L_2$, η συνάρτηση anc^{L2}_{L1} αντιστοιχίζει κάθε στοιχείο του $dom(L_1)$ σε ένα στοιχείο του $dom(L_2)$.
- Ένα σύνολο δεδομένων DS πάνω σε ένα σχήμα $S=[L_1,\ldots,L_n,A_1,\ldots,A_m]$ είναι ένα πεπερασμένο σύνολο tuples πάνω στο S έτσι ώστε $[L_1,\ldots,L_n]$ είναι επίπεδα , οι υπόλοιπες

ιδιότητες είναι μέτρα και $[L_1,\ldots,L_n]$ είναι ένα πρωτεύον κλειδί. Ένα λεπτομερές σύνολο τιμών DS^0 είναι ένα σύνολο δεδομένων όπου όλα τα επίπεδα είναι στο κατώτερο σημείο των ιεραρχιών τους

- Μία συνθήκη επιλογής φ είναι μία φόρμουλα που περιλαμβάνει άτομα(atoms) και τους λογικούς συνδέσμους $\wedge$, $\vee$ και $\neg$. The atoms περιλαμβάνει επίπεδα, τιμές και ancestor συναρτήσεις, με όρους στη μορφή x ∂ y. Μία λεπτομερής συνθήκη επιλογής περιλαμβάνει επίπεδα στο κατώτερο σημείο των ιεραρχιών τους
- Ένας πρωτεύων κύβος c (πάνω στο σχήμα $[L_1,\ldots,L_n,M_1,\ldots,M_m]$), είνα μία έκφραση της μορφής $c=(DS^0,\varphi,[L_1,\ldots,L_n,M_1,\ldots,M_m],[agg_1(M^0_1),\ldots,agg_m(M^0_m)])$, όπου:
    - DS^0 *είναι* ένα λεπτομερές σύνολο τιμών πάνω στο σχήμα $S=[L^0_1,\ldots,L^0n,M^0_1,\ldots,M^0_k],m\leq k$.
    - φ είναι μία λεπτομερής συνθήκη επιλογής.
    - $M_1, \ldots, M_m$ είναι μέτρα.
    - L^0_i και L_i είναι επίπεδα έτσι ώστε $L^0_i \prec L_i$, $1\leq i\leq n$.
    - $agg_i \in \{sum,min,max,count\}$, $1\leq i\leq m$.

Οι περιορισμοί των πρωτευόντων κύβων συνίστανται στο γεγονός πως παρόλο που μοντελοποιούν με ακρίβεια επερωτήσεις του τύπου SELECT-FROM-WHERE-GROUPBY, αποτυγχάνουν στα (α) ταξινόμηση (ordering), (β) υπολογισμό τιμών μέσω συναρτήσεων (γ) επιλογή (Select) πάνω σε υπολογισμένες ή συναθροισμένες τιμές (π.χ τον όρο HAVING ενός SQL query). Για να ξεπεραστούν οι αδυναμίες αυτές , το προηγούμενο μοντέλο επεκτείνεται με τις ακόλουθες οντότητες :

- Έστω F ένα σύνολο που αντιστοιχίζει σύνολα γνωρισμάτων(attributes) σε γνωρίσματα. Γίνεται διαχωρισμός των ακολούθων κύριων κατηγοριών συναρτήσεων: συναρτήσεις ιδιοτήτων, αριθμητικές συναρτήσεις και συναρτήσεις ελέγχου. Για παράδειγμα, για το επίπεδο Day, μπορεί να υπάρχει η συνάρτηση ιδιότητας holiday(Day) που δείχνει αν μία μέρα είναι γιορτή ή όχι. Μία αριθμητική συνάρτηση για παράδειγμα είναι η Profit=(Price-Cost)*Sold_Items. Οι συναρτήσεις ελέγχου προσομοιώνουν τις εντολές ελέγχου των γλωσσών προγραμματισμού
- Μία δευτερεύουσα συνθήκη επιλογής ψ είναι μία φόρμουλα σε διαχωριστική κανονική μορφή. Ένα άτομο μίας δευτερεύουσας συνθήκης επιλογής είναι σωστό,λάθος ή μία έκφραση της μορφής x θ y, όπου x και y μπορούν να είναι ένα από τα ακόλουθα: (α) ένα

γνώρισμα A_i (συμπεριλαμβανομένου RANK), (β) μία τιμή l , μία έκφραση της μορφής $f_i(A_i)$, όπου A_i είναι ένα σύνολο γνωρισμάτων (επίπεδα και μέτρα) και (γ) θ είναι ένας τελεστής από το σύνολο $(>, <, =, \geq, \leq, \neq)$. Με τον τρόπο αυτό, είναι δυνατόν να υπολογιστούν σχέσεις μεταξύ των μέτρων (Cost>Price), επιλογές ταξινόμησης(Rank) και εύρους(Range) (ORDER BY...;STOP after 200, RANK[20:30]), επιλογές μέτρου (sales>3000), ιδιότητες βασισμένες στην επιλογή (Color(Product)='Green')

- Έστω ένα σύνολο δεδομένων DS πάνω στο σχήμα $[A_1, A_2, \ldots, A_z]$. Χωρίς βλάβη της γενικότητας, έστω ένα μη κενό υποσύνολο του σχήματος $S=A_1,\ldots,A_k, k\leq z$. Έπειτα, υπάρχει ένα σύνολο λειτουργιών ταξινόμησης O^{θ}_S, το οποίο ταξινομεί τις τιμές του συνόλου δεδομένων, ανάλογα με το σύνολο γνωρισμάτων που υπάρχουν στο S. Το θ ανήκει στο σύνολο $\{<,>,\square\}$ για να υποδηλώσει αύξουσα σειρά, κατιούσα σειρά και καμία σειρά, αντίστοιχα. Μία αύξουσα λειτουργία εφαρμόζεται σε ένα σύνολο δεδομένων και επιστρέφει ένα άλλο σύνολο δεδομένων το οποίο υποχρεωτικά περιλαμβάνει το μέτρο RANK.
- Ένας δευτερεύων κύβος πάνω στο σχήμα $S=[L_1,\ldots,L_n,M_1,\ldots,M_m,A_{m+1},\ldots,A_{m+p}, RANK]$ είναι μία έκφραση που έχει τη μορφή

  $s=[c,[A_{m+1}:f_{m+1}(\mathbf{A}_{m+1}),\ldots,A_{m+p}:f_{m+p}(\mathbf{A}_{m+p})],O\theta\mathbf{A},\psi]$ όπου $c=(DS^0,\varphi,$

  $[L_1,\ldots,L_n,M_1,\ldots,M_m],[agg_1(M01),\ldots,agg_m(M0m)])$ είναι ένας πρωτεύων κύβος,

  $[\mathbf{A}_{m+1},\ldots,\mathbf{A}_{m+p}]\subseteq[L_1,\ldots,L_n,M_1,\ldots,M_m], \mathbf{A}\subseteq S-\{RANK\}, f_{m+1},\ldots,f_{m+p}$ είναι συναρτήσεις που ανήκουν στο F και ψ είναι μία δευτερεύουσα συνθήκη επιλογής
- Ένας δευτερεύων κύβος έχει την ακόλουθη τυπική σημασιολογία :

  $s=\{x\in Tup(L_1,\ldots,L_n,M_1,\ldots,M_m,A_{m+1},\ldots,A_{m+p},RANK) \mid \exists$ σύνολο δεδομένων DS^s ορισμένο πάνω στο σχήμα $[L_1,\ldots,L_n,M_1,\ldots,M_m,A_{m+1},\ldots,A_{m+p}], y\in DS^s, y_1\in c^1 : y[L_i]=y_1[L_i],$

  $1\leq i\leq n,\ y[M_i]=y_1[M_i], 1\leq i\leq m,\ y[A_{m+i}]=f_{m+i}[\mathbf{A}_{m+i}], 1\leq i\leq p, x\in\psi(O\theta A(DS^s))\}$

Με τις επεκτάσεις αυτές , οι πρωτεύοντες κύβοι επεκτείνονται σε δευτερεύοντες οι οποίοι εμπεριέχουν: (α) υπολογισμό νέων γνωρισμάτων (A_{m+i}) μέσω των αντιστοίχων συναρτήσεων (f_{m+i}), (β) ταξινόμηση $(O\theta\mathbf{A})$ και (γ) τον όρο HAVING μέσω της δευτερεύουσας συνθήκης επιλογής ψ.

Η αναφορά στο λογικό στρώμα θα σταματήσει στο σημείο τούτο μιας και μία πιο λεπτομερής αναφορά ξεφεύγει από το θέμα της διπλωματικής. Για περισσότερα ο αναγνώστης παραπέμπεται στα (Maniatis et al., 2003, Maniatis et al., 2003a, Maniatis, 2003, Maniatis, 2004). Ακολουθεί η ανάλυση του παρουσιαστικού στρώματος.

3.2.3 Παρουσιαστικό στρώμα του CPM

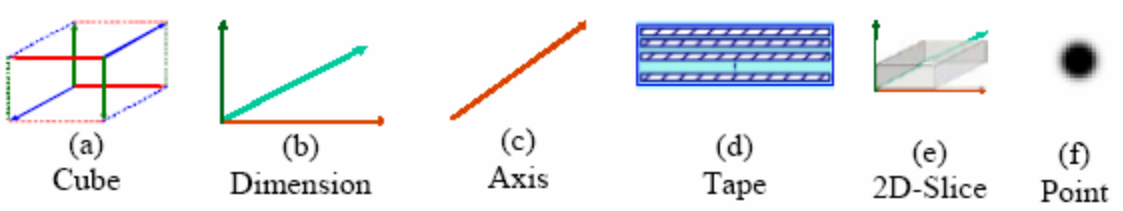
Στην παράγραφο τούτη θα περιγραφεί το παρουσιαστικό στρώμα του CPM, το οποίο και χρησιοποιήσαμε στην εφαρμογή μας. Αρχικά θα παρατεθούν οι βασικές οντότητες του μοντέλου με τη βοήθεια του παραδείγματος του Σχ.3.1 και στη συνέχεια θα δοθούν οι πιο τυπικοί ορισμοί των διαφόρων οντοτήτων του μοντέλου.

3.2.3.1 Βασικές οντότητες

Οι βασικότερες οντότητες του παρουσιαστικού στρώματος του CPM απεικονίζονται στα Σχ.3.4 και Σχ.3.5 και είναι οι εξής:

- **Σημεία(Points):** Ένα σημείο πάνω σε έναν άξονα αντιπροσωπεύει την αντίστοιχη κλασσική μαθηματική έννοια. Στη απλούστερη περίπτωση ένα σημείο χαρακτηρίζεται από μία συνθήκη επιλογής ισότητας πάνω σε ένα επίπεδο. Όμως επειδή ομαδοποιούμε περισσότερα από ένα γνωρίσματα σε κάθε άξονα, δηλαδή ένας άξονας παρουσίασης μπορεί να περιέχει περοσσότερες από μία λογικές διαστάσεις (με σκοπό την απεικόνισή τους σε διδιάστατη μορφή), τυπικά ένα σημείο είναι ένα ζευγάρι που αποτελείται από ένα σύνολο γνωρισμάτων ομάδας(με ένα από αυτά να λειτουργεί ως των πρωτεύον κλειδί) και ένα σύνολο συνθηκών επιλογής ισότητας για κάθε κλειδί

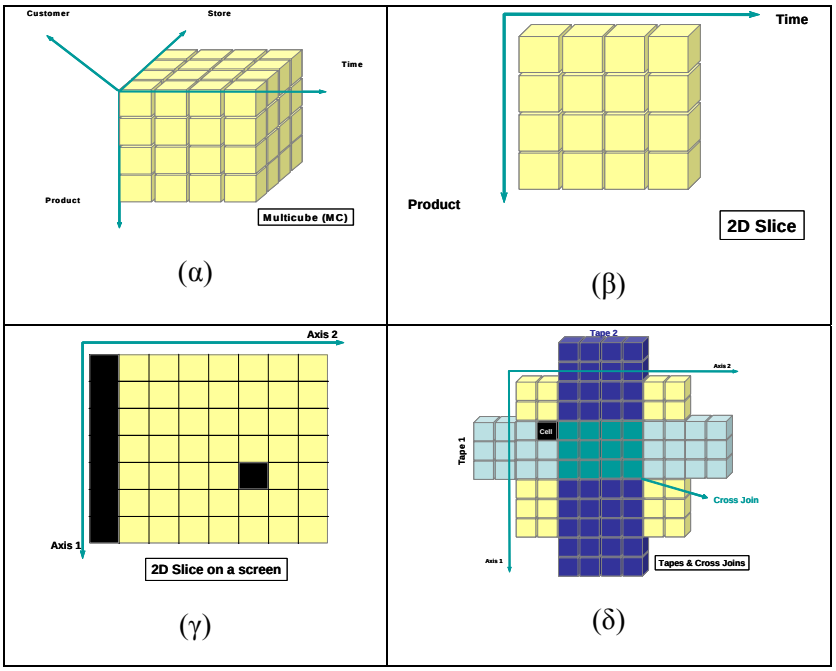
- **Άξονας(Axis):** Τα OLAP συστήματα βασίζονται στην πολυδιάστατη διαχείριση της πληροφορίας. Κάθε διάσταση ενός τέτοιου συστήματος αντιπροσωπεύει εννοιολογικά μία διαφορετική κατηγοριοποίηση πληροφορίας. Ιδεατά θα θέλαμε κάθε άξονας να περιέχει μία μόνο διάσταση. Εν τούτοις στο CPM, για λόγους παρουσίασης κάτι τέτοιο δε συμβαίνει αφού όπως θα δούμε στη συνέχεια ένας άξονας μπορεί να περιέχει περισσότερες από μία διαστάσεις
- **Πολυκύβοι(Multicubes):** Για την υλοποίηση ενός πολυδιάστατου χώρου είναι αναγκαίος ο συνδυασμός αξόνων. Οι τιμές των σημείων του πολυδιάστατου χώρου είναι οι συναθροισμένες τιμές των μέτρων που υπολογίζονται από το λογικό μοντέλο



Σχήμα 3. 4 -- CPM Οντότητες

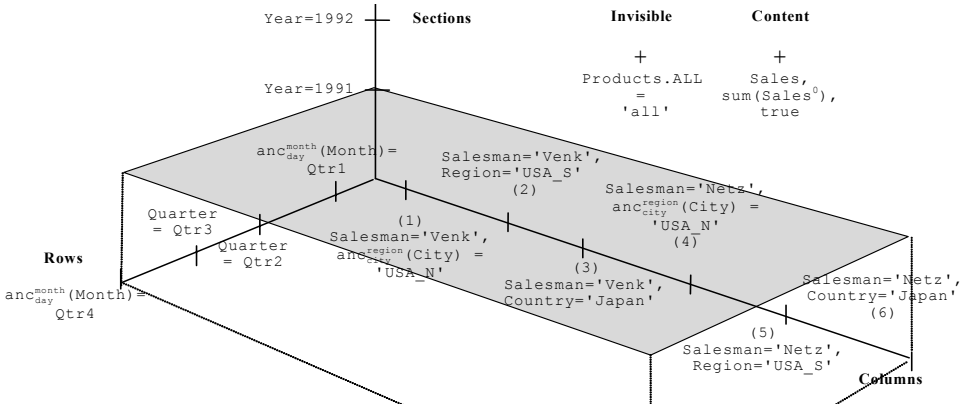
- **Διδιάστατη τομή(2D-slice):** Το CPM προσανατολίζεται σε διδιάστατες απεικονίσεις . Μία διδιάστατη τομή είναι ένα στρώμα του πολυδιάστατου χώρου που μπορεί να απεικονιστεί διδιάστατα. Έστω ένας multicube MC, αποτελούμενος από K άξονες. Μία διδιάστατη τομή *στον MC* είναι ένα σύνολο (K-2) σημείων, το καθένα από ξεχωριστό άξονα. Πρακτικά , η διδιάστατη τομή τοποθετεί τους άξονες του multicube σε συγκεκριμένα σημεία εκτός των δύο αξόνων οι οποίοι και απεικονίζονται στην οθόνη
- **Ταινία(Tape):** Πρακτικά μία ταινία είναι μία στήλη ή μία γραμμή μίας διδιάστατης τομής. Έστω μία διδιάστατη τομή SL πάνω σε ένα multicube MC, αποτελούμενο από K άξονες. Μία ταινία πάνω στην *SL* είναι ένα σύνολο (K-1) σημείων, όπου (K-2) σημεία είναι σημεία της SL. Μία ταινία είναι πάντα παράλληλη σε έναν συγκεκριμένο άξονα: Από τους δύο ελεύθερους άξονες της διδιάστατης τομής , 'καρφιτσώνουμε' τον έναν από αυτούς σε ένα συγκεκριμένο σημείο

1. **Τομή(Cross-join):** Πάλι πρακτικά αν πάρουμε μία παράλληλη στον οριζόντιο άξονα και μία δεύτερη παράλληλη στον κάθετο, η τομή τους είναι ένα κελί, ή στη γενικότερη περίπτωση ένα σύνολο κελιών. Και στις δύο περιπτώσεις θεωρούνται Cross-joins. Οι τομές είναι πολύ σημαντικές για τη σύνδεση λογικού και παρουσιαστικού στρώματος γιατί ουσιαστικά αποτελούν δευτερεύοντες κύβους



Σχήμα 3. 5 -- Αντιστοίχιση CPM οντοτήτων σε 3D και 2D μορφές/απεικονίσεις

Ας έρθουμε όμως στο παράδειγμα του Σχ.3.1 για να δείξουμε την αντιστοιχία με τις CPM οντότητες. Είναι φανερό πως στον οριζόντιο άξονα έχουμε δύο διαστάσεις λόγω του CROSSJOIN. Μας ενδιαφέρει στο σημείο αυτό να καθορίσουμε τί θα δει ο τελικός χρήστης. Όπως προαναφέρθηκε αυτό θα είναι μία διδιάστατη τομή (2D-slice). Σχηματικά αυτό φαίνεται παρακάτω :



Σχήμα 3.6: Η διδιάστατη τομή(2D-Slice) SL για το παράδειγμα του Σχ.3.1

Στην SL υπάρχουν 4 οριζόντιες ταινίες, οι `R1, R2, R3 and R4`, και 6 κάθετες ταινίες. Για τις οριζόντιες το νόημα είναι ξεκάθαρο, μιας και αναπαριστούν τη διάσταση `Quarter`, είτε ως

quarters ή ως months. Για τις κάθετες ταινίες το νόημα είνα κάπως πιο περίπλοκο αφού αναπαριστούν το συνδυασμό των διαστάσεων `Salesman` και `Geography`, όπου η `Geography` εκφράζεται στα επίπεδα `City`, `Region` και `Country`. Η διάσταση `Year` είναι καθορισμένη σε συγκεκριμένο σημείο ενώ η διάσταση `Product` αγνοείται.

Σε αυτό τον πολυδιάστατο χώρο των 5 αξόνων, οι ταινίες `C1` και `R1` για παράδειγμα ορίζονται ως εξής :

C1 = [(Salesman='Venk'∧anc$^{region;city}$(city)='USA_N'),(Year='1991'),
anc$^{ALL;item}$(Products)='all'),(Sales,sum(Sales))]

R1 = [(anc$^{month;day}$(Month)='Qtr1'∧Year='1991'),(Year='1991'),
(anc$^{ALL;item}$(Products)='all'),(Sales,sum(Sales))]

‘Εστω τώρα η τομή cross-join `t1` που την ορίζουμε από τα κοινά κελιά των `R1` και `C1`. Αυτή ορίζεται ως εξής:

t1 =([SalesCube,(Salesman='Venk'∧anc$^{region;city}$(city)='USA_N ∧
anc$^{month;day}$(Month)='Qtr1'∧Year='1991'∧anc$^{ALL;item}$
(Products)='all'),
[Salesman,City,Month,Year,Products.ALL,Sales],sum],[Size(City)],true)

3.2.3.2 Ο Πολυδιάστατος χώρος

Στην ενότητα τούτη θα αναλυθεί ο πολυδιάστατος χώρος του παρουσιαστικού στρώματος του CPM.Ουσιαστικά πρόκειται για δύο οντότητες, τους άξονες και τα σημεία. Ο τυπικός ορισμός τους προϋποθέτει την εισαγωγή κάποιων βοηθητικών οντοτήτων, οι οποίες είναι απαραίτητες για την συνύπαρξη περισσοτέρων από μία διαστάσεων σε έναν άξονα.

Έστω τα ακόλουθα διμελή ανεξάρτητα και άπειρα μετρήσιμα σύνολα : ένα σύνολο σημείων $U_{\mathcal{P}}$, ένα σύνολο αξόνων $U_{\mathcal{A}}$, και ένα σύνολο από multicubes $U_{\mathcal{MC}}$.

Έστω επίσης τα πεπερασμένα σύνολά τους , **P** για τα σημεία, **A** για τους άξονες και **MC** για τους multicubes, κάθε φορά που αντιμετωπίζουμε μία συγκεκριμένη περίπτωση.

Στο σημείο αυτό θα δείξουμε την επέκταση του ορισμού των διαστάσεων .Με τον τρόπο αυτό η έννοια της διάστασης ενσωματώνει κάθε είδος γνωρίσματος (*attribute*), για παράδειγμα αποτελέσματα συναρτήσεων, μέτρα κ.τ.λ. Επίσης, κάθε γνώρισμα που δεν ανήκει σε κάποια διάσταση, ανήκει σε μία ενός-επιπέδου διάσταση(με το ίδιο όνομα με το γνώρισμα), χωρίς κάποια ancestor συνάρτηση ή ιδιότητα ορισμένη σε αυτό. Γίνεται διαχωρισμός των διαστάσεων που αποτελούνται από επίπεδα (*level dimensions*) και αυτών που αποτελούνται από εξαρτώμενα

γνωρίσματα (*attribute dimensions*). Οι διαστάσεις που εμπεριέχουν αριθμητικά μέτρα λέγονται *measure dimensions*.

Μετά από την επέκταση αυτή, η συνύπαρξη πολλών διαστάσεων σε έναν άξονα είναι εφικτή. Όπως προαναφέρθηκε αυτό είναι αναγκαίο για διδιάστατες απεικονίσεις. Για παράδειγμα στο Σχ.3.7 οι λογικές διαστάσεις `Salesman` και `Geography` βρίσκονται στον ίδιο (παρουσιαστικό) άξονα. Αυτό βέβαια πρακτικά σημαίνει την επανάληψη των τιμών της `Geography` για κάθε διαφορετική τομή της `Salesman`. Για το λόγο αυτό η αναπαράσταση τέτοιων δομών απαιτείται ο καθορισμός ομάδων γνωρισμάτων(*attribute groups*), που να συνυπάρχουν στον ίδιο άξονα.

Ένα *attribute group* `AG` είναι ένα ξευγάρι [**A,DA**], όπου **A** είναι μία λίστα γνωρισμάτων (που ονομάζεται κλειδί της ομάδας) και **DA** είναι μία λίστα από attributes *εξαρτώμενα από τα* attributes του **A**. Ο όρος εξαρτώμενα σημαίνει (α) μέτρα εξαρτώμενα από τα αντίστοιχα επίπεδα του συνόλου δεδομένων (β) αποτελέσματα συναρτήσεων εξαρτώμενα από τα ορίσματα της συνάρτησης. Παράδειγμα attribute group :

ag_1=([Sales,Expenses],[Profit]) , ag_2=([City],[Size(City)])

Μία ομάδα διαστάσεων (*dimension group)* `DG` είναι ένα ζευγάρι [**D,DD**], όπου **D** είναι μία λίστα διαστάσεων (που ονομάζεται κλειδί της ομάδας) και **DD** είναι μία λίστα από διαστάσεις *εξαρτώμενες από τις* διαστάσεις του **D** , όμοια ουσιαστικά με τα attribute groups.

Τώρα μπορούμε να δώσουμε τους ακόλουθους τυπικούς ορισμούς:

- Ένα σχήμα άξονα (*axis schema)* είναι ένα ζευγάρι [**DG,AG**], όπου **DG** είναι μία λίστα `K` dimension groups και **AG** είναι μία ταξινομημένη λίστα από `K` πεπερασμένες ταξινομημένες λίστες από attribute groups, όπου τα κλειδιά κάθε (εσωτερικής) λίστας ανήκουν στην ίδια διάσταση, και βρίσκονται στην ίδια θέση στην **DG**, όπου `K>0`. Τα μέλη κάθε ταξινομημένης λίστας δεν είναι αναγκαστικά διαφορετικά. Παριστάνουμε ένα axis schema ως εξής : $AS_K=([DG_1 \times DG_2 \times \ldots \times DG_K],[[ag^{1;1},ag^{2;1},\ldots,ag^{k_1;1}]\times[ag^{1;2},ag^{2;2},\ldots,ag^{k_2;2}]\times\ldots\times[ag^{1;k},ag^{2;k},\ldots,ag^{k_k;k}])\}$.

Στο παράδειγμά μας έχουμε τα εξής δύο axes schemata, με τους ακόλουθους ορισμούς:

Row_S = {[Quarter],[Month,Quarter,Quarter,Month]}
Column_S = {[Salesman×Geography], [Salesman]×[[City,Size(City)], Region, Country]}

- Ένα σημείο (*point)* ανήκει στο σύνολο U_p

2. Ένα σημείο σε ένα *axis schema* AS, είναι ένα σημείο μαζί με ένα σύνολο συνθηκών επιλογής ισότητας , ένα για κάθε attribute group του axis schema

Για παράδειγμα , στο axis schema [Salesman,[City,Size(City)]], ένα σημείο ορίζεται ως :

$\mathbf{p_1}$=([Salesman='Venk',anc$^{region;city}$(City)='USA_N'])

ή αν θέλουμε να ενσωματώσουμε και το axis schema στον ορισμό :

$\mathbf{p_1}$=([Salesman,[City,Size(City)]], [Salesman='Venk',anc$^{region;city}$(City)='USA_N'])

- Ένας άξονας (*axis) πάνω σε ένα axis schema* AS, είναι μία πεπερασμένη λίστα σημείων, όλων ορισμένων στο axis schema AS.

Πρακτικά ένας άξονας προκύπτει από τον περιορισμό ενός axis schema σε συγκεκριμένες τιμές, μέσω της εισαγωγής συγκεκριμένων περιορισμών στις εμφανίσεις κάθε επιπέδου.

a = (AS_K,[$\varphi_1,\varphi_2,...,\varphi_K$]),K≤N ή

a={[$DG_1 \times DG_2 \times \ldots \times DG_K$],[[$ag^{1;1},ag^{2;1},\ldots,ag^{k_1;1}$]×[$ag^{1;2},ag^{2;2},\ldots,ag^{k_2;2}$]×…×[$ag^{1;k},ag^{2;k},\ldots,ag^{k_k;k}$]], [[$\varphi^{1;1},\varphi^{2;1},\ldots,\varphi^{k_1;1}$]×[$\varphi^{1;2},\varphi^{2;2},\ldots,\varphi^{k_2;2}$]×...×[$\varphi^{1;k},\varphi^{2;k},\ldots,\varphi^{k_k;k}$]]}

Ένα σύνολο από dimension groups από κάθε άξονα παριστάνεται ως dim(a).

Στο παράδειγμά μας έχουμε , τους εξής δύο άξονες :

Rows = {Row_S,[anc$^{month;day}$(Month)=Qtr1,Quarter=Qtr2,
Quarter=Qtr3,anc$^{month;day}$(Month)=Qtr4]}
Columns = {Column_S,{[Salesman='Venk',Salesman='Netz'],
[anc$^{region;city}$(City)='USA_N', Region='USA_S', Country='Japan']}

Σχετικά με τα σημεία τα χωρίζουμε σε δύο κατηγορίες τα atomic και hierarchically decomposable σημεία Τα πρώτα ορίζονται πάνω σε ένα μόνο επίπεδο ή τιμή μέτρου ενώ τα δεύτερα πάνω σε σύνολα τιμών.

- *Atomic points* : Χαρακτηρίζονται από το γεγονός πως όλες οι συνθήκες επιλογής ισότητας για τα attribute groups αναφέρονται σε ένα μόνο attribute . Με άλλα λόγια, τα atomic points είναι της μορφής Level=constant ή Measure=constant.
- *Hierarchically decomposable points:* Χαρακτηρίζονται από το γεγονός πως η συνθήκη επιλογής ενός ή περισσοτέρων attribute groups περιέχει μία ancestor συνάρτηση.

Για παράδειγμα ,το p_1 είναι ένα hierarchically decomposable point

p_1=([Salesman,[City,Size(City)]], [Salesman='Venk',anc$^{region;city}$ (City)='USA_N'])

ενώ το p_2 είναι ένα atomic point

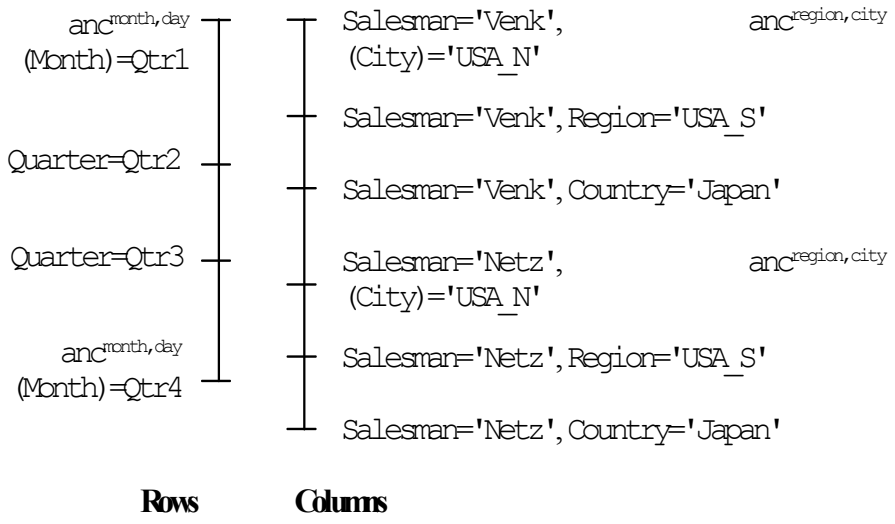
p_2=([Time,[Quarter], [Quarter=Qtr2])

Φυσικά ένα hierarchically decomposable point αντιστοιχεί σε ένα πεπερασμένο σύνολο από atomic points (κάτι που εξηγείται από το πεπερασμένο πεδίο των ancestor συναρτήσεων). Έτσι, το σημείο p_1 αντιστοιχεί στα σημεία $p_{1.1}$ και $p_{1.2}$, ορισμένα στο επίπεδο City .

$p_{1.1}$=([Salesman,[City,Size(City)]], [Salesman='Venk',City='Seattle'])

$p_{1.2}$=([Salesman,[City,Size(City)]], [Salesman='Venk',City='Boston'])

- Ένας άξονας που αποτελείται μόνο από atomic points ονομάζεται *atomic-level* άξονας. Ένας atomic-level άξονας X_a που αποτελείται από atomic points που προέρχονται από την ιεραρχική αποσύνθεση (hierarchical decomposition) των σημείων ενός άξονα X, ονομάζεται *atomic-level ισοδύναμος (equivelant)* του X.
- Ένα *axis tag* είναι ένας χαρακτηρισμός ενός άξονα αναφορικά με (α) τις φυσικές του ιδιότητες (β) το αν μπορεί να απεικονιστεί σε μία διδιάστατη οθόνη ή όχι Με τον τρόπο αυτό ένας άξονας χαρακτηρίζεται ως :
  - Συντεταγμένων(coordinate) ή μέτρου(measure), ανάλογα με το αν παριστάνει τις συντεταγμένες ή τα εσωτερικά σημεία του πολυδιάστατου χώρου
  - Ορατός(visible) ή αόρατος (invisible), ανάλογα με το αν απεικονίζεται στη διδιάστατη οθόνη ή απλά ¨καρφιτσώνουμε¨ τιμές των σημείων του χωρίς αυτός να εμπλέκεται στην απεικόνιση. Στο παράδειγμά μας, υπάρχει ένας αόρατος άξονας, που ¨καρφιτσώνει¨ το Year στο 1991 και το Products στο ALL.



Σχήμα 3.7 -- Τα σημεία των αξόνων Rows και Columns.

3.2.3.3 Τα περιεχόμενα του πολυδιάστατου χώρου

Στην ενότητα τούτη θα αναλυθούν τα περιεχόμενα του πολυδιάστατου χώρου του παρουσιαστικού στρώματος του CPM. Θεμελιώδης οντότητα είναι οι *multicubes* που λίγο πολύ αντιστοιχούν σε δομές n διαστάσεων. Επίσης σημαντικές οντότητες είναι τα *2D-slices*, που αντιστοιχούν σε διδιάστατες απεικονίσιμες δομές , οι *tapes*, που είναι μονοδιάστατες οντότητες και τα *cross-joins* που είναι περιοχές μίας 2D-slice όπου συναντώνται δύο *tapes*.

- Ένα σχήμα πολυκύβου (*multicube schema)* $MC^S=[\mathcal{A}^S]$ είναι ένα πεπερασμένο σύνολο από axis schemata $\mathcal{A}^S$
- Ένας πολυκύβος (*multicube)* $MC=[\mathcal{A}, \mathbf{f}]$, όπου $\mathcal{A}$ είναι ένα πεπερασμένο σύνολο αξόνων και **f** είναι μία συνάρτηση περιεχομένων, που αντιστοιχίζει συντεταγμένες σε μέτρα. Απαιτείται ότι $\mathcal{A} = \mathcal{C} \cup \{\mathbf{M}\}$, όπου $\mathcal{C}$ είναι οι *coordinate* άξονες και **M** είναι ο *measure* άξονας του multicube

Έστω επίσης **M** ένας measure άξονας. Τα σημεία του **M** θα υπολογιστούν μέσω ερωτήσεων στην βάση δεδομένων. Εν τούτοις πρόκειται για έναν κανονικό άξονα με τη διαφορά πως ένα σημείο (π.χ., `Sales = 40`) μπορεί να επαναλαμβάνεται. Ας θυμηθούμε πως οι άξονες είναι πεπερασμένες λίστες σημείων.

Στην απλούστερη περίπτωση, ένα σημείο χαρακτηρίζεται από μία απλή ισότητα της μορφής `[measure=constant]`. Παρόλα αυτά υπάρχει η δυνατότητα συνύπαρξης περισσοτέρων από ένός λογικού μέτρου σε έναν measure άξονα όπου κάθε σημείο του measure άξονα

χαρακτηρίζεται από ένα σύνολο ισοτήτων της μορφής $[measure_1=constant_1,\ldots,measure_k=constant_k]$, ανάλογα με το attribute/dimension group που καθορίζει τον άξονα..

Στην περίπτωση ενός measure άξονα, μπορούμε να τοποθετήσουμε στο σχήμα του άξονα μία συναθροιστική συνάρτηση για κάθε διάσταση του σχήματος.. Επίσης είναι δυνατή η επισύναψη μίας δεύτερης συνθήκης στο σχήμα, η οποία θα ενεργεί ως φίλτρο των δεδομένων που ανακτούνται.

Στο παράδειγμά μας υπάρχει ένας measure άξονας, ο `Content`, αποτελούμενος από 64 σημεία, όσα και τα κελιά τιμών του πίνακα δηλαδή. Ο `Content` ορίζεται με βάση τα atomic-level ισοδύναμα των αξόνων συντεταγμένων. Ο άξονας `Time` αποσυντίθεται ιεραρχικά σε 8 τιμές και ο `Geography×Salesman` άξονας σε επίσης 8 σημεία. Στο measure axis schema τοποθετείται το άθροισμα $Sales=sum(Sales^0)$ και η δευτερεύουσα συνθήκη επιλογής `true`.

Έστω μία συνάρτηση περιεχομένων για το **M**. Πρακτικά η συνάρτηση περιεχομένων αρχικοποιεί τα σημεία του άξονα measure, υπολογίζοντάς τα από ερωτήσεις στο σύνολο δεδομένων. Κάθε σημείο στον άξονα measure τότε εξαρτάται από τα σημεία των atomic-level ισοδυνάμων των αξόνων συντεταγμένων που είναι υπεύθυνοι για την ταυτοποίησή του.

$$contents_M(\mathcal{C}) : \mathcal{C} \rightarrow U_{\mathcal{P}}.$$

Με άλλα λόγια, έστω πως υπάρχουν $K-1$ άξονες στο $\mathcal{C}$, και έστω πως ορίζεται $contents_M(\mathcal{C})$ ως $[\mathbf{A}_1,\ldots,\mathbf{A}_{K-1}]$, τότε για κάθε συνδυασμό σημείων $[p_1,\ldots,p_{K-1}]$ (κάθε σημείο p_i προερχόμενο από άξονα $\mathbf{A}_i$) υπάρχει ένα σημείο μ στο $U_{\mathcal{P}}$., σαν το αποτέλεσμα της $contents_M(\mathcal{C})$ συνάρτησης. Βασισμένοι στο γεγονός πως το $\mathcal{C}$ αποτελείται από έναν πεπερασμένο αριθμό σημείων, η $contents_M(\mathcal{C})$ συνάρτηση, επιστρέφει επίσης έναν πεπερασμένο αριθμό σημείων. Εν τούτοις, περισσότερες από μία συντεταγμένες μπορούν να αντιστοιχούν στην ίδια τιμή μέτρου. Έτσι αποκλείεται η πιθανότητα εύρεσης μίας αντίστροφης συνάρτησης. Για την αντιμετώπιση του μειονεκτήματος αυτού υποθέτουμε αντιστοιχιζόμενες συντεταγμένες $coordinates(\mu)$, έτσι ώστε $coordinates(\mu)=[p_1,\ldots,p_{K-1}]$.

Στο παράδειγμά μας , υπάρχουν τα ακόλουθα axes schemata και άξονες :

Row_S = {[Quarter],[Month,Quarter,Quarter,Month]}

Column_S = {[Salesman×Geography], [Salesman]×[[City,Size(City)], Region, Country]}

Invisible_S = {[Product×Time],[[Product.ALL]×[Year]]}

Content_S = {[Sales],[Sales=sum(Sales^0), true]}

και οι αντίστοιχοι άξονες:

Rows={[anc$^{month;day}$(Month)=Qtr1,Quarter=Qtr2,Quarter=Qtr3, anc$^{month;day}$ (Month)=Qtr4]}

Columns = {[Salesman='Venk',Salesman='Netz'] × [anc$^{region;city}$(City)='USA_N', Region='USA_S', Country='Japan']}

Invisible = {[Year=1991] × [ALL='all']}

Content = {64 points}

Και τότε ο multicube MC θα ορίζεται ως :

MC = {Rows, Columns, Invisible, Content}

Έστω πως θέλουμε να παρουσιάσουμε δεδομένα σε ξεχωριστά φύλλα εργασίας (οθόνες), όπου κάθε φύλλο εργασίας αντιστοιχεί σε έναν χρόνο , π.χ., το πρώτο φύλλο το 1991 και το δεύτερο το 1992. Αυτό μπορεί να γίνει απλά με την προσθήκη ενός άξονα Sections (όπως φαίνεται στο Σχ.3.8). Θα έχουμε τις ακόλουθες αλλαγές:

Axes schemata:

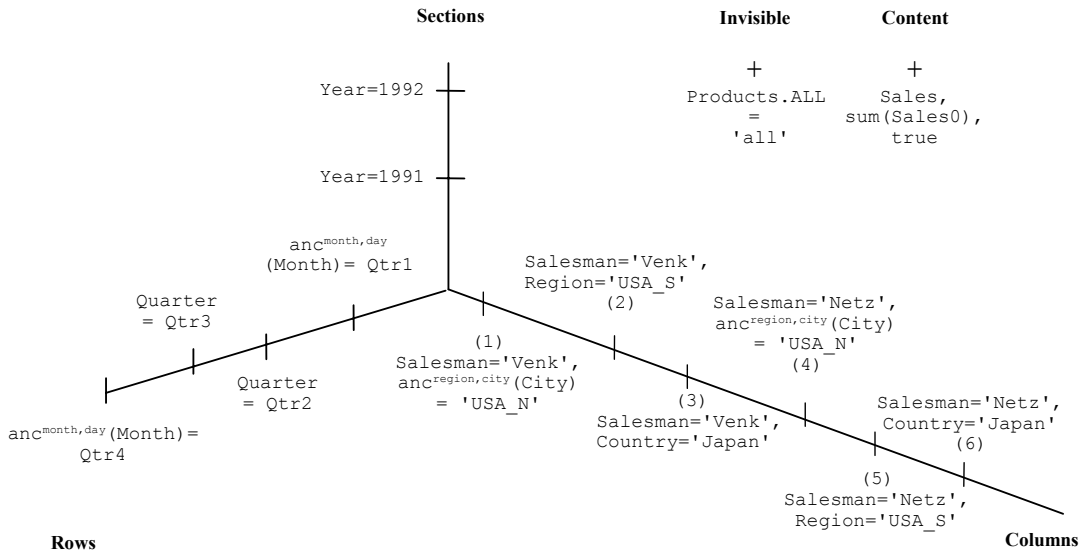
Section_S = {[Time],[Year]}
Invisible_S = {[Product],[Product.ALL]}

axes:

Sections = {Section_S,[Year=1991,Year=1992]}
Invisible = {Invisible_S,[ALL='all']}
Content = {128 points}

Και τότε ο multicube MC θα ορίζεται ως

MC = {Rows, Columns, Sections, Invisible, Content}



Σχήμα 3.8 -- Ο πολυδιάστατος χώρος για το παράδειγμά του Σχ.3.2 , επεκταμένο με Sections

- **Διδιάστατη τομή(2D-slice):** Έστω ένας multicube MC, αποτελούμενος από K άξονες. Μία διδιάστατη τομή *στον MC* είναι ένα σύνολο (K-2) σημείων, το καθένα από ξεχωριστό άξονα. Πρακτικά , η διδιάστατη τομή τοποθετεί τους άξονες του multicube σε συγκεκριμένα σημεία εκτός των δύο αξόνων οι οποίοι και απεικονίζονται στην οθόνη. Στο παράδειγμά μας τα Σχ.3.1 και Σχ.3.7 αναπαριστούν την ίδια 2D slice
- **Ταινία(Tape):** Έστω μία 2D slice διδιάστατη τομή SL πάνω σε ένα multicube MC, αποτελούμενο από K άξονες. Μία ταινία πάνω στην *SL* είναι ένα σύνολο (K-1) σημείων, όπου (K-2) σημεία είναι σημεία της SL. Μία ταινία είναι πάντα παράλληλη σε έναν συγκεκριμένο άξονα: Από τους δύο ελεύθερους άξονες της διδιάστατης τομής, 'καρφιτσώνουμε' τον έναν από αυτούς σε ένα συγκεκριμένο σημείο

Οι ταινίες ουσιαστικά αναπαριστούν μονοδιάστατα μέρη μίας 2D slice. Στην πραγματικότητα, από τους δύο ελεύθερους άξονες μίας 2D slice, μόνο ένας μένει ελεύθερος και ο άλλος 'καρφιτσώνεται' σε ένα συγκεκριμένο σημείο, έστω p. Όμως δεν μπορούν να θεωρηθούν γραμμές λόγω των hierarchically decomposable σημείων. Στην περίπτωση αυτή η απεικόνιση γίνεται με ένα σύνολο γραμμών. Επιπλέον αν μία 2D-slice έχει δύο άξονες a_1, a_2 με μέγεθος $size(a_1)$ και $size(a_2)$ σημείων η κάθεμία , τότε μπορούν να οριστούν $size(a_1)*size(a_2)$ ταινίες σε αυτήν τη 2D-slice.

Στο παράδειγμά μας οι C1, C2, C3, C4, C5, C6 και R1, R2, R3, R4 είναι ταινίες . Οι C1 και R1 όμως περιέχουν περισσότερες από μία γραμμές επειδή έχουν ΄καρφιτσωθεί σε ΄ hierarchically decomposable σημεία.

- **Τομή(Cross-join):** Έστω μία διδιάστατη τομή SL πάνω σε έναν multicube MC, αποτελούμενο από K άξονες και έστω δύο ταινίες t_1 και t_2 όχι παράλληλες στον ίδιο άξονα. Μία τομή στις t_1 *και* t_2 είναι ένα σύνολο K σημείων, όπου τα (K-2) σημεία ανήκουν στην SL και καθένα από τα δύο υπόλοιπα σημεία είναι σημείο σε διαφορετικό άξονα της τομής

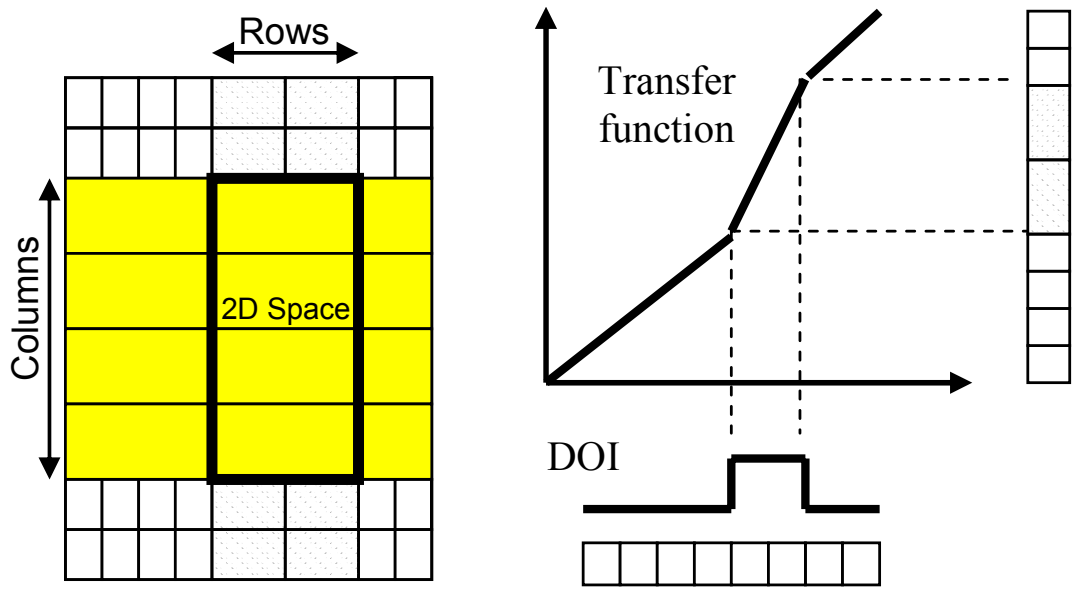
Πρακτικά ένα cross join είναι ένα σύνολο κελιών που προκύπτει από την τομή δύο ταινιών. Αν οι δύο ταινίες έχουν οριστεί πάνω σε atomic σημεία , το cross-join περιέχει ένα μόνο κελί. (π.χ.στην περίπτωση των R2 και C3).διαφορετικά προκύπτει ένα σύνολο κελιών (π.χ.στην περίπτωση των R2 και C1).

Για περισσότερες λεπτομέρειες ο αναγνώστης παραπέμπεται στα (Maniatis et al., 2003, Maniatis et al., 2003a, Maniatis, 2003, Maniatis, 2004).

3.2.4 Table Lens

Στην προηγούμενη ενότητα δείξαμε πως το CPM μπορεί να απεικονίσει αποτελέσματα πολλών ερωτήσεων. Στο σημείο αυτό είναι δυνατόν να εισάγουμε στο μοντέλο αυτό κάποιες τεχνικές οπτικοποίησης. Δεν είναι αναγκαστική η απεικόνιση μέσω κάποιας απλής πινακοειδής μορφής, αν και αυτό θα ήταν το αυτονόητο. Στην ενότητα αυτή θα δείξουμε πως οι δομές του CPM μπορούν να συνδυαστούν με μία μη παραδοσιακή τεχνική από το χώρο του HCI που ονομάζεται Table Lens (TL) , [RC94],[PR96]. Η τεχνική αυτή βασίζεται στην επικέντρωση σε κάποιες περιοχές που παρουσιάζουν περισσότερο ενδιαφέρον, μία λογική που αποδεικνύεται κατάλληλη για OLAP εφαρμογές.

Χρησιμοποιώντας Table Lens, μπορούμε εύκολα να κάνουμε μεγένθυνση(zoom) σε κάποιες περιοχές χωρίς όμως να χάνεται η γενική εικόνα των δεδομένων μας.



Σχήμα 3.9 -- Ένα παράδειγμα από Table Lens: (α) ένα focus παράθυρο 2x4 ορισμένο σε ένα χώρο 8x8 σημείων (β) Table Lens παραμόρφωση του άξονα Rows

3.2.4.1 Αντιστοίχιση του CPM στο Table Lens

Οι βασικές οντότητες της Table Lens τεχνικής είναι οι ακόλουθες :

- **Άξονες(Axes)**: Η Table Lens τεχνική βασίζεται σε δύο άξονες . Θα χρησιμοποιούμε τις ονομασίες Rows και Columns όπως στο Σχ.3.9
- **Διδιάστατος χώρος(2D Space)**: Ο διδιάστατος χώρος προκύπτει από το καρτεσιανό γινόμενο των δύο αξόνων και ουσιαστικά πρόκειται για ένα πεπερασμένο σύνολο κελιών. Η βασική ιδέα είναι πως δε θεωρούνται όλα τα κελιά (παρουσιαστικά) ισοδύναμα. Κάποια κελιά καταλαμβάνουν περισσότερο παρουσιαστικό χώρο, πρακτικά υπάρχει zoom σε κάποιες επιφάνειες της απεικόνισης
- **Συνάρτηση βαθμού ενδιαφέροντος (Degree of Interest Function -DOI)**: Η DOI είναι μία συνάρτηση που αντιστοιχίζει κάθε σημείο ενός άξονα σε μία τιμή που δείχνει το ενδιαφέρον για εκείνο το σημείο. Για κάθε άξονα υπάρχει και μία διαφορετική DOI, έτσι ο διδιάστατος χώρος χαρακτηρίζεται από διδιάστατα παράθυρα με focus. Στην απλούστερη περίπτωση , κάθε συνάρτηση DOI είναι μία απλή "pulse" συνάρτηση , δηλαδή έχει μία σταθερή τιμή εκτός για κάποια συγκεκριμένα σημεία που έχουν κάποια μεγαλύτερη τιμή. Στο Σχ.3.9α , απεικονίζεται ένας 8x8 χώρος με ένα 2x4 focus παράθυρο. Στο Σχ.3.9β, δείχνουμε πως τα αρχικά ισοδύναμα κελιά του Rows άξονα αποκτούν διαφορετική παρουσιαστική βαρύτητα μέσω της συνάρτησης DOI (σε δύο κελιά ή γραμμές έχουμε μεγαλύτερη τιμή)

- **Συνάρτηση μεταφοράς(Transfer Function)**: Η συνάρτηση μεταφοράς τοποθετεί κάθε κελί στις φυσικές του διαστάσεις, δείχνοντας το μέγεθος του zoom για κάθε κελί Πρακτικά η συνάρτηση αυτή είναι η DOI εκφρασμένη σε"pixel" . Στο Σχ.3.9β απεικονίζεται η συνάρτηση αυτή, σε αναλογία βέβαια με τη DOI

Η αντιστοίχιση οντοτήτων CPM και Table Lens είναι σχετικά απλή. Τα σημεία των αξόνων του CPM αντιστοιχίζονται στα σημεία των αξόνων του Table Lens και μία 2D Slice του CPM υλοποιείται ως ένας διδιάστατος χώρος στο Table Lens. Η συνάρτηση περιεχομένων (contents function) παρέχει τις τιμές για τα κελιά του διδιάστατου χώρου. Επειδή το CPM είναι αρκετά γενικό για να συμπεριλάβει την παραμόρφωση που προκαλεί μία συνάρτηση DOI , ο πιο απλός τρόπος είναι να καθορίζει ο χρήστης το παράθυρο με focus

3.2.4.2 Επιλογή παραθύρου με το περισσότερο ενδιαφέρον

Αναφορικά με την επιλογή του focused παραθύρου, υπάρχουν δύο επιλογές :

- Ο χρήστης επιλέγει μόνος του την περιοχή αυτή
- Η διαδικασία να γίνεται πιο αυτοματοποιημένα. Για παράδειγμα είναι δυνατόν να εφαρμόζονται κάποιοι αλγόριθμοι ,των οποίων οι βασικοί παράμετροι θα καθορίζονται από τον χρήστη

Μιας και στην εφαρμογή που υλοποιείται στην παρούσα διπλωματική, ο χρήστης έχει σχετικά περιορισμένες επιλογές καθορισμού του παραθύρου αυτού, ο αναγνώστης παραπέμπεται στο [MVSV03] για περισσότερες λεπτομέρειες σχετικά με τον καθορισμό αλγορίθμων που υλοποιούν τις απαιτήσεις αυτές.

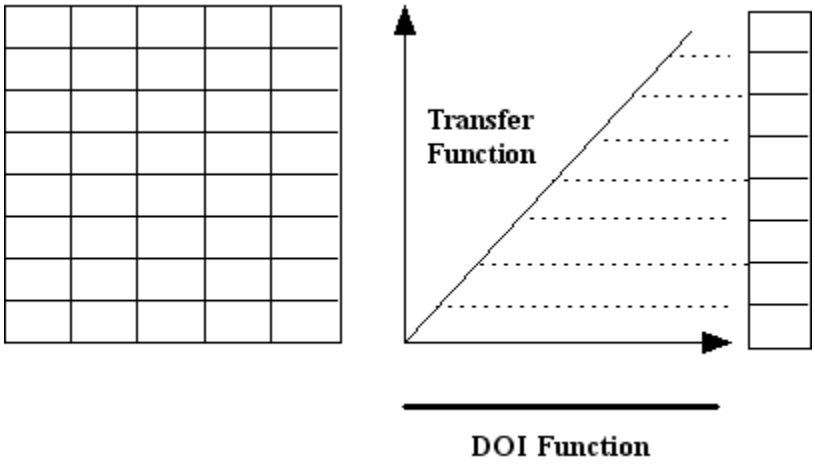
3.2.5 OLAP Lens

Το εργαλείο που υλοποιείται στα πλαίσια της διπλωματικής τούτης ονομάζεται OLAP Lens. Όπως γίνεται εύκολα κατανοητό η ονομασία προέρχεται αφενώς από τις OLAP λειτουργίες που πραγματοποιεί και αφετέρου από τη σχέση του με την τεχνική Table Lens.

Ας επικεντρώσουμε την προσοχή μας στη σχέση με την τεχνική Table Lens. Στην τεχνική αυτή είδαμε πως βασικό ρόλο παίζουν οι περιοχές μεγίστου ενδιαφέροντος. Στον OLAP Lens κυρίαρχο ρόλο παίζει το κελί μεγίστου ενδιαφέροντος.

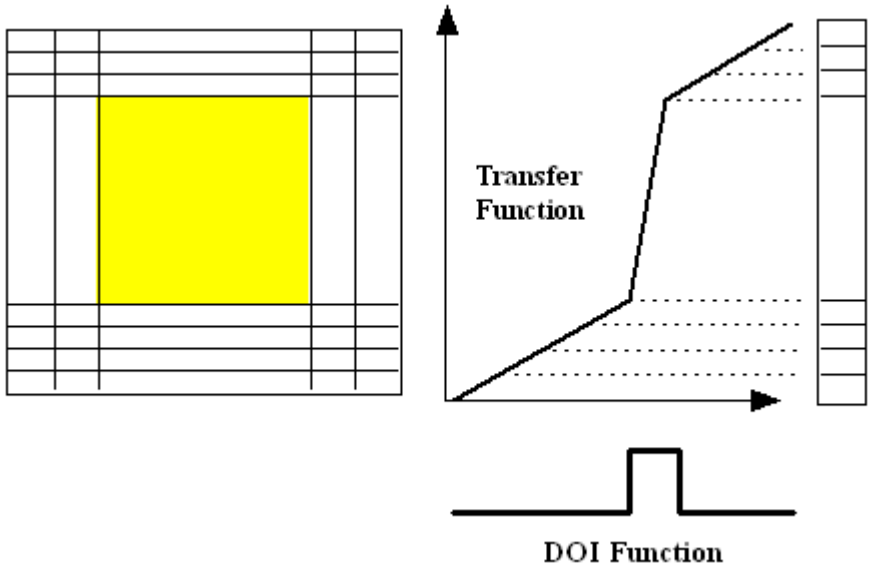
Όπως θα γίνει πιο σαφές στη συνέχεια, κατά την απεικόνιση μίας OLAP οθόνης, ένα κελί θα αποκτά μεγαλύτερη ΄΄παρουσιαστική΄΄ βαρύτητα από τα υπόλοιπα. Αυτό πρακτικά μεταφράζεται στη μεγένθυνση και τον χρωματισμό του. Ταυτόχρονα όλα τα υπόλοιπα κελιά αποκτούν μικρότερη ΄΄παρουσιαστική΄΄ βαρύτητα, πρακτικά σμικρύνονται.Η αντιστοιχία με την τεχνική

Table Lens φαίνεται στα ακόλουθα σχήματα. Αρχικά όλα τα κελιά είναι παρουσιαστικά ισοδύναμα. Αυτό αντικατοπτρίζεται στη συνάρτηση μεταφοράς η οποία είναι μία απλή ευθειά.



Σχήμα 3.10 (α): Συνάρτηση μεταφοράς και συνάρτηση DOI αρχικά

Στη συνέχεια είτε με παρέμβαση του χρήστη, επιλέγοντας ο ίδιος ένα συγκεκριμένο κελί, ή μέσω της χρήσης κάποιων συναρτήσεων(min,max,avg) κάποιο κελί αποκτά μεγαλύτερη παρουσιαστική ισχύ. Η συνάρτηση μεταφοράς τώρα είναι διαφορετική. Επίσης η συνάρτηση DOI αποκτά έναν παλμό που αντιστοιχεί στο μεγενθυμένο κελί.



Σχήμα 3.10 (β): Συνάρτηση μεταφοράς και συνάρτηση DOI μετά τη μεγένθυνση

Η μεγένθυνση εκτός από την αυτονόητη σημασία της για το συγκεκριμένο κελί, λειτουργεί όπως θα δούμε παρακάτω και ως το πρώτο από τα δύο βήματα που απαιτούνται για την πράξη της αναλυτικής καθόδου(drill-down).

Η βασική ιδέα πίσω από τον OLAP Lens κρύβεται σε μία μελλοντική επέκταση σε ασύρματες συσκευές. Σε ένα τέτοιο περιβάλλον το περιορισμένο μέγεθος της οθόνης αποτελέι βασικό σχεδιαστικό παράγοντα. Με τον τρόπο αυτό πολύτιμος παρουσιαστικός χώρος μπορεί να εξοικονομηθεί, έτσι ώστε να βοηθήσει το χρήστη να εκτελέσει κάποιες λειτουργίες πολύ πιο γρήγορα και αποτελεσματικά.

3.3 Απαιτήσεις χρήστη

Στην παράγραφο αυτή, και έπειτα από το λεπτομερές θεωρητικό υπόβαθρο που παρουσιάστηκε στις προηγούμενες παραγράφους, θα καταγραφούν οι απαιτήσεις χρήστη για την εργαλείο που θέλουμε να υλοποιήσουμε και το οποίο ονομάζεται OLAP Lens.

- **Ανεξαρτησία από την πηγή δεδομένων** : Το εργαλείο συνδέεται με κάποια πηγή δεδομένων, δηλαδή έναν OLAP server (η πιο συνηθισμένη είναι ο Microsoft® SQL Server™ 2000 Analysis Services). Η εφαρμογή είναι ανεξάρτητη από την επιλογή του κύβου. Ο χρήστης επιλέγοντας το κατάλληλο κουμπί(button) από το toolbar έχει τη δυνατότητα να διαλέξει με τη βοήθεια κατάλληλης φόρμας αρχικά τη βάση(τον κατάλογο) του server με τον οποίο θέλει να γίνει σύνδεση. Αμέσως εμφανίζονται οι κύβοι που υπάρχουν σε αυτή τη βάση και ο χρήστης επιλέγοντας έναν από αυτούς τους κύβους με τον οποίο θέλει να εργαστεί, πραγματοποιεί την σύνδεση καθώς και την αρχικοποίηση κάποιων δομων που σχετίζονται με το σχήμα του κύβου. Η εναλλάγή κύβων ή βάσεων γίνεται απλά με τη χρήση της ίδιας επιλογής η οποία αυτόματα επαναφέρει τις ρυθμίσεις στην αρχική τους μορφή
- **OLAP navigation :** Φιλικό προς τον χρήστη interface. Είναι απαραίτητο ο χρήστης να μπορεί με ευκολία να προσθέτει/αφαιρεί στο σχήμα τις διάφορες διαστάσεις, τόσο αυτές που απεικονίζονται στην οθόνη όσο και αυτές που λειτουργούν ως φίλτρα . Για το σκοπό αυτό πάλι μεσω κατάλληλου κουμπιού από το toolbar εμφανίζεται μία φόρμα η οποία περιέχει σε δενδρική μορφή όλες τις διαστάσεις του κύβου καθώς και την ιεραρχία που διέπει αυτές. Επίσης στην φόρμα εμφανίζονται δύο πίνακες που αντιστοιχούν στους δύο παρουσιαστικούς άξονες Rows κια Columns. Με τη βοήθεια ειδικών κουμπιών η πρόσθεση και αφαίρεση διαστάσεων στο σχήμα γίνεται απλά και γρήγορα. Ο χρήστης μπορεί να επιλέξει οποιοδήποτε επίπεδο κάποιας διάστασης εκτός από το τελευταίο, μιας και στην απεικόνιση των τελευταίων επιπέδων ακολουθείται μία διαφορετική λογική όπως θα δούμε στη συνέχεια. Ακριβώς όμοια γίνεται η διαδικασία φιλτραρίσματος του αποτελέσματος,η προσθαφαίρεση δηλαδή

διαστάσεων στον άξονα Sections του παρουσιαστικού μοντέλου. Μετά την επιλογή των διαστάσεων ο χρήστης πάλι μέσω κουμπιού από το toolbar μπορεί να ξεκινήσει την απεικόνιση

- **OLAP πράξεις :** Οι βασικές πράξεις γίνονται σύμφωνα με την τεχνική Table Lens που παρουσιάστηκε στην προηγούμενη ενότητα. Πιο συγκεκριμένα θα γίνονται τα εξής :
    - **Αναλυτική Κάθοδος (Drill down*)***: Κρατώντας τις ίδιες διαστάσεις θέλουμε μετάβαση σε πιο λεπτομερές(κατώτερο) επίπεδο. Όταν ο χρήστης κάνει left click πάνω σε ένα κελί τότε αυτό αυτόματα μεγενθύνεται και χρωματίζεται, δηλαδή κάνουμε focus στο κελί αυτό. Ταυτόχρονα τα υπόλοιπα κελιά θα σμικρύνονται. Με το δεύτερο left click πάνω στο κελί που είναι focused (ή εναλλακτικά με τη χρήση κουμπιού για το σκοπό αυτό) θα γίνεται και το drill-down. Η αναλυτική κάθοδος θα γίνεται και στους δύο παρουσιαστικούς άξονες αν αυτό είναι δυνατό. Αν σε κάποιον άξονα δεν είναι δυνατή ή αναλυτική κάθοδος (διότι έχουμε φτάσει στο κατώτερα ορισμένο επίπεδο), τότε το drill γίνεται μόνο στον ένα άξονα. Επίσης στην περίπτωση που ο χρήστης έχει επιλέξει περισσότερες από μια διαστάσεις στον ίδιο άξονα τότε το drill γίνεται στην τελευταία διάσταση
    - **Αναλυτική Άνοδος (Drill up):** Κρατώντας τις ίδιες διαστάσεις θέλουμε μετάβαση σε πιο γενικό (συναθροιστικό) επίπεδο.Για το σκοπό αυτό ο χρήστης έχει πάλι δύο επιλογές. Η αναλυτική άνοδος γίνεται είτε με right click σε οποιοδήποτε σημείο της απεικόνισης είτε με χρήση κατάλληλου κουμπιού από το toolbar. Φυσικά όταν φτάσουμε στο ανώτερο επίπεδο (όχι στο σχήμα του κύβου αλλά στο σχήμα που δημιουργήσαμε με τις επιλογές μας στα επίπεδα των αρχικών διαστάσεων) δεν είναι δυνατή η λειτουργία αυτή
    - **Αναλυτική μετάβαση (Drill within*)*:** Η λειτουργία αυτή αναφέρεται στην επιλογή διαφορετικού μέλους στο ίδιο επίπεδο. Αναφορικά με τους απεικονιζόμενους άξονες η λογική είναι απλή: Ως προς το κατώτερα απεικονιζόμενο επίπεδο όλα τα μέλη βρίσκονται στην απεικόνιση οπότε δε δημιουργείται πρόβλημα. Στην περίπτωση που ο χρήστης θέλει να κάνει drill-within σε κάποιο ανώτερο επίπεδο, θα πρέπει να κάνει πρώτα drill-up να επιλέξει το μέλος που επιθυμεί και να συνέχισει έτσι την ανάλυσή του. Αναφορικά με τον άξονα Sections , και για τον οποίο είναι πιο σημαντική η λειτουργία αυτή, υπάρχει ειδικό combobox το οποίο θα έχει ως τιμή

απεικόνισης το παρών φίλτρο. Επιλέγοντας κάποια άλλη τιμή στο combobox αυτόματα ανανεώνονται οι απεικονιζόμενες τιμές

- **Περιστροφή (Pivot):** Αναδιάταξη της 2D προβολής(2D slice) στην οθόνη. Η λειτουργία αυτή είναι ιδιαίτερα χρήσιμη και συμβάλλει συχνά στο τελικό αισθητικό αποτέλεσμα. Πραγματοποιείται απλά και γρήγορα με χρήση κατάλληλου κουμπιού από το toolbar. Η αναδιάταξη δεν προκαλεί καμία αλλαγή στα δεδομένα της εφαρμογής (δηλαδή στην υλοποιούμενη CPM δομή) αλλά μόνο στον τρόπο που η οθόνη 'χειρίζεται' τους δύο άξονες του μοντέλου. Με άλλα λόγια η λειτουργία αυτή γίνεται άμεσα χωρίς ο χρήστης να περιμένει μία χρονοβόρα αναδιάταξη της υλοποιούμενης δομής. Η αναδιάταξη βέβαια είναι μόνιμη και δεν ισχύει μόνο για την απεικόνιση του συγκεκριμένου στιγμυοτύπου. Η διάταξη των αξόνων παραμένει και μετά μία ενδεχόμενη αναλυτική άνοδο ή κάθοδο

- **OLAP patterns** : Συχνά ο όγκος των δεδομένων που απεικονίζονται είναι μεγάλος. Χρήσιμη είναι η δυνατότητα γρήγορης και άμεσης αναφοράς σε κάποια συγκεκριμένα κελιά (Table Lens). Θα είναι διαθέσιμες κάποιες λειτουργίες/συναρτήσεις πάνω στα δεδομένα της οθόνης:
    - Find Maximum:Χρησιμοποιώντας την επιλογή αυτή θα γίνεται αυτόματα focus στο κελί με τη μέγιστη τιμή
    - Find Minimum: Χρησιμοποιώντας την επιλογή αυτή θα γίνεται αυτόματα focus στο κελί με την ελάχιστη τιμή
    - Find Average: Χρησιμοποιώντας την επιλογή αυτή θα γίνεται αυτόματα focus στο κελί με την πλησιέστερη στον μέσο όρο τιμή
- **Επιλογές αρχικοποίησης :** Πολύ σημαντικό στοιχείο σε μία OLAP εφαρμογή παίζει η απόδοση της. Επειδή το εργαλειο συνδέεται με οποιαδήποτε πηγή δεδομένων, είναι αναγκαίο να εξασφαλίσουμε οτι η απόδοσή της θα βρίσκεται σε ικανοποιητικά επίπεδα. Στο συγκεκριμένο εργαλείο η απόδοση σχετίζεται αποκλειστικά με τον όγκο των δεδομένων που επιλέγει ο χρήστης να ανακτήσει (και να τοποθετήσει με τον τρόπο αυτό στην υλοποιούμενη CPM δομή) κατά την αρχικοποίηση. Για το λόγο αυτό μέσω ειδικής φόρμας ο χρήστης έχει τη δυνατότητα να :
    - επιλέξει τον αριθμό των επιπέδων που ανακτούνται κατά την αρχικοποίηση
    - αν κατά τη διάρκεια της ανάλυσης επιθυμεί να είναι διαθέσιμη η λειτουργία drill-within. Αυτό είναι βαρύνουσας σημασίας μιας και με τον τρόπο αυτό μπορεί να αποφευχθεί η ενσωμάτωση μίας ακόμη διάστασης στην

υλοποιούμενη δομή, που όπως είναι γνωστό αυξάνει εκθετικά τον χρόνο επεξεργασίας

- **Δυνατότητα εκκίνησης νέας ανάλυσης** : Χρησιμοποιώντας την επιλογή αυτή ο χρήστης μπορεί να ξεκινήσει μία νέα ανάλυση στον ίδιο κύβο χωρίς να είναι απαραίτητη η αρχικοποίηση του υλοποιούμενου σχήματος
- **Παρουσιαστικό μοντέλο** : Αν και η αυτόνομη παρουσία του δεν είναι άμεσα αντιληπτή από το χρήστη, αξίζει να υπογραμμίσουμε πως το εργαλείο χρησιμοποιεί ένα αυτόνομο και διακριτό μοντέλο, το CPM.

Όπως προαναφέρθηκε, μελοντικό σκοπό αποτελεί η επέκταση του εργαλείου αυτού σε ασύρματες συσκευές (κινητά τηλέφωνα, Smartphones, PDAs, Pocket PCs). Αυτός είναι άλλωστε ο κυριότερος λόγος για τον οποίο προτιμήθηκε αυτή η τεχνική οπτικοποίησης που βασίζεται στο focus συγκεκριμένων κελιών.

Βέβαια για το σχεδιασμό ενός αντίστοιχου εργαλείου ανακύπτουν διάφορα νέα θέματα που αξίζουν προσοχή. Στο σημείο αυτό θα παρουσιάσουμε συνοπτικά κάποια από αυτά :

- Το πολύ μικρό μέγεθος της οθόνης των συσκευών αυτών είναι ο σημαντικότερος παράγοντας διαφοροποίησης της προσέγγισης. Η εφαρμογή θα πρέπει να προσαρμόζει τον αλγόριθμο μεγένθυνσης και σμίκρυνσης των κελιών ανάλογα με το μέγεθος της οθόνης της συσκευής. Ένα επιπλέον θέμα είναι πως η ανάλυση των οθονών αυτών δεν μπορεί να συγκριθεί με αυτήν μιας τυπικής οθόνης ενός υπολογιστή. Επίσης περιορισμένες είναι οι λειτουργικές δυνατότητες του χρήστη πάνω στην οθόνη αυτή .Και αν στην περίπτωση της απεικόνισης ενός κειμένου υπάρχει η δυνατότητα ο χρήστης να το δει σε περισσότερες από μία φάσεις (με κάποιου είδος navigation), στην απεικόνιση διαγραμμάτων για παράδειγμα κάτι τέτοιο δεν είνα επιθυμητό
- Ο εγγενής περιορισμένος χώρος μνήμης. Αν και οι ασύρματες συσκευές αποκτούν ολοένα και περισσότερες δυνατότητες, εντούτοις δεν μπορούν προς το παρόν να συναγωνιστούν desktop συστήματα, ειδικά σε θέματα μνήμης. Όπως φάνηκε από την ανάλυση της σχεδιάσης του OLAP Lens, εφόσον η υλοποιούμενη δομή στήνεται στο χώρο της δυναμικής μνήμης, θα πρέπει να επανεξεταστούν κάποια θέματα διαχείρισης μνήμης
- Η επικοινωνία με τον OLAP server γίνεται αναγκαστικά με τη χρήση καποιου ασύρματου δικτύου. Το περιορισμένο εύρος ζώνης των δικτύων αυτών είναι ένας βασικός περιορισμός. Ένα επιπλέον ζήτημα είναι η offline επεξεργασία . Θα πρέπει

να ληφθεί υπόψη η περίπτωση που για κάποιο λόγο διακόπτεται η σύνδεση με το δίκτυο και οι συνέπειες της αναξιοπιστίας αυτής στο σχεδιασμό της εφαρμογής

4 *Υλοποίηση*

Στο παρόν κεφάλαιο, στην παράγραφο 4.1 περιγράφεται η αρχιτεκτονική της εφαρμογής που υλοποιήθηκε, στην παράγραφο 4.2 γίνεται μία περιληπτική παρουσίαση των εργαλείων που χρησιμοποιήθηκαν στη φάση της υλοποίησης ενώ στην παράγραφο 4.3 περιγράφονται οι τεχνικές υλοποίησης. Στην παράγραφο 4.4 γίνεται μία οπτική παρουσίαση του τελικού αποτελέσματος. Στην παράγραφο 4.5 γίνεται ο έλεγχος της ορθότητας της εφαρμογής.Στην τελευταία παράγραφο του κεφαλαίου παρατίθενται κάποιες πειραματικές μετρήσεις.

4.1 Αρχιτεκτονική

4.1.1 Υλοποιούμενη Αρχιτεκτονική

Έχοντας περιγράψει στο προηγούμενο κεφάλαιο το CPM, το μοντέλο παρουσίασης που θα χρησιμοποιηθεί, καθώς και τις απαιτήσεις χρήστη, είμαστε έτοιμοι να περιγράψουμε την αρχιτεκτονική του OLAP Lens. Σύμφωνα με την αρχιτεκτονική αυτή θα προκύψουν οι ανάγκες για τα εργαλεία που θα πρέπει να χρησιμοποιηθούν κατά τη φάση της υλοποίησης, τα οποία και θα παρουσιαστούν στην επόμενη παράγραφο.

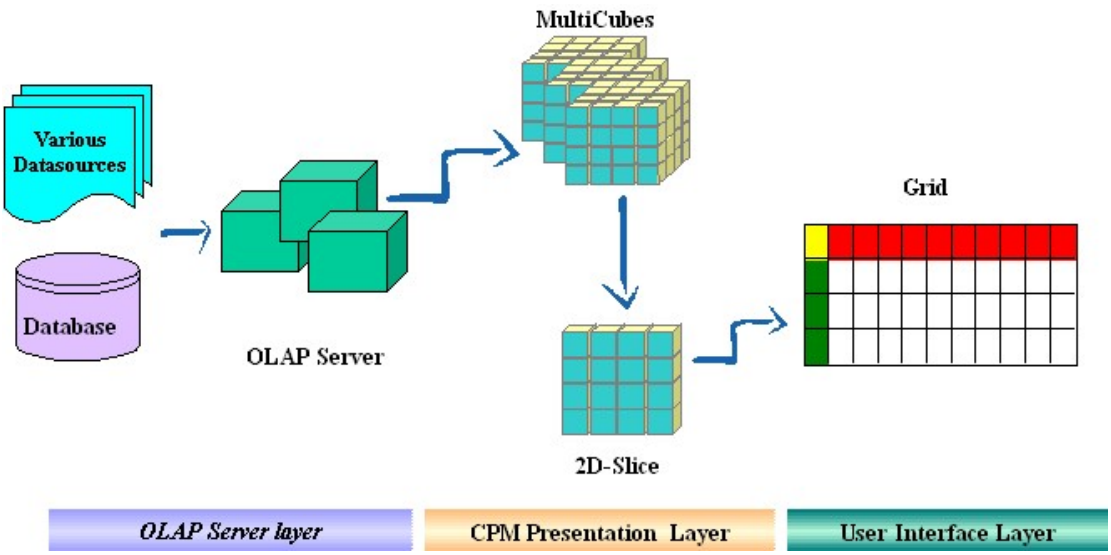
Το πρώτο θέμα που ανακύπτει είναι η πηγή δεδομένων. Όπως προαναφέρθηκε, επειδή σκοπός της παρούσας διπλωματικής είναι κυρίως θέματα οπτικοποίησης, δε θα χρησιμοποιηθεί το λογικό στρώμα του CPM για τη δημιουργία των κύβων. Αντιθέτως θα χρησιμοποιηθεί ένας OLAP server για τη δημιουργία των κύβων καθώς και για την

επικοινωνία με την εφαρμογή. Στο σημείο αυτό αξίζει να υπενθυμιστεί πως απαιτείται η λειτουργία της εφαρμογής με οποιοδήποτε κύβο του συγκεκριμένου server.

Μετά την ανάκτηση των δεδομένων από τον OLAP server θε πρέπει αυτά να αποθηκεύονται δυναμικά στη μνήμη(RAM) υλοποιώντας μία CPM δομή. Πιο συγκεκριμένα θα υλοποιείται το παρουσιαστικό στρώμα του CPM. Όπως είδαμε κατά την παρουσίαση του παρουσιαστικού στρώματος του CPM, βασική οντότητα αποτελεί ο πολυκύβος(multicube). Στην πραγματικότητα θα δημιουργήσουμε πολυκύβους οι οποίοι θα συνδέονται μεταξύ τους, με έναν τρόποπου θα δούμε στη συνέχεια.

Μετά τη δημιουργία και οργάνωση της CPM δομής τα δεδομένα θα οπτικοποιούνται μέσω μίας πινακοειδούς απεικόνισης. Η απεικόνιση αυτή θα είναι βέβαια το τελικό στάδιο, δηλαδή μέρος του user interface. Μέσω του user interface ο χρήστης θα μπορεί να επέμβει στην υλοποιούμενη CPM δομή καθώς επίσης και να ζητήσει την ανάκτηση επιπρόσθετων δεδομένων από τον OLAP server.

Στο παρακάτω σχήμα απεικονίζεται μία απλοποιημένη αρχιτεκτονική του εργαλείου μας. Περισσότερες λεπτομέρειες θα δοθούν στη συνέχεια.



Σχήμα 4.1 -- Απλοποιημένη Αρχιτεκτονική του OLAP Lens

Παρατηρούμε την ύπαρξη τριών ουσιαστικά φάσεων (στρωμάτων) από όπου προκύπτουν ανάλογες απαιτήσεις εργαλείων για την υλοποίηση :

- **Στρώμα OLAP server**: Περιλαμβάνει τη δημιουργία και τη διαχείριση των πολυδιάστατων δομών (κύβων). Το εργαλείο συνδέεται με τον server. Επομένως

χρειάζόμαστε έναν server, και ένα μέσο με τον οποίο η εφαρμογή θα επικοινωνεί με αυτόν

- **Στρώμα CPM** : Το μοντέλο θα υλοποιείται στο χώρο της δυναμικής μνήμης. Το περιβάλλον θα μας το παρέχει βέβαια η προγραμματιστική πλατφόρμα που θα επιλεγεί. Η ανάλυση γίνεται σε επόμενη παράγραφο
- **Στρώμα διαπροσωπείας χρήστη** : Πρόκειται για το πιο σημαντικό στρώμα της εφαρμογής. Πρόκειται όχι μόνο για την πινακοειδή απεικόνιση των δεδομένων μας αλλά και για ολόκληρο το user interface μέσω του οποίου ο χρήστης θα ελέγχει την εφαρμογή. Οι απαιτήσεις χρήστη που έχουν καθοριστεί απαιτούν προσεκτική επιλογή των εργαλείων που θα υλοποιήσουν την απεικόνιση

4.2 Εργαλεία

Στην ενότητα τούτη θα γίνει μία σύντομη περιγραφή των εργαλείων που χρησιμοποιήθηκαν για την υλοποίηση του OLAP Lens. Αξίζει να τονιστεί πως η επιλογή των εργαλείων δεν ήταν από την αρχή της σχεδίασης ξεκάθαρη, αλλά αντιθέτως αποτέλεσε ένα βασικό κομμάτι αυτής. Στα πλαίσια της παρούσας διπλωματικής έγινε ξεκάθαρο πόσο σημαντική είναι η αναζήτηση και επιλογή των κατάλληλων εργαλείων, πριν το ξεκίνημα του υπολοίπου σχεδιασμού.

4.2.1 Έρευνα και δοκιμή εργαλείων

Στην υποπαράγραφο τούτη θα γίνει μία σύντομη αναφορά στον τρόπο με τον οποίο επιλέχθηκαν τα διάφορα εργαλεία που χρησιμοποιήθηκαν κατά τη διάρκεια της υλοποίησης. Όπως προαναφέρθηκε η διαδικασία αυτή αποδείχτηκε επίπονη και χρονοβόρα.

Καθώς ο κύριος στόχος της διπλωματικής είναι οι τεχνικές οπτικοποίησης, και πιο συγκεκριμένα η υλοποίηση της τεχνικής που προβλέπει μεγένθυνση συγκεκριμένων περιοχών οι οποίες ενδιαφέρουν περισσότερο τον τελικό χρήστη (όπως περιγράφηκε στο προηγούμενο κεφάλαιο), αρχικά προτιμήσαμε να δοκιμάσουμε και να δουλέψουμε με κάποια υπάρχοντα opensource εργαλεία. Με τον τρόπο αυτό θα εξοικονομούσαμε χρόνο ο οποίος θα καταναλωνόταν στο κομμάτι της ανάκτησης των δεδομένων από την πηγή, έτσι ώστε να δοθεί αρχικά έμφαση στο μέρος της οπτικοποίησης και δευτερευόντως στην επέκταση σε ασύρματες συσκευές. Η προσπάθεια αυτή δεν στέφθηκε με επιτυχία, αξίζει όμως μία σύντομη αναφορά.

Στο πλαίσιο αυτό έγινε μία σχετική έρευνα των διαθέσιμων επιλογών. Καταλήξαμε στο εργαλείο JPivot. Προκειται ουσιαστικά για μία JSP tag βιβλιοθήκη, η οποία οπτικοποιεί έναν OLAP πίνακα και επιτρέπει στο χρήστη να εκτελέσει τις βασικότερες OLAP πράξεις όπως

είναι η αναλυτική κάθοδος ή άνοδος. Το JPivot είναι σχεδιασμένο έτσι ώστε να μπορεί θεωρητικά να λειτουργήσει με αρκετούς OLAP servers. Πρακτικά έχει σχεδιαστεί ώστε να επικοινωνεί με έναν συγκεκριμένο server τον Mondrian.

Ο Mondrian είναι ένας OLAP server γραμμένος σε Java. Εκτελεί ερωτήσεις γραμμένες στη γλώσσα MDX, διαβάζοντας δεδομένα από μία σχεσιακή βάση. Άρα πρόκειται για μία ROLAP προσέγγιση. Επιπλέον παρέχει ένα Java API για την παρουσίαση των δεδομένων σε πολυδιάστατη μορφή.

Στα πλαίσια αυτά ο Mondrian ουσιαστικά θα υλοποιούσε το λογικό στρώμα του CPM χωρίς εμείς να τροποποιήσουμε οτιδήποτε σε αυτόν. Η προσπάθεια θα επικεντρωνόταν στο JPivot και πιο συγκεκριμένα στον τρόπο με τον οποίο αυτό παρουσίαζε τελικά τα δεδομένα που έπερνε από τον Mondrian στην οθόνη.

Η προσπάθεια αυτή ήταν τελικά ανεπιτυχής για τους ακόλουθους λόγους: Πρώτον η τεκμηρίωση (documentation) του κώδικα του JPivot είναι ελλιπής καθιστώντας την κατανόησή του αρκετά δύσκολη. Επιπλέον ο τρόπος με τον οποίο τα δεδομένα παρουσιάζονται στην οθόνη δεν είναι μόνο πολύπλοκος αλλά όπως αποδείχτηκε ακατάλληλος για την τεχνική που θέλουμε να υλοποιήσουμε. Πιο συγκεκριμένα το JPivot οπτικοποιεί χρησιμοποιώντας XSLT πάνω σε XML αρχεία. Απόσο γνωρίζουμε η μεγένθυνση κελιών ενός πίνακα που προκύπτει από μία XSLT απεικόνιση δεν είναι δυνατή. Ακόμα πιο δύσκολη ήταν η υλοποίηση του παρουσιαστικού στρώματος του CPM. Εκτός των προηγουμένων η προσέγγιση αυτή είναι μάλλον περιορισμένης χρηστικότητας μιας και είναι άρρηκτα συνδεδεμένη με τον Mondrian. Η επέκταση σε κάποιον διαφορετικό OLAP server είναι επίπονη και θα απαιτούσε πολλές αλλαγές. Επιπλέον οι δυνατότητες επέμβασης στο user interface είναι επίσης περιορισμένες όταν αναφερόμαστε σε μία web application όπως αυτή.

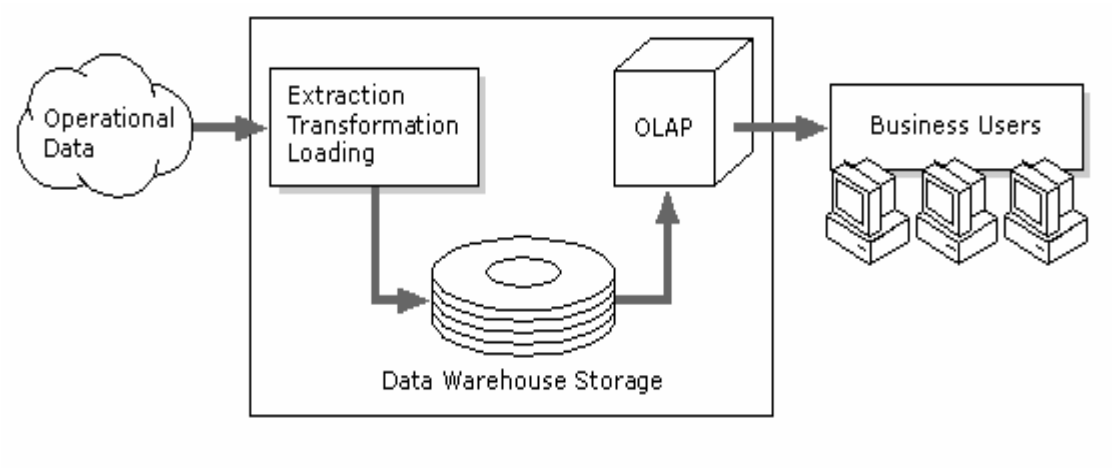
Με άλλα λόγια η χρήση του JPivot μάλλον θα επιβράδυνε παρά θα επιτάχυνε την εργασία μας. Για το λόγο αυτό ακολουθήθηκε μία προσέγγιση που ουσιαστικά ξεκίνησε από το μηδέν, χωρίς να χρησιμοποιηθεί κάτι έτοιμο. Με τον τρόπο αυτό επιτύχαμε να κάνουμε το εργαλείο μας αφενώς επεκτάσιμο, μιας και όπως θα δειχθεί στη συνέχεια η χρησιμοποίηση ενός εναλλακτικού OLAP server είναι σχετικά απλή, και αφετέρου αποτελεί μία καλή αφετηρία για επέκταση σε ασυρματες συσκευές.

Στις αμέσως επόμενες παραγράφους ακολουθεί μία σύντομη περιγραφή των εργαλείων που τελικά χρησιμοποιήθηκαν.

4.2.2 Analysis Sevices

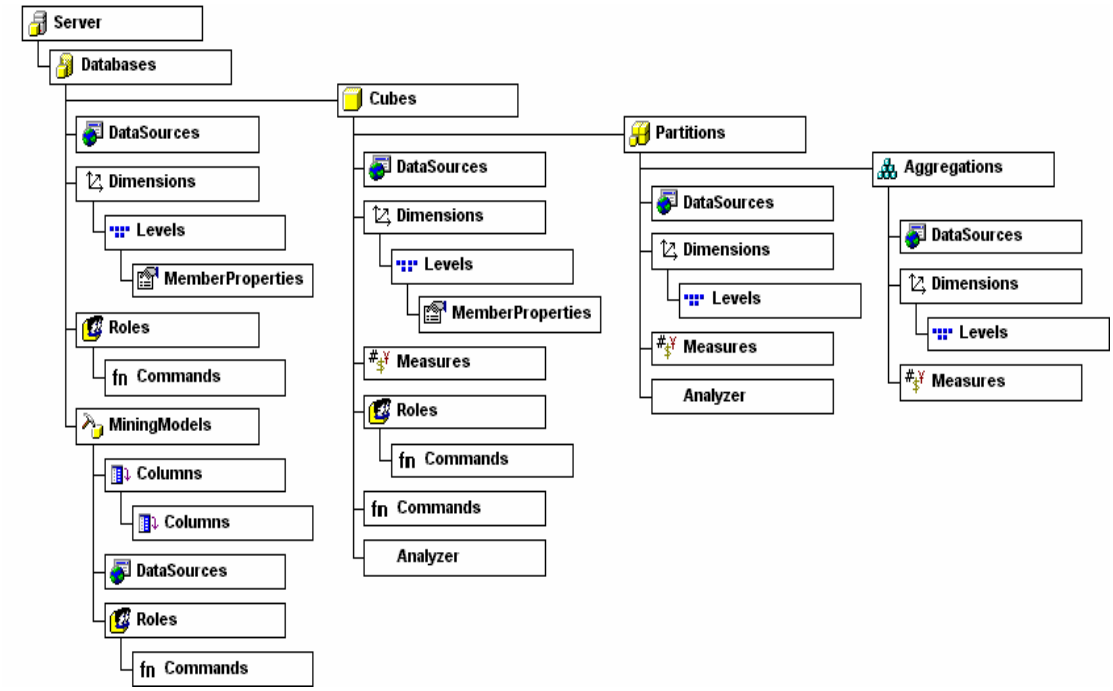
Όπως προαναφέρθηκε σκοπός της παρούσας διπλωματικής είναι κυρίως η οπτικοποίηση των δεδομένων. Για το λόγο αυτό προτιμήθηκε η επιλογή ενός εμπορικού εργαλείου για τη δημιουργία των κύβων , παρά η υλοποίηση του λογικού στρώματος του CPM.

Ο Microsoft® SQL Server™ 2000 Analysis Services παρέχει γρήγορη και αναλυτική πρόσβαση σε δεδομένα αποθηκών δεδομένων. Για το σκοπό αυτό ο Analysis Services δημιουργεί πολυδιάστατους κύβους από τα fact δεδομένα της αποθήκης και τους πίνακες διαστάσεων. Κατά τη δημιουργία των κύβων οι αριθμητικές τιμές συναθροίζονται. Οι κύβοι αποθηκεύονται σε πολυδιάστατες δομές, σχεδιασμένες με τέτοιον τρόπο ώστε να έχουν γρήγορη απόκριση στις ερωτήσεις, συνδυάζοντας τις συναθροισμένες τιμές αλλά και απλές fact τιμές. Με τον τρόπο αυτό μπορούν να απαντήσουν σε μία ποικιλία ερωτήσεων.



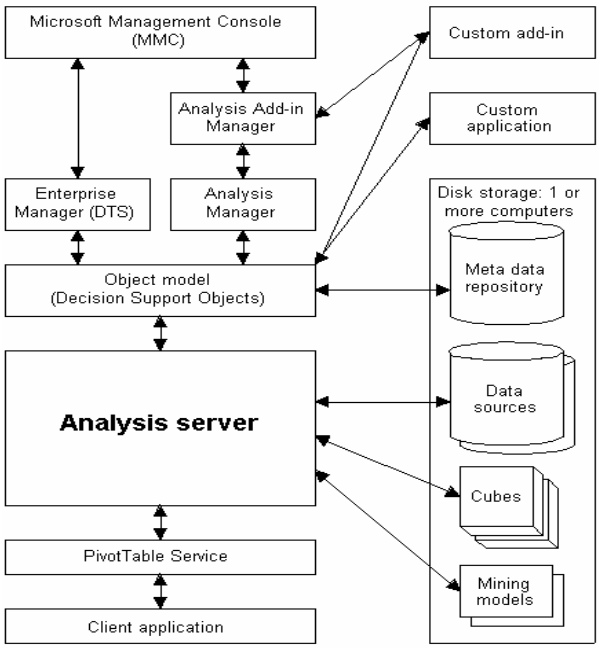
Σχήμα 4.2 --Ένα τυπικό OLAP σύστημα

Οι Analysis Services παρέχουν μία ποικιλία αντικειμένων (objects) για την υλοποίηση OLAP ή data mining εφαρμογών σε μία αποθήκη δεδομένων. Το κύριο αντικείμενο για OLAP είναι βέβαια ο κύβος (cube), ο οποίος περιέχει την αναλυτική πληροφορία που ενδιαφέρει τον τελικό χρήστη. Τα αντικείμενα που χρησιμοποιούνται στην υποστήριξη OLAP και data mining εφαρμογών αναπαριστούνται σε μία ιεραρχία η οποία χρησιμοποιείται για τη διατήρηση πολύπλοκων σχέσεων μεταξύ διαφόρων αντικειμένων όπως κύβοι, διαστάσεις κτλ. Το ακόλουθο διάγραμμα δείχνει τις θέσεις όλων των αντικειμένων στην ιεραρχία αυτή :



Σχήμα 4.3 -- Η ιεραρχία αντικειμένων στις Analysis Services

Μία πηγή δεδομένων (data source) περιέχει την πληροφορία που είναι απαραίτητη για ένα αντικείμενο (π.χ κύβος) έτσι ώστε να έχει πρόσβαση στα δεδομένα .Ουσιαστικά πρόκειται για τη γέφυρα επικοινωνίας με τα δεδομένα και όχι για τα ίδια τα δεδομένα .Μία data source καθορίζει έναν OLE DB provider και κάποιες άλλες ρυθμίσεις για την επίτευξη της επικοινωνίας.



Ο Analysis server είναι ο εξυπηρετητής (server) των Analysis Services. Είναι ειδικά σχεδιασμένος για να δημιουργεί και να διατηρεί δομές πολυδιάστατων δεδομένων και να επιστρέφει τα δεδομένα αυτά σε ερωτήσεις πελατών. Ο Analysis Manager παρέχει ένα user interface που τρέχει στην Microsoft Management Console (MMC). Επιπλέον , οι Data Transformation Services (DTS) κομμάτι του SQL Server Enterprise Manager, παρέχει ένα user interface μαζί με τα server

Σχήμα 4.4 -- Αρχιτεκτονική Analysis Server

Programming interfaces έτσι ώστε να είναι εφικτή η επικοινωνία των εφαρμογών με το object model καθώς και με τον Analysis Manager.

4.2.3 .NET

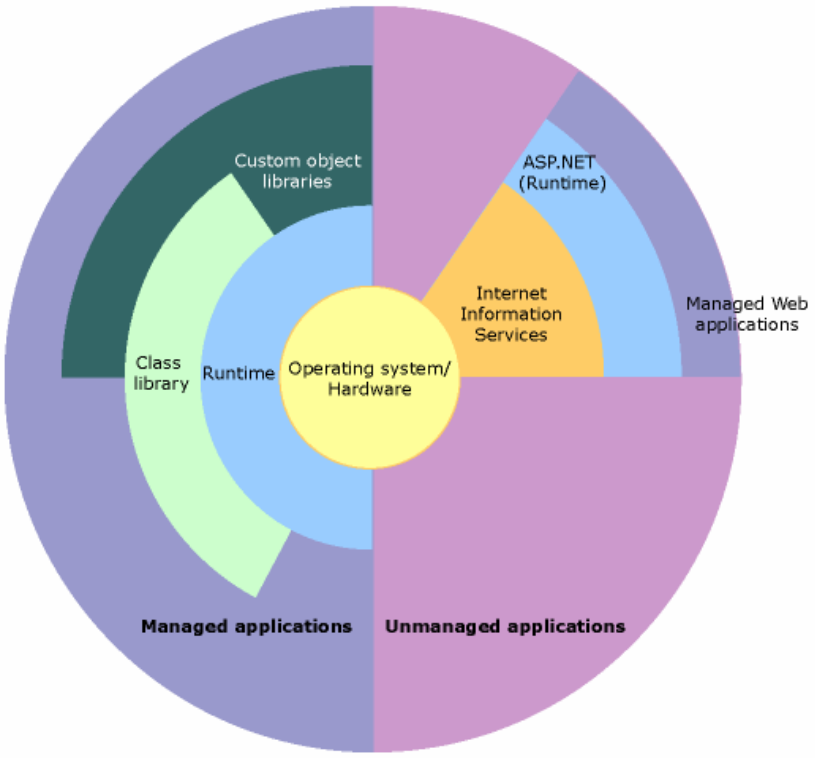
Το Περιβάλλον .NET είναι το αποτέλεσμα της προσπάθειας της Microsoft να αντιμετωπίσει τα αρκετά προβλήματα που δημιουργήθηκαν από μία βασική της σχεδιαστική αρχή. Κυρίαρχο ρόλο στο πέρασμα των χρόνων στις τεχνολογίες των Windows ήταν η συμβατότητα με παλαιότερες τεχνολογίες. Παρόλο το αυτονόητο της επιλογής αυτής, κάθε φορά που κάποια τεχνολογία εξελισσόταν και προσέθετε νέα χαρακτηριστικά, η Microsoft έπρεπε να επεκτείνει και η ίδια με τη σειρά της τα σχεδιαστικά εργαλεία και τις γλώσσες προγραμματισμού ώστε να ικανοποιούνται τόσο οι νέες προδιαγραφές στο hardware όσο και η συμβατότητα με παλαιότερες τεχνολογίες. Μετά από κάποιο σημείο ήταν σαφές πως το σκηνικό αυτό δε μπορούσε να συνεχιστεί επάπειρον, και έπρεπε να σχεδιαστεί κάτι από την αρχή.

4.2.3.1 .NET Framework

Το .NET Framework είναι μία νέα προγραμματιστική πλατφόρμα η οποία έχει ως σκοπό την απλούστερη ανάπτυξη εφαρμογών λογισμικού, τόσο για την ανάπτυξη παραθυρικών εφαρμογών(Windows Applications) όσο και για ανάπτυξη εφαρμογών στο κατανεμημένο περιβάλλον του διαδικτύου. Το .NET Framework αποτελείται από δύο βασικά συστατικά : το *common language runtime(CLR)* και την *.NET Framework* βιβλιοθήκη κλάσεων (Framework Class Library-FCL).

Το CLR είναι ο θεμέλιος λίθος του .NET Framework. Το CLR μπορεί να περιγραφεί σαν ένας πράκτορας(agent) που διαχειρίζεται τον κώδικα κατά τον χρόνο εκτέλεσης, παρέχοντας υπηρεσίες όπως είναι η διαχείριση μνήμης, η διαχείριση των διαφόρων threads καθώς επίσης και η ασφάλεια στις μετατροπές τύπων, κλάσεων και άλλων οντοτήτων έτσι ώστε να εξασφαλίζεται η αξιοπιστία της εφαρμογής. Στην πραγματικότητα η σημαντικότερη λειτουργία του CLR είναι η διαχείριση του κώδικα. Κώδικας που αναφέρεται στον χρόνο εκτέλεσης ονομάζεται διαχειριζόμενος κώδικας(managed code), ενώ διαφορετικά ονομάζεται μη διαχειριζόμενος κώδικας(unmanaged code).

Η βιβλιοθήκη κλάσεων, το άλλο βασικό συστατικό του .NET Framework, είναι μία περιεκτική, αντικειμενοστρεφής συλλογή επαναχρησιμοποιήσιμων τύπων, που μπορούν να χρησιμοποιηθούν για την ανάπτυξη οποιασδήποτε εφαρμογής .



Σχήμα 4. 5 -- Το .NET Framework με μια ματιά

Τα βασικότερα πλεονεκτήματα της πλατφόρμας αυτής είναι τα ακόλουθα :

- **Αντικειμενοστρεφής προγραμματισμός :**το .NET είναι εξολοκλήρου βασισμένο στο αντικειμενοστρεφές μοντέλο, με τα γνωστα πλεονεκτήματα του
- **Ανεξαρτησία γλωσσών :** Ίσως το σημαντικότερο καινούριο χαρακτηριστικό. Όλες οι γλώσσες του .NET (VB,C#,J#) μεταγλωττίζονται σε μία κοινή , ενδιάμεση γλώσσα (MicroSoft Intermediate Language – MSIL), η οποία κατά την ώρα της εκτέλεσης της εφαρμογής μεταγλωττίζεται σε γλώσσα μηχανής
- **Μοιραζόμενος κώδικας :** Πέρα από την ύπαρξη των παραδοσιακών DLL εισάγεται η έννοια της συλλογής(assembly) με την οποία αντιμετωπίζεται το πρόβλημα των διαφορετικών εκδόσεων(γνωστό και ως DLL Hell)
- **Visual Studio.NET :** Πρόκειται για ένα καινούριο γραφικό περιβάλλον ανάπτυξης εφαρμογών , το οποίο είναι και ενιαίο για όλες τις γλώσσες
- **C#:** Πρόκειται για μία νέα γλώσσα προγραμματισμού κατάλληλη κυρίως για γραφικές εφαρμογές

4.2.3.2 Visual Studio.NET

Το Visual Studio.NET είναι το γραφικό περιβάλλον (IDE) με το οποίο έγινε η ανάπτυξη της εφαρμογής. Αποτελεί τη συνέχεια του Visual Studio 6 και παρέχει έναν ενιαίο τρόπο

ενσωμάτωσης όλων των υποστηριζομένων γλωσσών. Οι δυνατότητες αποσφαλμάτωσης (debugging) που προσφέρει αποδείχτηκαν πολύτιμες

4.2.3.3 C#.NET

Κατά μία έννοια η C# είναι για τις προγραμματιστικές γλώσσες ότι και το .NET για το περιβάλλον των Windows. Ακριβώς όπως η Microsoft την τελευταία δεκαετία προσέθετε νέα χαρακτηριστικά στα Windows και στο Windows API, με τον ίδιο τρόπο επεκτάθηκαν η Visual Basic και η C++. Παρά το γεγονός πως το αποτέλεσμα της διαδικασίας αυτής ήταν η ισχυροποίηση των γλωσσών αυτών, ταυτόχρονα κληρονομήθηκαν αρκετά προβλήματα.

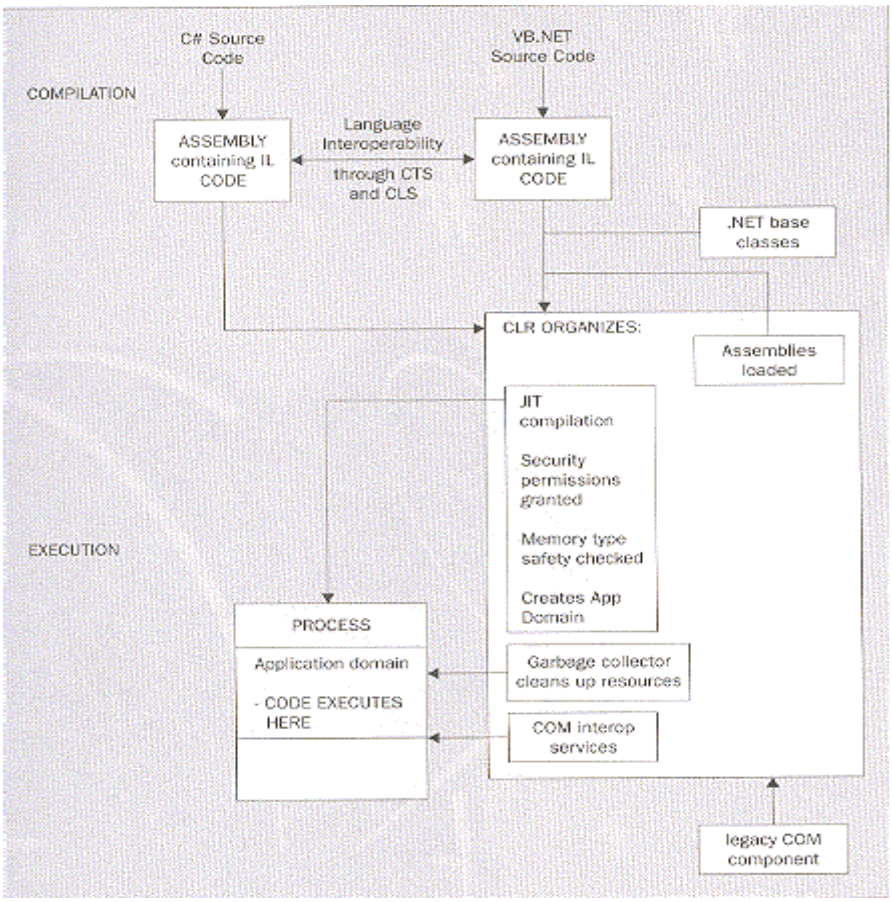
Στην περίπτωση της Visual Basic, το κύριο πλεονέκτημα είναι η ευκολία στην κατανόησή της, καθώς και το γεγονός πως πολλές προγραμματιστικές ενέργειες γίνονται απλά και γρήγορα, αποκρύπτοντας τις λεπτομέρειες του Windows API και των COM αντικειμένων από τον προγραμματιστή. Εντούτοις η Visual Basic ποτέ δεν υπήρξε μία πραγματικά αντικειμενοστρεφής γλώσσα, με αποτέλεσμα πολλές εφαρμογές γρήγορα να αποδιοργανώνονται, και να γίνονται δύσκολες στη συντήρησή τους. Επιπλέον το συντακτικό της έχει κληρονομηθεί από τις πρώτες εκδόσεις της Basic(η οποία σχεδιάστηκε έτσι ώστε να είναι περισσότερο κατάλληλη για αρχάριους προγραμματιστές) με αποτέλεσμα να είναι ακόμα δυσκολότερη μία αντικειμενοστρεφής προσέγγιση.

Η C++ από την άλλη μεριά έχει τις ρίζες της στην ANSI C++. Δεν είναι τελείως συμβατή με το ANSI πρότυπο μιας και η Microsoft σχεδίασε τον πρώτο της C++ compiler πριν τον τελικό καθορισμό του ANSI προτύπου, αλλά πλησιάζει αρκετά. Δύο είναι τα βασικά μειονεκτήματα. Πρώτον η ANSI C++ έχει τις ρίζες της σε τεχνολογίες που επικρατούσαν πριν από μία δεκαετία με αποτέλεσμα να μην μπορεί να υποστηρίξει μοντέρνες έννοιες (όπως για παράδειγμα τα Unicode strings ή XML τεχνολογίες). Έπειτα η Microsoft προσπαθούσε ταυτόχρονα να αναπτύξει την C++ ώστε να γίνει μία υψηλής απόδοσης γλώσσα για απαιτητικές διεργασίες των Windows, με αποτέλεσμα την πρόσθεση ενός μεγάλου αριθμού αναγνωριστικών(keywords) καθώς και διαφόρων άλλων βιβλιοθηκών. Το αποτέλεσμα ήταν χαοτικό.

Για τους παραπάνω λόγους η Microsoft αποφάσισε να δώσει στην προγραμματιστική κοινότητα μία εναλλακτική γλώσσα, σχεδιασμένη από την αρχή και ειδικά για το .NET. Το αποτέλεσμα ήταν η C# η οποία περιγράφεται από την ίδια την εταιρία ως μία απλή, μοντέρνα και ασφαλής γλώσσα προερχόμενη από τις C και C++. Κάποιος ανεξάρτητος παρατηρητής θα υπογράμμιζε επίσης την στενή της σχέση με τη Java. Συντακτικά, η C# μοιάζει πολύ με τις C++ και Java, σε τέτοιο βαθμό μάλιστα που πολλά αναγνωριστικά είναι ταυτόσημα. Τα κυριότερα χαρακτηριστικά της C# είναι τα ακόλουθα :

- Πλήρης υποστήριξη κλάσεων και αντικειμενοστρεφών εννοιών, συμπεριλαμβάνοντας κληρονομικότητα διαπροσωπειών(Interfaces) και τύπων
- Συνεπές και πολύ καλά ορισμένο σύνολο βασικών τύπων
- Εγγενής υποστήριξη δημιουργίας XML εγγράφων
- Αυτόματη αποδέσμευση δυναμικά δεσμευμένης μνήμης
- Πλήρης πρόσβαση στις βιβλιοθήκες του .NET καθώς και εύκολη πρόσβαση στο Windows API(αν και το τελευταίο δεν είναι σχεδόν ποτέ αναγκαίο)
- Οι δείκτες και η απευθείας πρόσβαση στη μνήμη είναι διαθέσιμοι αν είναι απαραίτητο, αλλά η γλώσσα έχει σχεδιαστεί με τέτοιον τρόπο ώστε να μην είναι απαραίτητο σχεδόν σε καμία περίπτωση
- Με μία απλή αλλαγή στις επιλογές του compiler το αποτέλεσμα μπορεί να είναι είτε ένα εκτελέσιμο πρόγραμμα ή μία βιβλιοθήκη

Στο ακόλουθο διάγραμμα απεικονίζεται ο τρόπος με τον οποίο τα κυριότερα συστατικά του .NET ενεργούν κατά τη διάρκεια της μεταγλώττισης και της εκτέλεσης



Σχήμα 4.6 -- Βασικά συστατικά του .NET κατά το χρόνο μεταγλώτισσης και εκτέλεσης

4.2.4 MDX

4.2.4.1 Γενικά

Η MDX (**M**ulti**d**imensional E**x**pressions) , είναι ένα συντακτικό που υποστηρίζει τον ορισμό και τη δημιουργία πολυδιάστατων αντικειμένων και δεδομένων. Η MDX είναι κατά πολλές έννοιες όμοια με την Structured Query Language (SQL), αλλά όχι μία επέκτασή της. Μάλιστα κάποιες λειτουργίες που γίνονται με την MDX μπορούν να γίνουν και με την SQL αλλά λιγότερο αποδοτικά Όπως και με ένα SQL query, κάθε MDX query απαιτεί την αναζήτηση πληροφορίας (με τον όρο SELECT), μία αφετηρία (με τον όρο FROM), και ένα φίλτρο (με τον όρο WHERE).

```
SELECT [<axis_specification>
       [, <axis_specification>...]]
  FROM [<cube_specification>]
WHERE [<slicer_specification>]]
```

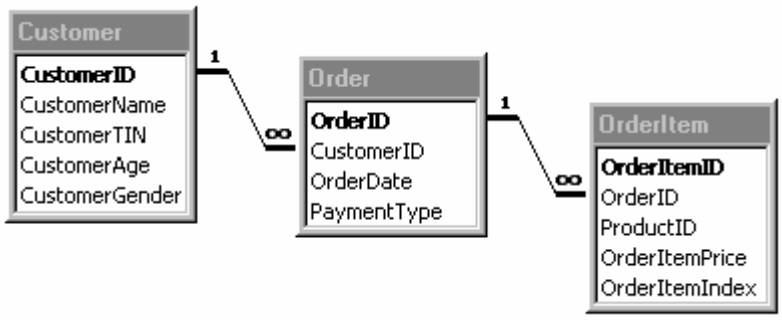
Αυτές καθώς και άλλες είναι οι λέξεις κλειδιά για την ανάκτηση συγκεκριμένης πληροφορίας από έναν κύβο. Η MDX παρέχει επίσης ένα σύνολο συναρτήσεων για τη διαμόρφωση των ανακτόμενων δεδομένων καθώς και τη δυνατότητα επέκτασης με ορισμό νέων συναρτήσεων από τον χρήστη.

Η MDX, όμοια με την SQL, παρέχει μία γλώσσα ορισμού δεδομένων (DDL) για τη διαχείριση δομών δεδομένων. Υπάρχουν MDX εντολές για τη δημιουργία ή διαγραφή κύβων, διαστάσεων, μέτρων. Όπως γίνεται κατανοητό σκοπός της MDX είναι να κάνει ευκολότερη και πιο ενστικτώδη τη διαχείριση πολυδιάστατης πληροφορίας

4.2.4.2 Βασικές οντότητες

Οι περισσότερες γλώσσες που χρησιμοποιούνται για ορισμό και διαχείριση πληροφορίας, όπως και η SQL, έχουν σχεδιαστεί ώστε να ανακτούν τα δεδομένα σε δύο διαστάσεις, συνήθως σε γραμμές και στήλες

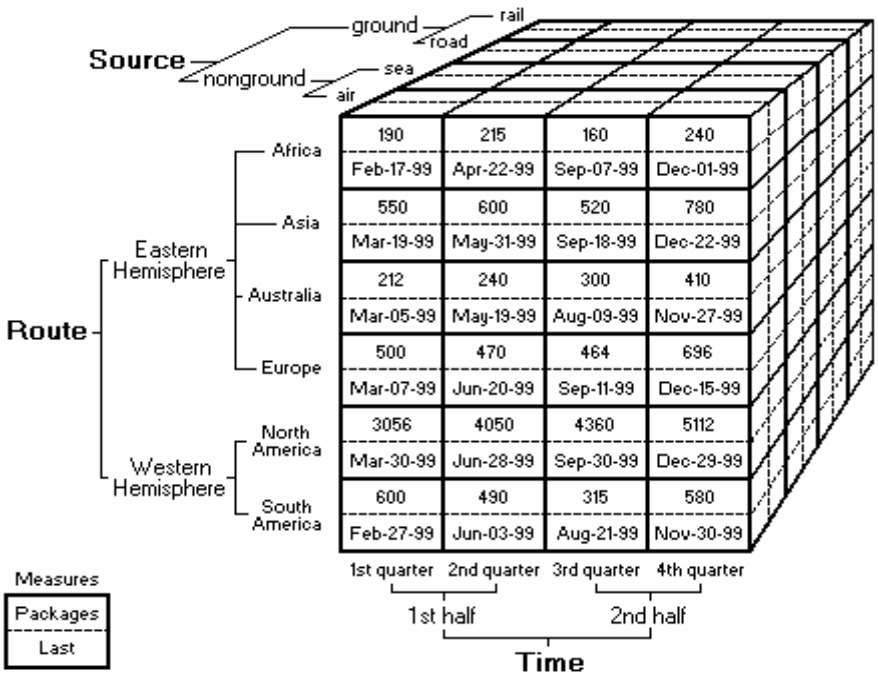
Στο Σχ.4.7 απεικονίζεται ένα παραδοσιακό σχήμα σχεσιακής βάσης για την αποθήκευση παραγγελιών



Σχήμα 4.7 -- Ένα κλασσικό σχεσιακό σχήμα

Κάθε πίνακας αναπαριστά διδιάστατα δεδομένα. Στην τομή κάθε γραμμής και στήλης υπάρχει ένα μοναδικό στοιχείο πληροφορίας που ονομάζεται πεδίο. Οι συγκεκριμένες στήλες που προκύπτουν από ένα SQL query καθορίζονται από τον όρο SELECT, ενώ οι γραμμές περιορίζονται από τον όρο WHERE.

Αντίθετα τα πολυδιάστατα δεδομένα μπορούν να αναπαρασταθούν από δομές με περισσότερες διαστάσεις Αυτές οι δομές είναι οι κύβοι.. Σε τομή διαστάσεων ενός κύβου είναι δυνατόν να υπάρχουν περισσότερα από ένα στοιχεία πληροφορίας. Τα στοιχεία ατά ονομάζονται μέτρα Το Σχ.4.8 απεικονίζει έναν κύβο τριών διαστάσεων Route, Service και Time και δύο μέτρων Packages και Last. Κάθε διάσταση αποτελείται από επίπεδα και τα επίπεδα αποτελούνται από τα μέλη τους



Σχήμα 4.8 -- Παράδειγμα ενός κύβου με 3 διαστάσεις και 2 μέτρα

Όπως γίνεται κατανοητό από το σχήμα ακόμα και απλές ερωτήσεις δεν είναι δυνατόν να διατυπωθούν εύκολα με την SQL. Μία πιο λεπτομερής ανάλυση ξεφεύγει από το σκοπό της διπλωματικής. Για περισσότερες πληροφορίες ο αναγνώστης παραπέμπεται στα αντίστοιχα εγχειρίδια της Microsoft ή στο Help file του Analysis Manager.

4.2.5 ADOMD.NET

Μέχρι τώρα περιγράψαμε την προγραμματιστική πλατφόρμα καθώς και τον OLAP server που θα χρησιμοποιηθούν στο εργαλείο μας. Ένα επιπλέον θέμα που πρέπει να καθοριστεί είναι ο τρόπος με τον οποίο θα επικοινωνεί η εφαρμογή με τον server μας. Και αυτό γιατί το .NET framework μας παρέχει μεν τη δυνατότητα να επικοινωνήσουμε με τον Analysis Server, μέσω του data provider ADO.NET, όμως το τελευταίο είναι ένα μοντέλο ακατάλληλο για διαχείριση πολυδιάστατης πληροφορίας. Με απλά λόγια αν επιθυμούσαμε την σύνδεση με ένα απλό σχεσιακό σχήμα, εκτελώντας μόνο SQL ερωτήσεις, το ADO.NET θα ήταν κατάλληλο. Στην περίπτωσή μας όμως, όπου θέλουμε να εκτελέσουμε MDX ερωτήσεις σε έναν κύβο είναι αναγκαίο ένα πολυδιάστατο μοντέλο.

Η βιβλιοθήκη ADOMD.NET είναι παρεμφερής με την ActiveX Data Objects Multidimensional (ADO MD) βιβλιοθήκη , μία επέκταση της ActiveX Data Objects (ADO) βιβλιοθήκης που χρησιμοποιείται για πρόσβαση σε πολυδιάστατα δεδομένα σε Component Object Model (COM) client εφαρμογές.

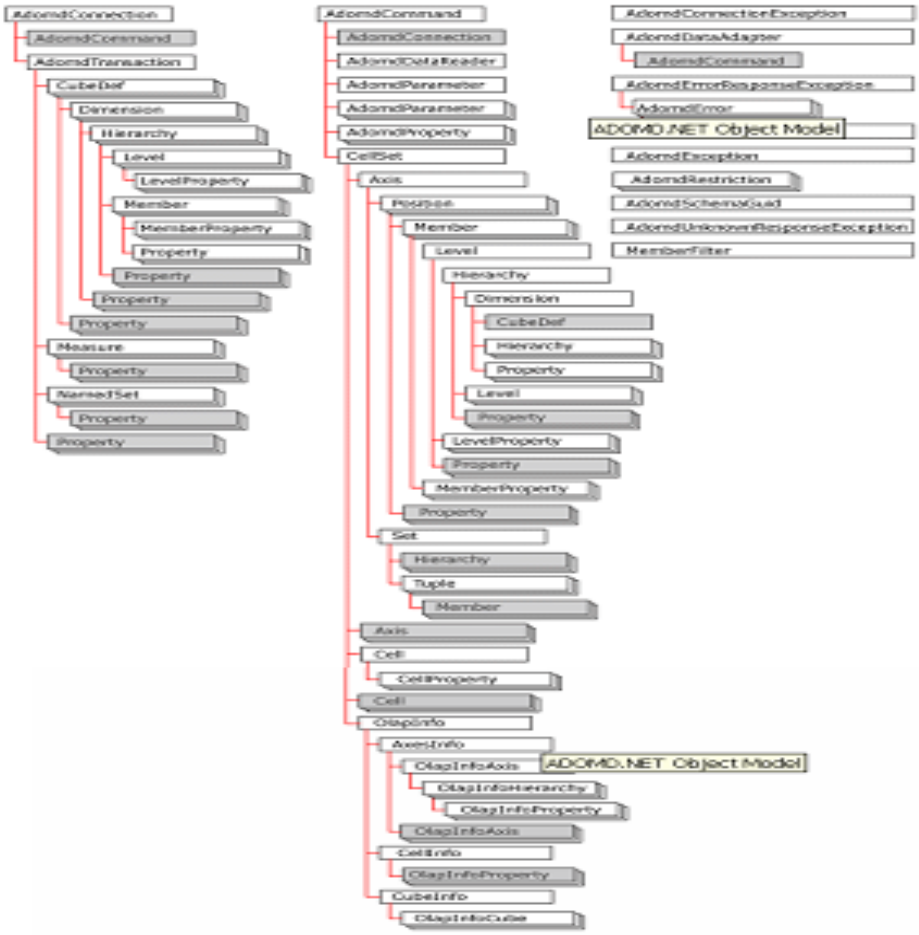
Το ADOMD.NET (Active X Data Objects- Multidimensional) παρέχει πρόσβαση σε πολυδιάστατες πηγές δεδομένων όπως οι Analysis Services. Οι client εφαρμογές μπορούν να χρησιμοποιήσουν το ADOMD.NET για τη σύνδεση με πολυδιάστατες πηγές δεδομένων, την ανάκτηση και διαχείριση δεδομένων και μεταδεδομένων.

Το ADOMD.NET είναι ένας προτυποποιημένος .NET data provider, και χρησιμοποιείται μαζί με άλλους παροχείς που υποστηρίζουν το πρότυπο XML for Analysis έκδοση 1.1 για την επικοινωνία με πηγές δεδομένων, χρησιμοποιώντας είτε TCP/IP ή HTTP ρεύματα για την αποστολή και λήψη SOAP αιτήσεων(requests) και απαντήσεων(responses) που είναι συμβατές με το XML for Analysis πρότυπο. Τα δεδομένα μπορούν να ανακτηθούν σε πολυδιάστατη μορφή και να εξεταστούν χρησιμοποιώντας το ADOMD.NET object model. Το ADOMD.NET, παρέχει τη δυνατότητα διαχείρισης των δεδομένων με δύο τρόπους : είτε μέσω συμβατών με OLEDB rowsets ή χρησιμοποιώντας το ADOMD.NET object model το οποίο και απεικονίζεται στο Σχ. 4.9 .

Το ADOMD.NET έχει σχεδιαστεί με τέτοιον τρόπο ώστε να λειτουργεί μόνο με providers συμβατούς με το πρότυπο XML for Analysis compliant και που υποστηρίζουν sessions .Τα Sessions (σύνοδοι) χρησιμοποιούνται για να 'πλαισιώσουν' το πεδίο και την εμβέλεια των

εντολών και δοσοληψιών.Εαν χρησιμοποιηθεί provider που δεν υποστηρίζει sessions τότε η επικοινωνία αποτυγχάνει. Στην εφαρμογή μας χρησιμοποιούμε τον msolap που υποστηρίζει sessions.

Υπάρχουν 3 διαφορετικά αντικείμενα με τα οποία μπορεί να γίνει ανάκτηση δεδομένων α) CellSet β)AdomdDataReader γ) XmlReader. Το καθένα από αυτά παρέχει ξεχωριστές δυνατότητες. Στο εργαλείο μας θα χρησιμοποιήσουμε τον AdomdDataReader. Ο AdomdDataReader παρέχει μία καλή ισορροπία μεταξύ αλληλεπίδρασης(interactivity) και κόστους (overhead) όταν διαχειρίζεται πολυδιάστατα δεδομένα.. Το βασικό του πλεονέκτημα είναι πως μπορεί να επεξεργάζεται πολύ γρήγορα και αποδοτικά μεγάλο όγκο δεδομένων. Το μειονέκτημα είναι πως οι δυνατότητες τροποποίησης των δεδομένων αυτών είναι περιορισμένες. Σε μία εφαρμογή οπτικοποίησης όμως, όπου δεν χρειάζεται καμία τροποποίηση των δεδομένων ο AdomdDataReader αποδείχτηκε η καλύτερη επιλογή.



Σχήμα 4.9 -- Το object model του ADOMD.NET

4.2.6 Sourcegrid

Σκοπός του OLAP Lens είναι η διδιάστατη απεικόνιση OLAP αποτελεσμάτων. Γίνεται προφανής η ανάγκη να χρησιμοποιηθεί μία πινακοειδής μορφή. Δυστυχώς όμως , αν και στη βιβλιοθήκη του .NET υπάρχει ένα νέο control, το DataGrid, που χρησιμοποιείται ακριβώς για την πινακοειδή απεικόνιση πληροφορίας, στην περίπτωσή μας αποδεικνύεται ακατάλληλο. Παρόλα τα πολλά και ενδιαφέροντα χαρακτηριστικά του DataGrid, υπάρχει το μειονέκτημα πως ο πίνακς θεωρείται ως ένα σύνολο γραμμών και στηλών και όχι ως ένα σύνολο κελιών. Επίσης είναι ουσιαστικά σχεδιασμένο ώστε να χρησιμοποιεί ως πηγή δεδομένων το Dataset μία δομή δεδομένων του .NET . Με τον τρόπο αυτό γίνεται σαφές πως είναι αδύνατον να εφαρμοστεί μία τεχνική όπως το Table Lens. Όπως αναφέρθηκε στις απαιτήσεις χρήστη βασικό ρόλο παίζει η μεγένθυνση και ο χρωματισμός ενός συγκεκριμένου κελιού κάτι που δεν μπορεί να γίνει στο DataGrid, μιας και για να το επιτύχουμε αυτό θα έπρεπε ταυτόχρονα να γινόταν focus σε ολόκληρη τη στήλη ή σε ολόκληρη τη γραμμή.

Για το λόγο αυτό έπρεπε να βρεθεί μία εναλλακτική λύση. Είτε θα έπρεπε να σχεδιάσουμε από το μηδέν ένα control κάτι το οποίο θα ήταν χρονοβόρο και εκτός της λογικής της διπλωματικής ή θα έπρεπε να βρεθεί κάποιο ήδη υπάρχον control. Η λύση δόθηκε μέσω του SourceGrid. Το SourceGrid είναι ένα opensource control γραμμένο εξολοκλήρου σε C#. Είναι μεταγλωτισμένο με το Microsoft Framework. NET 1.1. Προκύπτει η βιβλιοθήκη SourceLibrary.dll 1.2.0.0.

Σε αντίθεση με το DataGrid στο SourceGrid ο πίνακας μπορεί να θεωρηθεί και ως ένα σύνολο κελιών κάτι που μας επιτρέπει να εφαρμόσουμε μη παραδοσιακές τεχνικές οπτικοποίησης.

4.3 Τεχνικές υλοποίησης

Στην παράγραφο τούτη θα αναλυθούν οι τεχνικές υλοποίησης της εφαρμογής μας. Έμφαση θα δοθεί στη φιλοσοφία της υλοποίησης και όχι σε μία αναλυτική επεξήγηση του κώδικα. Όπου κρίθηκε σκόπιμο παρατίθενται απλά διαγράμματα των κλάσεων που συμμετέχουν στην υλοποίηση. Ολόκληρος ο κώδικας περιέχεται στο CD-ROM που συνοδεύει τον τόμο τούτο.

4.3.1 Σύνδεση με την πηγή δεδομένων

Η σύνδεση με την πηγή δεδομένων είναι φυσικά το πρώτο πράγμα που πρέπει να γίνει. Ο OLAP server που χρησιμοποιείται είναι ο Analysis Server των Analysis Services. Στον server

είναι δυνατόν να υπάρχουν περισσότερες από μία βάσεις (ή καλύτερα πηγές δεδομένων). Έτσι με την εκκίνηση του OLAP Lens, πρέπει να ανακτηθεί μία λίστα με τις διαθέσιμες βάσεις. Αυτό αν και φαινομενικά απλό αποδείχτηκε αγκάθι της σχεδίασης όπως θα δούμε στη συνέχεια.

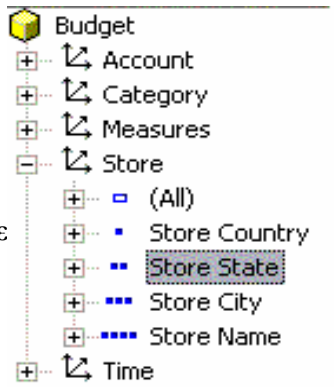
Ο κλασσικός τρόπος ανάκτησης των διαθέσιμων βάσεων σε έναν database server είναι με την εγκατάσταση μίας ADODB σύνδεσης. Δυστυχώς όμως η βιβλιοθήκη για τον ADODB provider προέρχεται από ένα COM component με αποτέλεσμα η C# να αντιμετωπίζει πρόβλημα στη κλήση της αντίστοιχης βιβλιοθήκης. Οπότε προέκυψε μία καλή ευκαιρία να δοκιμάσουμε στην πράξη την ανεξαρτησία των γλωσσών στο .NET. Έτσι δημιουργήσαμε με τη βοήθεια της Visual Basic.NET μία βιβλιοθήκη , την server_databases.dll, η οποία απλά συνδέεται με τον Analysis Server και αποθηκεύει σε έναν πίνακα τις διαθέσιμες βάσεις δεδομένων. Αυτό ήταν και το μοναδικό σημείο που δεν χρησιμοποιήθηκε η C#.

Η λίστα με τις διαθέσιμες βάσεις εμφανίζεται στην οθόνη του χρήστη, ο οποίος με τη σειρά του επιλέγει την επιθυμητή. Αμέσως εμφανίζονται στην οθόνη οι κύβοι που έχουν δημιουργηθεί από δεδομένα της βάσης αυτής. Ο χρήστης επιλέγει τον επιθυμητό κύβο και ξεκινάει η αρχικοποίηση της CPM δομής. Η σύνδεση βέβαια για την ανάκτηση των διαθέσιμων κύβων γίνεται με τον ADOMD.NET provider.

4.3.2 Ανάκτηση δεδομένων

Η ανάκτηση των δεδομένων από τον Analysis Server γίνεται εφιτκή μέσω του ADOMD.NET. Στην υποπαράγραφο αυτό θα δείξουμε τον τρόπο με τον οποίο ανακτούμε τα δεδομένα.

Η ανάκτηση των δεδομένων γίνεται μέσω της εκτέλεσης διαδοχικών MDX queries. Ο τρόπος είναι σχετικά απλός χάρη στο πλουσιο συντακτικό των MDX. Μέρος του συντακτικού αυτού είναι και η έκφραση AllMembers. Όπως γίνεται κατανοητό από το όνομα της, η έκφραση αυτή αναφέρεται σε όλα τα μέλη ενός επιπέδου.



Για παράδειγμα ας θεωρήσουμε τον κύβο Budget του οποίου οι διαστάσεις και τα επίπεδα απεικονίζονται στο διπλανό σχήμα . Στη διάσταση Store διακρίνουμε 5 επίπεδα. Η έκφραση [Store].[Store State]. AllMembers αναφέρεται σε όλες τις πολιτείες, όλων των χωρών. Αν θέλαμε να αναφερθούμε στις πολιτείες μίας συγκεκριμένης χώρας θα έπρεπε να χρησιμοποιηθεί η έκφραση Children, για παράδειγμα η έκφραση [Store].[All Stores].[Canada]. Children δίνει τις πολιτείες του Καναδά ενώ δε θα μπορούσαμε να γράψουμε [Store].[All Stores].[Canada]. AllMembers

μιας και ο όρος [Store].[All Stores].[Canada] δεν είναι επίπεδο.

Αν απλά θέλαμε να απεικονίζουμε τα αποτελέσματα ενός καινούριου MDX query κάθε φορά που γινέται ένα drill-down ο όρος Children θα ήταν ο ενδεδειγμένος. Για τη δημιουργία των πολυκύβων στο πλαίσιο του CPM θα χρησιμοποιηθεί ο όρος AllMembers.

4.3.3 Υλοποιούμενη δομή CPM

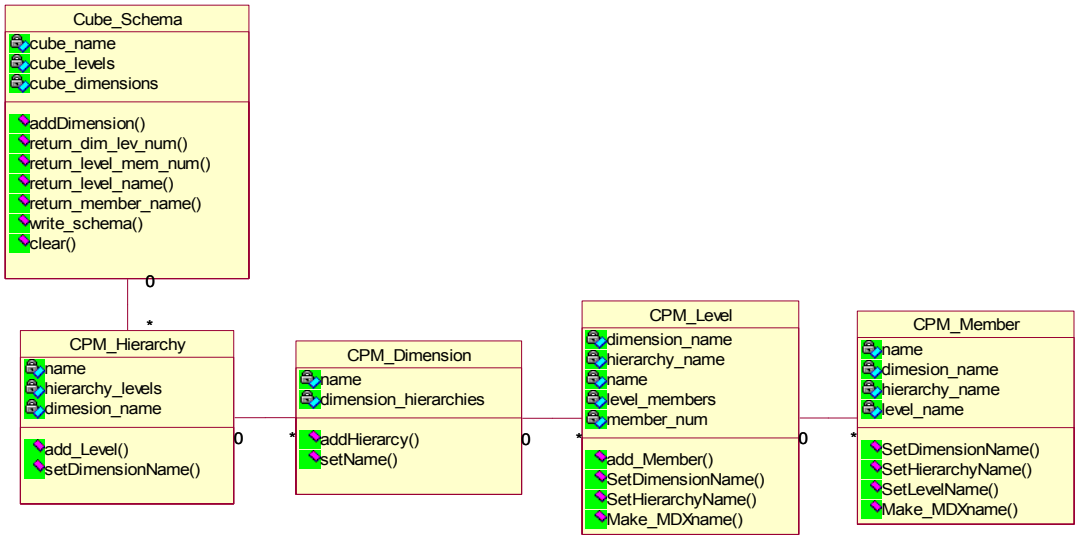
Στην παρούσα παράγαραφο θα αναλυθεί ο τρόπος με τον οποίο υλοποιείται το παρουσιαστικό στρώμα του CPM. Σκοπός είναι βέβαια να συμπεριλάβουμε τις βασικές οντότητες του στρώματος αυτού σε μία αντικειμενοστρεφή υλοποίηση.

4.3.3.1 Αρχικοποίηση

Ο χρήστης αφού συνδεθεί με τον κύβο που επιθυμεί, είναι σε θέση να επιλέξει τις διαστάσεις. Πριν όμως γίνει αυτό, δηλαδή ταυτόχρονα με τη σύνδεση με τον κύβο γίνεται μία αρχικοποίηση σε μία κλάση που ονομάζεται Cube_Schema και η οποία περιέχει πληροφορίες σχετικά με το σχήμα του κύβου. Η αρχικοποίηση αυτή γίνεται μέσω της κλάσης Connect_Navigate. Πιο συγκεκριμένα η Cube_Schema περιέχει τις ακόλουθες πληροφορίες :

- Το όνομα του κύβου με τον οποίο υπάρχει σύνδεση
- Τα ονόματα των διαστάσεων που υπάρχουν στον κύβο
- Τις ιεραρχίες που διέπουν κάθε διάσταση
- Τα επίπεδα κάθε ιεραρχίας

Η αρχικοποίηση αυτή είναι απαραίτητη μιας και ο χρήστης θα πρέπει να βλέπει όλες αυτές τις πληροφορίες στην φόρμα που θα επιλέξει τις διαστάσεις. Οι διαστάσεις, οι ιεραρχίες, τα επίπεδα καθώς και τα μέλη αναπαριστώνται αντίστοιχα από τις κλάσεις CPM_Dimension, CPM_Hierarchy, CPM_Level και CPM_Member. Η δημιουργία των διαφόρων στιγμυοτύπων κάθε κλάσης γίνεται βέβαια δυναμικά, με τη βοήθεια μίας νέας δομής που παρέχει η C#, της ArrayList. Η σχέση μεταξύ των κλάσεων αυτών απεικονίζεται στο Σχ.4.10:



Σχήμα 4.10 -- Cube_Schema: Δομή αρχικοποίησης

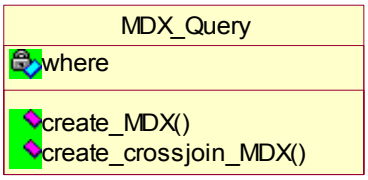
Μία από τις βασικές λειτουργίες των κλάσεων αυτών είναι η επιστροφή των ονομάτων επιπέδων και μελών σε κατάλληλη μορφή παρουσίασης, για παράδειγμα αν έχουμε το επίπεδο [Product].[All Products].[Non-Consumable].[Health and Hygiene].[Decongestants] γίνεται εύκολα κατανοητό πως στην απεικόνισή μας θέλουμε να δείξουμε μόνο το Decongestants.

4.3.3.2 Δημιουργία multicubes

Η υλοποίηση της βασικής οντότητας του παρουσιαστικού στρώματος του CPM, των multicubes, αναλύεται στην παράγραφο τούτη.

Αφού ο χρήστης επιλέξει τις διαστάσεις που επιθυμεί, με τη χρήση του κουμπιού εκκίνησης(run) δημιουργείται η CPM δομή με την εκτέλεση ενός στιγμυοτύπου της κλάσης Initialize. Ας θυμηθούμε στο σημείο αυτό πως ο χρήστης μπορεί μέσω αρχικών επιλογών να καθορίσει τον αριθμό των επιπέδων στα οποία θέλει να κάνει drill-down και συνεπώς να καθορίσει τον αριθμό των πολυκύβων που θα δημιουργηθούν.

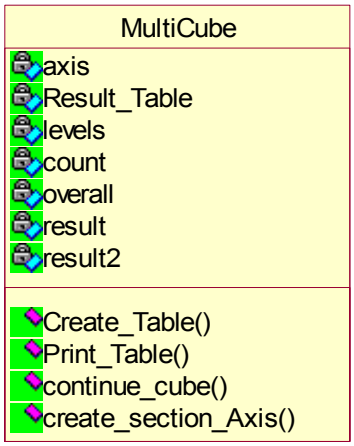
Για τη δημιουργία κάθε πολυκύβου εκτελείται ένα MDX query, που αντιπροσωπεύεται από την κλάση MDX_Query. Η ανάκτηση των αποτελεσμάτων γίνεται μέσω του ADOMD.NET provider και συγκεκριμένα του αντικειμένου AdomdDataReader που μας επιτρέπουν πολυδιάστατη διαχείριση της πληροφορίας. Το αντικείμενο AdomdDataReader μετά την ολοκλήρωση της δοσοληψίας περιέχει όλη την απαραίτητη πληροφορία για τη δημιουργία ενός πολυκύβου.



Σχήμα 4.11 -- MDX_Query:Κλάση ερωτήσεων

Ο πολυκύβος αντιπροσωπεύεται στον OLAP Lens από την κλάση Multicube . Η κλάση δέχεται ως είσοδο το αντίστοιχο στιγμυότυπο της κλάσης MDX_Query που δημιουργήθηκε για την εκτέλεση της ερώτησης. Οι κυριότερες ιδιότητες της κλάσης Multicube είναι οι εξής:

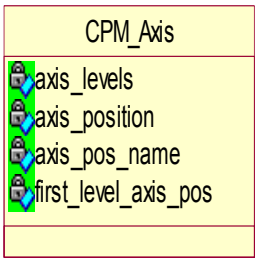
- Ο πίνακας Result_Table που αντιπροσωπεύει τα περιεχόμενα του πολυδιάστατου χώρου
- Οι άξονες που αποτελούν στιγμυότυπα της κλάσης CPM_Axis και που ορίζουν τον πολυδιάστατο χώρο



Σχήμα 4.12 -- Κλάση MultiCube

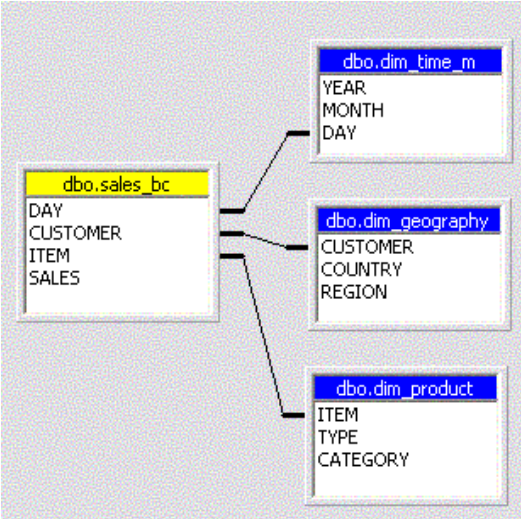
Το παρουσιαστικό στρώμα του CPM μπορεί να χωριστεί όπως είδαμε σε δύο μεγάλες κατηγορίες: τον πολυδιάστατο χώρο και τα περιεχόμενά του. Αναφορικά με τα περιεχόμενά του ακολουθείται η εξής διδικασία: Τα δεδομένα από τον AdomdDataReader αντιγράφονται στον πίνακα Result_Table. Ο Result_Table είναι στη γενική περίπτωση τριδιάστατος. Οι δύο άξονες αντιπροσωπεύουν τους δύο παρυσιαστικούς άξονες Rows και Columns και ο τρίτος δημιουργείται στην περίπτωση που έχει επιλεχθεί φιλτράρισμα των τιμών οπότε αντιπροσωπεύει τον άξονα Sections. Τα δεδομένα στον AdomdDataReader βρίσκονται επίσης σε μία πινακοειδή δομή (όχι πίνακα όμως), η οποία μοιάζει αρκετά στην οργάνωσή της με το μοντέλο μας, και την οποία την οποία σαρώνουμε γραμμή γραμμή ώστε να γίνει σωστή αντιστοίχιση στον πίνακα Result_Table.

Πριν αναφερθούμε στον πολυδιάστατο χώρο θα πρέπει να παρουσιάσουμε την κλάση CPM_Axis η οποία αντιπροσωπεύει τους άξονες του CPM.



Στη συγκεκριμένη εφαρμογή πρόκειται για άξονες συντεταγμένων (coordinate axes). Ένας άξονας αποτελείται από σημεία που αντιπροσωπεύονται από το διδιάστατο στη γενική περίπτωση πίνακα `axis_position`.Όπως τονίστηκε επανειλημμένως μας ενδιαφέρει η συνύπαρξή περισσοτέρων από μίας διαστάσεων στον ίδιο άξονα. Έτσι στον πίνακα `axis_position` η μία διάσταση αντιπροσωπεύει τον αριθμό των πραγματικών σημείων που πρέπει να παρουσιαστούν. Στην άλλη διάσταση του `axis_position` συνυπάρχουν όλες οι υπόλοιπες λογικές διαστάσεις . Αυτό που απεικονίζεται είναι βέβαια μία 2D-slice(ή ένα cross-join) και όχι ολόκληρος ο multicube. Πρακτικά η 2D-slice δεν είναι τίποτα άλλο από μία τομή του τριδιάστατου στη γενική περίπτωση Result_Table. Ας δώσουμε όμως ένα παράδειγμα για να γίνει η διαδικασία πιο κατανοητή.

Έστω για παράδειγμα ο κύβος Sales με τις διαστάσεις που φαίνονται παρακάτω :



Έχουμε 3 διαστάσεις και ένα μέτρο. Έστω επίσης οτι ο χρήστης έχει επιλέξει να δημιουργούνται οι μέγιστοι δυνατοί κύβοι, δηλαδή να φτάνουμε στο προτελευταίο επίπεδο της ιεραρχίας και για τους δύο παρουσιαστικούς άξονες. Έστω επίσης πως οι επιλογές του χρήστη είναι οι ακόλουθες :

- Rows: [Geography].[All Geography]
- Columns : [Product].[All Product].[Category] και [Time].[All Time].[Year]1992]

Η Initialize θα προκαλούσε την εκτέλεση των παρακάτω ερωτήσεων :

```
select ( crossjoin
({[Product].[Category].AllMembers},{[Time].[Year].AllMembers}) ) ON
columns, [Geography].[(All)].AllMembers ON rows from Sales
select ( crossjoin
({[Product].[Category].AllMembers},{[Time].[Month].AllMembers}) ) ON
columns, [Geography].[Region].AllMembers ON rows from Sales
```

```
select ( crossjoin ({[Product].[Category].AllMembers},{[Time].[Month].AllMembers}) ) ON columns, [Geography].[Country].AllMembers ON rows from Sales
```

Θα δημιουργηθούν 3 multicubes ένας για κάθε query. Ο κάθε multicube θα είχε 2 άξονες μιας και δεν έγινε επιλογή φιλτραρίσματος. Στον άξονα Rows ο πίνακας `axis_position` θα ήταν μονοδιάστατος μιας και τα σημεία του προκύπτουν μόνο από τη λογική διάσταση `Geography`.

Στον άξονα Columns όμως έχουμε δύο λογικές διαστάσεις τις `Time` και `Product`. Στην περίπτωση αυτή κάθε σημείο του άξονα δεν μπορεί να προκύπτει μόνο από τη μια διάσταση. Για το λόγο αυτό ο πίνακας είναι διδιάστατος και ισχύει :

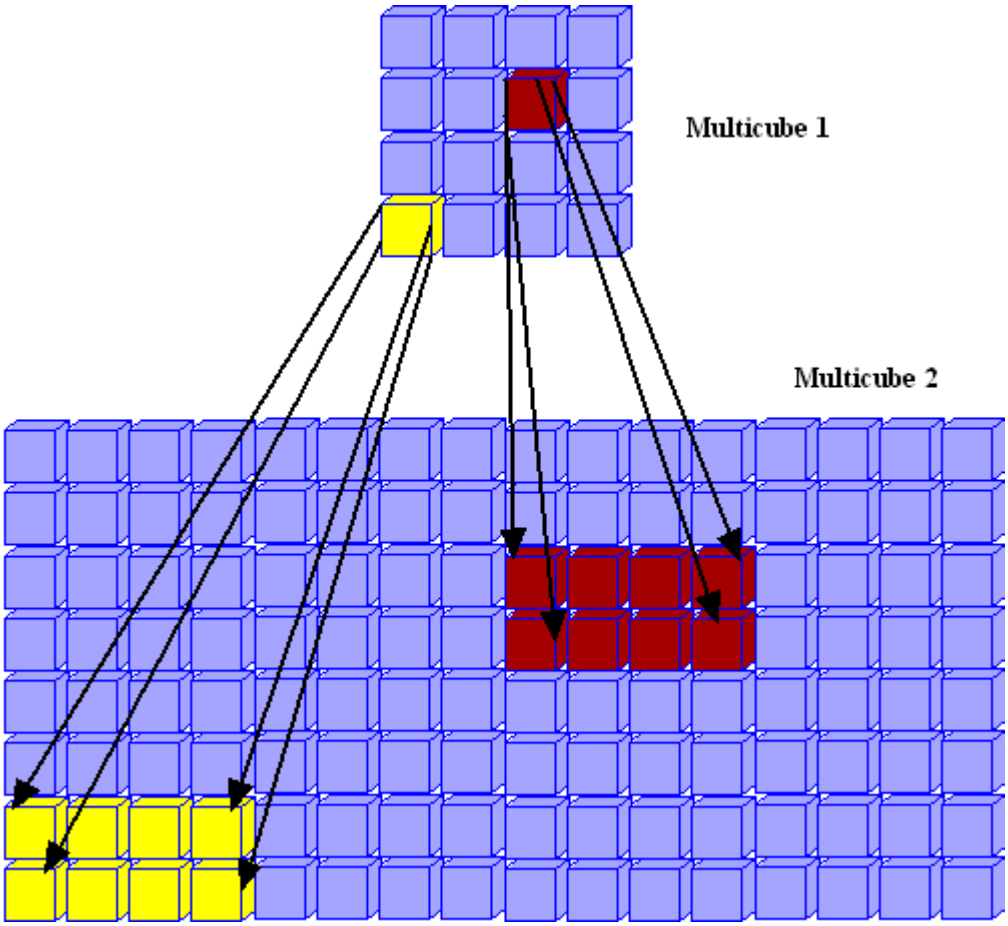
```
axis_position[0][0]=" ECONOMY"
axis_position[1][0]=" 1992"
axis_position[0][1] =" ECONOMY"
axis_position[1][1]=" 1993" ...
axis_position[0][7] =" LARGE"
axis_position[1][7] ="1992"
axis_position[0][8] =" LARGE"
axis_position[1][8] ="1993"
```

Δηλαδή στη διάσταση του `axis_position` που επιλέχθηκε πρώτη και στην οποία δεν κάνουμε drill έχουμε επαναλαμβανόμενες τιμές.

Τα σημεία που απαρτίζουν κάθε άξονα μπορούν να θεωρηθούν ως hierarchically decomposable. Τυπικά για να συνέβαινε κάτι τέτοιο θα έπρεπε να προκύπτουν από μία συνθήκη επιλογής που θα περιείχε μία ancestor συνάρτηση. Στην περίπτωσή μας δεν χρησιμοποιούμε το λογικό στρώμα του CPM ώστε να προκύπτουν με τον τρόπο αυτό, αλλά σημασιολογικά χρησιμοποιώντας τον όρο AllMembers του MDX συντακτικού πλησιάζουμε τη λογική αυτή. Κατά αυτή την έννοια τα σημεία που παρουσιάζονται τελικά είναι τα atomic σημεία στα οποία αναλύονται τα hierarchically decomposable σημεία. Με τον τρόπο αυτό οι άξονες που δημιουργούμε μπορούν να θεωρηθούν ως οι atomic ισοδύναμοι των αρχικών.

Αφού δημιουργηθούν οι multicubes θα πρέπει με κάποιον τρόπο να επιτευχθεί η αναμεταξύ τους σύνδεση. Κάτι τέτοιο προκύπτει από την τεχνική οπτικοποίσης που έχουμε επιλέξει. Εφόσον όταν ο χρήσης κάνει drill-down γίνεται μετάβαση στον επόμενο κύβο, θα πρέπει να ξέρουμε σε ποια σημεία του νέου multicube πρέπει να αναφερθούμε. Και αυτό γιατί μετά το πρώτο drill-down αυτό που απεικονίζεται δεν είναι μία 2D-slice αλλά ένα crossjoin.

Για το λόγο αυτό κάθε στοιχείο του Result_Table δεν είναι ένα απλό string. Κάθε στοιχείο του Result_Table είναι μία δομή(struct) η οποία εκτός από το string περιέχει 4 ακόμα ακεραίους που είναι οι συντεταγμένες εκκίνησης και τερματισμού στους άξονες του επομένου multicube. Πρακτικά οι ακέραιοι αυτοί λειτουργούν ως δείκτες. Την δεικτοδότηση αναλαμβάνει να κάνει η κλάση Map στηριζόμενη στο σχήμα του κύβου που έχει αρχικοποιηθεί όπως προαναφέρθηκε στην κλάση Cube__Schema.



Σχήμα 4.13 -- Δεικτοδότηση ανάμεσα σε multicubes

4.3.4 Παράμετροι συστήματος

Οι παράμετροι του συστήματος καθορίζονται από τον ίδιο το χρήστη και παίζουν καθοριστικό ρόλο στην απόδοση. Το εργαλείο μας έχει ως κατευθυντήρια γραμμή την αρχικοποίση μέσω της ανάκτησης των δεδομένων όλων των επιπέδων που βρίσκονται στις διαστάσεις που επιλέγονται από το χρήστη. Αυτή η επιλογή όμως μπορεί να αποδειχτεί ανασταλτικός παράγοντας στην περίπτωση που ο όγκος των δεδομένων που είναι προς

ανάλυση από το χρήστη , είναι ένα μικρό ποσοστό των συνολικών δεδομένων. Στην περίπτωση αυτή η CPM δομή γίνεται ανούσια μεγάλη σε όγκο.

Για το λόγο αυτό ο χρήστης έχει τη δυνατότητα να επέμβει στον όγκο των αρχικά ανακτώμενων πληροφοριών.Πιο συγκεκριμένα έχει τις εξής επιλογές :

- Ανάκτηση των δεδομένων μέχρι και ένα επίπεδο πριν το κατώτερο και για τους δύο (παρουσιαστικούς)άξονες .Αποτελεί την default επιλογή μιας και αυτή είναι η φιλοσοφία του OLAP Lens
- Ανάκτηση των δεδομένων μόνο των 3 πρώτων επιπέδων. Στην περίπτωση αυτή ο χρήστης θα μπορεί να κάνει driil-down δύο φορές χρησιμοποιώντας δεδομένα των πολυκύβων
- Ανάκτηση των δεδομένων μόνο των 2 πρώτων επιπέδων. Στην περίπτωση αυτή ο χρήστης θα μπορεί να κάνει driil-down μόνο μία φορά χρησιμοποιώντας δεδομένα των πολυκύβων

Από τα παραπάνω προκύπτει όμως ένα θέμα σχετικά με την αντιμετώπιση των επομένων (των κατωτέρων από των αρχικά επιλεγμένων)επιπέδων. Δηλαδή το πρόβλημα του τι γίνεται αν ο χρήστης αποφασίσει να κατέβει σε κατώτερο επίπεδο από αυτό που οι αρχικές του επιλογές έχουν καθορίσει. Στην περίπτωση αυτή το εργαλείο εκτελεί ένα συγκεκριμένο MDX query του οποίου τα αποτελέσματα απεικονίζονται άμεσα στην οθόνη. Ενδιάμεσα γίνεται μία μοντελοποίηση στα γενικότερα πλαίσια του CPM αλλά όχι στην τυπική μορφή που περιγράψαμε. Σκοπός στο συγκεκριμένο σημείο είναι απλά η εξυπηρέτηση του χρήστη. Οπτικά βέβαια ο χρήστης δε διακρίνει καμία διαφορά.

Μία άλλη παράμετρος που μπορεί να καθορίσει ο χρήστης αφορά τον αόρατο(invisible) κατά CPM άξονα Sections. Υπάρχουν περιπτώσεις που ο χρήστης επιλέγει μία επιλογή φίλτρο χωρίς να σκοπεύει να την αλλάξει κατά τη διάρκεια της ανάλυσης. Στην περίπτωση αυτή θα ήταν αναίτιο και χρονοβόρο να προσθέσουμε στη CPM δομή μας μία επιπλέον διάσταση, αυξάνοντας εκθετικά το χρόνο επεξεργασίας. Μπορούμε δηλαδή να κάνουμε τον άξονα Sections να αποτελείται μόνο από ένα σημείο , το σημείο που έχει επιλέξει ο χρήστης και όχι και τα υπόλοιπα μέλη του επιπέδου. Η default βέβαια τιμή της παραμέτρου αυτή προσανατολίζεται στη γενική περίπτωση.

4.3.5 Απεικόνιση

Μετά τη δημιουργία της CPM δομής, τα δεδομένα είναι έτοιμα να οπτικοποιηθούν. Η οπτικοποίηση γίνεται στην παρούσα φάση σε desktop περιβάλλον. Για το σκοπό αυτό δημιουργείται πρακτικά μία κλασσική Windows απεικόνιση με τα εξής βασικά χαρακτηριστικά :

- Ένα βασικό menu με τις επιλογές File, Select, Options, Functions, Help
- Ένα toolbar το οποίο αποτελείται από κουμπιά για τις κυριότερες λειτουργίες του OLAP Lens όπως είναι η επιλογή βάσης και κύβου, η επιλογή των διαστάσεων, η επιλογή των φίλτρων, οι παράμετροι του συστήματος καθώς και κουμπιά για τις βασικές OLAP πράξεις, την αναλυτική κάθοδο και την αναλυτική άνοδο
- Μία επιφάνεια εργασίας όπου απεικονίζεται ο πίνακας
- Ένα statusbar στο κάτω μέρος της οθόνης στο οποίο εμφανίζονται χρήσιμες πληροφορίες για την κατάσταση της εφαρμογής

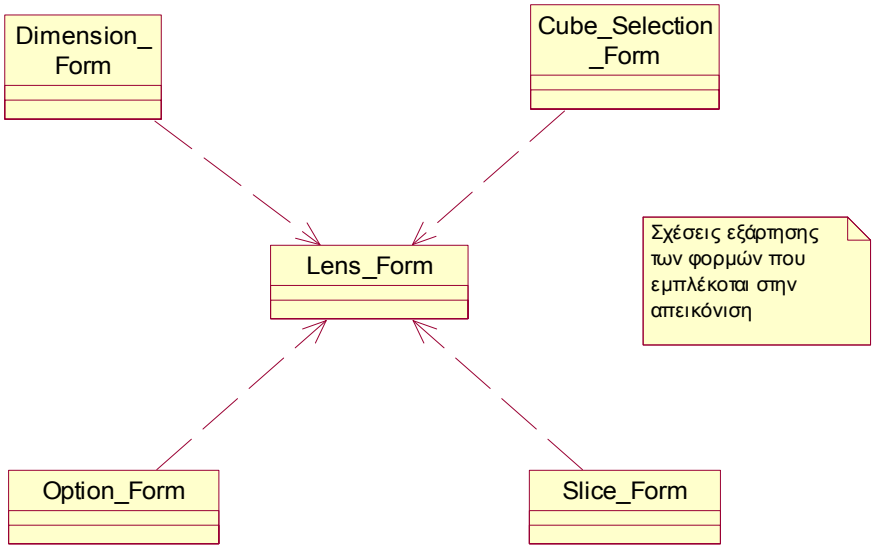
Ένα απλό δείγμα φαίνεται στο Σχ.4.14 :

Σχήμα 4.14 -- Αρχική οθόνη κατάσταση OLAP Lens

‘Οπως αναφέρθηκε στο προηγούμενι κεφάλαιο,σκοπός μας είναι και η χρήση της τεχνικής Table Lens. Στον OLAP Lens έχουμε εστίαση σε ένα συγκεκριμένο κελί το οποίο μεγενθύνεται είτε διότι επιλέγει κάτι τέτοιο ο χρήστης ή λόγω της εκτέλεσης κάποιου OLAP pattern πάνω στα απεικονιζόμενα δεδομένα.

Αναλυτική παρουσίαση οθονων από το OLAP Lens ακολουθεί στην αμέσως επόμενη παράγραφο. Στο σημείο αυτό θα γίνει ανάλυση των διαφόρων αντικειμένων που συνεργάζονται για την υλοποίηση της απεικόνισης. Στην πραγματικότητα υπάρχει μία βασική φόρμα (Windows Form) που εκτελεί το κύριο μέρος της υλοποίσης και μερικές βοηθητικές φόρμες. Πιο συγκεκριμένα έχουμε τις ακόλουθες φόρμες :

- **Lens_Form :** Πρόκειται για τη βασική φόρμα. Είναι αυτή που αναλαμβάνει να πάρει τα σωστά δεδομένα από τους multicubes που έχουν δημιουργηθεί, και να δημιουργήσει το grid της απεικόνισης.
- **`Cube_selection_Form:`** Είναι η φόρμα που επιτρέπει τη σύνδεση με τον Analysis Server, την επιλογή αρχικά της επιθυμητής βάσης και στη συνέχεια του επιθυμητού κύβου
- **`Dimension_Form:`** Επιτρέπει την επιλογή των διαστάσεων στους παρουσιαστικούς άξονες Rows και Columns
- **`Slice_Form:`** Επιτρέπει την επιλογή των διαστάσεων-φίλτρων
- **`Option_Form:`** Από τη φόρμα αυτή ο χρήστης επιλέγει τις παραμέτρους του συστήματος



Σχήμα 4.15 -- Φόρμες που συμμετέχουν στην απεικόνιση

Ας εστιάσουμε την προσοχή μας στην Lens_Form, μιας και οι υπόλοιπες φόρμες δεν παρουσιάζουν κάτι ιδιαίτερο. Η Lens_Form στηρίζεται σε μία λογική, που επιτρέπει εξοικονόμηση υπολογιστικών πράξεων. Πρακτικά δημιουργείται κατά την ανάλυση του κύβου από το χρήστη, μία λίστα από grids. Η δημιουργία κάθε grid επιμερίζεται στις εξής φάσεις

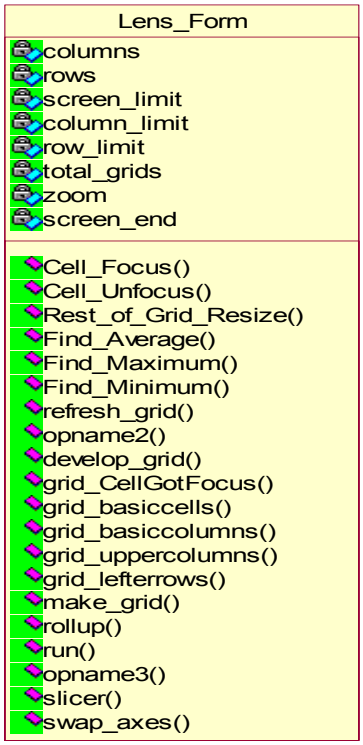
- Δημιουργία των γραμμών του πίνακα που υλοποιείται από τις διαδικασίες `grid_basicrows` και `grid_lefterrows`
- Δημιουργία των στηλών του πίνακα που υλοποιείται από τις διαδικασίες `grid_basiccolumns` και `grid_uppercolumns`

- Δημιουργία των βασικών κελιών του πίνακα που υλοποιείται από τη διαδικασία `grid_basiccells`

Το κάθε grid δεν είναι μία ανεξάρτητη οντότητα, μιας και η δημιουργία του ν-οστού grid σχετίζεται άμεσα με τη δημιουργία του (ν-1)-οστού. Όταν ο χρήστης κάνει drill-down αμέσως δημιουργείται και απεικονίζεται ένα νέο grid. Αυτό όμως δε γίνεται με αντικατάσταση του προηγουμένου. Το προηγούμενο και οι πόροι του συνεχίζουν να υπάρχουν στη μνήμη, απλά δεν απεικονίζονται. Και αυτό διότι αν και όταν ο χρήστης αποφασίσει να κάνει roll-up, θα πρέπει να εμφανιστεί ξανά αυτό το grid. Έτσι αν το είχαμε σβήσει θα έπρεπε να το ξαναδημιουργήσουμε, κάτι που θα επιβάρυνε την εφαρμογή μας.

Αυτό βέβαια ισχύει στη γενική περίπτωση. Υπάρχουν περιπτώσεις που η 'επέμβαση' του χρήστη σε κάποιο συγκεκριμένο επίπεδο των δεδομένων, δηλαδή σε κάποιο συγκεκριμένο grid, επιφέρει αλλαγές και στα προηγούμενα που είναι ήδη αποθηκευμένα στη μνήμη. Για παράδειγμα αν ο χρήστης επιλέξει περιστροφή των αξόνων και στη συνέχεια κάνει roll-up θα πρέπει η περιστροφή αυτή να ισχύσει και στο προηγούμενο grid. Αυτό επιτυγχάνεται με τη διαδικασία refresh_grid η οποία ελέγχει αν για κάποιο λόγο πρέπει να ανανεωθούν κάποια δεδομένα για τη δημιουργία ενός ήδη υπάρχοντος grid.

Η διαδικασία της ανανέωσης των δεδομένων εκείνων που καθορίζουν την μορφή του grid προϋποθέτει το σώσιμο των δεδομένων αυτών, δηλαδή επιβεβαιώνεται η αλληλεξάρτηση των διαφόρων απεικονίσεων.



Σχήμα 4.16 -- Φόρμα Lens_Form

Αξίζει να αναφερθεί επίσης ο πλήρης διαχωρισμός των αξόνων του CPM και των στηλών, γραμμών του πίνακα που εμφανίζουμε. Με άλλα λόγια δεν είναι υποχρεωτικό πως στις γραμμές του πίνακα θα υπάρχουν τα σημεία του άξονα Rows και πως στις στήλες του πίνακα θα υπάρχουν τα σημεία του άξονα Columns. Αν συνέβαινε κάτι τέτοιο είτε δε θα ήταν δυνατή η περιστροφή των αξόνων ή θα έπρεπε η περιστροφή να προκαλούσε αναδιάταξη ολόκληρης της CPM δομής κάτι που δε θα ήταν καθόλου αποδοτικό. Η περιστροφή των αξόνων απλά αλλάζει τον τρόπο με τον οποίο η φόρμα μας ΄βλέπει΄ τα δεδομένα του μοντέλου παρουσίασης.

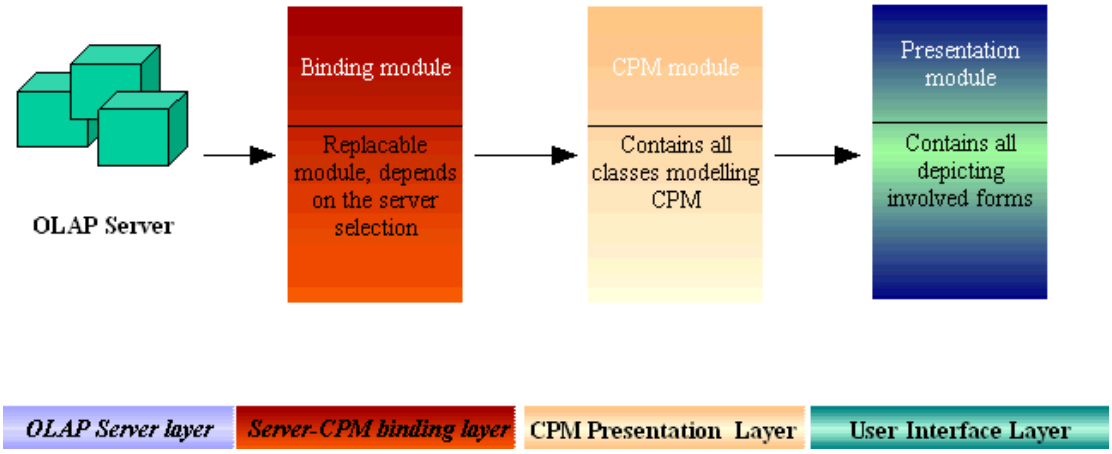
4.3.6 Επεκτασιμότητα εφαρμογής

Ο OLAP Lens λειτουργεί στην παρούσα υλοποίηση χρησιμοποιώντας τον Analysis Server της Microsoft. Το γεγονός πως είναι δυνατή η επιλογή οποιουδήποτε κύβου στον server αυτό είναι ικανοποιητικό. Τί γίνεται όμως αν θελήσουμε να χρησιμοποιήσουμε έναν διαφορετικό OLAP Server, όπως για παράδειγμα τον server της Oracle; Με άλλα λόγια πόσο εύκολη είναι η σύνδεση των δύο στρωμάτων του CPM;

Ο τρόπος με τον οποίο έχουμε σχεδιάσει τον OLAP Lens μας επιτρέπει μία εύκολη σχετικά τροποποίηση. Παρατηρούμε τα εξής :

- Ως προς το κομμάτι της απεικόνισης των δεδομένων στην οθόνη, και το οποίο αναλαμβάνουν οι φόρμες που φαίνονται στο Σχ.4.15 δεν απαιτείται καμία αλλαγή
- Ως προς την υλοποιούμενη CPM δομή, δεν υπάρχουν επίσης αλλαγές. Αυτό που πρέπει να αλλάξει είναι ο τρόπος που η δομή αυτή παίρνει τα δεδομένα. Όπως δείξαμε στην παρούσα υλοποίηση αυτό επιτυγχάνεται μέσω της κλάσης MDX_Query δηλαδή με την εκτέλεση διαδοχικών MDX ερωτήσεων. Όταν υπάρχει διαφορετικός OLAP Server αυτό φυσικά θα πρέπει να αλλάξει. Τον τρόπο με τον οποίο η εφαρμογή θα 'ζητά' και θα ανακτά δεδομένα από τον server, τον καθορίζει ο ίδιος ο server.(Στην ιδανική περίπτωση που υποστηρίζονται MDX ερωτήσεις δεν απαιτείται καμία αλλαγή)

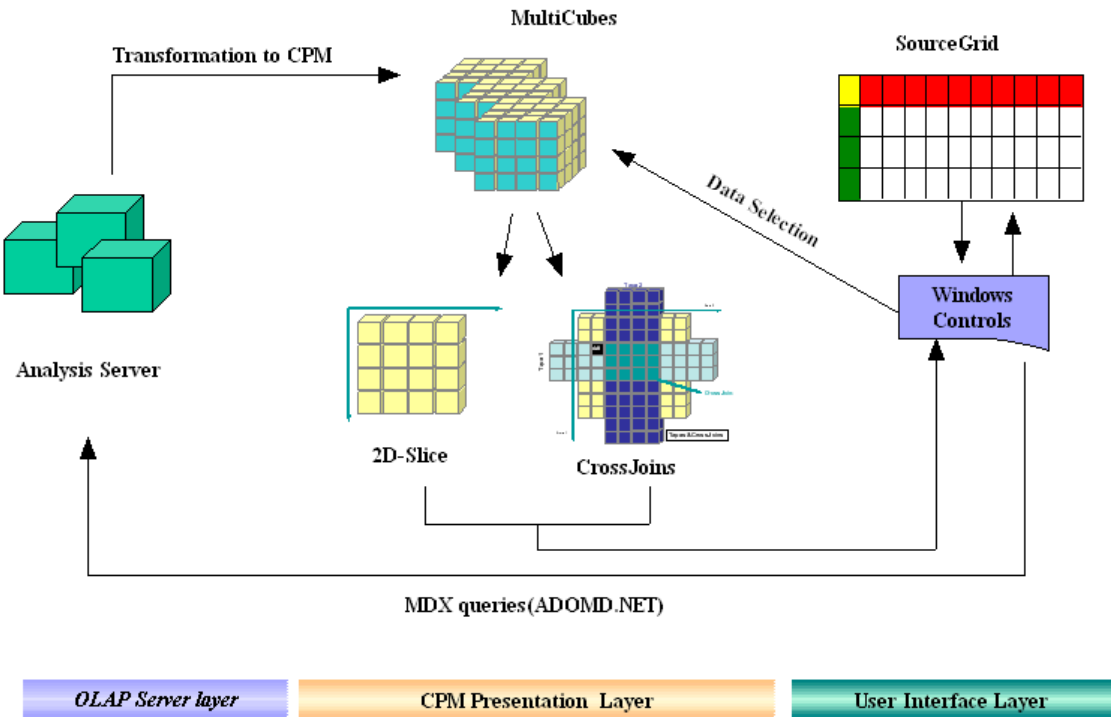
Τα παραπάνω φαίνονται ξεκάθαρα και στο παρακάτω σχήμα. Τα modules της CPM δομής καθλως και των φορμών που υλοποιούν την απεικόνιση δεν χρειάζεται να τροποποιηθούν. Αντίθετα το module της σύνδεσης της CPM δομής με τον server πρέπει να τροποποιηθεί. Όμως αφού αυτό αντιπροσωπεύεται μόνο από την κλάση MDX_Query, η τροποποίηση κρίνεται απλή και η επεκτασιμότητα ευέλικτη.



Σχήμα 4.17 -- OLAP Lens software architecture

4.4 Παρουσίαση και ανάλυση υλοποίησης

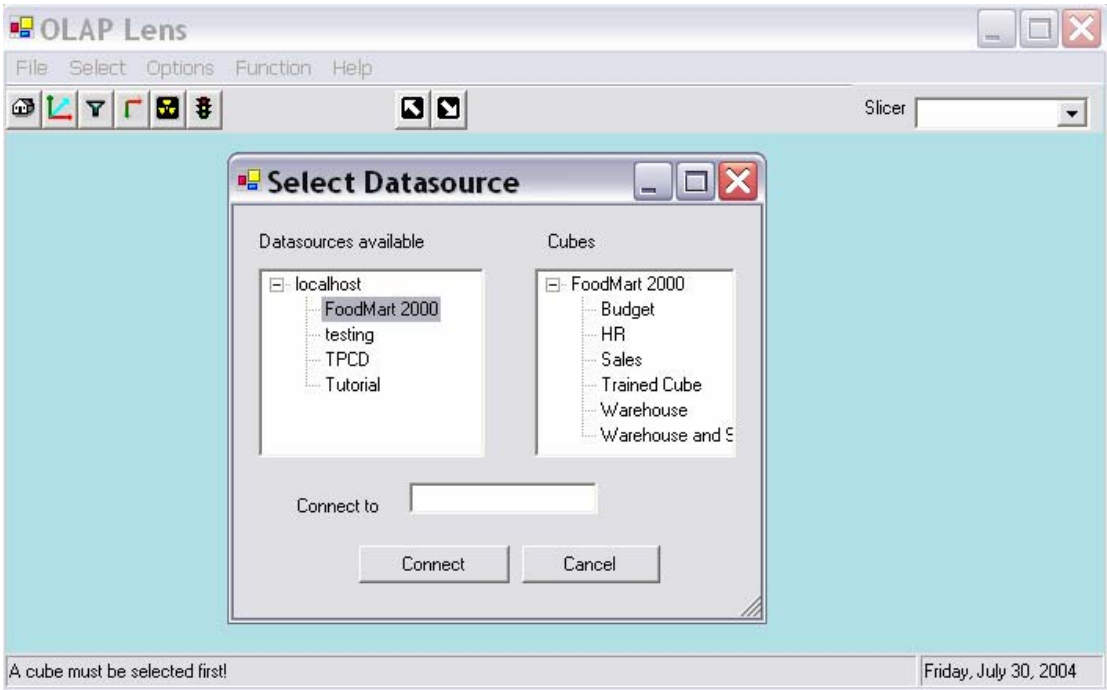
Στην παράγραφο τούτη θα προσπαθήσουμε να γίνουμε πιο παραστατικοί, παρουσιάζοντας στιγμυότυπα από τις βασικότερες λειτουργίες του OLAP Lens, σχολιάζοντας παράλληλα τον τρόπο με τον οποίο αυτές υλοποιούνται .



Σχήμα 4. 18 -- Διάγραμμα ροής OLAP Lens

4.4.1 Οθόνη επιλογής πηγής δεδομένων

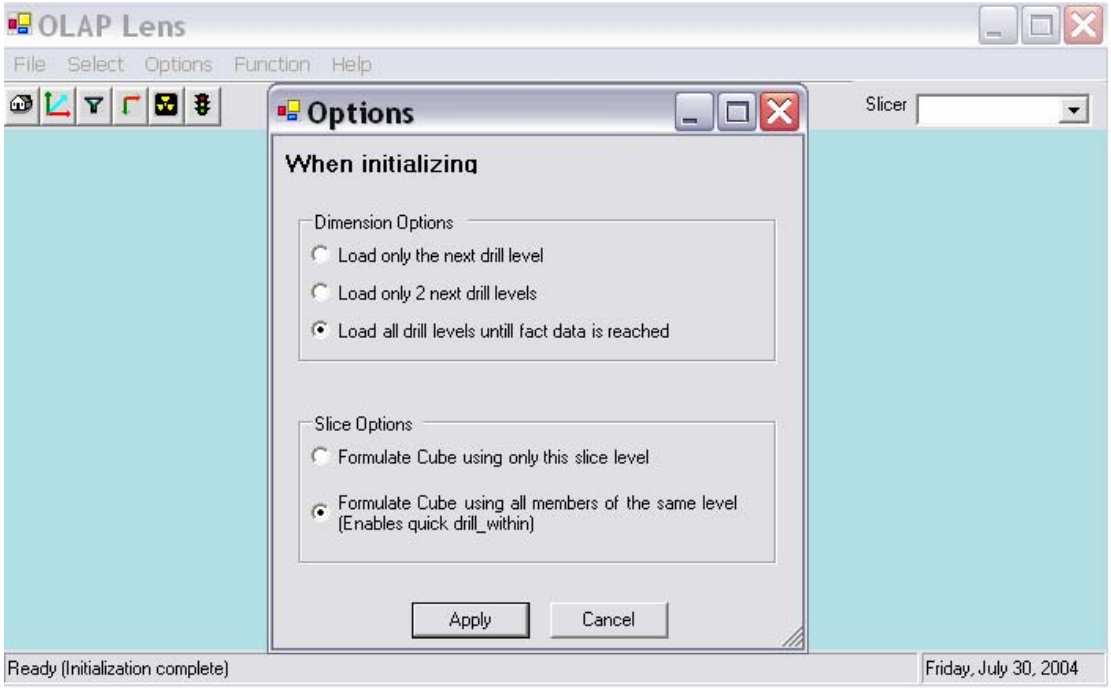
Ο χρήστης πρέπει πριν από όλα να επιλέξει την πηγή δεδομένων με την οποία θέλει να συνδεθεί. Αφού πατήσει το κατάλληλο κουμπί από το toolbar εμφανίζεται η φόρμα `Cube_selection_Form` οπότε ο χρήστης βλέπει στην οθόνη του κάτι σαν το πρακάτω σχήμα :



Σχήμα 4.19 -- Οθόνη σύνδεσης με την πηγή δεδομένων

4.4.2 Οθόνη επιλογής αρχικών παραμέτρων

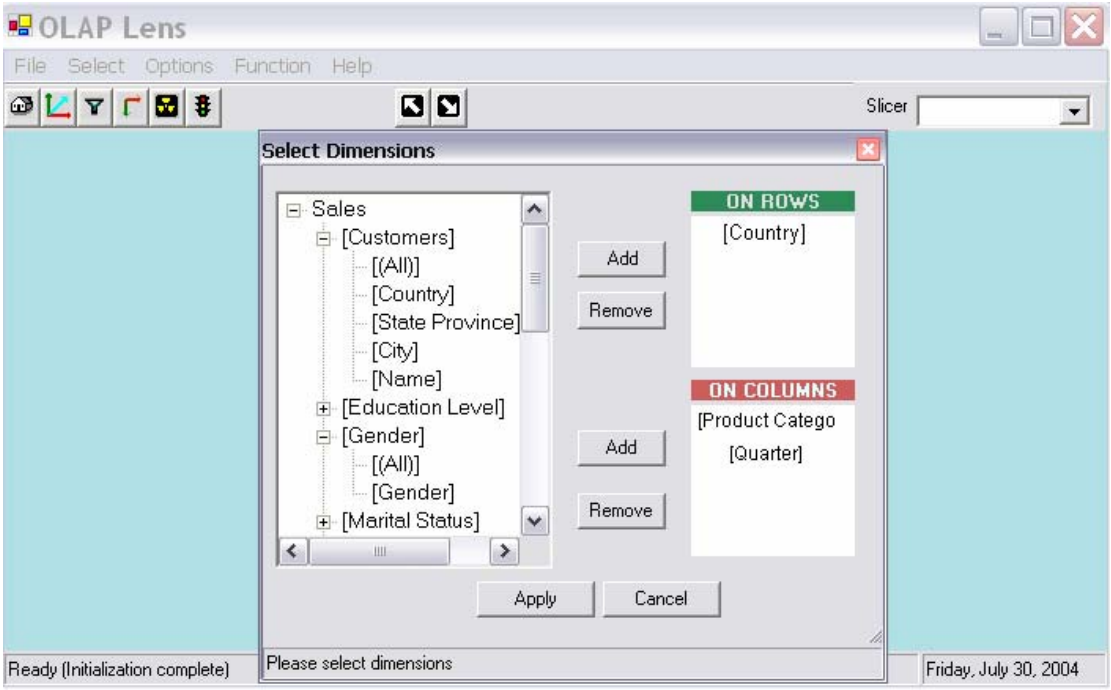
Σε οποιοδήποτε στάδιο πριν την εκκίνηση της απεικόνισης, δηλαδή πρίν πατηθεί το κουμπί Run, ο χρήστης έχει τη δυνατότητα να τροποποιήσει τις αρχικές παραμέτρους του συστήματος. Αφού πατήσει το κατάλληλο κουμπί από το toolbar εμφανίζεται η φόρμα Option_`Form` οπότε εμφανίζεται η οθόνη που απεικονίζεται στο Σχ.4.20



Σχήμα 4.20 -- Οθόνη επιλογής αρχικών παραμέτρων

4.4.3 Οθόνη επιλογής διαστάσεων

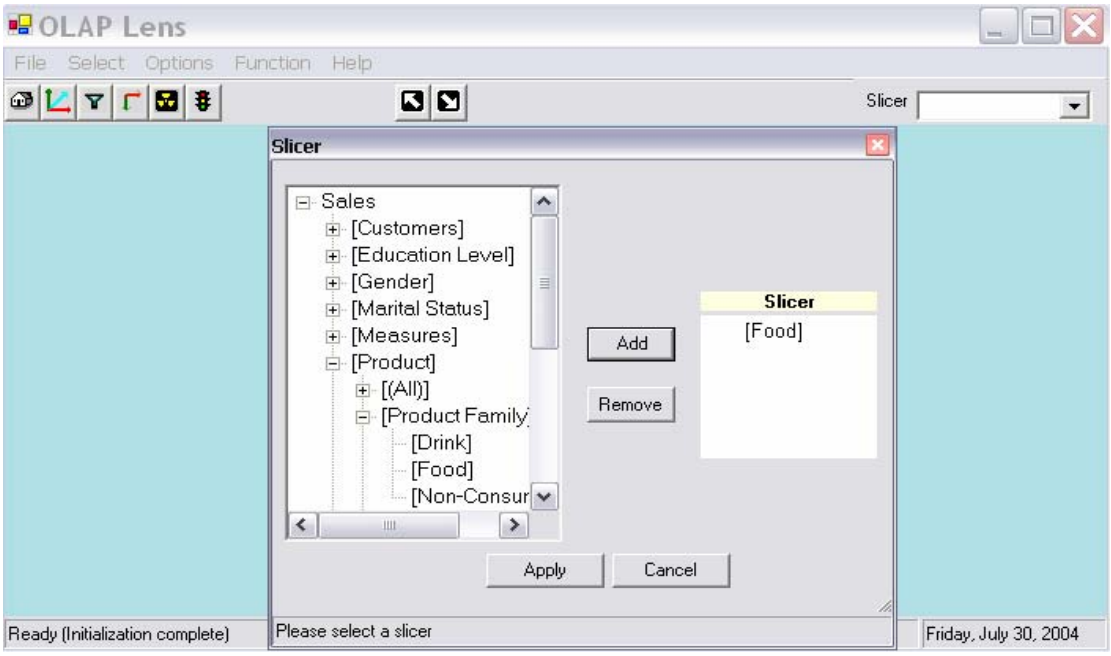
Η επιλογή των διαστάσεων γίνεται με τη βοήθεια της φόρμας Dimension_`Form`. Επειδή ο χρήστης κατά βάση χρησιμοποιεί συχνά την επιλογή αυτή, θα πρέπει να είναι σχεδιασμένη απλά και αποδοτικά. Η απεικόνιση των διαστάσεων γίνεται με τη βοήθεια μίας δενδρικής δομής που ενημερώνεται με βάση τις πληροφορίες της κλάσης Cube_Schema. Με τη χρήση των δύο κουμπιών πρόσθεσης και αφαίρεσης η διαδικασία γίνεται απλά. Τα κατώτερα επίπεδα των διαστάσεων δεν είναι δυνατόν να επιλεγούν, όπως καθορίστηκε στις απαιτήσεις χρήστη. Επίσης στην περίπτωση που επιλεγούν επίπεδα της ίδιας διάστασης είτε στον ίδιο ή σε διαφορετικούς άξονες, το statusbar που βρίσκεται στο κάτω μέρος της φόρμας ενημερώνει τον χρήστη ώστε να διορθώσει τις επιλογές του. Ο έλεγχος αυτός γίνεται στο σημείο αυτό χωρίς να γίνεται η επικοινωνία με τη βάση και να γίνεται εκ των υστέρων αντιληπτό το λάθος.



Σχήμα 4.21 -- Οθόνη επιλογής διαστάσεων

4.4.4 Οθόνη επιλογής φίλτρου

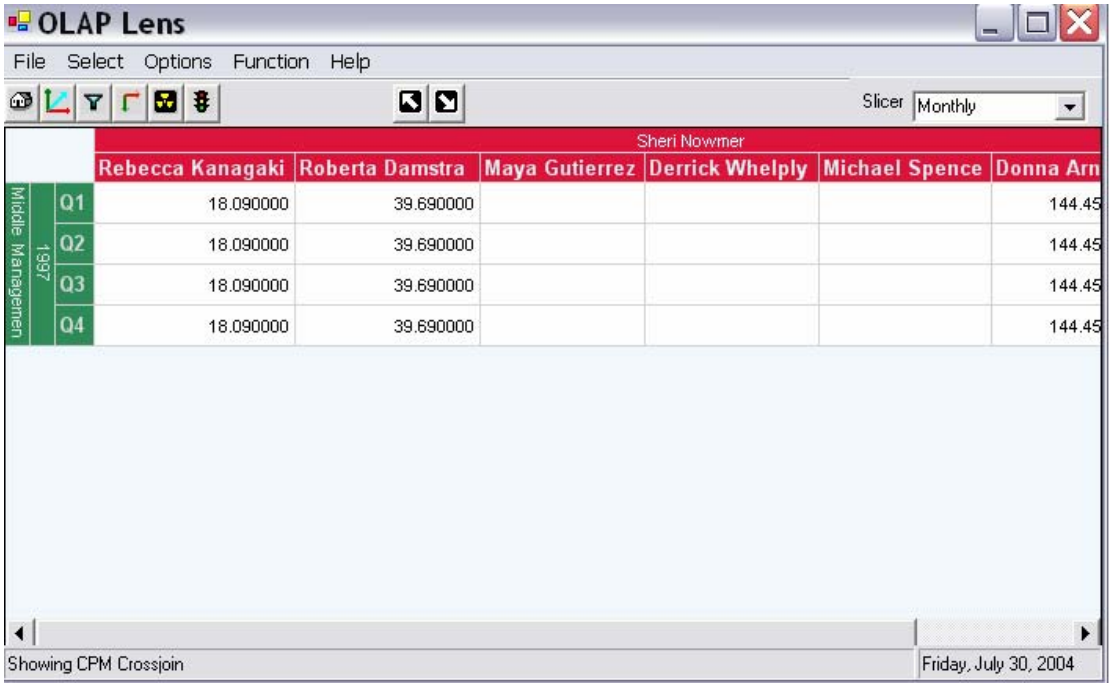
Ουσιαστκά η λειτουργία της φόρμας Slice_Form είναι παρόμοια με αυτήν της Dimension_Form. Μόνο που στην περίπτωση αυτή ο χρήστης δεν μπορεί να επιλέξει επίπεδο διάστασης, αλλά μέλος κάποιου επιπέδου. Η επιλογή του φίλτρου θα εμφανίζεται σε ένα combobox που έχει δημιουργηθεί για το σκοπό αυτό στο πανω δεξιά μερος της οθόνης



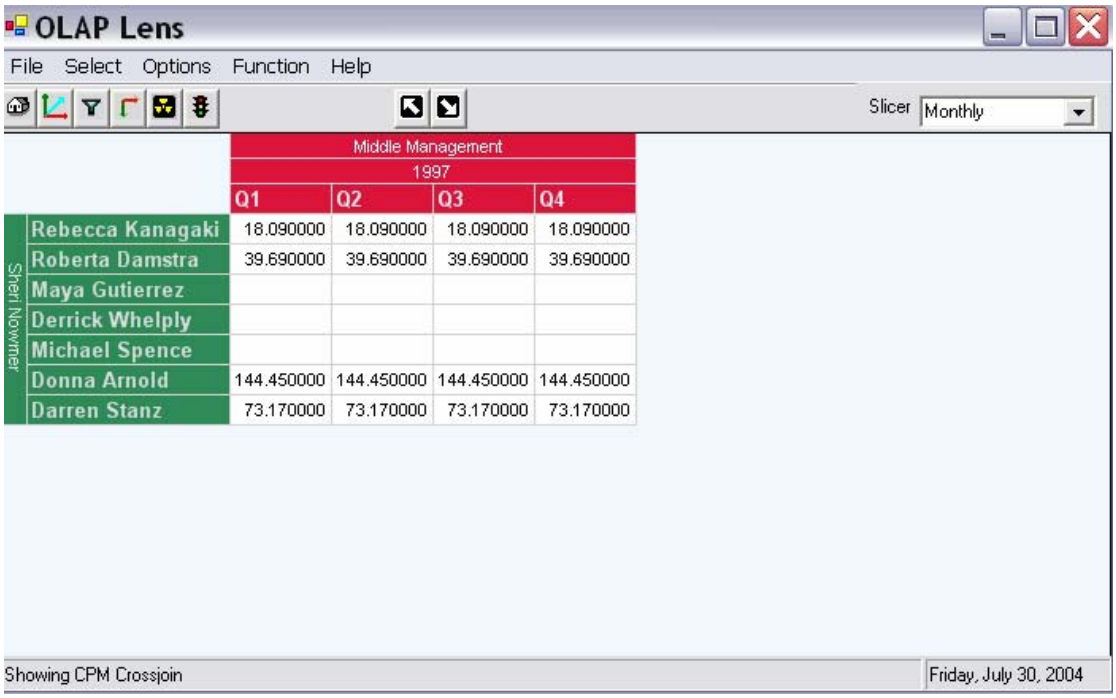
Σχήμα 4.22 -- Οθόνη επιλογής φίλτρου

4.4.5 Οθόνη αλλαγής αξόνων

Οι δύο παρακάτω οθόνες δείχνουν ένα τυπικό παράδειγμα αναδιάταξης των αξόνων Rows και Columns. Όπως περιγράφηκε σε προηγούμενη ενότητα η διαδικασία γίνεται απλά και γρήγορα, αλλάζοντας όχι ολόκληρη τη δομή, αλλά τον τρόπο με τον οποίο η κυρίως φόρμα Lens_Form 'βλέπει' τους άξονες του CPM. Ο χρήστης δε χρειάζεται να περιμένει καθόλου.



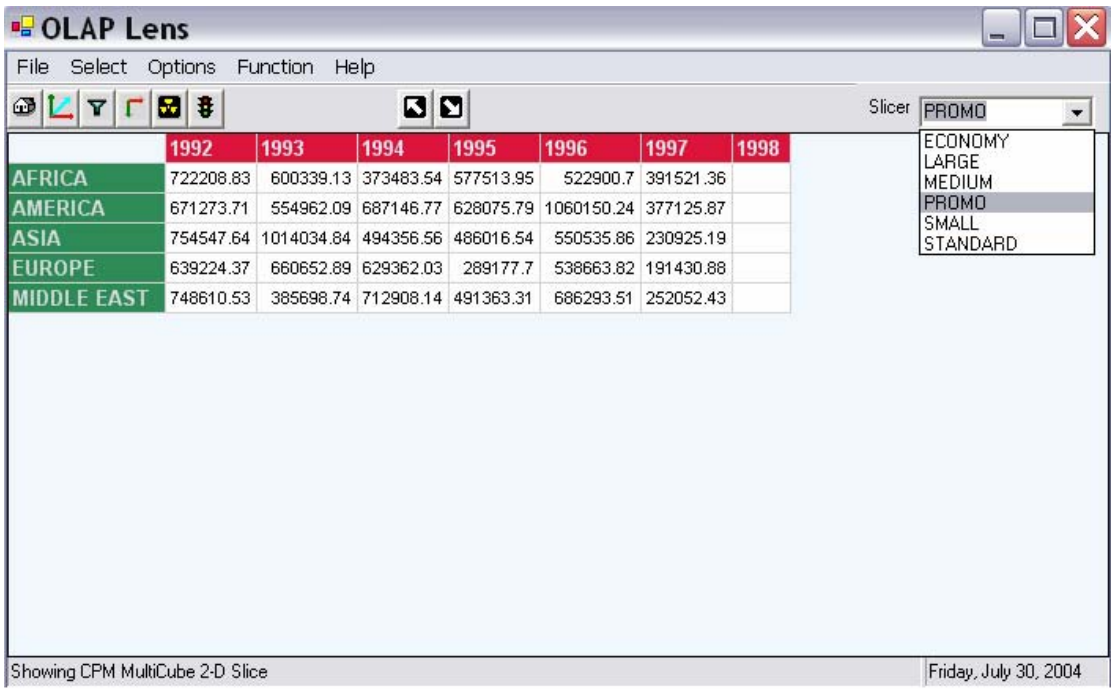
Σχήμα 4.23 -- Οθόνη αλλαγής αξόνων(α)



Σχήμα 4.23 -- Οθόνη αλλαγής αξόνων(β)

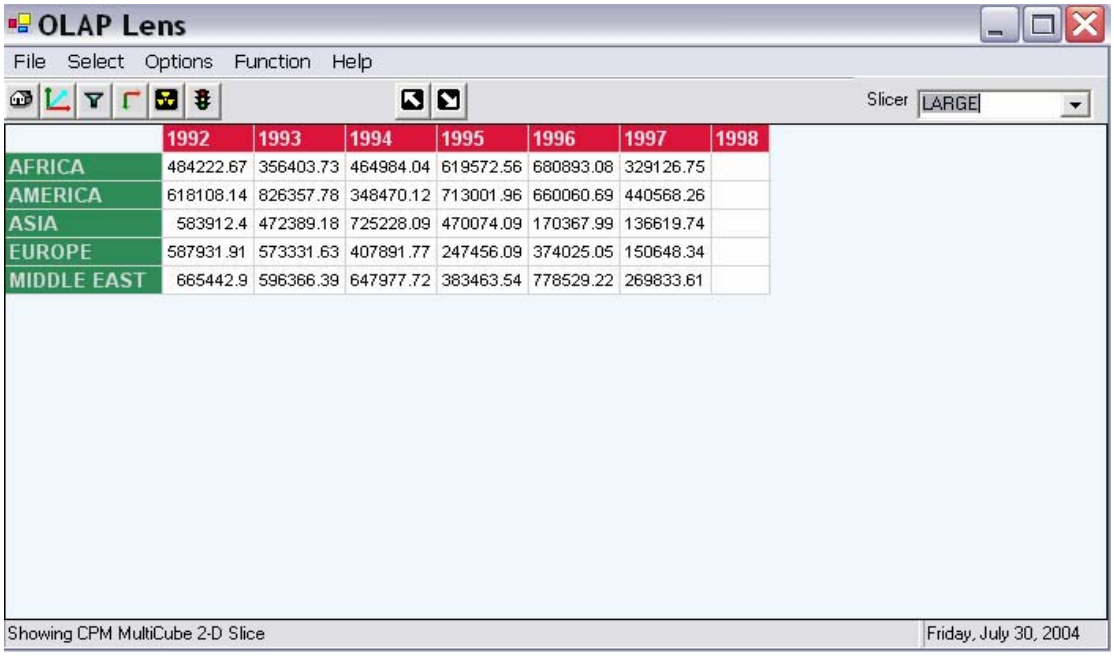
4.4.6 Οθόνη drill within

‘Οταν ο χρήστης συμπεριλάβει στο προς παρουσίαση σχήμα κάποια τιμή φίλτρου, τότε στην default περίπτωση ανακτούνται και αποθηκεύονται στη CPM δομή μας, και οι τιμές για όλες τα μέλη που βρίσκονται στο ίδιο επίπεδο με το επιλεγμένο μέλος(για την ακρίβεια έχουν και τον ίδιο πατέρα). Με τον τρόπο αυτό το drill-within γίνεται κατά τη διάρκεια της ανάλυσης πολύ γρήγορα μιας και το μόνο που απαιτείται είναι η επιλογή ενός διαφορετικού 2D-slice ή crossjoin. Γραφικά η δυνατότητα αυτή αναπαριστάται με ένα combobox



	1992	1993	1994	1995	1996	1997	1998
AFRICA	722208.83	600339.13	373483.54	577513.95	522900.7	391521.36	
AMERICA	671273.71	554962.09	687146.77	628075.79	1060150.24	377125.87	
ASIA	754547.64	1014034.84	494356.56	486016.54	550535.86	230925.19	
EUROPE	639224.37	660652.89	629362.03	289177.7	538663.82	191430.88	
MIDDLE EAST	748610.53	385698.74	712908.14	491363.31	686293.51	252052.43	

Σχήμα 4. 24 -- Οθόνη drill-within (α)

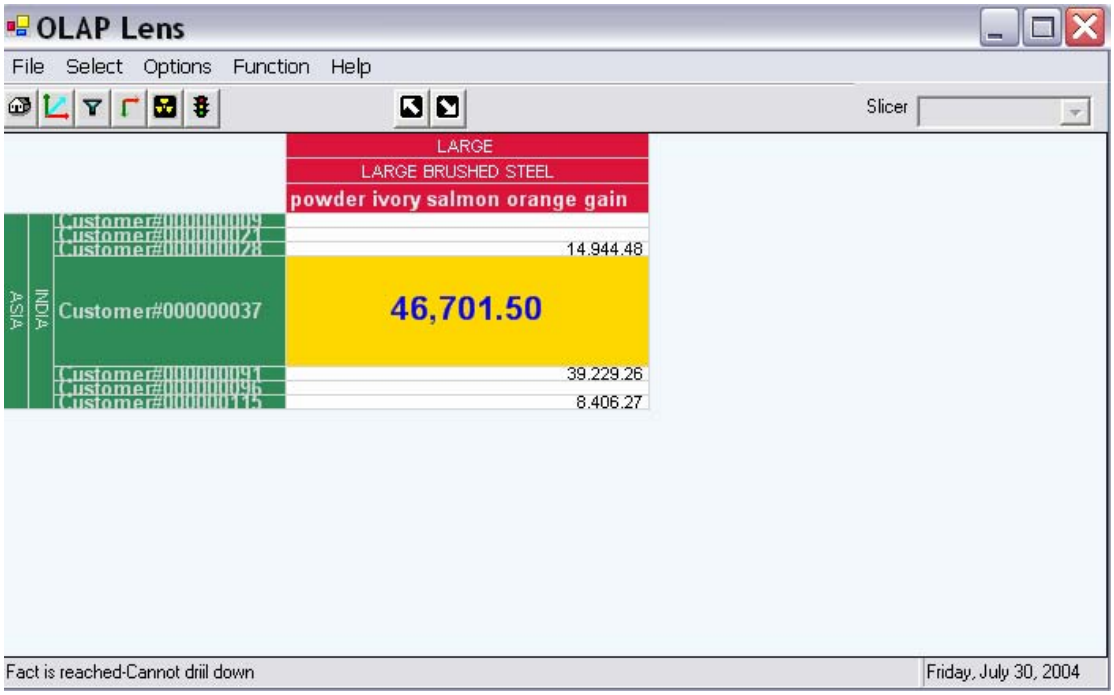


	1992	1993	1994	1995	1996	1997	1998
AFRICA	484222.67	356403.73	464984.04	619572.56	680893.08	329126.75	
AMERICA	618108.14	826357.78	348470.12	713001.96	660060.69	440568.26	
ASIA	583912.4	472389.18	725228.09	470074.09	170367.99	136619.74	
EUROPE	587931.91	573331.63	407891.77	247456.09	374025.05	150648.34	
MIDDLE EAST	665442.9	596366.39	647977.72	383463.54	778529.22	269833.61	

Σχήμα 4.24 -- Οθόνη drill-within (β)

4.4.7 Οθόνη κατώτερου επιπέδου της ιεραρχίας

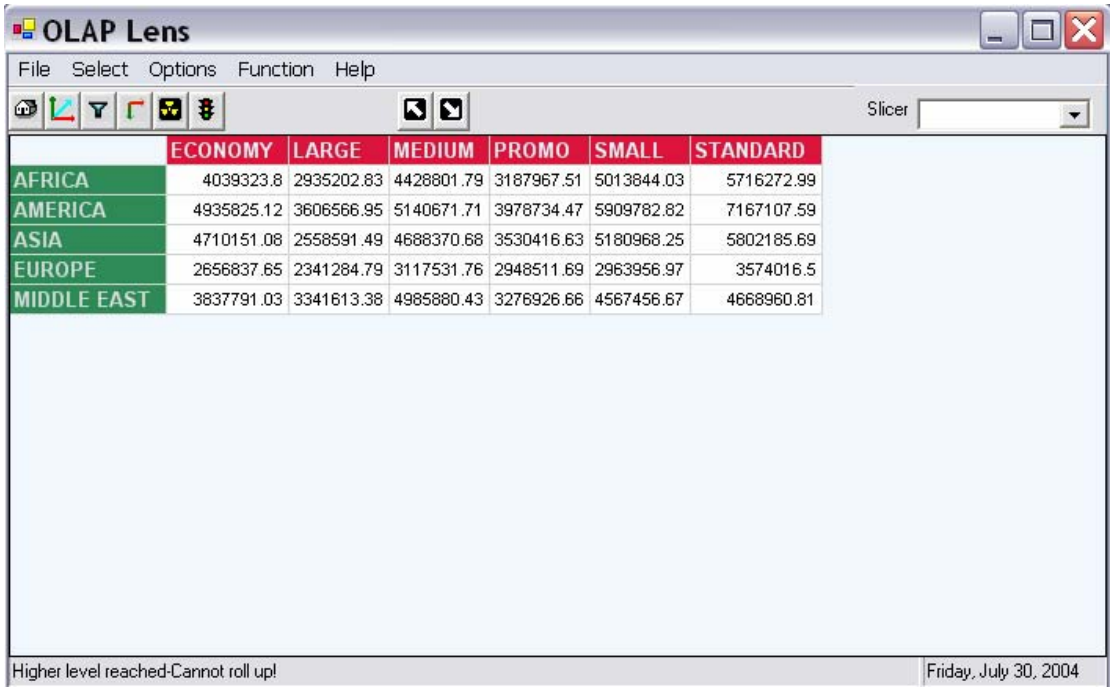
Όταν ο χρήστης φτάσει στο κατώτερο επίπεδο της ιεραρχίας μίας (ή και των δύο) διαστάσεων δεν είναι δυνατή η αναλυτική κάθοδος. Στην περίπτωση που ο χρήστης προσπαθήσει να κάνει κάτι τέτοιο, ειδοποιείται με κατάλληλο μήνυμα στο statusbar που βρίσκεται στο κάτω νέρος της φόρμας. Ένα παράδειγμα απεικονίζεται στο Σχ.4.25



Σχήμα 4. 25-- Οθόνη κατώτερου επιπέδου της ιεραρχίας

4.4.8 Οθόνη ανώτερου επιπέδου της ιεραρχίας

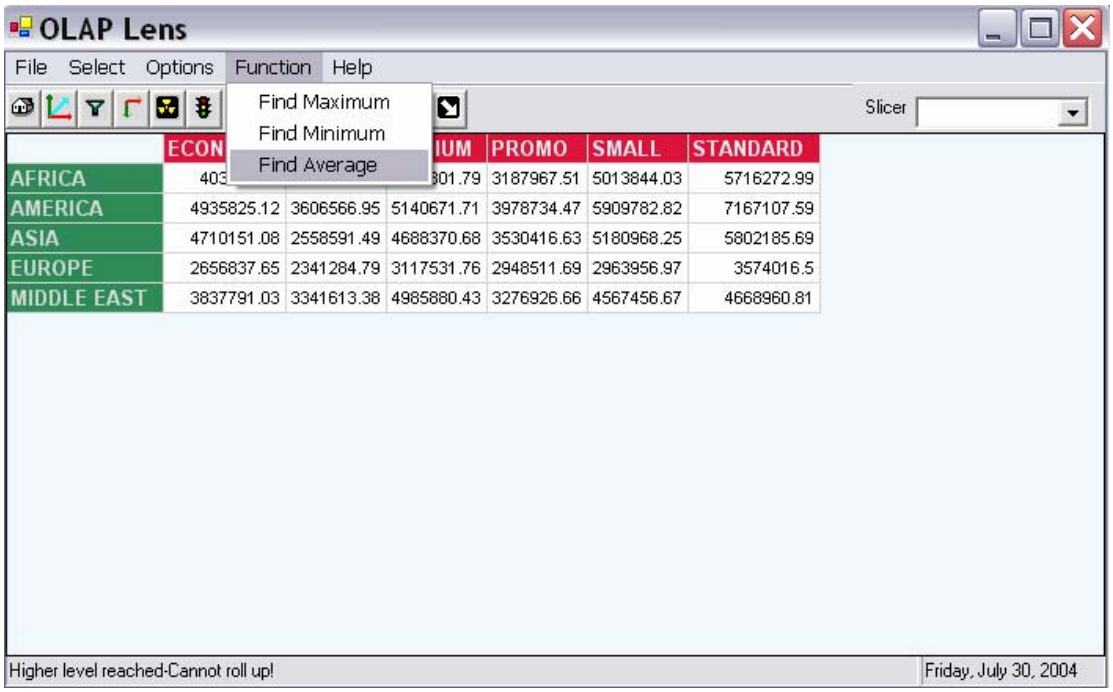
Τα ίδια με αμέσως προηγουμένως ισχύουν όταν ο χρήστης φτάσει στο ανώτερο επίπεδο της ιεραρχίας των δύο διαστάσεων. Η αναλυτική άνοδος δεν είναι πλέον δυνατή:



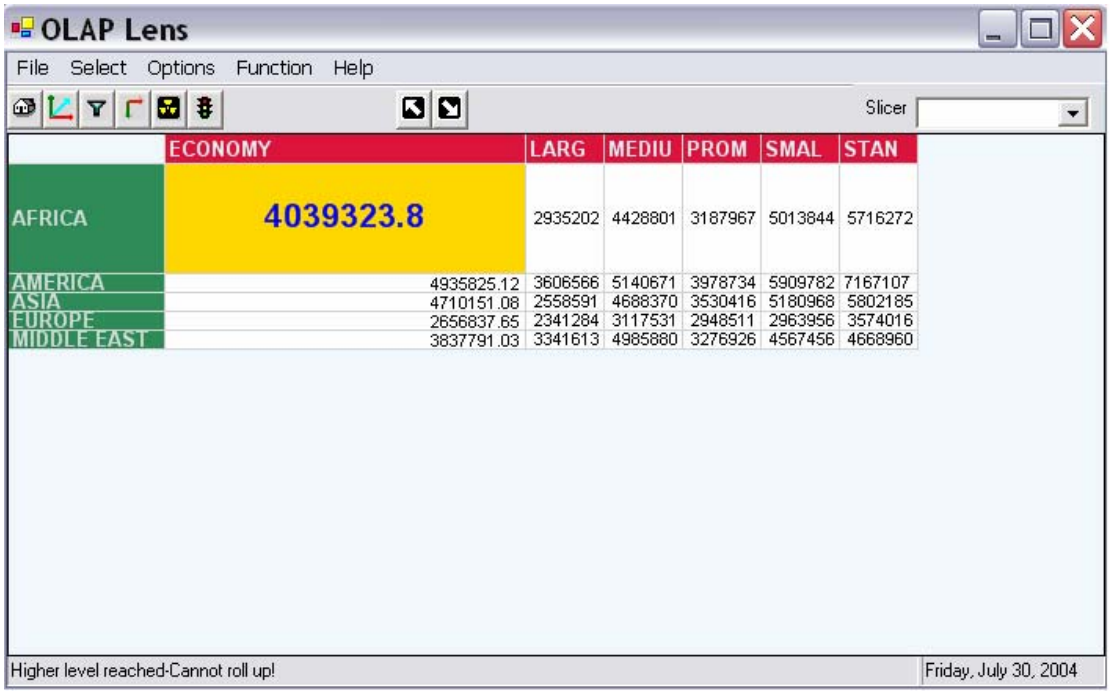
Σχήμα 4.26 -- Οθόνη ανώτερου επιπέδου της ιεραρχίας

4.4.9 Οθόνη επιλογής συνάρτησης

Ένα ιδιαίτερο χαρακτηριστικό είναι η δυνατότητα εκτέλεσης κάποιων συναρτήσεων (OLAP patterns) πάνω στα δεδομένα. Στα δύο σχήματα που ακολουθούν βλέπουμε τον τρόπο με τον οποίο ο χρήστης μπορεί να βρει την μέση τιμή των δεδομένων που απεικονίζονται. Για την ακρίβεια γίνεται focus στο κελί εκείνο του οποίου η τιμή απέχει λιγότερο από το μέσο όρο.



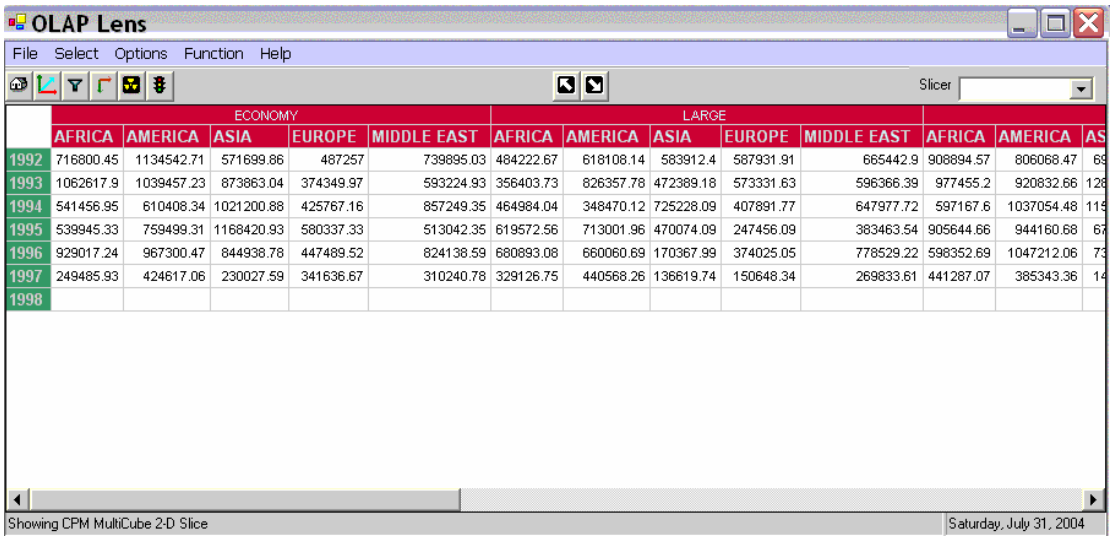
Σχήμα 4.27 : Οθόνη επιλογής συνάρτησης(α)



Σχήμα 4.27 -- Οθόνη επιλογής συνάρτησης(β)

4.5 Έλεγχος ορθότητας

Ο έλεγχος του συστήματος πραγματοποιήθηκε με χρήση σεναρίων λειτουργίας. Έγινε προσπάθεια να καλυφθούν όσο το δυνατόν περισσότερα σενάρια Ως προς την ορθότητα των δεδομένων που απεικονίζονται ήταν δυνατή η σύγκριση με ένα εργαλείο που περιέχεται στα Analysis Services, που ονομάζεται MDX Sample Application. Το εργαλείο αυτό αν και ελάχιστα μοιάζει με τον OLAP Lens, βοήθησε σημαντικά στον έλεγχο. Το MDX Sample Application απεικονίζει τα αποτελέσματα από MDX queries που πληκτρολογεί ο χρήστης. Στο πλαίσιο αυτό χρησιμοποιήσαμε τον κύβο Sales από μία αποθήκη δεδομένων που ονομάζεται Foodmart2000. Σε κάθε στάδιο της ανάλυσης γινόταν αντιπαραβολή με τα αποτελέσματα του MDX Sample Application. Αυτό γινόταν εφικτό με την εκτέλεση του αντίστοιχου MDX query. Για παράδειγμα έστω το παρακάτω στιγμυότυπο από τον OLAP Lens



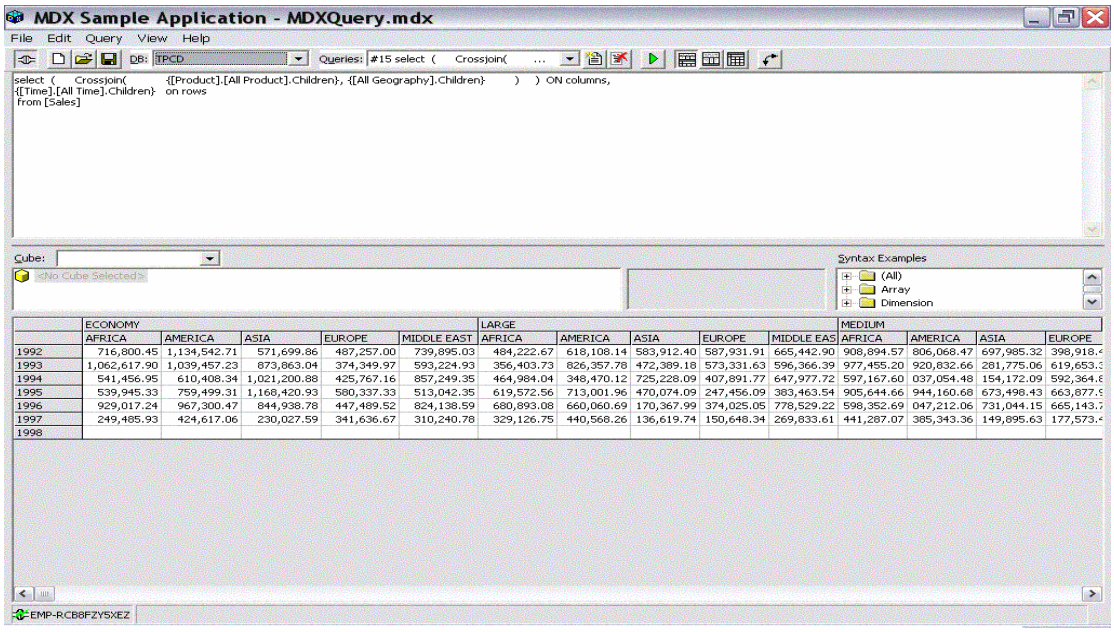
	ECONOMY					LARGE							
	AFRICA	AMERICA	ASIA	EUROPE	MIDDLE EAST	AFRICA	AMERICA	ASIA	EUROPE	MIDDLE EAST	AFRICA	AMERICA	AS
1992	716800.45	1134542.71	571699.86	487257	739895.03	484222.67	618108.14	583912.4	587931.91	665442.9	908894.57	806068.47	69
1993	1062617.9	1039457.23	873863.04	374349.97	593224.93	356403.73	826357.78	472389.18	573331.63	596366.39	977455.2	920832.66	128
1994	541456.95	610408.34	1021200.88	425767.16	857249.35	464984.04	348470.12	725228.09	407891.77	647977.72	597167.6	1037054.48	115
1995	539945.33	759499.31	1168420.93	580337.33	513042.35	619572.56	713001.96	470074.09	247456.09	383463.54	905644.66	944160.68	67
1996	929017.24	967300.47	844938.78	447489.52	824138.59	680893.08	660060.69	170367.99	374025.05	778529.22	598352.69	1047212.06	73
1997	249485.93	424617.06	230027.59	341636.67	310240.78	329126.75	440568.26	136619.74	150648.34	269833.61	441287.07	385343.36	14
1998													

Σχήμα 4.28 -- Στιγμυότυπο OLAP Lens

Στο απεικονιζόμενο σχήμα διακρίνουμε 3 λογικές διαστάσεις, 2 στον άξονα Columns και έναν στον άξονα Rows. Για τον έλεγχο της ορθότητας των δεδομένων πληκτρολογούμε στο MDX Sample Application το ακόλουθο MDX query :

```
select ( Crossjoin( {[Product].[All Product].Children},
{[All Geography].Children} ) ) ON columns,
{[Time].[All Time].Children} on rows
 from [Sales]
```

οπότε στην οθόνη μας βλέπουμε τα εξής (επαληθευόμενα) αποτελέσματα :



	ECONOMY					LARGE					MEDIUM			
	AFRICA	AMERICA	ASIA	EUROPE	MIDDLE EAST	AFRICA	AMERICA	ASIA	EUROPE	MIDDLE EAS	AFRICA	AMERICA	ASIA	EUROPE
1992	716,800.45	1,134,542.71	571,699.86	487,257.00	739,895.03	484,222.67	618,108.14	583,912.40	587,931.91	665,442.90	908,894.57	806,068.47	697,985.32	398,918.
1993	1,062,617.90	1,039,457.23	873,863.04	374,349.97	593,224.93	356,403.73	826,357.78	472,389.18	573,331.63	596,366.39	977,455.20	920,832.66	281,775.06	619,653.
1994	541,456.95	610,408.34	1,021,200.88	425,767.16	857,249.35	464,984.04	348,470.12	725,228.09	407,891.77	647,977.72	597,167.60	037,054.48	154,172.09	592,364.
1995	539,945.33	759,499.31	1,168,420.93	580,337.33	513,042.35	619,572.56	713,001.96	470,074.09	247,456.09	383,463.54	905,644.66	944,160.68	673,498.43	663,877.
1996	929,017.24	967,300.47	844,938.78	447,489.52	824,138.59	680,893.08	660,060.69	170,367.99	374,025.05	778,529.22	598,352.69	047,212.06	731,044.15	665,143.
1997	249,485.93	424,617.06	230,027.59	341,636.67	310,240.78	329,126.75	440,568.26	136,619.74	150,648.34	269,833.61	441,287.07	385,343.36	149,895.63	177,573.
1998														

Σχήμα 4.29 -- Στιγμυότυπο MDX Sample Application

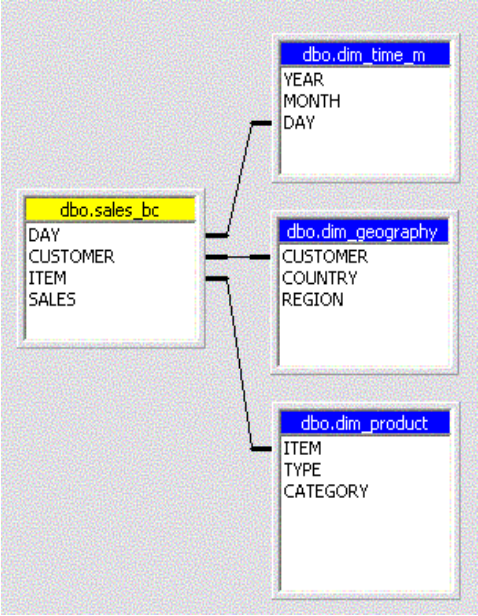
Η διαδικασία αυτή είναι ενδεικτική. Σε καμία περίπτωση δεν εξασφαλίζει απόλυτα την ορθότητα του OLAP Lens. Το κύριο μέρος του ελέγχου πραγματοποιήθηκε μέσω πολλών και διαφορετικών σεναρίων, και μάλιστα με τη βοήθεια κάποιων συναδέλφων που προθυμοποιήθηκαν να λάβουν μέρος στη διαδικασία αυτή. Ως γνωστόν ο δημιουργός μίας εφαρμογής δεν είναι ο κατάλληλος για τον έλεγχο του.

4.6 Μέτρηση απόδοσης

Στην παράγραφο τούτη θα παρουσιαστούν κάποια πειραματικά αποτελέσματα που αφορούν την απόδοση του OLAP Lens. Μετά από την εκτέλεση κάποιων σεναρίων καταφέραμε να βγάλουμε μερικά χρήσιμα συμπεράσματα.

Βασικός παράγοντας στην απόδοση του συστήματος είναι βέβαια ο όγκος της πληροφορίας, δηλαδή το σχήμα και το μέγεθος του κύβου προς απεικόνιση..

Για αρχή πειραματιστήκαμε με τον κύβο Sales που δημιουργήσαμε με τη βοήθεια των Analysis Services. Ως πηγή δεδομένων χρησιμοποιήθηκε ένα υποσύνολο του TPCD, ενός γνωστού εμπορικού benchmark. Ο κύβος του οποιου το σχήμα απεικονίζεται στο διπλανό σχήμα αποτελείται από 3 διαστάσεις, τις Time,Geography, Product και η κάθεμία από αυτές από 3 επίπεδα. Κάναμε μερικές μετρήσεις χρησιμοποιώντας τον απλό αυτό κύβο του οποίου το μέγεθος δεν ξεπερνάει τα 10 Mb. Στο σημείο αυτό όμως θα πρέπει να εξηγηθεί το τι ακριβώς θα μετρηθεί στην εκτέλεση των σεναρίων.



Σχήμα 4.30 – TPCD κύβος

Στο πλαίσιο αυτό χωρίσαμε την εκτέλεση μίας ανάλυσης στις εξής επιμέρους φάσεις :

1. Φάση αρχικοποίσης δεδομένων, δηλαδή η φάση κατά την οποία η κλάση Cube_Schema συλλέγει όλες τις πληροφορίες σχετικά με το σχήμα του κύβου. Ο χρόνος που καταναλώνεται στη φάση αυτή δεν

2. Φάση υλοποίησης CPM δομής. Είναι η πιο σημαντική φάση. Εξαρτάται άμεσα από τις επιλογές του χρήστη, τόσο αναφορικά με τις διαστάσεις που επιλέγει όσο και από τις παραμέτρους του συστήματος
3. Φάση απεικόνισης. Αντιπροσωπεύει το χρόνο που απαιτείται έτσι ώστε η κύρια φόρμα Lens_Form να απεικονίσει τον πίνακα.

Θα μπορέσουμε να διαπιστώσουμε επίσης τη χρησιμότητα της ενσωμάτωσης των παραμέτρων του εργαλείου.

Ας εκτελέσουμε μία υποθετική ανάλυση. Έστω οτι οι επιλογές του χρήστη είναι οι ακόλουθες

- Rows: [Geography].[All Geography]
- Columns: [Time].[All Time]

Ακολουθεί ένας πίνακας με δύο μετρήσεις συναρτήσει των παραμέτρων του συστήματος αναφορικά με τον αριθμό των επιπέδων που συμμετέχουν στο σχήμα.

	Επιλογή όλων των διαθέσιμων επιπέδων	**Επιλογή μόνο και του επόμενου επιπέδου**
Φάση 1	2,226 sec	2,226 sec
Φάση 2	1,547 sec	1,235 sec
Φάση 3	0,172 sec	0,172 sec

Η φάση 1 δηλαδή η φάση της αρχικοποίησης του σχήματος ανεξάρτητα από τις επιλογές του χρήστη είναι όπως αναμενόταν σταθερή. Στη φάση 2 αν και η επιτάχυνση μπορεί να φαίνεται μικρή, στην πραγματικότητα είναι ικανοποιητική. Μιας και κάθε διάσταση αποτελείται από 3 επίπεδα, η αλλαγή της παραμέτρου προκαλεί την αφαίρεση ενός μόνο επιπέδου. Αν η διάσταση είχε για παράδειγμα 6 επίπεδα θα αφαιρούταν 4 επίπεδα και η επιτάχυνση θα ήταν ακόη μεγαλύτερη.

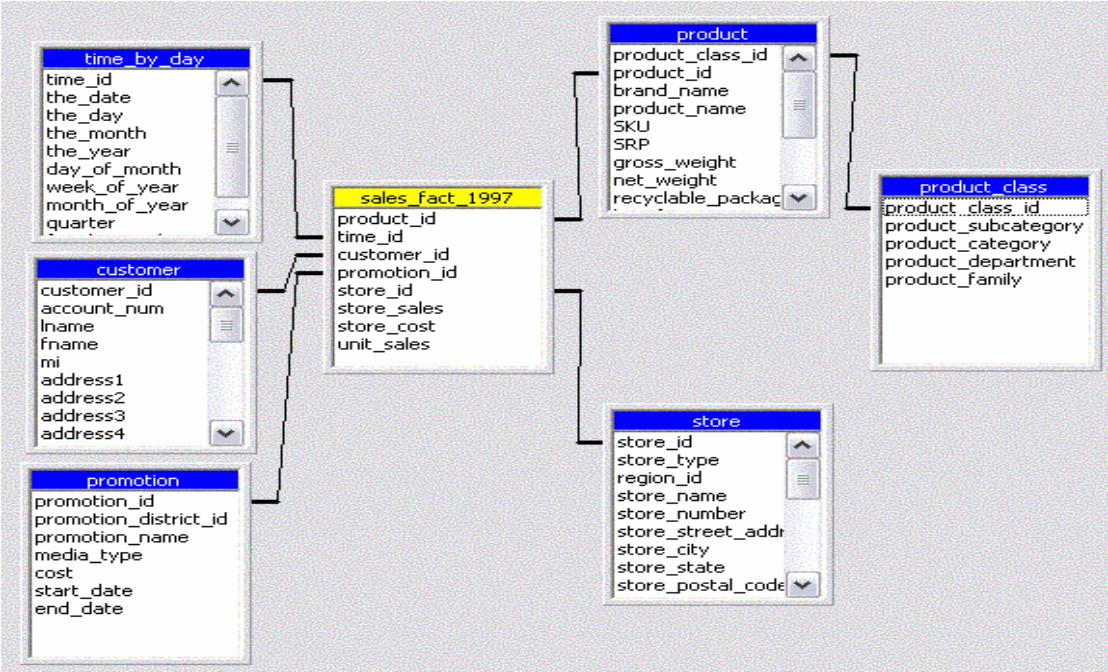
Έστω τώρα οτι οι επιλογές του χρήστη είναι οι ακόλουθες έτσι ώστε να συμπεριληφθεί και ένα φίλτρο

- Rows: [Geography].[All Geography]
- Columns:[Product].[All Product].[Category]
- Section: [Time].[All Time].[Year].[1992]

	Αρχικοποίηση με χρησιμοποίηση μόνο της τιμής φίλτρου για τον άξονα Section	**Αρχικοποίηση με όλα τα μέλη του ίδιου επιπέδου**
Φάση 1	2,226 sec	2,226 sec
Φάση 2	1,547 sec	2,708 sec
Φάση 3	0,047 sec	0,047 sec

Πάλι στη φάση 2 διαπιστώνουμε την επιτάχυνση που προκαλείται από την πρόσθεση στον Multicube μόνο της επιλεγόμενης τιμής φίλτρου η οποία στη συγκεκριμένη περίπτωση αγγίζει το 43%.

Στη συνέχεια αποφασίσαμε να χρησιμοποιήσουμε έναν πιο μεγάλο κύβο. Έτσι επιλέξαμε τον κύβο Sales που περιέχεται στα Analysis Services. Ο κύβος αυτός αποτελείται από 12 λογικές διαστάσεις, καθεμία από τις οποίες αποτελείται από 2 έως 7 επίπεδα. Με τον κύβο αυτό αποφασίσαμε να κάνουνε μετρήσεις αναφορικά με τις διαστάσεις που συμμετέχουν στο προς απεικόνιση σχήμα και όχι αναφορικά με τις παραμέτρους του συστήματος όπως στην προηγούμενη μέτρηση. Οι παράμετροι του συστήματος έχουν τις default τιμές τους.



Σχήμα 4.31 –FoodMart2000 κύβος

Στο σημείο αυτό θα πρέπει να τονιστεί πως οι μετρήσεις αυτές είναι ενδεικτικές. Φυσικά δεν παίζει ρόλο μόνο ο αριθμός των διαστάσεων αλλά και το μέγεθός τους και το επίπεδό το οποίο επιλέγεται από το χρήστη.

Στις στήλες αναφέρεται ο αριθμός των διαστάσεων που έχουν επιλεγεί σε καθέναν από τους 3 άξονες του CPM. Σε σχέση με τις προηγούμενες μετρήσεις βλέπουμε τον εκθετικό τρόπο με τον οποίο ο αριθμός των διαστάσεων επηρεάζει την απόδοση. Επίσης αρχίζουν και γίνονται κάπως πιο ξεκάθαρες κάποιες μεταβολές στη φάση 3, δηλαδή στη φάση όπου δημιουργείται η πινακοειδής απεικόνιση μιας και ο όγκος των προς απεικόνιση αποτελεσμάτων γίνεται μεγαλύτερος. Βέβαια σε όλες τις μετρήσεις η φάση της απεικόνισης αναφέρεται στη δημιουργία του πρώτου grid, ο οποίος είναι μικρότερος από το δευτερο κ.τ.λ.

	Rows: 1 **Columns:1** **Sections: 0**	**Rows: 1** **Columns:1** **Sections: 1**	**Rows: 2** **Columns:1** **Sections: 0**	**Rows: 2** **Columns:2** **Sections: 0**	**Rows: 2** **Columns:2** **Sections: 1**
Φάση 1	5,565 sec	5,565 sec	5,565 sec	5,565 sec	5,565 sec
Φάση 2	1,422 sec	1,782 sec	3,532 sec	5,578 sec	9,781 sec
Φάση 3	0,047 sec	0,047 sec	0,062 sec	0,069 sec	0,069 sec

Θα ήταν ιδιαίτερα χρήσιμη μία δοκιμή με έναν κύβο πολύ μεγαλύτερου μεγέθους, έτσι ώστε να ανταποκρίνεται σε πραγματικές συνθήκες. Δυστυχώς αυτό δεν έγινε εφικτό στα πλαίσια της διπλωματικής.

5 *Συμπεράσματα-Μελλοντική Δουλειά*

Στο τελευταίο τούτο κεφάλαιο συνοψίζουμε τα αποτελέσματα της παρούσας διπλωματικής. Στην παράγραφο 5.1 γίνεται μια αρκετά τεκμηριωμένη αναφορά στη μελλοντική δουλειά που ανακύπτει μέσα από την εργασία τούτη ενώ στην παράγραφο 5.2. περιγράφουμε το τι καταφέραμε να επιτύχουμε στα πλαίσια της διπλωματικής.

5.1 Μελλοντική δουλειά

Όπως ήδη αναφέραμε σε προηγούμενα κεφάλαια, η εργασία αυτή μπορεί να αποτελέσει τη βάση για μία επέκταση σε ασύρματο περιβάλλον. Άλλωστε σημαντικές σχεδιαστικές επιλογές του desktop συστήματος έλαβαν υπόψη την παράμετρο αυτή. Για παράδειγμα προς αυτή την κατεύθυνση επιλέχθηκε η τεχνική Table Lens, μία τεχνική που σε οθόνες περιορισμένου μεγέθους αποκτά μεγαλύτερη αξία και χρηστικότητα. Ακολουθεί μία αρκετά τεκμηριωμένη αναφορά σε αυτήν την πιθανή επέκταση, μαζί με το απαραίτητο θεωρητικό υπόβαθρο. Πρώτα θα γίνει μία αναφορά σε γενικές αρχές που θα πρέπει να διέπουν μία αρχιτεκτονική και στη συνέχεια θα δείξουμε πως όλα αυτά μπορούν να συνδεθούν με την παρούσα εργασία.

5.1.1 Χαρακτηριστικά κινητών συσκευών που επηρεάζουν τις εφαρμογές χρήστη

Η δημιουργία εφαρμογών για κινητές συσκευές και ο τρόπος αλληλεπίδρασης του χρήστη με τη συσκευή αποτελεί ένα ξεχωριστό πεδίο για τον τομέα της αλληλεπίδρασης ανθρώπου υπολογιστή. Η ιδιομορφία και η δυσκολία στη δημιουργία εύχρηστων εφαρμογών οφείλεται στη φορητότητα, στο υλισμικό και τους τρόπους εισόδου και εξόδου δεδομένων στις συσκευές αυτές, όπως αναφέρεται στο [ERA 01].

Οι βασικοί περιοριστικοί παράγοντες για την ανάπτυξη του γραφικού περιβάλλοντος, σύμφωνα με [PRR 02], είναι :

- Η περιορισμένη ανάλυση των οθονών σε κινητές συσκευές είναι ένας σημαντικός παράγοντας που πρέπει να ληφθεί υπ' όψιν από το σχεδιαστή. Για ένα κινητό τηλέφωνο η ανάλυση κυμαίνεται στα 120*120 pixel, ενώ για ένα PDA στα 240*320 pixel.
- Το μικρό μέγεθος της οθόνης είναι άλλη μια παράμετρος, ιδιαίτερα σημαντική για εφαρμογές που απευθύνονται σε ομάδες χρηστών με μεγάλη ποικιλομορφία, όπου τυχόν θα υπάρχουν και προβλήματα όρασης.
- Η περιορισμένη υπολογιστική ισχύς των κινητών συσκευών περιορίζει σημαντικά τη χρήση κινούμενων γραφικών σε πραγματικό χρόνο και γενικά τη χρήση πολύπλοκων γραφικών αναπαραστάσεων. Ειδικά η δημιουργία τρισδιάστατων αλληλεπιδρώντων περιβαλλόντων είναι σχεδόν αδύνατη αφού δεν υπάρχει το απαραίτητο υλικό που να υποστηρίζει την απόδοση τρισδιάστατων γραφικών. Ο περιορισμός αυτός καθιστά αδύνατη τη χρήση υπαρχόντων γραφικών περιβαλλόντων που στηρίζονται σε τρισδιάστατα γραφικά.
- Τέλος υπάρχουν περιορισμοί στη μνήμη, λόγω της αδυναμίας ενσωμάτωσης μεγαλύτερου μεγέθους σε τόσο μικρές συσκευές, γεγονός που υπαγορεύει τη δημιουργία γραφικού περιβάλλοντος με μικρές απαιτήσεις μνήμης.

Επίσης ιδιαιτερότητες των κινητών συσκευών υπάρχουν και στον τρόπο αλληλεπίδρασης με το χρήστη, στην εισαγωγή και έξοδο δεδομένων. Πιο συγκεκριμένα η εξαιτίας της φορητότητας έχουμε :

- Έλλειψη τυποποίησης. Σε αντίθεση με τους προσωπικούς υπολογιστές, πολλές συσκευές εισόδου/ εξόδου είναι προσανατολισμένες για χρήση με συγκεκριμένες συσκευές και αυτό προκαλεί προβλήματα στη δημιουργία μιας εφαρμογής που να τρέχει σε πολλές συσκευές.
- Η εισαγωγή δεδομένων δε γίνεται μέσω ενός τυπικού πληκτρολογίου ηλεκτρονικού υπολογιστή, αλλά μέσω πληκτρολογίων περιορισμένου αριθμού πλήκτρων τα οποία είναι λιγότερο χρηστικά από ένα συνηθισμένο πληκτρολόγιο.
- Δεν υπάρχει ποντίκι που διευκολύνει την πλοήγηση σε δισδιάστατες γραφικές εφαρμογές, αφού οι περισσότερες εφαρμογές έχουν σχεδιαστεί με γνώμονα τη συσκευή αυτή. Σε κινητές συσκευές συνήθως χρησιμοποιούνται οθόνες αφής ή κάποια λαβίδα, όπως π.χ. στα PDA.
- Υπάρχουν εξειδικευμένες μέθοδοι εισόδου/ εξόδου όπως οι φωνητικές εντολές, η ηχητική αναπαράσταση κειμένου κ.α. που δεν χρησιμοποιούνταν σε προσωπικούς

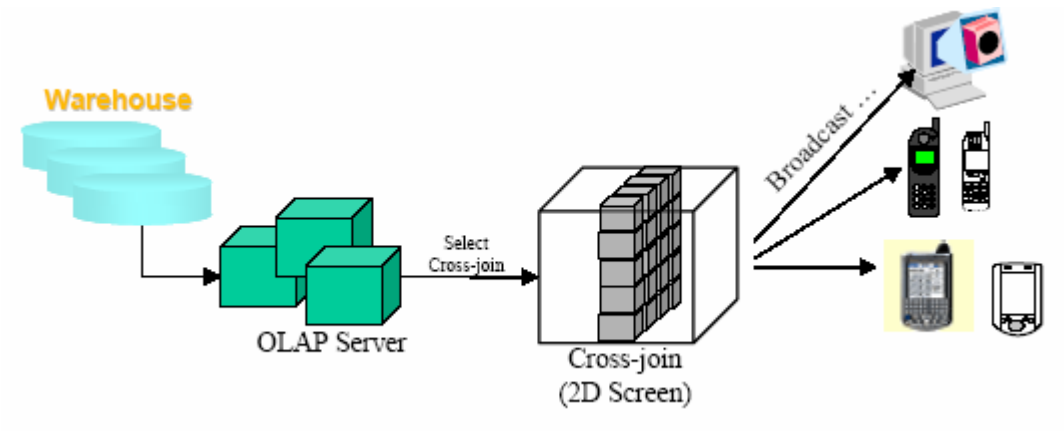
υπολογιστές και γι' αυτό υπάρχουν περιορισμένες τεχνικές ανάπτυξης εφαρμογών που να βασίζονται σε αυτά τα εργαλεία.

Τέλος περιορισμοί υπάρχουν λόγω του περιβάλλοντος χρήσης της συσκευής. Αναλυτικότερα :

- Ιδιαιτερότητες εξαιτίας του ηχητικού περιβάλλοντος. Συνήθως ένα κινητό τηλέφωνο ή ένα PDA χρησιμοποιούνται σε εξωτερικό χώρο, και αυτό δημιουργεί πρόβλημα είτε εξαιτίας του θορύβου αφού π.χ. δυσκολεύει την αλληλεπίδραση με τον χρήστη μέσω φωνητικών εντολών, είτε λόγω της ανεπιθύμητης έντασης ήχου από τη συσκευή π.χ. σε μια διάλεξη ή ένα μουσείο.
- Το περιβάλλον χρήσης των συσκευών πρέπει συνυπολογίζεται κατά το σχεδιασμό των εφαρμογών. Οι κινητές συσκευές δεν χρησιμοποιούνται σε ελεγχόμενο φωτιστικά περιβάλλον, όπως ένας υπολογιστής, αλλά σε μια ποικιλία από εξωτερικούς και εσωτερικούς χώρους με διαφορετικές συνθήκες φωτεινότητας που μπορεί να κυμαίνονται από σκοτάδι έως πολύ υψηλή φωτεινότητα. Οι συνθήκες αυτές θα έχουν μεγάλο αντίκτυπο στην ευκρίνεια της γραφικής αναπαράστασης.
- Το επίπεδο προσοχής του χρήστη συνήθως είναι μειωμένο κατά τη χρήστη κινητών συσκευών, σε αντίθεση με το συνηθισμένο περιβάλλον εργασίας του γραφείου ή του σπιτιού που χρησιμοποιούνται οι προσωπικοί υπολογιστές. Πολλές φορές χρησιμοποιούνται παράλληλα με άλλες δραστηριότητες ή διακοπτόμενα. Έτσι πρέπει να σχεδιάζονται οι εφαρμογές χωρίς να έχουν υψηλές γνωστικές απαιτήσεις από το χρήστη, καθώς επίσης να του δίνουν τις απαραίτητες πληροφορίες για να συνεχίσει εύκολα μετά από μια διακοπή.

5.1.2 Αρχιτεκτονική Mobile OLAP

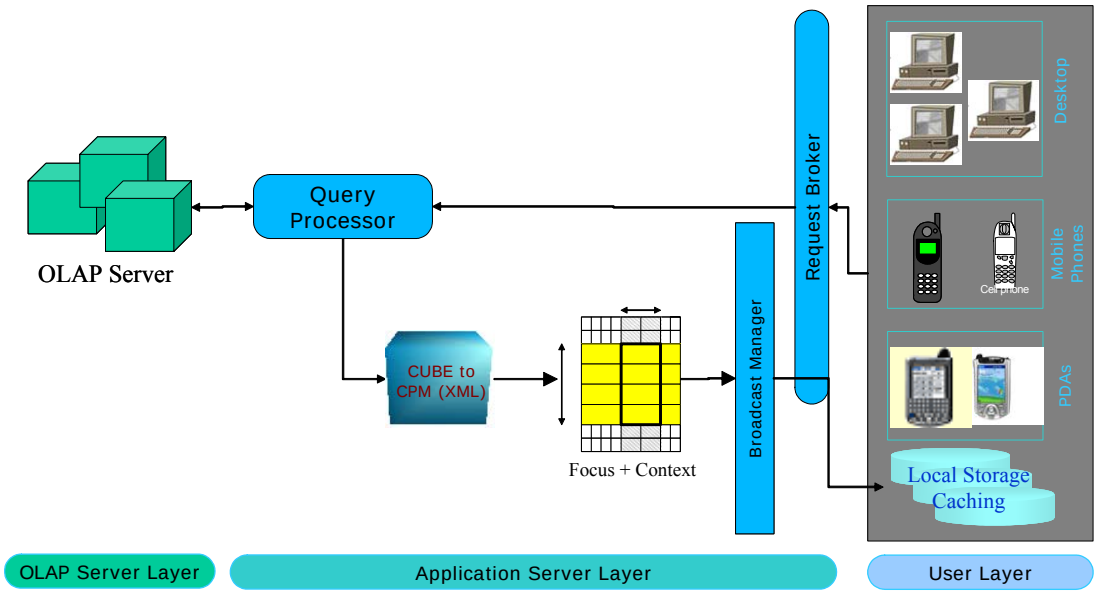
Οι εγγενείς περιορισμοί των ασύρματων συσκευών, όπως περιγράφηκαν στην προηγούμενη παράγραφο αναπόφευκτα δημιουργούν νέα θέματα στο σχεδιασμό ενός αντίστοιχου με τον OLAP Lens εργαλείου. Μία απλοποιημένη αρχιτεκτονική απεικονίζεται στο Σχ.5.1. Η γενική ιδέα είναι πως ο χρήστης χρησιμοποιεί ένα συγκεκριμένο user interface στη συσκευή του για να κάνει την ανάλυση των δεδομένων. Τα δεδομένα αυτά είναι αποθηκευμένα τοπικά, δηλαδή στην ίδια τη συσκευή, σε συναθροιστική μορφή. Αυτή η λογική είναι συχνά επαρκής μιας και τα συναθροισμένα δεδομένα εμπεριέχουν μεγάλο όγκο δεδομένων.



Σχήμα 5.1 -- Απλοποιημένη Αρχιτεκτονική ενός MOLAP συστήματος

Στην περίπτωση που υπάρχει ανάγκη για επιπλέον δεδομένα, τότε το user interface δημιουργεί το αντίστοιχο query, το οποίο προωθείται σε έναν ενδιάμεσο Application Server ο οποίος με τη σειρά του το προωθεί στο OLAP server. Μετά την ανάκτηση των δεδομένων από των OLAP server ο Application Server οργανώνει τα δεδομένα σε μία κατάλληλη μορφή για παρουσίαση και τα στέλνει τελικά στο user interface.

Μία τέτοια προσέγγιση παρμένη από το [MVSVMM04] απεικονίζεται στο Σχ.5.2. Αναφέρεται στο Cube View, μία ακαδημαϊκή πλατφόρμα για OLAP απεικονίσεις.



Σχήμα 5.2 -- Αρχιτεκτονική Cube View

5.1.3 MOLAP Lens

Στην υποπαράγραφο τούτη θα προσπαθήσουμε να περιγράψουμε μία αρχιτεκτονική βασισμένη στον OLAP Lens. Στόχος είναι βέβαια να χρησιμοποιήσουμε όσο το δυνατόν

περισσότερα κομμάτια της υπάρχουσας υλοποίησης. Ας δούμε όμως τί αλλάζει, τί είναι αυτό που μας αναγκάζει να τροποποιήσουμε κάποιες σχεδιαστικές επιλογές.

Στο desktop σύστημα που υλοποιήσαμε η επικοινωνία με τον OLAP server ήταν αξιόπιστη και συνεχής, μιας και ο server είναι τοπικός. Στην περίπτωση των ασύρματων συσκευών η σύνδεση αφενώς δεν είναι αξιόπιστη μιας και οποιαδήποτε στιγμή ο χρήστης είναι δυνατόν να βρεθεί εκτός δικτύου, αφετέρου το εύρος ζώνης είναι περιορισμένο.

Ένα άλλο πολύ σημαντικό ζήτημα είναι το περιορισμένο μέγεθος της μνήμης. Οι κατασκευαστές ασύρματων συσκευών παρουσιάζουν μοντέλα με ολοένα και αυξημένες δυνατότητες, παρόλα αυτά η διαθέσιμη μνήμη των συσκευών αυτών δεν μπορεί ακόμα να συγκριθεί με την μνήμη desktop συστημάτων.

Αυτοί οι δύο περιορισμοί κρίνονται ιδιαίτερα σημαντικοί στην περίπτωσή μας. Όπως είδαμε στον OLAP Lens η υλοποιούμενη CPM δομή οργανώνεται δυναμικά στην μνήμη (RAM) του συστήματος. Ανάλογα με τον όγκο των δεδομένων οι απαιτήσεις μνήμης μπορεί να είναι μεγαλύτερες από τις διαθέσιμες. Επίσης ένα θέμα δημιουργείται στην περίπτωση που η συσκευή για κάποιον λόγο είτε κλείνει είτε βγαίνει εκτός δικτύου. Θα πρέπει τα δεδομένα που ήδη υπάρχουν στη μνήμη της συσκευής να μην χάνονται έτσι ώστε μετά την αποκατάσταση της λειτουργίας της συσκευής, η ανάλυση να μπορεί να συνεχιστεί χωρίς επιπλοκές.

Ένα επίσης κρίσιμο θέμα, είναι αυτό της υπολογιστικής ισχύος. Η υλοποίηση της CPM δομής απαιτεί (ανάλογα και με τον όγκο των δεδομένων) μάλλον μεγαλύτερη υπολογιστική ισχύ από αυτήν μίας κοινής ασύρματης συσκευής.

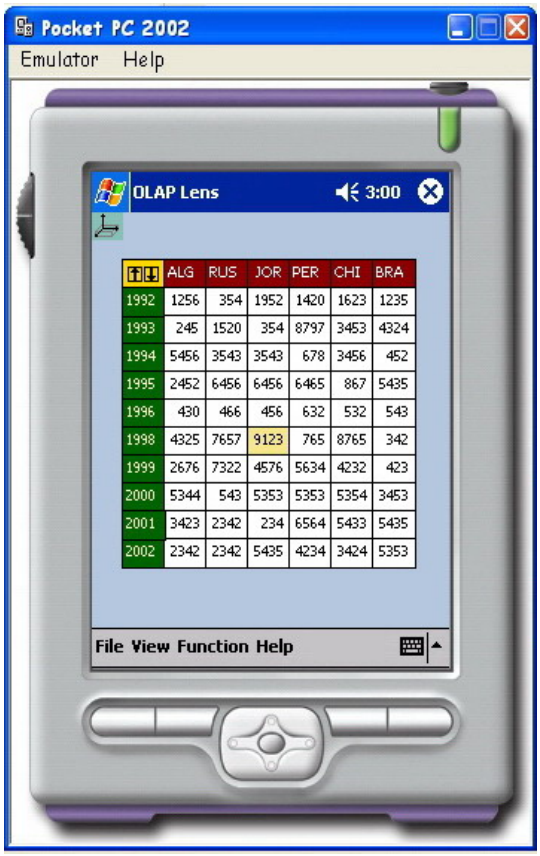
Στα πλαίσια της διπλωματικής μπορούμε απλά να κάνουμε κάποιες προτάσεις για την αντιμετώπιση των νέων αυτών θεμάτων :

- Αναφορικά με το θέμα της μνήμης, η λύση μπορεί να δοθεί μέσω των συστήματων βάσεων δεδομένων κατάλληλων για ασύρματες συσκευές. Σχεδόν όλοι οι μεγάλοι κατασκευαστές ΣΔΒΔ όπως η Oracle και η Microsoft διαθέτουν πλέον απλές εκδόσεις των desktop αντίστοιχων εκδόσεων, με περιορισμένες βέβαια δυνατότητες. Με τη χρήση μιας τέτοιας έκδοσης θα μπορούσε να αντιμετωπιστεί το πρόβλημα της απώλειας δεδομένων. Βέβαια απαιτείται δουλειά ως προς τον τρόπο που τα CPM (προς παρουσίαση) οργανωμένα δεδομένα μπορούν να αποθηκευτούν σε ένα σχεσιακό σχήμα στο ΣΔΒΔ και αντίστροφα.
- Αναφορικά με το θέμα της υπολογιστικής ισχύος είναι σαφές πως αυτή δεν μπορεί να παρασχεθεί από την ίδια συσκευή. Όπως και στο Cube View είναι μάλλον αναπόφευκτη η χρήση ενός ενδιάμεσου Application Server που θα οργανώνει τα

δεδομένα σε μία κατάλληλη μορφή για παρουσίαση και τα στέλνει στη συσκευή ήδη σε CPM μορφή.

Από τα παραπάνω προκύπτει εύλογα το ερώτημα, του τί ακριβώς μπορεί να χρησιμοποιηθεί από τον OLAP Lens. Η απάντηση είναι το user interface και φυσικά ο τρόπος με τον οποίο αυτό δέχεται τα κατά CPM οργανωμένα δεδομένα.

Μάλιστα στα πλαίσια της προσπάθειας αυτής πειραματιστήκαμε έτσι ώστε να διπαιστώσουμε στην πράξη το κατά πόσο είναι δυνατόν να πραγματοποιήσουμε την επέκταση αυτή. Έτσι χρησιμοποιήσαμε το .NET Compact Framework , το οποίο είναι ένα υποσύνολο του .NET Framework ,το οποίο χρησιμοποιήσαμε στην ανάπτυξη του OLAP Lens, και που απευθύνεται σε εφαρμογές ασύρματων συσκευών. Πιο συγκεκριμένα χρησιμοποιήσαμε τον Pocket PC 2002 Emulator. Δυστυχώς προς το παρόν τα αποτελέσματα δεν είναι ικανοποιητικά. Αν και η Microsoft διαφημίζει πως το .NET Compact Framework περιέχει σχεδόν όλες τις βασικές βιβλιοθήκες κλάσεων, εν τούτοις στην πράξη συναντήσαμε πολλές δυσκολίες. Οι δυνατότητες ανάπτυξης ενός user interface εφάμιλλου με τα αντίστοιχα desktop user interfaces είναι περιορισμένες. Στο Σχ.5.3.α απεικονίζεται ένα απλό στιγμυότυπο. Πολλά από τα κυριότερα controls του .NET Framework απουσιάζουν στο .NET Compact Framework, καθιστώντας μία πιθανή υλοποίηση μάλλον αδύνατη, προς το παρόν τουλάχιστον. Η Microsoft έχει ήδη εξαγγείλει πως σύντομα θα ενσωματώσει επιπλέον χαρακτηριστικά στο .NET Compact Framework. Κάτι τέτοιο θα διευκολύνει σημαντικά μία πιθανή ανάπτυξη. Το ενθαρρυντικό είναι πως το SourceGrid θεωρητικά λειτουργεί και στο .NET Compact Framework.

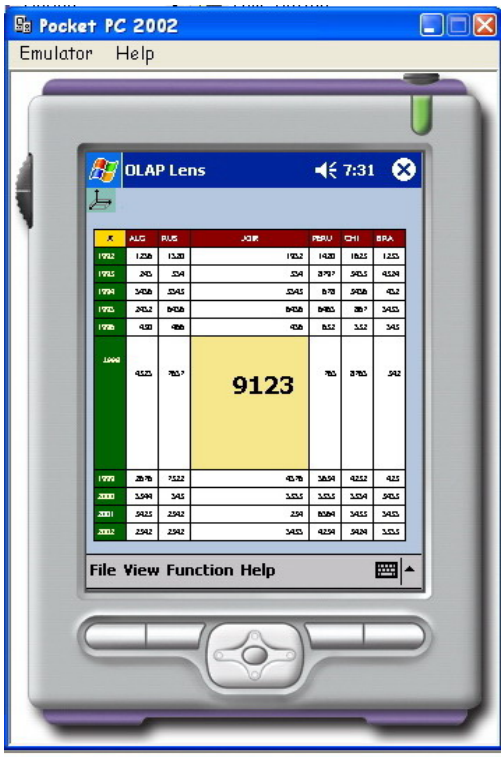


	ALG	RUS	JOR	PER	CHI	BRA
1992	1256	354	1952	1420	1623	1235
1993	245	1520	354	8797	3453	4324
1994	5456	3543	3543	678	3456	452
1995	2452	6456	6456	6465	867	5435
1996	430	466	456	632	532	543
1998	4325	7657	9123	765	8765	342
1999	2676	7322	4576	5634	4232	423
2000	5344	543	5353	5353	5354	3453
2001	3423	2342	234	6564	5433	5435
2002	2342	2342	5435	4234	3424	5353

Σχήμα 5. 3α: MOLAP Lens

Στα Σχ.5.3(α)(β)(γ) απεικονίζονται κάποια στιγμυότυπα στον Pocket PC 2002 Emulator κατά τα πρότυπα του OLAP Lens. Αξίζει να σημειώσουμε τον τρόπο με τον οποίο στο Σχ.5.3(β) ένα συγκεκριμένο κελί αποκτά μεγαλύτερη παρουσιαστική βαρύτητα, δηλαδή μεγενθύνεται

και αποκτά διαφορετικό χρωματισμό. Σε αυτό το περιορισμένο μέγεθος οθόνης τόσο το αισθητικό αποτέλεσμα όσο και η πρακτικότητα είναι ιδιαίτερα ικανοποιητικά.



Σχήμα 5. 3 (β) (γ): MOLAP Lens

5.2 Συμπεράσματα και σύνοψη διπλωματικής

Στα πλαίσια της παρούσας διπλωματικής εργασίας, καθορίστηκαν οι απαιτήσεις, σχεδιάστηκε και τελικά υλοποιήθηκε ένα εργαλείο, το οποίο παρέχει τη δυνατότητα οπτικοποίησης πολυδιάστατων δεδομένων χρησιμοποιώντας μία μη παραδοσιακή τεχνική οπτικοποίησης..

Αρχικά κάναμε μία εισαγωγή στο χώρο της αλληλεπίδρασης ανθρώπου-υπολογιστή(HCI), δίνοντας έμφαση συγκεκριμένα σε συστήματα βάσεων δεδομένων. Αναδείξαμε την οπτικοποίηση ως ένα βασικό κομμάτι του HCI. Στη συνέχεια περιγράψαμε κάποιες βασικές απαιτήσεις σε εφαρμογές οπτικοποίησης για OLAP μετά και από σχετική έρευνα των διαθέσιμων εμπορικών εργαλείων. Από την έρευνα αυτή πιστοποιήθηκε η απουσία ενός αυτόνομου και διακριτού μοντέλου παρουσίασης, μία ενδεχόμενη παρουσία του οποίου θα καθιστούσε πιο ευέλικτο το σχεδιασμό του καθαρά παρουσιαστικού μέρους ενός συστήματος.

Ακολούθησε η σχεδίαση του εργαλείου μας το οποίο και ονομάσαμε OLAP Lens. Καθορίστηκαν οι απαιτήσεις χρήστη για το εργαλείο αυτό. Βασικός σκοπός ήταν αφενώς η υλοποίηση του CPM(Cube Presentation Model), ενός αυτόνομου και διακριτού μοντέλου παρουσίασης και αφετέρου μία τεχνική οπτικοποίησης που σχετίζεται με την γνωστή τεχνική Table Lens. Παράλληλα έγινε μία εκτενής αναφορά στο CPM. Έγινε προσπάθεια το εργαλείο αυτό να είναι όσο το δυνατόν πιο ευέλικτο και επεκτάσιμο. Ακολούθησε η ανάλυση και περιγραφή της υλοποίησης του εργαλείου. Παρουσιάστηκε η αρχιτεκτονική του συστήματος και με τον τρόπο αυτό προέκυψαν οι ανάγκες των προγραμματιστικών εργαλείων, η επιλογή των οποίων τεκμηριώθηκε στη συνέχεια. Η επιλογή των εργαλείων αποτέλεσε ένα πολύ σημαντικό κομμάτι της εργασίας τούτης. Έπειτα αναλύθηκαν οι τεχνικές υλοποίησης. Δείξαμε στην πράξη την υλοποίηση του CPM, και τον τρόπο με τον οποίο μπορούμε να παρουσιάσουμε πολυδιάστατα δεδομένα σε μία διδιάστατη απεικόνιση. Εφαρμόσαμε με επιτυχία μία τεχνική βασισμένη στην Table Lens. Επίσης δείξαμε τον τρόπο με τον οποίο μπορεί να αντικατασταθεί ο επιλεγμένος OLAP server με κάποιον άλλο. Έπίσης δείξαμε πως η αρχιτεκτονική αυτή επιτρέπει ευέλικτες τροποποιήσεις. Στην παρούσα υλοποίηση ναι μεν χρησιμοποιήθηκε ένας OLAP server της Microsoft, αλλά μία πιθανή αντικατάσταση του server αυτού με κάποιον άλλο είναι σχετικά απλή. Στη συνέχεια απεικονίστηκαν στιγμυότυπα οθονών του εργαλείου καθώς και κάποια θέματα σχετικά με την απόδοση του.

Στο τελευταίο μέρος του τόμου τούτου, όχι μόνο προσεγγίσαμε κάποιες πιθανές μελλοντικές επεκτάσεις που αφορούν ασύρματες συσκευές, αλλά επιπρόσθετα παραθέσαμε κάποιες πρώτες σκέψεις στην κατεύθυνση αυτή, παράλληλα με κάποιες πειραματικές προσπάθειες χρήσης κάποιων κατάλληλων εργαλείων.

6 *Βιβλιογραφία*

- [MVSVMM] Andreas Maniatis, Panos Vassiliadis, Spiros Skiadopoulos, Yannis Vassiliou,George Mavrogonatos, Ilias Michalarias. *A Presentation Model and Non-Traditional Visualization for OLAP* (to appear)
- [VaSe99]P. Vassiliadis, T. Sellis: *A Survey of Logical Models for OLAP Databases.*SIGMOD Record 28(4), Dec. 1999.
- [Vass98]P. Vassiliadis: *Modeling Multidimensional Databases, cube and cube operations*. Proc. of the SSDBM 1998.
- [VaSk00] Panos Vassiliadis, Spiros Skiadopoulos: *Modeling and Optimization Issues for Multidimensional Databases*. Proc. of CAiSE-00, Stockholm, Sweden 2000
- [Man03]Maniatis, A. (2003). *OLAP Presentation Modeling with UML and XML*. In Proceedings of BCI 2003, pp. 232 - 241, Thessaloniki, Greece.
- [Man04]Maniatis, A. (2004). *The Case for Mobile OLAP*. Proceedings of the 1st International Workshop on Pervasive Information Management (in conjunction with EDBT '04), pp. 103 – 114, Heraklion, Greece.
- [PR96]Pirollo, P., & Rao, R. (1996). *Table Lens as a Tool for Making Sense of Data*. In Proceedings of the AVI 1996 Workshop, pp. 67 – 80, Gubbio, Italy.
- [RC94]Rao, R., & Card, S.K. (1994). *The Table Lens: Merging Graphical and Symbolic Representations in an effective Focus + Context Visualization for Tabular Information*. In Proceedings of ACM SIGCHI 1994, Boston, USA.
- [RR00]Rodriguez-Martinez, M., & Rossopoulos, N. (2000). *MOCHA: A Self-ExtensibleDatabase Middleware System for Distributed Data Sources*. In Proceedings of ACM SIGMOD 2000, pp 213 – 224, Dallas, USA.
- [MCR98]Microsoft (1998). *OLEDB for OLAP*, Available at: http://www.microsoft.com/data/oledb/olap/
- [MVSV03]Maniatis, A., & Vassiliadis, P., & Skiadopoulos, S., & Vassiliou, Y. (2003). *CPM: A Cube Presentation Model for OLAP*. In Proceedings of DaWaK 2003, pp. 4 – 13, Prague, Czech Republic.
- [MVSV03A]Maniatis, A., & Vassiliadis, P., & Skiadopoulos, S., & Vassiliou, Y. (2003a). *Advanced Visualization for OLAP*. In Proceedings of DOLAP 2003, pp. 9 – 16, New Orleans, USA.
- [BaBu87] R. M. Baecker and W. A. S. Buxton, editors, *Readings in Human-Computer Interaction: A Multi-disciplinary Approach*, Los Altos, CA: Morgan Kaufmann 1987.
- [BC99] Stephen A Brewster, Peter G Cryer : *Maximising Screen-Space on Mobile Computing Devices* ACM, 1999.
- [BR02] P. Brusilovsky, R.Rizzo : *Map-based Access to Multiple Educational On-Line Resources from Mobile Wireless Devices*, Proceedings of the 4th International Symposium on Mobile Human-Computer Interaction, 2002.
- [BRIO03] Brio Product Line: *www.brio.com* , 2003.
- [CMS99] Stuart K. Card, Jock D. Mackinlay, and Ben Shneiderman. *Readings in Information Visualization: Using Vision to Think*. Morgan Kaufmann Publishers, 1999.

- [ERA01] Eran Aharonson : *Human user interface for mobile devices*, Electronic Engineering Times, www.eetasia.com , 2001.
- [HBC+02] Hewett, Baecker, Card, Carey, Gasen, Mantei, Perlman, Strong and Verplank : *Curricula for Human-Computer Interaction ,* ACM SIGCHI, 2002.
- [HiRe00]David M. Hilbert and David F. Redmiles. *Extracting usability information from user interface events*. *ACM Computing Surveys*, 32(4):384–421, December 2000.
- [HPRN03] Hyperion Product Line: *www.hyperion.com* , 2003.
- [Inmo96] W.H. Inmon: *Building the Data Warehouse*. John Wiley & Sons, 1996.
- [KK97] Michael Klein, Birgitta Konig-Ries : *An Ontology-Based Document-Space as an Adaptable User Interface for Mobile Information Systems*, CiteSeer.IST Scientific Literature Digital Library, 1997.
- [MSTR03] Microstrategy Inc. Product Line: *www.microstrategy.com* , 2003.
- [PaTe04] PARIS Technologies : www.paristech.com , 2004.
- [PRR02] Volker Paelke, Christian Reimann, Waldemar Rosenbach : *A Visualization Design Repository for Mobile Devices* Proceedings of the 2nd international conference on Computer graphics, virtual Reality, visualisation and interaction in Africa, 2002.
- [SIG04] *ACM's Special Interest Group on Computer-Human Interaction* : www.sigchi.org , 2004.
- [StHa00] Chris Stolte, Pat Hanrahan: *Polaris: A System for Query, Analysis and Visualization of Multidimensional Relational Databases*. Proc. of the IEEE Symposium on Information Visualization 2000 (InfoVis'00).
- [UnMa01] University of Maryland : www.otal.umd.edu/hci-rm/index.html , 2001.
- [WF04] WAP Forum. http://www.wapforum.org
- [WAPF99] Wireless Application Protocol Forum,Ltd : *Wireless Application Protocol Wireless Markup Language Specification Version 1.1* , http://www.wapforum.org , 1999.
- [WML02] WML Online resource for markup language technologie : http://www.oasis-open.org/cover/wap-wml.html , 2002.