

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ
ΤΜΗΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ
ΔΙΚΤΥΟΚΕΝΤΡΙΚΑ ΣΥΣΤΗΜΑΤΑ



ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ
ΔΙΑΛΕΙΤΟΥΡΓΙΚΟΤΗΤΑ ΣΥΣΤΗΜΑΤΩΝ

Υπεύθυνος Διπλωματικής:
Αναπληρωτής καθηγητής Δεμέστιχας Π.

Φοιτητής: **Πέτρος Κοκκινάκης**
Αριθμός Μητρώου: **ME / 0676**
Email: **batspy@gmail.com**

Ακ. Έτος: 2007-2008

Περιεχόμενα

Εισαγωγή	9
1 Βιβλιογραφική Ανασκόπηση	14
1.1 Διαλειτουργικότητα Συστημάτων	15
1.2 Υπάρχουσες Υλοποιήσεις	27
1.2.1 IBM WID (Websphere Integration Developer)	28
1.2.2 Asynchronous Mediation for Integrating Business and Operational Processes	31
1.2.3 Ενδιάμεσο Λογισμικό Διαλειτουργικότητας	34
2 Σχεδίαση και Ανάπτυξη	38
2.1 Σχεδίαση Συστήματος	39
2.1.1 Γενική περιγραφή λογισμικού	40
2.1.2 Στόχοι και Κανόνες του Ultimate Mediator	43
2.1.3 Διαγράμματα και παραδείγματα επικοινωνίας	45
2.1.3.1 UML Διαγράμματα	46
2.1.3.2 Διαγράμματα κλάσεων	56
2.1.3.3 Παραδείγματα επικοινωνίας (sequence diagrams)	62
2.1.4 Μεταφορά δεδομένων σε πραγματικό χρόνο (Online transactions)....	65
2.2 Επιλογές Υλοποίησης	70
2.2.1 Visual Studio 2008	71
2.2.2 LINQ (Language Integrated Query)	74
2.2.3 XML και XML Schema	76
2.2.4 SOAP και HTTP	78
2.2.5 Μοντελοποίηση Βάσης Δεδομένων	81
3 Εγχειρίδιο χρήσης	91
3.1 Απαιτούμενο λογισμικό	91
3.2 Εγκατάσταση εφαρμογής	92
3.3 Είσοδος στο πρόγραμμα	95
3.4 Περιγραφή των menu της βασικής Φόρμας	97
3.5 Manage Connections	99
3.6 Manage Web Services	109
3.7 Mediator Designer	116
3.7.1 Manage Associations	120
3.7.2 Manage Conditions	124
3.7.3 Validation	137
3.7.4 Execution	141
3.7.5 Add/Set/Edit/View/Remove Connection	143
3.7.6 Add/Edit/View/Delete Node	146
3.7.6.1 Προσθήκη κόμβου (Add Node)	147
3.7.6.2 Επεξεργασία / Προβολή / Διαγραφή Κόμβου	156
3.7.7 Διαχείριση Σχημάτων στη φόρμα σχεδίασης	158
3.8 XML to Web Service Transporter	161
3.9 Help και About	166
4 Περιπτώσεις χρήσης	172
4.1 Μεταφορά από Βάση σε XML	173
4.2 XML σε Βάση Δεδομένων	179
4.3 XML σε XML	185
4.4 Μεταφορά δεδομένων από Βάση σε Βάση	189

4.5	Σύνθετη περίπτωση από Βάση σε XML	193
4.6	Σύνθετη περίπτωση από XML σε XML	200
4.7	Χρήση πολλαπλών συνδέσεων εισόδου	205
4.8	Άμεση αποστολή XML σε Web Service	208
5	Τεχνολογικά - Επιχειρηματικά Οφέλη	210
6	Προοπτικές Ανάπτυξης	213
	Παραπομπές	214

Λίστα Εικόνων

Εικόνα 1.1.1 Ρόλος του ενδιάμεσου επιπέδου / λογισμικού Διαλειτουργικότητας [11]	15
Εικόνα 1.1.2 Γενικότερη σύνδεση ενδιάμεσου λογισμικού με τα υπόλοιπα συστήματα[12].	16
Εικόνα 1.1.3 Οι γενιές του EAI σε κλίμακα χρόνου [15]	17
Εικόνα 1.1.4 Τα διαφορετικά επίπεδα της πρώτης γενιάς υπολογιστών [15]	18
Εικόνα 1.1.5 Message Broker και Δεύτερη γενιά [15]	19
Εικόνα 1.2.1 Οθόνη προγράμματα WID (Websphere Integration Developer)	28
Εικόνα 1.2.2 Δημιουργία εργασίας μέσω του BPEL συντάκτη	30
Εικόνα 1.2.3 Υποστήριξη κατά τον χρόνο εκτέλεσης.	31
Εικόνα 1.2.4 Ολοκληρωμένα μοντέλα παρουσίασης	32
Εικόνα 1.2.5 Αρχική οθόνη του ΕΛΔ	34
Εικόνα 1.2.6 Λειτουργίες του ΕΛΔ	35
Εικόνα 1.2.7 Διαχείριση Βάσεων Δεδομένων	35
Εικόνα 1.2.8 Κώδικας Java για την δημιουργία επεκτάσεων	36
Εικόνα 2.1.1 Το γενικότερο μοντέλο λειτουργίας του Ultimate Mediator	40
Εικόνα 2.1.2 Επικοινωνία οντοτήτων με το πρόγραμμα	41
Εικόνα 2.1.3 Διάγραμμα λειτουργίας προγράμματος (1 ^ο επίπεδο)	46
Εικόνα 2.1.4 Διάγραμμα εισόδου στο σύστημα	47
Εικόνα 2.1.5 Διάγραμμα σχεδίασης περιπτώσεων διαλειτουργικότητας	47
Εικόνα 2.1.6 Διάγραμμα διαχείρισης συνδέσεων	48
Εικόνα 2.1.7 Διάγραμμα διαχείρισης Web Services	48
Εικόνα 2.1.8 Διάγραμμα αποστολής δεδομένων σε Web Services	49
Εικόνα 2.1.9 Διάγραμμα περιπτώσεων αυτοματισμού	49
Εικόνα 2.1.10 Διάγραμμα χρήσης workflow υποσυστήματος	50
Εικόνα 2.1.11 Διάγραμμα κλάσεων υλοποίησης του Ultimate Mediator (α)	56
Εικόνα 2.1.12 Διάγραμμα κλάσεων υλοποίησης του Ultimate Mediator (β)	57
Εικόνα 2.1.13 Διάγραμμα κλάσεων υλοποίησης του Ultimate Mediator (γ)	58
Εικόνα 2.1.14 Sequence Diagram για την εξαγωγή και αποστολή δεδομένων σε Web Service	62
Εικόνα 2.1.15: Sequence Diagram για την απευθείας αποστολή δεδομένων σε web service	64
Εικόνα 2.1.16 Αρχιτεκτονική διασύνδεσης Ultimate Mediator	66
Εικόνα 2.1.17 Ultimate Mediator με Workflow Subsystem	68
Εικόνα 2.1.18 Ultimate Mediator με Workflow στον πραγματικό κόσμο	69
Εικόνα 2.2.1 Χρωματικές διαφοροποιήσεις μεταξύ λειτουργία auto-complete	72
Εικόνα 2.2.2 Οθόνη σχεδίασης παραθύρου μέσα από το Visual Studio	73
Εικόνα 2.2.3 Παράδειγμα χρήσης της Linq για την πραγματοποίηση ερωτημάτων σε xml δεδομένα	75
Εικόνα 2.2.4 Παράδειγμα πολλών Linq ερωτημάτων για τον συνδυασμό όλων των διαφορετικών πινάκων	75
Εικόνα 2.2.5 Παράδειγμα ενός XML	76
Εικόνα 2.2.6 Δομή ενός SOAP μηνύματος με ένα επισυναπτόμενο αρχείο	78
Εικόνα 2.2.7 Παράδειγμα ενός xml με τα πεδία ορισμένο με τις ετικέτες του soap	79
Εικόνα 2.2.8 Οι επικεφαλίδες αποστολής του HTTP πρωτοκόλλου	80
Εικόνα 2.2.9 Σχήμα Βάσης Δεδομένων του Ultimate Mediator	81
Εικόνα 2.2.10 Τα πεδία του πίνακα WebServices	82

Εικόνα 2.2.11 Τα πεδία του πίνακα WebServiceParams	82
Εικόνα 2.2.12 Τα πεδία του πίνακα Connections	83
Εικόνα 2.2.13 Τα πεδία του πίνακα Attributes	85
Εικόνα 2.2.14 Τα πεδία του πίνακα Conditions	85
Εικόνα 2.2.15 Τα πεδία του πίνακα ConnectionTypeSettings	87
Εικόνα 3.1.1 Μήνυμα της εφαρμογής όταν δεν είναι εγκατεστημένη η 3.5 έκδοση λογισμικού του .Net Framework	91
Εικόνα 3.2.1 Πρώτη οθόνη εγκατάστασης του λογισμικού	92
Εικόνα 3.2.2 Περίπτωση ήδη εγκατεστημένης εφαρμογής	93
Εικόνα 3.2.3 Μήνυμα που δηλώνει ότι υπάρχει ήδη μία εγκατεστημένη έκδοση του προγράμματος	93
Εικόνα 3.2.4 Οθόνη επιλογής διαδρομής και χρηστών συστήματος	94
Εικόνα 3.3.1 Οθόνη εισόδου κωδικού πρόσβασης	95
Εικόνα 3.3.2 Μήνυμα σφάλματος σε περίπτωση λάθους κωδικού	95
Εικόνα 3.3.3 Κύρια οθόνη εφαρμογής	96
Εικόνα 3.5.1 Οθόνη διαχείρισης Συνδέσεων	99
Εικόνα 3.5.2 Οθόνη διαχείρισης συνδέσεων με κουμπί επιλογής	100
Εικόνα 3.5.3 Οθόνη προσθήκης νέας σύνδεσης	101
Εικόνα 3.5.4 Απόκρυψη δεδομένων εισαγωγής XML λόγω επιλογής σύνδεσης τύπου Βάσης	102
Εικόνα 3.5.5 Δεδομένα εισαγωγής σε σύνδεση τύπου XML αρχείου	103
Εικόνα 3.5.6 Παράθυρο διαλόγου επιλογής αρχείου	104
Εικόνα 3.5.7 Εμφάνιση διαδρομής δίσκου κατά τη χειροκίνητη εισαγωγή αρχείου	105
Εικόνα 3.5.8 Παράδειγμα εμφάνισης Root Element	106
Εικόνα 3.5.9 Οθόνη εμφάνισης λάθους κατά την αποθήκευση των στοιχείων	106
Εικόνα 3.5.10 Οθόνη εμφάνισης στοιχείων επεξεργασίας Σύνδεσης	107
Εικόνα 3.5.11 Οθόνη προβολής Σύνδεσης	108
Εικόνα 3.5.12 Μήνυμα επικύρωσης διαγραφής Σύνδεσης	108
Εικόνα 3.6.1 Φόρμα Manage Web Services	109
Εικόνα 3.6.2 Φόρμα Manage Web Services με κουμπί επιλογής	110
Εικόνα 3.6.3 Φόρμα προσθήκης νέου Web Service	110
Εικόνα 3.6.4 Οθόνη προσθήκης παραμέτρου σε Web Service	111
Εικόνα 3.6.5 Περίπτωση που δηλώνεται και η στατική και η δυναμική τιμή	112
Εικόνα 3.6.6 Περίπτωση που δεν έχει οριστεί δυναμική ή στατική τιμή	112
Εικόνα 3.6.7 Ο χρήστης έχει βάλει παραπάνω από μία παραμέτρους με δυναμική τιμή	113
Εικόνα 3.6.8 Περίπτωση που δεν έχει οριστεί δυναμική ή στατική τιμή	113
Εικόνα 3.6.9 Επεξεργασία ενός Web Service	114
Εικόνα 3.6.10 Επεξεργασία παραμέτρου ενός Web Service	114
Εικόνα 3.6.11 Προβολή Web Service	115
Εικόνα 3.6.12 Προβολή παραμέτρου Web Service	115
Εικόνα 3.7.1 Αρχική οθόνη Mediator Designer	116
Εικόνα 3.7.2 Παράδειγμα εισαγωγής connection στη σχεδιαστική φόρμα	117
Εικόνα 3.7.3 Επιλογές πατώντας στον κενό χώρο	118
Εικόνα 3.7.4 Επιλογές που εμφανίζονται επάνω στο τελικό σχήμα μεταφοράς	119
Εικόνα 3.7.5 Διαχειριστική φόρμα Συσχετίσεων	120
Εικόνα 3.7.6 Προσθήκη νέας συσχέτισης	121
Εικόνα 3.7.7 Επεξεργασία υπάρχουσας συσχέτισης	122
Εικόνα 3.7.8 Εμφάνιση συσχετίσεων στις συνδέσεις της σχεδιαστικής φόρμας	123
Εικόνα 3.7.9 Φόρμα διαχείρισης συνθηκών	124

Εικόνα 3.7.10 Φόρμα προσθήκης συνθήκης	125
Εικόνα 3.7.11 Πατώντας στο «Root Condition» το κουμπί προσθήκης ενεργοποιείται	125
Εικόνα 3.7.12 Εισαγωγή κενής συνθήκης	126
Εικόνα 3.7.13 Έξτρα λειτουργίες της κενής συνθήκης	126
Εικόνα 3.7.14 Επίπεδα ιεραρχίας με βάση τους τελεστές	127
Εικόνα 3.7.15 Παράδειγμα δέντρου ιεραρχίας συνθηκών	128
Εικόνα 3.7.16 Φόρμα διαχείρισης συνθηκών	129
Εικόνα 3.7.17 Προσθήκη νέας απλής συνθήκης	130
Εικόνα 3.7.18 Επιλογή είδους ελέγχου συνθήκης	132
Εικόνα 3.7.19 Επιλογή πεδίου για τη σύγκρισή του με το αρχικό	133
Εικόνα 3.7.20 Επιλογή στατικής τιμής για τη σύγκρισή της με το αρχικό πεδίο	133
Εικόνα 3.7.21 Παράδειγμα ορισμού νέας συνθήκης	134
Εικόνα 3.7.22 Επιλογής της συνθήκης από τη λίστα διαχείρισης	135
Εικόνα 3.7.23 Χτίσιμο του δέντρου συνθηκών	135
Εικόνα 3.7.24 Προσθήκη δεύτερης συνθήκης	136
Εικόνα 3.7.25 Το δέντρο ιεραρχίας συνθηκών με δύο συνθήκες	136
Εικόνα 3.7.26 Φόρμα επικύρωσης (α)	137
Εικόνα 3.7.27 Φόρμα επικύρωσης (β)	138
Εικόνα 3.7.28 Περίπτωση μηνυμάτων με όλα τα φίλτρα ενεργά	139
Εικόνα 3.7.29 Περίπτωση μηνυμάτων με δύο ενεργά φίλτρα	139
Εικόνα 3.7.30 Τελικό μήνυμα επιβεβαίωσης πριν τη δημιουργία του πίνακα	139
Εικόνα 3.7.31 Εμφάνιση τελικού sql ερωτήματος	141
Εικόνα 3.7.32 Οθόνη εμφάνισης XML δεδομένων στη δενδρική τους μορφή	142
Εικόνα 3.7.33 Μενού επιλογών που εμφανίζεται στον κενό χώρο	143
Εικόνα 3.7.34 Εμφάνιση σύνδεσης στη φόρμα σχεδίασης	143
Εικόνα 3.7.35 Μενού που εμφανίζεται με το δεξί πλήκτρο του ποντικού επάνω στο τελικό σχήμα	144
Εικόνα 3.7.36 Επεξεργασία σύνδεσης από τη φόρμα σχεδίασης	144
Εικόνα 3.7.37 Απλή προβολή σύνδεσης από τη διαχειριστική φόρμα	145
Εικόνα 3.7.38 Φόρμα προσθήκης κόμβου	147
Εικόνα 3.7.39 Το group XML Conversions στα Predefined Statements	150
Εικόνα 3.7.40 Το group SQL Conversions στα Predefined Statements	151
Εικόνα 3.7.41 Προσθήκη ενός Predefined Statement	151
Εικόνα 3.7.42 Οθόνη της καρτέλας Column Modification	152
Εικόνα 3.7.43 Οθόνη της καρτέλας «XML Specific»	153
Εικόνα 3.7.44 Ορισμός XML Association (σύνδεσης στοιχείων σχήματος)	154
Εικόνα 3.7.45 Λάθη μη εισαγωγής δεδομένων στη δημιουργία νέου κόμβου	155
Εικόνα 3.7.46 Παράδειγμα επεξεργασίας κόμβου	156
Εικόνα 3.7.47 Παράδειγμα προβολής κόμβου	157
Εικόνα 3.7.48 Μετακίνηση πλαισίου προς τα αριστερά	158
Εικόνα 3.7.49 Μετακίνηση πλαισίων εκτός ορίων παραθύρου	159
Εικόνα 3.7.50 Αλλαγή μεγέθους σχημάτων	160
Εικόνα 3.8.1 Αρχική φόρμα αποστολής XML σε Web Service	161
Εικόνα 3.8.2 Διαχειριστική φόρμα συνδέσεων φιλτραρισμένη μόνο για XML	161
Εικόνα 3.8.3 Πληκτρολόγηση διαδρομής δίσκου με βοήθεια	162
Εικόνα 3.8.4 Μήνυμα σφάλματος όταν το σύστημα δεν μπορεί να επικοινωνήσει με κάποιο Web Service	162
Εικόνα 3.8.5 Οθόνη εμφάνισης XML δεδομένων στη δενδρική τους μορφή	163
Εικόνα 3.8.6 Οθόνη εμφάνισης XML δεδομένων σε μορφή κειμένου	163

Εικόνα 3.8.7 Παράθυρο επιλογής αρχείου για αποθήκευση δεδομένων.....	164
Εικόνα 3.9.1 Οθόνη βοήθειας με όλα τα διαθέσιμα topics.....	166
Εικόνα 3.9.2 Προβολής επιλεγμένης βοήθειας στον κενό χώρο.....	167
Εικόνα 3.9.3 Οθόνη που περιέχει το ερωτηματικό για την προβολή βοήθειας κάποιου πεδίου.....	168
Εικόνα 3.9.4 Προβολή βοήθειας μετά από την επιλογή του πεδίου Final Modification.....	169
Εικόνα 3.9.5 Επιλογή άλλης παραπλήσιας βοήθειας.....	170
Εικόνα 3.9.6 Μήνυμα που δείχνει ότι δε βρέθηκε βοήθεια για το πεδίο αυτό.....	170
Εικόνα 3.9.7 Φόρμα About του Ultimate Mediator.....	171
Εικόνα 4.1.1 Εμφάνιση σύνδεσης για τη βάση μίας τράπεζας.....	173
Εικόνα 4.1.2 Εικόνα προβολής πινάκων της σύνδεσης «Trapeza Alpha».....	174
Εικόνα 4.1.3 Μεταφορά του πίνακα customers στο τελικό σχήμα.....	174
Εικόνα 4.1.4 Τελικό σχήμα με τον πίνακα customers.....	175
Εικόνα 4.1.5 Δημιουργία σύνδεσης τύπου XML.....	176
Εικόνα 4.1.6 Εμφάνιση τελικού sql ερωτήματος.....	177
Εικόνα 4.1.7 Το xml που δημιουργείται μετά το τέλος της μεταφοράς δεδομένων..	177
Εικόνα 4.1.8 Στοιχεία του web service που δημιουργήθηκε.....	178
Εικόνα 4.2.1 Μεταφορά του πίνακα «TRANSACTIONS» στο τελικό σχήμα.....	179
Εικόνα 4.2.2 Αλλαγή του τύπου δεδομένων στο τελικό σχήμα.....	180
Εικόνα 4.2.3 Προσθήκη νέας ομάδας συνθηκών.....	180
Εικόνα 4.2.4 Δημιουργία απλής συνθήκης με βάση το CU_ID.....	181
Εικόνα 4.2.5 Πλήρης συνθήκη έχοντας μόνο μία απλή συνθήκη.....	182
Εικόνα 4.2.6 Προβλήματα κατά τον έλεγχο τελικού σχήματος.....	182
Εικόνα 4.2.7 Διαθέσιμοι πίνακες από την υπάρχουσα δομή της βάσης.....	183
Εικόνα 4.2.8 Εμφάνιση νέου πίνακα που δημιουργήθηκε στην βάση δεδομένων....	183
Εικόνα 4.2.9 Οθόνη ένδειξης ότι οι εγγραφές καταχωρήθηκαν.....	183
Εικόνα 4.2.10 Τα δεδομένα του xml αρχείου που χρησιμοποιήθηκε.....	184
Εικόνα 4.3.1 Μεταφορά του xml πίνακα «customers».....	185
Εικόνα 4.3.2 Προσθήκη νέου κόμβου σε xml.....	186
Εικόνα 4.3.3 Καθορισμός τιμής στο νέο κόμβο.....	187
Εικόνα 4.3.4 Επιστρεφόμενο xml αποτέλεσμα που φαίνεται η τιμή του νέου πεδίου.....	188
Εικόνα 4.4.1 Δημιουργία σύνδεσης με MSSQLServer.....	189
Εικόνα 4.4.2 Μεταφορά πίνακα customers στο τελικό σχήμα.....	190
Εικόνα 4.4.3 Ορισμός κρυφού κόμβου.....	191
Εικόνα 4.4.4 Ερώτημα επικύρωσης για δημιουργία πίνακα.....	191
Εικόνα 4.4.5 Εμφάνιση τελικού ερωτήματος για τη λήψη των δεδομένων.....	192
Εικόνα 4.4.6 Μήνυμα επιτυχούς αποστολής δεδομένων.....	192
Εικόνα 4.4.7 Πίνακας στη MySQL βάση.....	192
Εικόνα 4.5.1 Μεταφορά δύο πινάκων στο τελικό σχήμα.....	193
Εικόνα 4.5.2 Ορισμός ενός κόμβου ως ιδιότητα (Attribute).....	194
Εικόνα 4.5.3 Ορισμός του πατέρα συσχέτισης στο τελικό σχήμα.....	195
Εικόνα 4.5.4 Προσθήκη συσχέτισης μεταξύ των πινάκων στα σχήματα πηγής.....	196
Εικόνα 4.5.5 Εμφάνιση συσχέτισης στη φόρμα σχεδιασμού στο σχήμα πηγής.....	197
Εικόνα 4.5.6 Προσθήκη νέας συνθήκης με βάση το όνομα του πελάτη.....	198
Εικόνα 4.5.7 Εμφάνιση SQL ερωτήματος πριν την εκτέλεση για τη λήψη των δεδομένων.....	199
Εικόνα 4.5.8 Τελικό αποτέλεσμα εξόδου του τελικού σχήματος.....	199
Εικόνα 4.6.1 Πρώτο κομμάτι αρχείου XML με δεδομένα συναλλαγών.....	200

Εικόνα 4.6.2 Δεύτερο κομμάτι XML με δεδομένα συναλλαγών	201
Εικόνα 4.6.3 Φόρμα σχεδίασης μετά τον ορισμό όλων των συσχετίσεων.....	201
Εικόνα 4.6.4 Δεδομένα μετά από έξοδο όπως είναι από το αρχικό τους σχήμα	202
Εικόνα 4.6.5 Περίπτωση με δεδομένα μερικώς ορισμένα.....	203
Εικόνα 4.6.6 Αποτέλεσμα εκτέλεσης ολικής σύνδεσης με ελλιπές δεδομένα	204
Εικόνα 4.6.7 Έξοδος δεδομένων με αλλαγή του «Join Type» στις συσχετίσεις.....	204
Εικόνα 4.7.1 Οθόνη σχεδίασης με πολλαπλές συνδέσεις εισόδου	205
Εικόνα 4.7.2 Διαχειριστική φόρμα συσχετίσεων της περίπτωσης πολλαπλής εισόδου δεδομένων	206
Εικόνα 4.7.3 Αποτέλεσμα σε περίπτωση πολλαπλών συνδέσεων.....	207
Εικόνα 4.8.1 Φόρμα άμεσης αποστολής xml σε web service.....	208
Εικόνα 4.8.2 Οθόνη προβολής απάντησης από web service	208

Εισαγωγή

Οι συνεχόμενες εξελίξεις στον τομέα της Πληροφορικής καθώς και η ραγδαία διείσδυση των πληροφοριακών συστημάτων στις καθημερινές διαδικασίες του κύκλου ζωής των οργανισμών και επιχειρήσεων, οδήγησαν στην ύπαρξη πληθώρας εφαρμογών που εξυπηρετούν μεμονωμένες και εξειδικευμένες ανάγκες των επιχειρησιακών διαδικασιών [12][21][22][23]. Η συντριπτική πλειοψηφία των εν λόγω εφαρμογών αποκτάται τμηματικά και συνήθως παρουσιάζουν ανομοιογένεια αναφορικά με τις τεχνολογικές υποδομές που χρησιμοποιούν.

Με την πάροδο του χρόνου τα πληροφοριακά συστήματα γίνονται ολοένα και πιο απαραίτητα, τόσο για την απρόσκοπτη λειτουργία των επιχειρησιακών διαδικασιών, όσο και για την ταχύτερη διεκπεραίωσή τους, όπως πολύ εύστοχα διαπίστωσαν οι Philippe Lalanda, Luc Bellissard και Roland Balter [12]. Από τη μέχρι τώρα περιγραφή, ο αναγνώστης μπορεί να διαπιστώσει ότι παρουσιάζονται σημαντικά μειονεκτήματα [21][22][23]:

- Οι εφαρμογές που χρησιμοποιούνται σε καθημερινή χρήση από τους συνεργάτες / υπαλλήλους των οργανισμών / επιχειρήσεων, διαχειρίζονται μεμονωμένες πληροφορίες που διατηρούνται σε συγκεκριμένη μορφή και δομή με τρόπο που απαγορεύει τη διάχυση της αποθηκευμένης πληροφορίας σε τρίτα συστήματα [21][22][23],
- Δεν παρέχεται κεντρικός έλεγχος [23],
- Δεν υπάρχει τρόπος ενοποίησης της πληροφορίας που διαχειρίζονται μεμονωμένα οι εφαρμογές, πληροφορία που συνθέτει τη γνώση (know how) που υπάρχει σε κάθε οργανισμό / επιχείρηση [22].

Τα μειονεκτήματα που αναφέρθηκαν δυσκολεύουν την κεντρική διαχείριση (planning) των επιχειρησιακών διαδικασιών και σχεδόν εξαλείφουν τη δυνατότητα λήψης αποφάσεων (decision-making) από το management [24], αποτελώντας σημαντικό εμπόδιο στην ανάπτυξη και επέκταση των οργανισμών και επιχειρήσεων [12][21][22][23].

Αρκετοί οργανισμοί / επιχειρήσεις, προκειμένου να εξέλθουν από το αδιέξοδο της αποκεντρωμένης πληροφορίας, οδηγούνται στην υλοποίηση μιας ενιαίας εφαρμογής που καλύπτει όλες τις ανάγκες τους, συνδυάζει ή / και συνθέτει τη μεμονωμένη πληροφορία που υπάρχει σε κάθε τμήμα και προσωρινά λύνει το θέμα της κεντρικής διαχείρισης και λήψης αποφάσεων. Η ιστορία όμως έχει αποδείξει ότι η εν λόγω προσέγγιση παρουσιάζει κρίσιμα μειονεκτήματα που τις περισσότερες φορές οδηγεί σε ανεπιθύμητα αποτελέσματα [12]. Πιο συγκεκριμένα:

- Η ανάπτυξη πολύ ειδικευμένων / custom εφαρμογών τις περισσότερες φορές δεν προβλέπει τη δημιουργία νέων αναγκών, με αποτέλεσμα να έχουμε σαν σύνθετες φαινόμενο την εξολοκλήρου δημιουργία του.

- Η επέκταση αυτών των εφαρμογών στη συντριπτική τους πλειοψηφία εγκλωβίζει τον οργανισμό / επιχείρηση στη συνεργασία με συγκεκριμένο ανάδοχο που έχει πλέον το μονοπώλιο. Χάνεται έτσι η δυνατότητα ύπαρξης ανταγωνιστικότητας που θα έκανε τον οργανισμό / επιχείρηση πιο δυνατό στις προκλήσεις [24],
- Οποιαδήποτε επιχειρησιακή αλλαγή σε κάποιο τμήμα ενδέχεται να επηρεάσει και τα υπόλοιπα τμήματα [24],
- Η εφαρμογή σε συστήματα που διαχειρίζονται κρίσιμα και ενδεχομένως ευαίσθητα δεδομένα (π.χ. σύστημα τραπεζικών συναλλαγών) είναι απαγορευτική. Οι επιχειρήσεις / οργανισμοί τέτοιου κύρους δύσκολα θα προβούν σε αλλαγή των back-end συστημάτων τους που λειτουργούν σωστά, είναι δοκιμασμένα και τα εμπιστεύονται.

Όπως πολύ εύστοχα αναφέρει η Julie Gable [13] «Ζούμε σε μια εποχή που τα συστήματα που υλοποιούνται για να προσαρμόζονται είναι περισσότερο χρήσιμα και πολύτιμα από τα συστήματα που υλοποιούνται για να έχουν μεγάλη διάρκεια ζωής».

Λύση στο αδιέξοδο υπόσχονται να δώσουν οι λύσεις Enterprise Application Integration (εφεξής EAI) [7][23]. Υπάρχουν δύο είδη EAI:

- **Mediation:** Στην περίπτωση του mediation, τα EAI συστήματα δρουν ως ενδιάμεσοι πολλών εφαρμογών. Οποτεδήποτε ένα συμβάν λάβει χώρα (π.χ., είσοδος νέας πληροφορίας, ολοκληρωμένες συναλλαγές κ.λπ.) ένα υποσύστημα του EAI ενημερώνεται. Το υποσύστημα έπειτα διαδίδει τις αλλαγές που προέκυψαν από το συμβάν σε άλλες συνδεδεμένες εφαρμογές. Έτσι μπορούμε ταυτόχρονα να έχουμε πολλά διαφορετικά συστήματα ενημερωμένα με τα ίδια δεδομένα [7][23].
- **Federation:** Στην περίπτωση του federation, τα EAI συστήματα δρουν ως μοναδικό επίπεδο επικοινωνίας με τα εξωτερικά συστήματα και εφαρμογές που είναι δηλωμένα στο EAI. Επιπλέον το επίπεδο EAI αποτελεί κατά κάποιο τρόπο ένα τείχος προστασίας, διαχειρίζοντας τη συναλλασσόμενη πληροφορία και περιορίζοντας την πρόσβαση μόνο σε ένα υποσύνολο της πληροφορίας που ορίζουν οι διαχειριστές του [7][23].

Στην πράξη τις περισσότερες φορές εφαρμόζεται ο συνδυασμός των 2 ειδών, οι EAIs μπορεί να συγχρονίζουν πολλές εφαρμογές αλλά ταυτόχρονα να αποτελούν ενδιάμεσο επίπεδο επικοινωνίας με εφαρμογές εξωτερικών χρηστών [7].

Το συγκριτικό πλεονέκτημα που προσφέρουν οι EAIs γίνεται αντιληπτό αν αναλογιστεί κανείς τη μορφή με την οποία υφίστανται τα δεδομένα στις διάφορες εφαρμογές και συστήματα, τα οποία καλείται να ενοποιήσει ένα EAI σύστημα [23]. Συνήθως τα προς ενοποίηση δεδομένα δεν βρίσκονται μορφοποιημένα στην επιθυμητή μορφή και δομή, γεγονός που σηματοδοτεί την ανάγκη επεξεργασίας πριν τη μεταφορά για ενοποίηση [14][7][13][21].

Επιπλέον, τις περισσότερες φορές, κατά τη διάρκεια της ενοποίησης διαπιστώνεται από τους αναλυτές ότι η ύπαρξη του συνόλου των δεδομένων από όλες

τις διαθέσιμες βάσεις, όχι απλά δεν εξυπηρετεί, αλλά μάλλον οδηγεί στην ύπαρξη περιττής πληροφορίας, που μπορεί να οδηγήσει σε αλόγιστη σπατάλη πόρων συστήματος. Πιο συγκεκριμένα τα συνήθη προβλήματα που εμφανίζονται είναι τα ακόλουθα [14][7][13][21]:

- Ύπαρξη δεδομένων από διαφορετικές βάσεις (τύποι και κατασκευαστές ίσως ακόμα και εκδόσεις του ίδιου προϊόντος π.χ. να έχουμε από την μία SQL Server 2000 και από την άλλη τον SQL Server 2008 που είναι δύο γενιές μετά) και διαφορετικούς πίνακες.
- Πληροφορία που διατηρείται αποθηκευμένη σε XML αρχεία, είναι δύσκολο να χρησιμοποιηθεί αν δεν υπάρχει συγκεκριμένο πρόγραμμα που να μπορεί αν διαβάσει τα δεδομένα, να αποτυπώσει την δομή τους για να γίνουν οι απαραίτητες διαδικασίες επεξεργασίας.
- Ανάγκη για λήψη ενός υποσυνόλου των δεδομένων βάσει συγκεκριμένων συνθηκών, κάτι που σημαίνει παράλληλα την ύπαρξη ενός κοινού κώδικα για την υποστήριξη και βάσεων και xml αρχείων.
- Ανάγκη για απευθείας αποστολή των δεδομένων σε κάποιο Web Service.
- Πολλές φορές χρειάζεται να χρησιμοποιήσουμε παραπάνω από ένα πρόγραμμα για την επίτευξη όλων των παραπάνω. Αυτό αμέσως σημαίνει έξτρα κόστος για την αγορά των διαφόρων προγραμμάτων, κόστος με την μορφή εργατο-ωρών που θα δαπανηθούν λόγω εκμάθησης αυτών, και άλλα κόστη που προκύπτουν από τέτοιες διαδικασίες.

Με την παρούσα μελέτη, ο συγγραφέας αποσκοπεί στο να καλύψει τις παραπάνω ανάγκες. Το πρόγραμμα Ultimate Mediator, που υλοποιήθηκε στα πλαίσια της μελέτης, δίνει τη δυνατότητα επιλογής, συνδυασμού, μετατροπής, και μεταφοράς δεδομένων, μέσα από ειδικές και εύχρηστες φόρμες σχεδίασης.

Ο Ultimate Mediator υποστηρίζει τη μεταφορά πληροφορίας, μέσα από βάσεις xml και web services. Όσον αφορά τις βάσεις υποστηρίζονται:

- Oracle,
- Microsoft SQL Server,
- MySQL,
- Microsoft Access.

Η μεταφορά των δεδομένων υποστηρίζεται με τις ακόλουθες περιπτώσεις:

- Από βάσεις δεδομένων σε αρχεία xml,
- Από αρχεία xml σε βάσεις,
- Από xml αρχεία σε web services
- Από απλά δεδομένα κειμένου σε web services

Σκοπός του προγράμματος είναι να μπορέσει ένας χρήστης, με ελάχιστες τεχνικές γνώσεις σε θέματα Migration, να πραγματοποιήσει εύκολα και γρήγορα τα ακόλουθα:

- Να χρησιμοποιήσει μία μόνο ολοκληρωμένη εφαρμογή, που να περιέχει όλες εκείνες τις λειτουργίες που περιγράφηκαν νωρίτερα.
- Να μην χρειαστεί να κάνει χρήση άλλων διαφορετικών προγραμμάτων για την κάλυψη των αναγκών του.
- Να κάνει εύκολη την χρήση της εφαρμογής χωρίς πολύ εξειδικευμένες και χρονοβόρες διαδικασίες.
- Να πραγματοποιήσει την μεταφορά δεδομένων όσο το δυνατόν πιο απλά και εύκολα.
- Να έχει τη δυνατότητα επιλογής διαφόρων βάσεων και xml αρχείων ως είσοδο επιλογής των τελικών δεδομένων που θα εξαχθούν,
- Να μπορεί να αποστέλλεται σε πραγματικό χρόνο το παραγόμενο xml αρχείο σε κάποιο web service,
- Να υπάρχει η δυνατότητα επαναλαμβανόμενης αποστολής των xml αρχείων ή δεδομένων που λαμβάνει σαν απάντηση το λογισμικό στο ίδιο ή σε άλλο web service.
- Να μπορεί να αποθηκεύσει όσες φορές θέλει το xml αρχείο ή το επιστρεφόμενο αποτέλεσμα ενός web service,
- Να έχει την δυνατότητα να επεξεργαστεί την απάντηση του web service πριν την στείλει σε άλλο ή την αποθηκεύσει σε αρχείο.

Όλα τα παραπάνω επιτυγχάνονται με τη χρήση μίας σειράς από τεχνολογίες που υπάρχουν και είναι διαθέσιμες για να χρησιμοποιηθούν. Οι τεχνολογίες αυτές θα παρουσιαστούν σε άλλη ενότητα αναλυτικά για να φανούν τα πλεονεκτήματα που απορρέουν από την χρήση τους.

Ευχαριστίες

Η εκπόνηση της πτυχιακής αυτής χρειάστηκε ένα σημαντικό χρονικό διάστημα και αρκετή δουλειά για την επιτυχή ολοκλήρωσή της. Τα τέσσερα έτη του προπτυχιακού και τα δύο έτη του μεταπτυχιακού, του τμήματος Διδακτική της Τεχνολογίας και Ψηφιακών Συστημάτων, με βοήθησαν να αποκτήσω σωστά θεμέλια γνώσης και ακαδημαϊκής εμπειρίας.

Θα ήθελα να ευχαριστήσω όλους τους καθηγητές του τμήματος που στάθηκαν δίπλα μου και κυρίως τον Αναπληρωτή Καθηγητή κύριο Δεμέστιχα Παναγιώτη, που με βοήθησε στην εκπόνηση της διπλωματικής μου διατριβής.

Επίσης θέλω να ευχαριστήσω θερμά και τους συνεργάτες μου στην εταιρία EXIS I.T, που με στήριξαν σε όλα τα προβλήματα που συνάντησα και με βοήθησαν στην επαγγελματική μου εξέλιξη. Ονομαστικά από τους συνεργάτες μου θα ήθελα να αναφέρω τον κ. Κοτρώτσο Σεραφείμ, που με καθοδήγησε σε κάθε δύσκολο σημείο που προέκυψε, κατά την συγγραφή του κώδικα και του κειμένου της διπλωματικής, δίνοντας μου όλα τα κίνητρα και την απαραίτητη γνώση για την επιτυχή ολοκλήρωσή της.

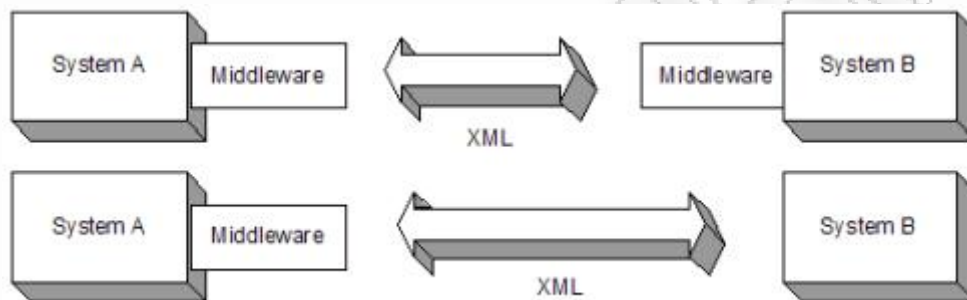
1 Βιβλιογραφική Ανασκόπηση

Σε αυτό το κεφάλαιο θα παρουσιαστεί μία βιβλιογραφική ανασκόπηση αναφορικά με τη Διαλειτουργικότητα και πιο συγκεκριμένα θα γίνει αναφορά στις έννοιες του Mediation και του Data Integration μέσω των EAI (Enterprise Application Integration) συστημάτων που αναφέρθηκαν στην *Εισαγωγή* [21][22][23]. Θα δοθεί ο ορισμός τους, ο βαθμός σημαντικότητάς τους και παράλληλα θα αναφερθούν οι εφαρμογές τους στα σημερινά συστήματα και στις λύσεις που παρέχονται από την επιστήμη της Πληροφορικής. Στη συνέχεια θα ακολουθήσει ανάλυση αναφορικά με τις προσπάθειες που έχουν γίνει για την ενσωμάτωση αυτών των εννοιών σε εφαρμογές και θα παρουσιαστούν σημαντικά προγράμματα που προσφέρουν λύσεις στον τομέα αυτό.

Επιπλέον, τα παραπάνω προγράμματα θα αναλυθούν περαιτέρω και θα παρουσιαστούν σε ξεχωριστή ενότητα, όπου και θα αποσαφηνιστούν τα προβλήματα που παρουσιάζουν και δυσχεραίνουν ή / και καθιστούν αδύνατη τη χρήση τους από ένα μεγάλο ποσοστό των οργανισμών / εταιριών που απευθύνονται [21][22][23][24].

1.1 Διαλειτουργικότητα Συστημάτων

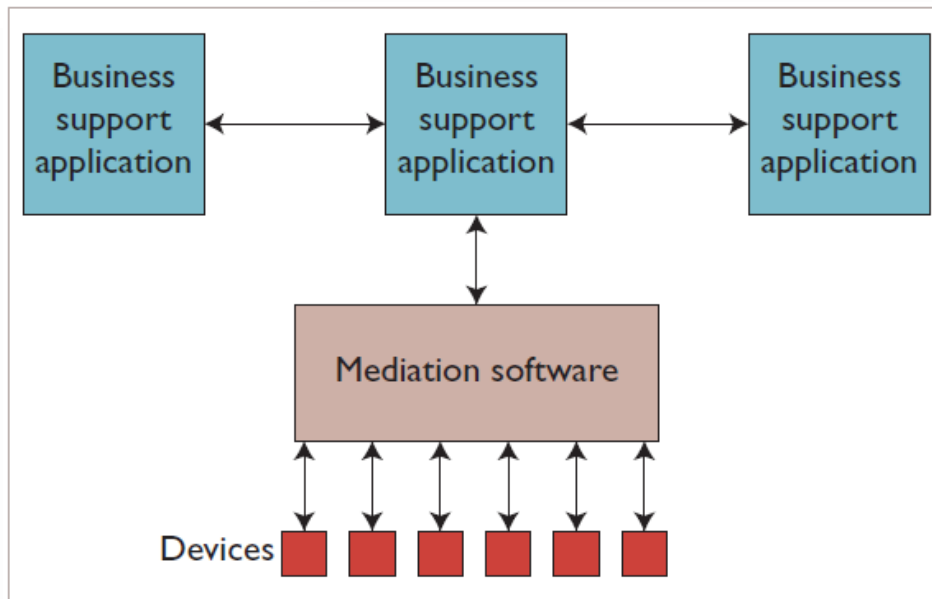
Ως Διαλειτουργικότητα Συστημάτων ορίζεται η δυνατότητα μεταφοράς, ενοποίησης και χρήσης της πληροφορίας με ενιαίο τρόπο από διαφορετικά πληροφοριακά συστήματα [1][23]. Είναι ένα από τα πιο κρίσιμα θέματα που απασχολούν τις σημερινές επιχειρήσεις και οργανισμούς, που χρειάζονται να έχουν πρόσβαση σε πληροφορίες από πολλαπλά και κυρίως ετερογενή συστήματα [23][24]. Για το λόγο αυτό χρησιμοποιούνται ενδιάμεσα επίπεδα / λογισμικά προκειμένου να επιτευχθεί η διαμόρφωση και τελική μεταφορά των δεδομένων, όπως παρουσιάζεται στο σχήμα της Εικόνα 1.1.1.



Εικόνα 1.1.1 Ρόλος του ενδιάμεσου επιπέδου / λογισμικού Διαλειτουργικότητας [11]

Στην παραπάνω εικόνα παρουσιάζεται πως ένα Σύστημα A μπορεί να λάβει και να στείλει δεδομένα με την χρήση της XML γλώσσας σε ένα άλλο Σύστημα B. Τα δεδομένα αυτά δεν είναι αναγκαίο να είναι ακριβώς όπως ήταν κατά την λήψη τους. Μπορούν να εφαρμοστούν, όπως θα δούμε και παρακάτω, κανόνες και σχέσεις σε αυτά όπως επίσης και να πραγματοποιηθεί κάποιο είδος επεξεργασίας πάνω τους.

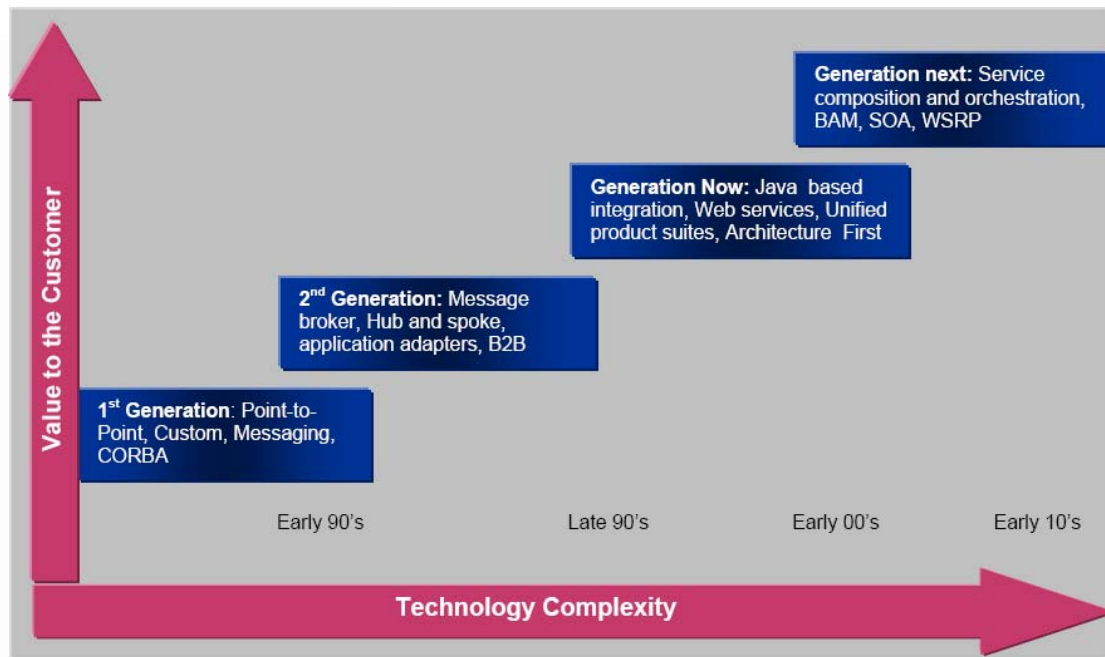
Η μεταφορά δεδομένων δεν στηρίζεται κατ' ανάγκη σε αρχεία xml ή σε WebServices. Μπορεί να πραγματοποιηθεί εκτελώντας τη μεταφορά δεδομένων μέσα από τις διάφορες συνδέσεις που δύναται να πραγματοποιηθούν χρησιμοποιώντας οποιοδήποτε από τα γνωστά πρωτόκολλα επικοινωνίας. Η Εικόνα 1.1.2 δείχνει ένα πιο γενικό μοντέλο που δεν περιορίζεται στην αποστολή αρχείων xml, αλλά λαμβάνει και αποστέλλει δεδομένα μέσω των συνδέσεων που έχουν οριστεί.



Εικόνα 1.1.2 Γενικότερη σύνδεση ενδιάμεσου λογισμικού με τα υπόλοιπα συστήματα[12].

Βασικό ρόλο στη ραγδαία ανάπτυξη του τομέα της Διαλειτουργικότητας διαδραμάτισε τα τελευταία χρόνια το EAI (Enterprise Application Integration) [21][22][23], που έχει βγάλει από το αδιέξοδο πολλές εταιρίες και οργανισμούς, κάνοντάς τους πιο ευέλικτους και πιο υπεύθυνους, σε ό,τι αφορά στις απαιτήσεις της αγοράς [24]. Ιδιαίτερα με την εμφάνιση του παγκόσμιου ιστού, οι εταιρίες αναγκάστηκαν να αυτοματοποιήσουν[22] και να ενοποιήσουν τις εργασίες τους και ταυτόχρονα τις να αναβαθμίσουν τις λειτουργίες των υπολογιστικών τους συστημάτων. Αυτό γιατί συχνά έρχονται αντιμέτωπες με το πρόβλημα των διαφορετικών εφαρμογών και των διαφορετικών βάσεων δεδομένων. Το EAI επιτρέπει σε διαφορετικά συστήματα να συνδεθούν μεταξύ τους γρήγορα, έτσι ώστε να μοιραστούν δεδομένα και διεργασίες, να επικοινωνήσουν και να απομακρύνουν άχρηστες πληροφορίες που δυσχεραίνουν το έργο πολλών επιχειρήσεων / οργανισμών [14][7][24].

Το EAI, όπως είναι γνωστό σήμερα, έχει περάσει από πολλά διαφορετικά στάδια εξέλιξης. Το παρακάτω διάγραμμα Εικόνα 1.1.3 παρουσιάζει τις διάφορες «γενιές» του EAI:



Εικόνα 1.1.3 Οι γενιές του EAI σε κλίμακα χρόνου [15]

1η Γενιά: Age of Point – to – Point

Στην περίοδο αυτή, το EAI μόλις είχε γίνει γνωστό, με τα προϊόντα ενοποίησης (integration) και τις τεχνικές να στοχεύουν στην παροχή APIs και διεπαφών μεταξύ των συστημάτων [15]. Αυτή η διαδικασία πραγματοποιούνταν είτε μέσω απομακρυσμένων κλήσεων (Remote Procedure Calls-RPCs), είτε με τη βοήθεια των διαδεδομένων προτύπων (π.χ. CORBA), είτε μέσω προϊόντων αποστολής μηνυμάτων [15]. Το RPC και το CORBA είχαν ακριβώς τον ίδιο στόχο με διαφορετικό τρόπο υλοποίησης το καθένα. Το CORBA, όντας πιο δυνατό από ένα απλό RPC, ενσωμάτωνε συστήματα με τη βοήθεια της αίτησης-απάντησης [15].

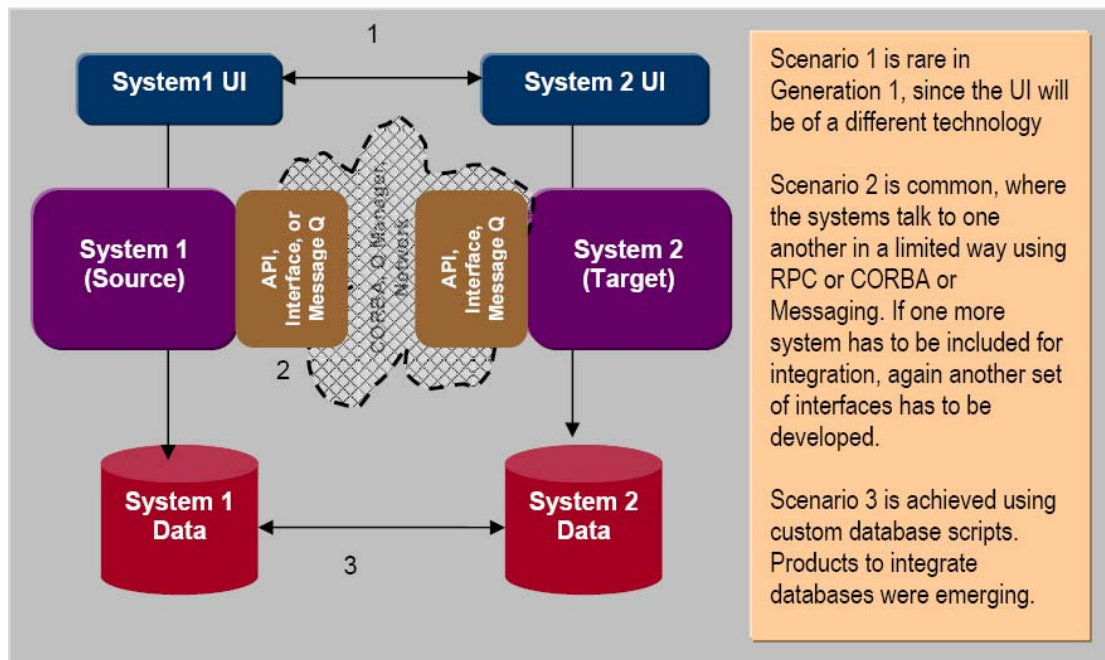
Συνοπτικά, τα κυριότερα επιτεύγματα της πρώτης γενιάς «integration» είναι τα ακόλουθα [15]:

1. Ενσωμάτωση εφαρμογών σε άλλα ανοιχτά συστήματα (βασισμένα σε περιβάλλοντα UNIX και Windows)
2. Μεγάλης κλίμακας integrations projects προήλθαν από τη χρήση του CORBA

Παρακάτω παρουσιάζονται τα πιο σημαντικά μειονεκτήματα της συγκεκριμένης Γενιάς [15]:

1. Απαιτήσεις μεγάλου όγκου κώδικα
2. Απουσία broker μηνυμάτων

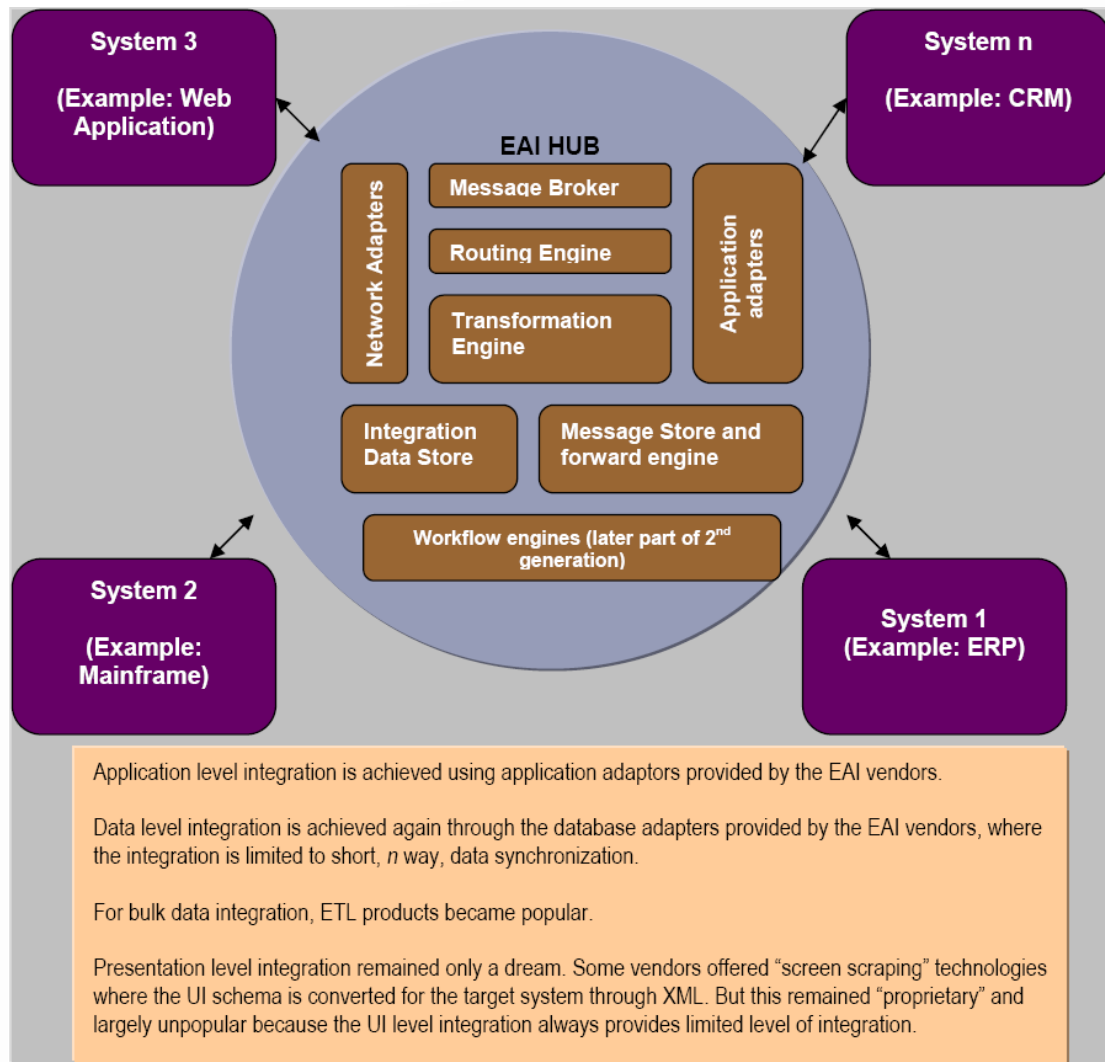
Το ακόλουθο διάγραμμα (Εικόνα 1.1.4) δείχνει τα διαφορετικά επίπεδα της πρώτης γενιάς υπολογιστών [15].



Εικόνα 1.1.4 Τα διαφορετικά επίπεδα της πρώτης γενιάς υπολογιστών [15]

2^η Γενιά: Age of Many – to – Many

Σε αυτήν την περίοδο, τα integration προϊόντα έφεραν την έννοια του message broker και προετοίμασαν το έδαφος για $n \times n$ integration εφαρμογές [15]. Ο message broker ήταν υπεύθυνος για τη μετάφραση μηνυμάτων μεταξύ των συστημάτων και τη δρομολόγηση αυτών στις κατάλληλες εφαρμογές (Εικόνα 1.1.5) [15]. Οι κανόνες δρομολόγησης και χαρτογράφησης, ορίζονταν μέσα στον broker μέσω της χρήσης visual application editor, ενώ η επεξεργασία μέσα στους brokers γινόταν μέσω της χρήσης scripting γλωσσών προγραμματισμού [15].



Εικόνα 1.1.5 Message Broker και Δεύτερη γενιά [15]

Πλεονεκτήματα δεύτερης Γενιάς [15]:

1. Τα συστήματα-εφαρμογές που συμμετέχουν στο «integration» προστατεύουν το ένα το άλλο.
2. Η λογική του integration (για παράδειγμα κανόνες δρομολόγησης) μπορούν να επαναχρησιμοποιηθούν και να μοιραστούν μεταξύ των πολλαπλών Projects.
3. Η λογική του integration μπορεί να μοντελοποιηθεί μέσω των visual editors με drag and drop χαρακτηριστικά και αυτό έχει σαν αποτέλεσμα να επιταχύνονται οι παραδόσεις μεταξύ των EAI projects.
4. Επίτευξη συγχρονισμού δεδομένων πραγματικού χρόνου

Μειονεκτήματα δεύτερης Γενιάς [15]:

1. Απαίτηση proprietary γλώσσας για την κωδικοποίηση – Αυτό άλλαξε κάπως κατά το τέλος της συγκεκριμένης γενιάς γιατί τα integration λογισμικά υιοθέτησαν την Java και την XML όπου οι συγκεκριμένες τεχνολογίες είχαν γίνει τότε ευρέως γνωστές.
2. Άλλη μία λιγότερο γνωστή επίδραση είναι ότι το EAI προϊόν έγινε το κέντρο όλων των εφαρμογών σε μία επιχείρηση. Το μέλλον μιας integration επιχείρησης εξαρτιόταν πλέον αποκλειστικά από τη δύναμη και τις προοπτικές που είχε ένας EAI πωλητής (vendor).
3. Δεν υιοθέτησαν όλοι οι vendors την οργανωσιακή μέθοδο εξέλιξης του (ενοποιημένου) integration λογισμικού. Σε πολλές περιπτώσεις οι integration brokers αγοράστηκαν και ενσωματώθηκαν σε εσωτερικά συστήματα αποστολής μηνυμάτων, γεγονός που οδήγησε σε μία περίοδο σύγχυσης που αφορούσε τη λογική τοποθέτηση, τον διπλασιασμό των υποδομών και τη συχνή αναβάθμιση των προϊόντων.

Η δεύτερη γενιά του EAI βοήθησε σε μεγάλο μέρος την προώθηση και την παγίωση των συναλλαγών του ηλεκτρονικού εμπορίου στο Διαδίκτυο [15]. Για παράδειγμα στη διαδικασία της κράτησης σε ένα online κατάστημα, ο κατάλογος του προϊόντος μπορεί να ελεγχθεί με ένα ERP σύστημα που χρησιμοποιεί EAI, και τότε η παραγγελία μπορεί να δρομολογηθεί βασιζόμενη στο σύστημα [15]. Έτσι, οι πληροφορίες της παραγγελίας, που μόλις πραγματοποιήθηκε, μπορούν να μεταφερθούν στην αποθήκη εμπορευμάτων [15]. Έτσι ο πελάτης έχει τη δυνατότητα να ενημερωθεί για την παραγγελία χρησιμοποιώντας EAI σε CRM εφαρμογές [15]. Στην πραγματικότητα, η δεύτερη γενιά έτεινε να γεφυρώσει το χάσμα μεταξύ του εσωτερικού της επιχείρησης και του εξωτερικού της περιβάλλον [15].

Generation Now: Age of Consolidation and Proliferation

Στη σημερινή γενιά πολλοί από τους πιο γνωστούς EAI vendors έχουν παγίσει τα προϊόντα τους και προσφέρουν Portal, Integration Server και Workflow Management σαν μία ακολουθία προϊόντων [15]. Κατά συνέπεια, αυτό οδηγεί την επιχείρηση να επενδύει πιο εύκολα σε τέτοιες τεχνολογίες πωλητών [15].

Επιπλέον η Java έχει εισέλθει στην επικρατούσα τάση του integration [15]. Οι application server vendors έχουν εισχωρήσει στο χώρο του integration και προσφέρουν μια ενοποιημένη ακολουθία προϊόντων: server application, integration, και workflow [15]. Τέλος η εμφάνιση του Service Oriented Architecture (SOA) και των Web Services έχουν οδηγήσει στην επανεξέταση του integration μεταξύ των ιθυνόντων. Η χρήση των Web Services σε αντίθεση με το σχετικά φτωχό ρόλο του integration έχει συζητηθεί πολύ πριν ένα integration project ξεκινήσει [15].

Πλεονεκτήματα Generation Now [15]:

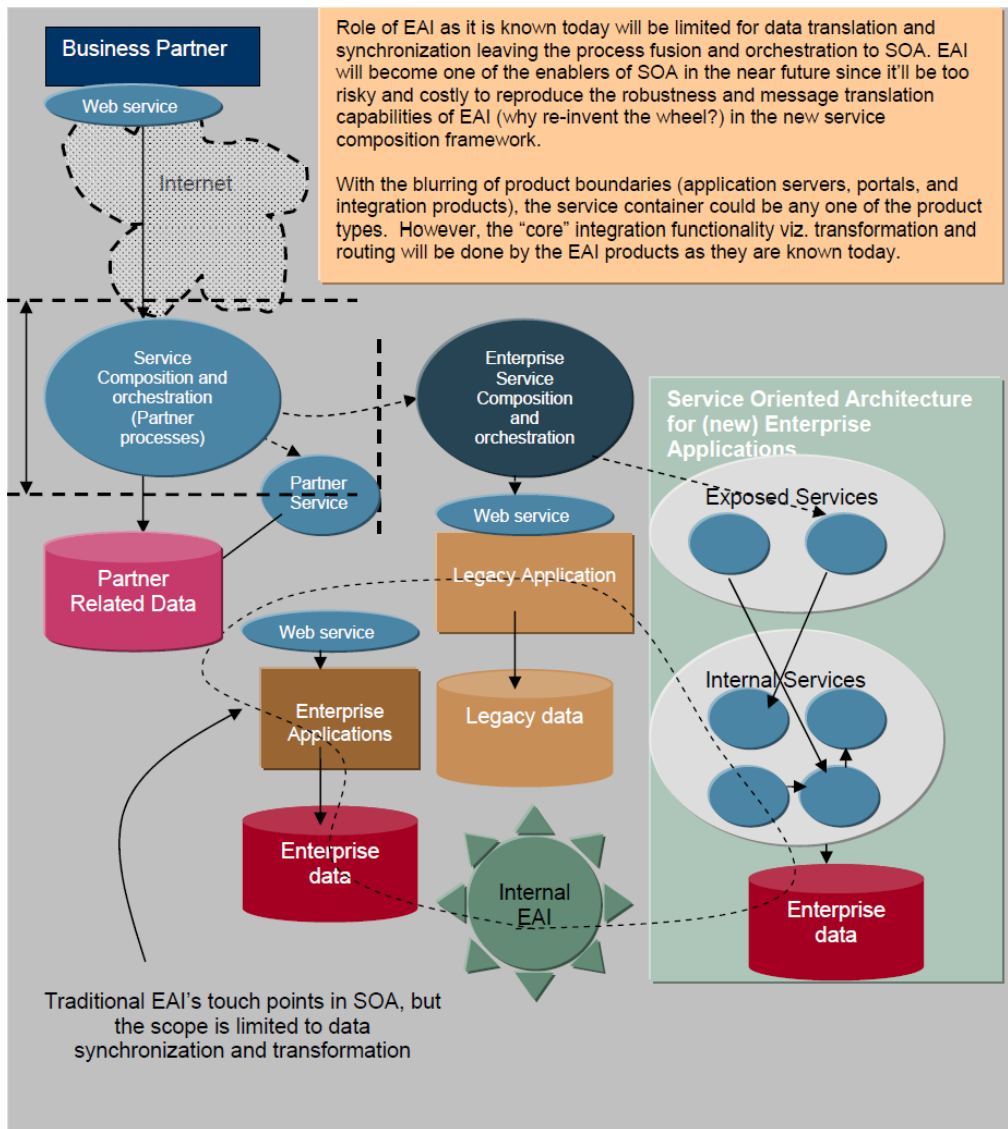
1. Το integration είναι πιο βελτιωμένο με πολλά είδη του (pure play, J2EE – based and web services - based) να έχουν προκύψει, προσφέροντας καλές ευκαιρίες και λύσεις στην επιχείρηση.
2. Υπάρχουσες EDI επενδύσεις που είναι ιδιοκτησία του EDIINT (Electronic Data Interchange Internet Integration) δεν χρειάζεται να απομακρυνθούν από την επιχείρηση.
3. Έχοντας βελτιωμένη τεχνολογία και προηγμένα εργαλεία μια επιχείρηση είναι εύκολο να συγκεντρώσει λύσεις Integration.

Generation Next

Η διαδικασία της ενοποίησης – integration, στην σημερινή εποχή έχει οριστεί ως μία γρήγορη λύση σε προβλήματα που παρουσιάζονται στις επιχειρήσεις / οργανισμούς, ακόμα και αν η λύση αυτή αναπτυχθεί σε υψηλό επίπεδο έτσι ώστε να καλύπτει ένα ευρύ φάσμα δυνατοτήτων [15].

Αυτή η διαδικασία όμως επρόκειτο να αλλάξει μέσα σε αυτήν την γενιά και ο λόγος είναι ότι οι ανησυχίες ολοκλήρωσης, θα εξεταστούν από την αρχή της επιχειρησιακής διαδικασίας και όχι από το τέλος της [15]. Αυτό προϋποθέτει και από το integration σύστημα να περιέχει υπηρεσίες καλά διαμορφωμένες, έτσι ώστε να ορίζονται ξεκάθαρα οι υπηρεσίες που χρησιμοποιούνται στο SOA μοντέλο της επιχείρησης. Επίσης θα πρέπει οι υπηρεσίες να μπορούν να επικοινωνήσουν με όλες τις πιθανές πλατφόρμες οι οποίες θα μπορούν να στέλνουν μηνύματα σε XML [15].

Με αυτόν τον τρόπο θα έχουμε μία διαλειτουργική σύνθεση υπηρεσιών που θα πετυχαίνει τους τελικούς στόχους. Ένα παράδειγμα του μοντέλου αυτής της γενιάς παρουσιάζεται στην Εικόνα 1.1.6 [15].



Εικόνα 1.1.6 Παράδειγμα μοντέλου γενιάς Generation Next [15].

Το EAI διαχωρίζεται σε 2 είδη, ανάλογα με τον τρόπο λειτουργίας του, κυρίως αναφορικά με τη διασύνδεσή του και τον τρόπο αλληλεπίδρασης με τις εξωτερικές / τρίτες εφαρμογές [15][7][23]:

- **Mediation:** Στην περίπτωση του mediation, τα EAI συστήματα δρουν ως ενδιάμεσοι πολλών εφαρμογών. Οποτεδήποτε ένα συμβάν λάβει χώρα (π.χ., είσοδος νέας πληροφορίας, ολοκληρωμένες συναλλαγές κ.λπ.) ένα υποσύστημα του EAI ενημερώνεται. Το υποσύστημα έπειτα διαδίδει τις αλλαγές που προέκυψαν από το συμβάν σε άλλες συνδεδεμένες εφαρμογές. Έτσι μπορούμε ταυτόχρονα να έχουμε πολλά διαφορετικά συστήματα ενημερωμένα με τα ίδια δεδομένα[7][23].
- **Federation:** Στην περίπτωση του federation, τα EAI συστήματα δρουν ως μοναδικό επίπεδο επικοινωνίας με τα εξωτερικά συστήματα και εφαρμογές που είναι δηλωμένα στο EAI. Επιπλέον το επίπεδο EAI αποτελεί κατά κάποιο

τρόπο ένα τείχος προστασίας, διαχειρίζοντας τη συναλλασσόμενη πληροφορία και περιορίζοντας την πρόσβαση μόνο σε ένα υποσύνολο της πληροφορίας που ορίζουν οι διαχειριστές του [7][23].

Στην πράξη τις περισσότερες φορές εφαρμόζεται ο συνδυασμός των 2 ειδών, οι EAIs μπορεί να συγχρονίζουν πολλές εφαρμογές αλλά ταυτόχρονα να αποτελούν ενδιάμεσο επίπεδο επικοινωνίας με εφαρμογές εξωτερικών χρηστών. Η παρούσα μελέτη δεν αποσκοπεί στη διάκριση των 2 ειδών EAI συστημάτων. Αντιθέτως η προσπάθεια που έγινε βασίστηκε σε ένα υβριδικό μοντέλο που συνδυάζει και τις δύο τεχνικές (Mediation/Federation). Εφεξής αναφορές θα γίνονται μόνο στον όρο Mediation και θα υπονοείται ο συνδυασμός των δύο παραπάνω τεχνικών [14][7][22][23].

Με τον όρο Mediation (μεσολάβηση) ορίζεται η διαδικασία σύνδεσης ανόμοιων πληροφοριών, σε σύντομο χρονικό διάστημα. Το λογισμικό (mediator) στοχεύει στην συλλογή δεδομένων, περιοδικά ή σε συγκεκριμένη χρονική στιγμή, από διαφορετικά συστήματα, δίνοντάς τα σαν είσοδο σε επιχειρηματικές διαδικασίες / προγράμματα [1][20][8]. Η χρήση της ενσωμάτωσης των διανεμημένων, ετερογενών δεδομένων αποτελεί καλή πρακτική που χρησιμοποιούν πολλές επιχειρήσεις / οργανισμοί σήμερα και οφείλεται στο γεγονός ότι μπορεί να επικοινωνήσει με πολλά διαφορετικά συστήματα, εξάγοντας ροές δεδομένων σε άλλα, ελαχιστοποιώντας με τον τρόπο αυτό τις συνδέσεις που θα έπρεπε να πραγματοποιηθούν χωρίς τη χρήση του μεσολαβητή [8].

Οι μεσολαβητές μέχρι σήμερα έχουν χρησιμοποιηθεί από διάφορους κλάδους ως ενδιάμεσοι, και συγκεκριμένα έχουν χρησιμοποιηθεί από [8]:

- **Μηχανικούς Λογισμικού** για να μαζεύουν δεδομένα από:
 - IT πόρους όπως βάσεις δεδομένων,
 - βάσεις γνώσεων – knowledge databases,
 - αρχεία συστήματος,
 - ψηφιακές βιβλιοθήκες,
 - συστήματα ηλεκτρονικού ταχυδρομείου – email.
- **Βιομηχανία των Τηλεπικοινωνιών** για σκοπούς:
 - Τιμολόγησης – Το πρόγραμμα του μεσολαβητή συλλέγει δεδομένα χρήσης υπηρεσιών του συνδρομητή από ένα ιδιωτικό δίκτυο και έπειτα τα ελέγχει και τα μετατρέπει σε ειδικό format τιμολόγησης για να τα δρομολογήσει στο σύστημα τιμολόγησης του operator – χειριστή.

Έχουν γίνει αρκετές προσπάθειες ανάπτυξης υποδομών όπως αυτών που περιλαμβάνουν σημασιολογικές (semantics) διαδικασίες για την δημιουργία ενός μεσολαβητή [17][8]. Αυτή όμως η διαδικασία δημιουργεί κάποια πρόσθετα ζητήματα όπως αυτό του αυτοματισμού της διαδικασίας ελέγχου συγκρούσεων μεταξύ των ετερογενών πληροφοριών [17][8][22]. Για αυτόν τον λόγο πολλές φορές γίνεται η χρήση των οντολογιών, όπως θα δούμε και στην συνέχεια [17][8]. Τέλος οι ενδιάμεσοι / μεσολαβητές περιλαμβάνουν διαδικασίες με τις οποίες δημιουργούν ένα είδος ερωτήματος, για την λήψη και μεταφορά δεδομένων σε διαφορετικές βάσεις

[17][8]. Όλες αυτές όμως οι διαδικασίες στηρίζονται στο γεγονός ότι θα πρέπει να γίνει σωστός προσδιορισμός του σχήματος της βάσης, όπως και των οντολογιών για να έχουμε το επιθυμητό αποτέλεσμα [17][8].

Τα σημασιολογικά σχήματα χρησιμοποιούνται για να ορίσουν επακριβώς το είδος των δεδομένων [17][8]. Έστω για παράδειγμα ότι έχουμε μία τιμή «37». Η τιμή αυτή μπορεί να σημαίνει πάρα πολλά πράγματα, αναλόγως με την κατάσταση, το περιβάλλον, και τον τρόπο με τον οποίο χρησιμοποιείται [8]. Έτσι μπορεί να ορίζουμε με τον αριθμό αυτό, τη θερμοκρασία που έχει κάποιο υλικό, ή την θερμοκρασία δωματίου [8]. Η θερμοκρασία όμως μετριέται διαφορετικά για κάποιον επιστήμονα που μπορεί να κρατάει τιμές σε βαθμούς Kelvin και με έναν απλό πολίτη που για εκείνον μπορεί να σημαίνει βαθμοί Κελσίου [8].

Αντίστοιχα μπορούμε να πούμε ότι η τιμή αυτή χαρακτηρίζει την ηλικία ενός ατόμου [8]. Η ηλικία όμως μετριέται και αυτή διαφορετικά σε κάποιες ανατολικές χώρες από ότι σε κάποιες δυτικές [8]. Για παράδειγμα στην Κορέα μετράνε την ηλικία του παιδιού από το 1 την ώρα που γεννιέται, και αυξάνουν το έτος σε κάθε νέα χρονιά [8]. Έτσι αν το μωρό γεννηθεί στις 31 Δεκεμβρίου τότε την επόμενη μέρα θα είναι δύο ετών [8].

Αυτή η διαφοροποίηση δεν ισχύει μόνο για χώρες με διαφορετικά ήθη και έθιμα. Μπορεί να συμβεί και σε έναν οργανισμό ή επιχείρηση που βρίσκεται στην ίδια χώρα ακόμα και στην ίδια πόλη [8]. Μπορεί να υπάρξουν διαδικασίες που εκτελούνται μέσα στην ίδια εταιρία αλλά να ορίζουν διαφορετική σημασία για μία ή και περισσότερες έννοιες από αυτήν που ορίζουν κάποιες άλλες [17][8].

Από τα παραπάνω συμπεραίνει κανείς την ύπαρξη προβλημάτων σε οργανισμούς και επιχειρήσεις δεδομένης της δημιουργίας πολλών τέτοιων διαφορετικών σημασιολογικών περιπτώσεων για την ίδια έννοια, με αποτέλεσμα να παρατηρείται το φαινόμενο του πολλαπλασιασμού αυτών. Για την αποφυγή τέτοιων προβλημάτων έγινε μία προσπάθεια κατηγοριοποίησης της σημασιολογικής διαλειτουργικότητας στα ακόλουθα επιμέρους τμήματα [8][20]:

1. **Mapping-based:** Μία προσπάθεια για την δημιουργία συσχετίσεων μεταξύ των διαφορετικών εννοιών, έχοντας ορίσει το σημασιολογικό σχήμα κάπου κεντρικά, και έπειτα δημιουργώντας επιμέρους σημασιολογικές συσχετίσεις και οντότητες που λειτουργούν σαν ενδιάμεσοι των πρωταρχικών με τα δεδομένα.
2. **Intermediary-based:** Σε αυτήν την περίπτωση χρησιμοποιούνται ενδιάμεσοι μηχανισμοί για να επιτύχουμε την διαλειτουργικότητα. Τέτοιοι ενδιάμεσοι περιλαμβάνουν γνώση, ή κανόνες για τις συγκεκριμένες αλλαγές στα δεδομένα που μπορεί να προκύψουν. Πολλές φορές σε αυτό το στάδιο χρησιμοποιούνται οντολογίες για να χρησιμοποιηθεί συγκεκριμένο λεξικό επικοινωνίας ή γίνεται η χρήση πρωτοκόλλων για να επιτυγχάνεται η επικοινωνία μεταξύ τους.
3. **Query-oriented:** Σε αυτήν την περίπτωση γίνεται η χρήση ειδικών γλωσσών, οι οποίες έχουν το συστατικό της διαλειτουργικότητας όπως είναι η SQL. Με αυτόν τον τρόπο μπορούν να χρησιμοποιούν την ίδια γλώσσα για την πραγματοποίηση ερωτημάτων σε διαφορετικές βάσεις. Θα πρέπει όμως να μπορούν να αντιληφθούν τέτοιες σημασιολογικές συγκρούσεις δεδομένων που μπορεί να προκύψουν. Αυτό όμως δημιουργεί πρόβλημα στους χρήστες

οι οποίοι θα πρέπει να κατανοήσουν τις βάσεις δεδομένων για να μπορέσουν να επιτύχουν σωστή διαλειτουργικότητα.

Μπορεί να προκύψουν διάφορα είδη «συγκρούσεων» μεταξύ δεδομένων αλλά και μεταξύ σημασιολογικών εννοιών. Έχει γίνει ήδη μία προσπάθεια ως προς την κατηγοριοποίηση τους. Υφίστανται δύο είδη «συγκρούσεων» και περιέχουν υποκατηγορίες, όπως ορίζονται παρακάτω [8][20]:

1. Data Level: Αυτών των ειδών συγκρούσεις μπορεί να προκύψουν από διαφοροποιήσεις των δεδομένων που προκαλούν οι πολλαπλές αναπαραστάσεις και ερμηνείες που μπορούν να έχουν τα δεδομένα.

- **Data-value Conflicts:** Έννοιες που έχουν διαφορετικό νόημα ανάλογα με το που χρησιμοποιούνται.
- **Data-representation Conflicts:** όταν για παράδειγμα ο τρόπος παρουσίασης των δεδομένων διαφέρει (π.χ. ο τρόπος παρουσίασης της ημερομηνίας μπορεί για ένα σύστημα να είναι η έτος – μήνας - μέρα (όλα χαρακτηρισμένα με νούμερα), αλλά για κάποιο άλλο να είναι μέρα / μήνας / έτος έχοντας τον μήνα αποτυπωμένα με γράμματα).
- **Data-unit Conflicts:** Προβλήματα στο είδος αντιπροσώπευσης των τιμών των δεδομένων. Για παράδειγμα μπορεί την θερμοκρασία να την χρησιμοποιεί ένα σύστημα σε βαθμούς κελσίου ενώ ένα άλλο σε βαθμού Kelvin.
- **Data-precision Conflicts:** Όταν κάτι για παράδειγμα μπορεί να χαρακτηριστεί σε κλίμακα του «τέλειου», «καλού», «φτωχού», ενώ αντίστοιχα άλλη διαδικασία χρησιμοποιεί το «Α», «Β», «Γ».

2. Schema Level: Τέτοια προβλήματα συγκρούσεων παρουσιάζονται από τις διαφορές σε λογικές δομές ή / και λόγων ασυνεπειών σε μεταδεδομένα της ίδιας εφαρμογής. Υποκατηγοριοποιήσεις τέτοιων συγκρούσεων είναι οι ακόλουθες:

- **Naming Conflicts:** Αυτές προκύπτουν όταν η ονοματολογία των στοιχείων των σχημάτων, ορίζονται αυθαίρετα από τους σχεδιαστές τους, με αποτέλεσμα να μπορεί να έχουμε δύο διαφορετικές έννοιες ορισμένες στην ίδια οντολογία.
- **Entity-identifier Conflicts:** Αυτές προκαλούνται συνήθως όταν ορίζονται διαφορετικά κλειδιά π.χ. για κάποιον πίνακα, που χρησιμοποιούνται με την ίδια λογική (concept) σε διαφορετικές βάσεις.
- **Schema-isomorphism Conflicts:** Δημιουργούνται όταν η ίδια έννοια (concept), περιγράφεται από ένα ανάμοιο σύνολο ιδιοτήτων, δηλ., η ίδια έννοια αντιπροσωπεύεται από διάφορες

διαφορετικές ιδιότητες και, επομένως, τα σύνολα οντοτήτων δεν είναι συμβατά με το σύστημα και τις διαδικασίες.

- **Generalization Conflicts:** Προκύπτουν από τις διαφορετικές σχεδιαστικές επιλογές κατά την κατηγοριοποίηση των κλάσεων οντοτήτων. Αν για παράδειγμα έχουμε μία βάση η οποία ορίζει ένα σχήμα «Απόφοιτος», και ένα δεύτερο σχήμα «Προπτυχιακός Φοιτητής», τότε αν θέλουμε να ενοποιήσουμε την βάση αυτή με μία άλλη βάση που ορίζει μόνο μία έννοια αυτή του «Φοιτητής», τότε θα έχουμε δύο προβλήματα. Πρώτον θα πρέπει να ορίσουμε τις αντιστοιχίες μεταξύ των δύο σχημάτων και του τρίτου (της άλλης βάσης), και δεύτερον θα πρέπει να ορίσουμε κανόνες που θα μπορούν να ξεχωρίσουν ποιες από τις οντότητες μέσα στο σχήμα «Φοιτητές», ανήκουν σε ποιο σχήμα.
- **Aggregation Conflicts:** Αυτές οι συγκρούσεις προκύπτουν όταν μια συνάθροιση χρησιμοποιείται σε μια βάση δεδομένων για να προσδιορίσει ένα σύνολο οντοτήτων, σε μια άλλη βάση δεδομένων. Επομένως οι ιδιότητες και οι τιμές τους σε μια βάση δεδομένων, μπορούν να αθροίσουν τις αντίστοιχες ιδιότητες και τις τιμές του συνόλου οντοτήτων μιας άλλης βάσης δεδομένων.
- **Schematic discrepancies:** Μπορούν να προκύψουν, όταν η λογική δομή ενός συνόλου ιδιοτήτων και οι τιμές τους ανήκουν σε μία κλάση οντοτήτων της μίας βάσης δεδομένων, οργανώνονται για να διαμορφώσουν μία διαφορετική δομή μίας άλλης βάσης δεδομένων.

1.2 Υπάρχουσες Υλοποιήσεις

Σε αυτήν την ενότητα θα παρουσιαστούν εμπορικά λογισμικά που έχουν ήδη υλοποιηθεί και βοηθούν στη διεκπεραίωση της διαλειτουργικότητας μεταξύ διαφόρων συστημάτων. Με βάση τη Διαλειτουργικότητα Συστημάτων που παρουσιάστηκε σε προηγούμενη ενότητα, στη συνέχεια θα πραγματοποιηθεί προσπάθεια σύνδεσης, με τρόπο ώστε να γίνουν κατανοητά στον αναγνώστη τα πιθανά μειονεκτήματα που απορρέουν από την χρήση αυτών των προγραμμάτων. Παράλληλα θα αναφερθεί ο τρόπος με τον οποίο η χρήση του Ultimate Mediator θα συνδράμει στην επίλυση αυτών. Επιπλέον στοιχεία για την εφαρμογή Ultimate Mediator θα περιγραφούν και θα παρουσιαστούν σε επόμενη ενότητα της μελέτης.

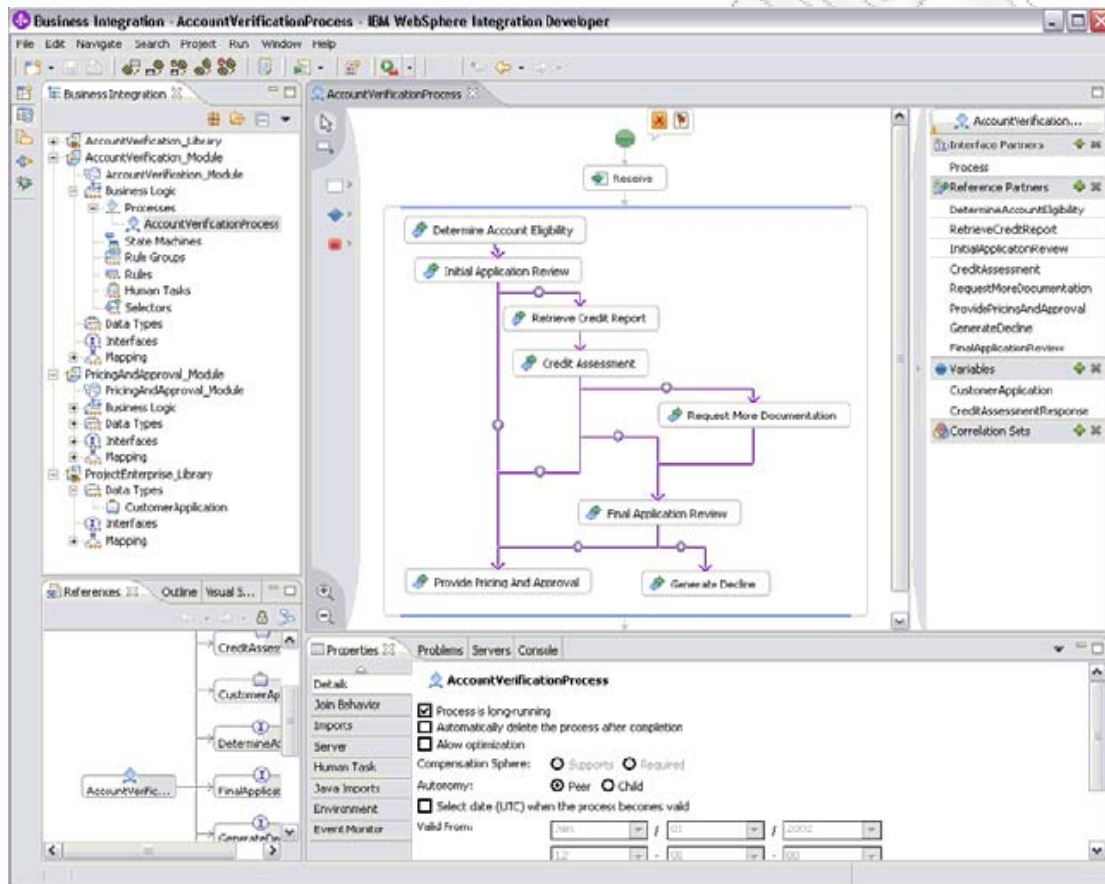
Οι εφαρμογές που θα παρουσιαστούν στην παρούσα ενότητα αναφέρονται στη συνέχεια:

1. IBM WID (Websphere Integration Developer) [9]
2. μία ερευνητική προσπάθεια από την Κοινωνία της Πληροφορίας σε συνεργασία με την εταιρεία EBS (<http://www.ebs.gr>), με τίτλο «Ενδιάμεσο Λογισμικό Διαλειτουργικότητας (Interoperability Middleware)» [11]
3. και τέλος το «Asynchronous Mediation for Integrating Business and Operational Processes» [12]

1.2.1 IBM WID (Websphere Integration Developer)

Το συγκεκριμένο λογισμικό[9] είναι ένα περιβάλλον δημιουργίας και συγγραφής, που παρέχει βήμα προς βήμα διαδικασίες για την ενσωμάτωσή του στην αρχιτεκτονική, προσανατολισμένη σε υπηρεσίες, μίας εταιρίας.

Είναι ένα εργαλείο βασισμένο στο λογισμικό Eclipse¹, για την δημιουργία μεταξύ άλλων, λύσεων ενοποίησης (solutions integration) [9]. Αυτό επιτυγχάνεται μέσω της χρησιμοποίησης άλλων προϊόντων της Websphere όπως WebSphere Process Server, WebSphere ESB, and WebSphere Adapters [18][9].



Εικόνα 1.2.1 Οθόνη προγράμματα WID (Websphere Integration Developer)

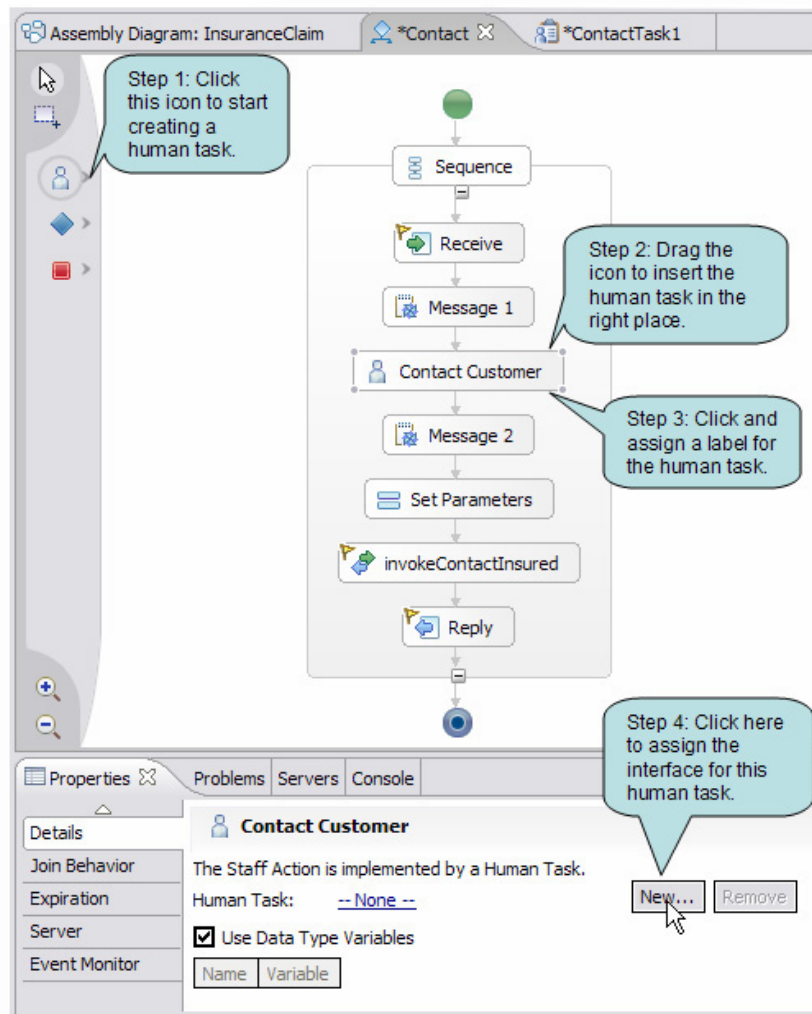
Οι λύσεις που προτείνει όμως το εργαλείο είναι ημιαυτόματες (semi-automatic) [9]. Απαιτούν την ύπαρξη προγραμματιστών, για να αναπτύξουν κομμάτια κώδικα που θα βοηθήσουν την ομοιογενοποίηση των διαφορετικών σχημάτων μεταδεδομένων, όπως επίσης και στην επίλυση των προβλημάτων που πηγάζουν από τα διαφορετικά πρωτόκολλα επικοινωνίας [9].

¹ Eclipse is an open source community, whose projects are focused on building an open development platform comprised of extensible frameworks, tools and runtimes for building, deploying and managing software across the lifecycle. (We started with the best Java IDE ever and we've grown from there.) [10]

Υψιστο στόχο τους αποτελεί η δημιουργία ημι-αυτόματων μεσολαβητών υπηρεσιών, οι οποίοι να μπορούν να συσχετίσουν όλους τους πιθανούς συνδυασμούς μεσολάβησης. Εστιάζουν κυρίως στην επίλυση του προβλήματος των διαφορετικών πρωτοκόλλων (ο τρόπος που τα συστήματα κάνουν την αρχική επικοινωνία / χειραγία, τα μηνύματα επικοινωνίας κ.λπ.) μεταξύ των διαφόρων πιθανών συστημάτων[9].

Πλεονεκτήματα συστήματος [9]:

1. Δημιουργία ενός τύπου SOA (Service Oriented Architecture) BPM (Business Process Model). Υποστηρίζει την δημιουργία ενός μοντέλου προσανατολισμένο σε υπηρεσίες το οποίο είναι πολύ χρήσιμο αφού όλες οι επιχειρήσεις / οργανισμοί τείνουν να υιοθετούν το μοντέλο SOA. Επίσης τις διαδικασίες που υποστηρίζει τις μοντελοποιεί ως επιχειρησιακές διαδικασίες, εισάγοντας την έννοια του Workflow στο σύστημα.
2. Ύπαρξη πολλών ειδικευμένων οθονών για όλων των ειδών διαδικασιών που προσφέρει το πρόγραμμα (όπως human-related tasks, business state machines, business rules, data mapping), οι οποίες ελαχιστοποιούν την ανάγκη για δημιουργία κώδικα σε Java.
3. Δυνατότητα δημιουργίας εκτελέσιμων διαδικασιών βασισμένων σε WS-BPEL (Εικόνα 1.2.2).
4. Είναι εμπορικό πακέτο το οποίο σημαίνει ότι αναπτύσσεται και βελτιώνεται συνέχεια μέσα από τις απαιτήσεις που έχουν επιχειρήσεις / οργανισμοί που το μπορεί αν το χρησιμοποιούν.
5. Δυνατότητα εισαγωγής κώδικα γραμμένο στην COBOL για το WebSphere MQ, το JMS ή ακόμα και ftp συνδέσεις / δεσμεύσεις, χωρίς να γραφτεί επιπλέον άλλος κώδικας.
6. Υποστήριξη IDE (Integrated Development Environment), για γρήγορη και εύκολη δημιουργία κώδικα, όταν αυτό κρίνεται αναγκαίο.



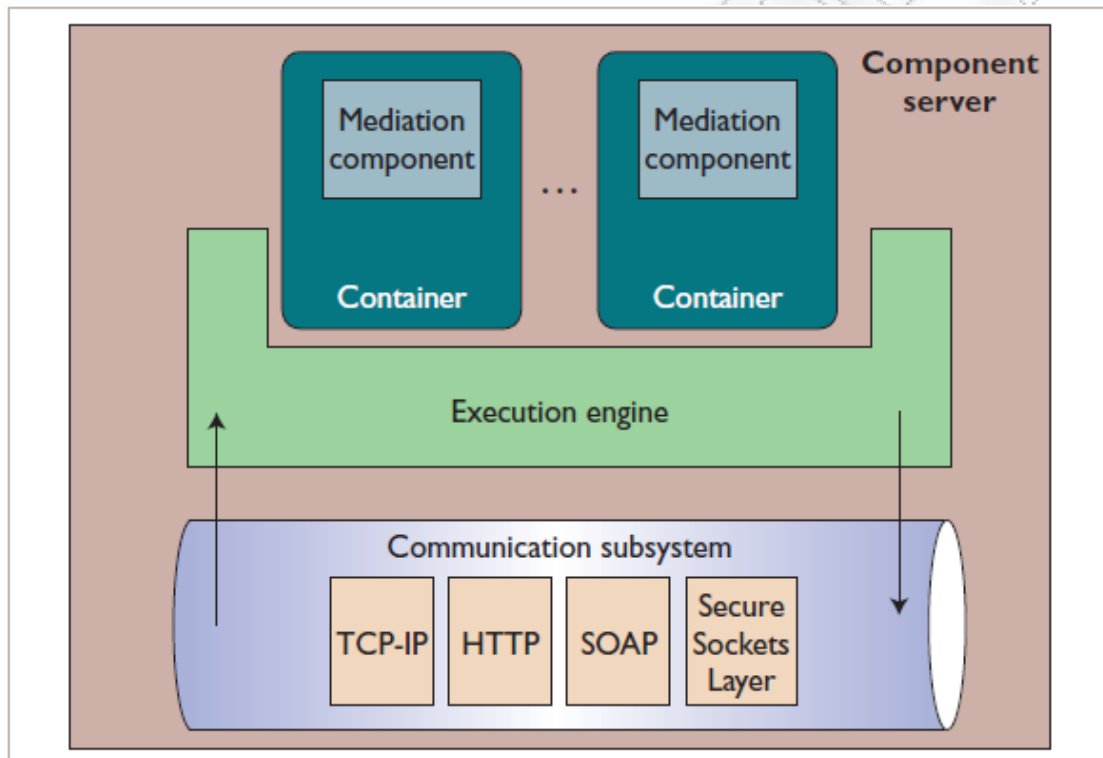
Εικόνα 1.2.2 Δημιουργία εργασίας μέσω του BPEL συντάκτη

Μειονεκτήματα συστήματος [9]:

1. Οι συνδέσεις / connectors είναι κλάσεις γραμμένες σε Java, τις οποίες χρειάζεται προγραμματιστείς για να τις δημιουργήσει.
2. Δεν αναφέρεται η δυναμική ενημέρωση του σχήματος πηγής των δεδομένων.
3. Δεν μπορεί να γίνει η διαμόρφωση / επεξεργασία των δεδομένων αλλά χρησιμεύει μόνο για μεσολάβηση (migration).
4. Δεν αναφέρεται η δυνατότητα της επιλογής πολλών ταυτόχρονα σχημάτων πηγών δεδομένων είτε μέσα από βάσεις είτε από xml αρχεία, για την εξαγωγή και μεταφορά αυτών.
5. Δεν παρέχεται η δυνατότητα σύνθεσης δεδομένων και αντιμετώπισής τους ως μια αδιαίρετη και ενιαία πηγή (π.χ. join 2 πινάκων και εξαγωγή μόνο της επιθυμητής πληροφορίας)

1.2.2 Asynchronous Mediation for Integrating Business and Operational Processes

Η προσέγγιση του Asynchronous Mediation for Integrating Business and Operational Processes [12] αποτελεί έρευνα που πραγματοποιήθηκε από τον Philippe Lalanda του Grenoble University σε συνεργασία με την εταιρεία ScalAgent Distributed Technologies και τους συνεργάτες Luc Bellissard και Roland Balter. Η θεώρηση των προαναφερθέντων στηρίχθηκε στην ιδέα τους για ανάγκη ύπαρξης mediators με αυξημένες δυνατότητες επέκτασης (scalability) (Εικόνα 1.2.3).



Εικόνα 1.2.3 Υποστήριξη κατά τον χρόνο εκτέλεσης.

Η μελέτη στηρίχθηκε σε ένα μοντέλο δομικών στοιχείων (components), που εκτελούνται ανεξάρτητα και σε διαφορετικούς εξυπηρετητές. Τα δομικά στοιχεία επικοινωνούν μεταξύ τους, ανταλλάσσοντας τις απαραίτητες επεξεργασμένες πληροφορίες που μεταφέρουν σε επόμενο επίπεδο (π.χ. στην εφαρμογή που ζητά τα στοιχεία από τις διάφορες πηγές)[12].

Τα δομικά στοιχεία που χρησιμοποιούνται χωρίζονται σε δύο επιμέρους κατηγορίες [12]:

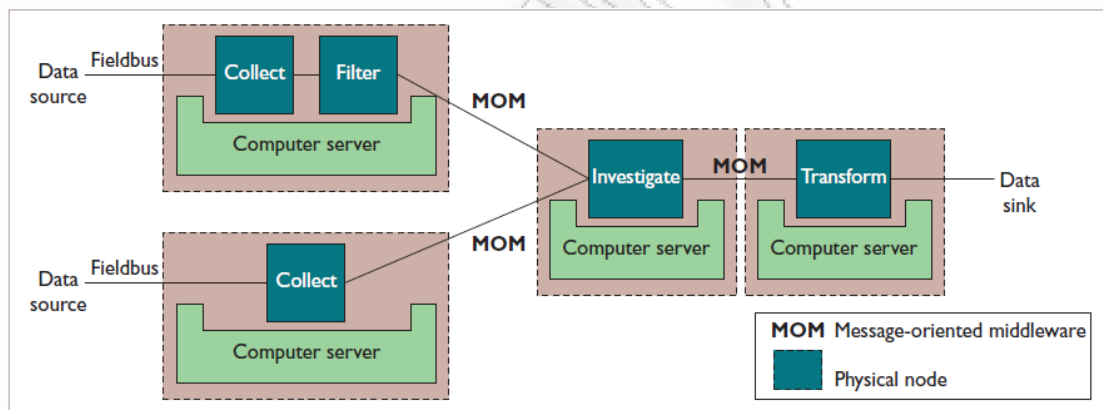
- **Functional interfaces:** Τα functional interfaces καθορίζουν τις υπηρεσίες που παρέχονται σε άλλα δομικά στοιχεία (components) και μπορεί να αποτελούν είτε διεπαφές εξυπηρετητή (server interfaces), είτε διεπαφές πελάτη (client interfaces).

- **Control interfaces:** Τα control interfaces παρέχουν τη δυνατότητα, σε διαχειριστικές οντότητες (ανθρώπους ή μηχανές), τόσο της παραμετροποίησης των δομικών στοιχείων όσο και τον καθορισμό του κύκλου ζωής τους (life cycle).

Στο σημείο αυτό αξίζει να αναφερθεί ότι τα functional interfaces αποτελούν κλάσεις της γλώσσας προγραμματισμού Java, που καλείται να υλοποιήσει ο εκάστοτε προγραμματιστής που αναλαμβάνει το mediation[12].

Η επικοινωνία μεταξύ των δομικών στοιχείων πραγματοποιείται με τεχνικές ισχυρής προστασίας και ασφάλειας, γεγονός που αποτελεί άμεση ανάγκη δεδομένης της κατακεκομμένης αρχιτεκτονικής.

Στο διάγραμμα που ακολουθεί (βλ. εικόνα) παρουσιάζεται ολοκληρωμένα το μοντέλο λειτουργίας του Asynchronous Mediation for Integrating Business and Operational Processes. Διαφορετικά δομικά στοιχεία (π.χ. συλλογή, επιλογή, μετατροπή κ.α.) φιλοξενούνται σε διαφορετικούς εξυπηρετητές και επικοινωνώντας με μηνύματα (MOM Message Oriented Middleware) συνθέτουν το τελικό αποτέλεσμα το οποίο και διοχετεύουν σε επόμενο επίπεδο[12].



Εικόνα 1.2.4 Ολοκληρωμένα μοντέλα παρουσίασης

Πλεονεκτήματα

Η προαναφερθείσα λύση παρέχει [12]:

- Ισχυρούς ελέγχους προστασίας κατά την επικοινωνία των διάφορων δομικών στοιχείων μεταξύ τους
- Ταχύτητα διάδοσης των δεδομένων λόγω της κατακεκομμένης αρχιτεκτονικής
- Αξιόπιστο και εύχρηστο περιβάλλον διαχείρισης
- Προσαρμογή σε διαφορετικά περιβάλλοντα (π.χ. λειτουργικά συστήματα, τεχνικές κ.α.)

- Αυξημένες δυνατότητες επέκτασης (παράλληλη προσθήκη επιπλέον δομικών στοιχείων)

Μειονεκτήματα

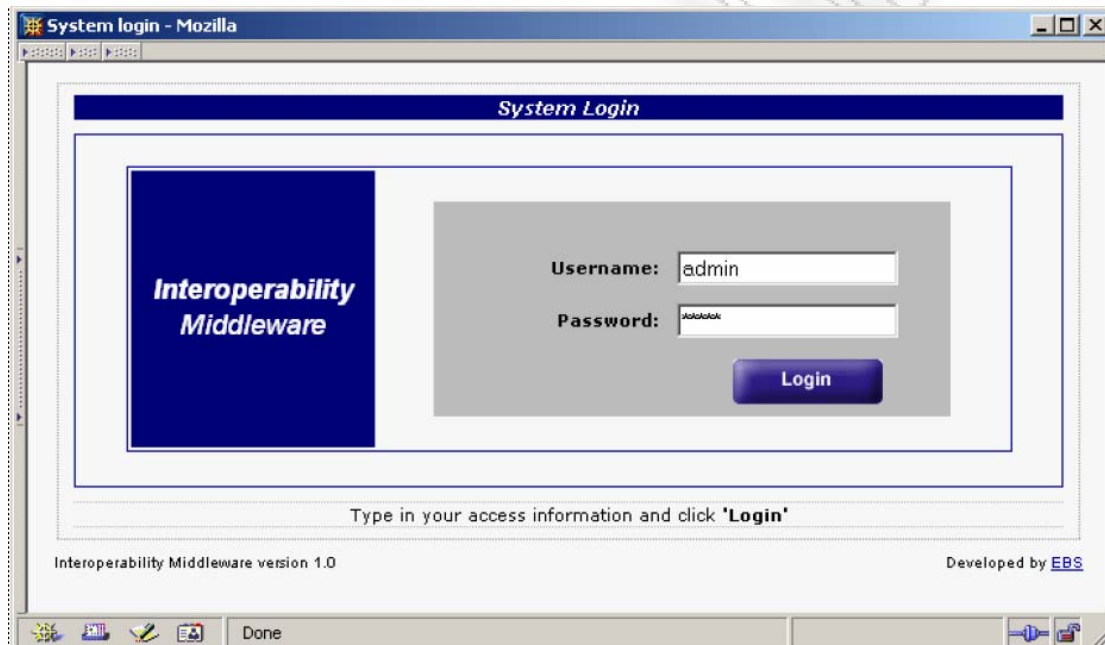
Αν και αποτελεί μια από τις πιο αξιοσημείωτες μελέτες στο χώρο του mediation, ο παρατηρήθηκαν τα ακόλουθα μειονεκτήματα[12]:

- Όπως προαναφέρθηκε τα functional interfaces, που αποτελούν το βασικότερο κομμάτι λειτουργίας του μοντέλου, δεν προσαρμόζονται αυτόματα στις πηγές δεδομένων. Η ανάπτυξή τους απαιτεί την ύπαρξη εξειδικευμένου τεχνικού που θα υλοποιήσει τις διεπαφές σε Java Classes
- Από το παραπάνω συμπέρασμα επιπλέον απορρέει η έλλειψη αυτόματης προσαρμογής στην περίπτωση μεταβολής (π.χ. αλλαγή σχήματος αρχείου XML ή βάσης δεδομένων) μιας ή περισσότερων πηγών πληροφορίας
- Δεν αναφέρονται οι δυνατότητες αυτοματοποιημένης επεξεργασίας των δεδομένων κατά την έλευσή τους από τα δομικά στοιχεία. Ο συγγραφέας υποθέτει ότι οι ερευνητές αποσκοπούν στην επίλυση του εν λόγω προβλήματος κατά την υλοποίηση των functional interfaces. Η προσέγγιση αυτή όμως μπορεί να γίνει ιδιαίτερα «ακριβή» στην περίπτωση μεγάλων και πολύπλοκων πηγών. Επιπλέον σε ενδεχόμενη μεταβολή μιας ή περισσότερων πηγών πληροφορίας, η προσαρμογή των functional interfaces μπορεί να είναι ανέφικτη.
- Δεν παρέχεται η δυνατότητα σύνθεσης δεδομένων και αντιμετώπισής τους ως μια αδιαίρετη και ενιαία πηγή (π.χ. join 2 πινάκων και εξαγωγή μόνο της επιθυμητής πληροφορίας)

1.2.3 Ενδιάμεσο Λογισμικό Διαλειτουργικότητας

Σε αυτήν την ενότητα θα παρουσιαστεί το λογισμικό Ενδιάμεσο Λογισμικό Διαλειτουργικότητας (Interoperability Middleware)[11], το οποίο είναι ένα ενδιάμεσο λογισμικό διαλειτουργικότητας που κατασκευάστηκε από την εταιρία EBS το 2004. Είναι φτιαγμένο σε γλώσσα προγραμματισμού «Java», έτσι ώστε να είναι συμβατό με όλα τα κοινά λειτουργικά συστήματα που είναι διαθέσιμα στην αγορά. Έχει σαν στόχο του, να κάνει όσο το δυνατόν πιο εύκολη τη δημιουργία εφαρμογών διαλειτουργικότητας.

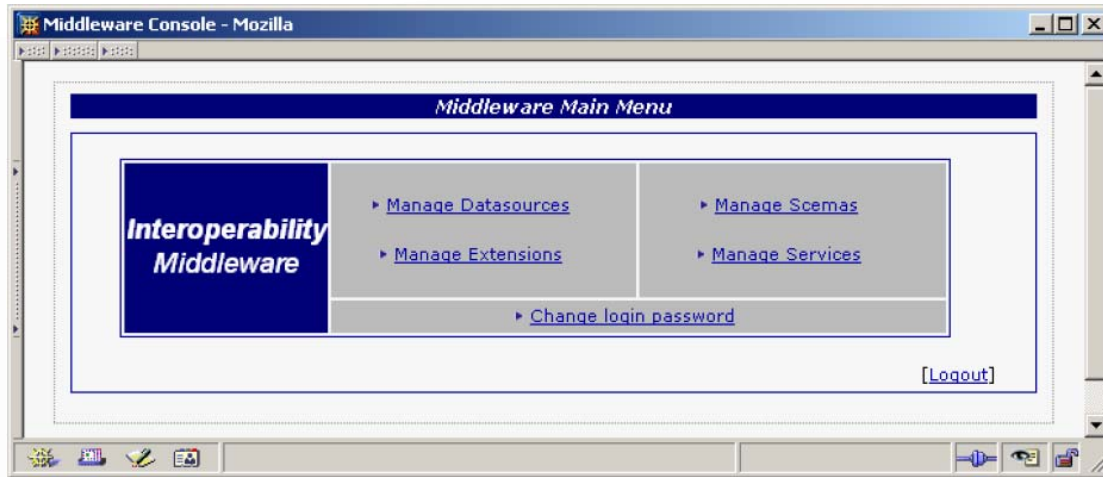
Όσον αφορά την πρόσβαση στο ΕΛΔ μπορεί να γίνει μέσω της εκκίνησης ενός web browser πληκτρολογώντας το ακόλουθο URL: **http://(server hostname):8080/mw/main**. Η αρχική του οθόνη παρουσιάζεται παρακάτω στην Εικόνα 1.2.5:



Εικόνα 1.2.5 Αρχική οθόνη του ΕΛΔ

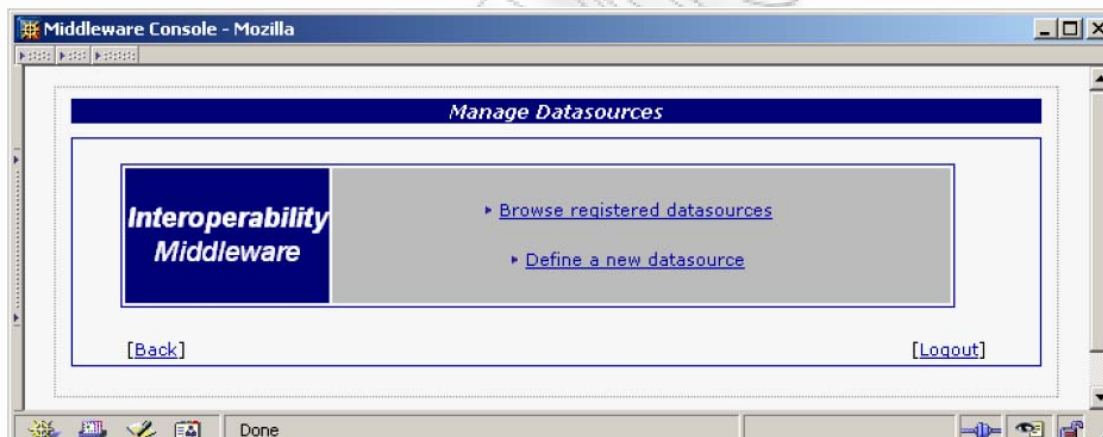
Οι διαθέσιμες λειτουργίες του ΕΛΔ [11] παρουσιάζονται όλες σε μία φόρμα όπως απεικονίζεται στην Εικόνα 1.2.6, και αναφέρεται στη συνέχεια:

1. Manage Datasources
2. Manage Extensions
3. Manage Schemas
4. Manage Services
5. Logout
6. Change login password



Εικόνα 1.2.6 Λειτουργίες του ΕΛΔ

Οι επεκτάσεις του συστήματος [11] αφορούν πολύπλοκους ορισμούς και υλοποιήσεις υπηρεσιών που προσφέρουν πολύ περισσότερες δυνατότητες από τις συνδέσεις μέσω ΒΔ. Οι υπόλοιποι ορισμοί γίνονται μέσα από την οθόνη που παρουσιάζεται στην Εικόνα 1.2.7.



Εικόνα 1.2.7 Διαχείριση Βάσεων Δεδομένων

Οι πολύπλοκοι ορισμοί [11] γίνονται με τη χρήση Java μέσω επεκτάσεων, όπως δείχνει το παράδειγμα της Εικόνα 1.2.8.

```

package gr.egif.testmbean;
public class MyClass implements MyClassMBean
{
    public String findNameByAFM(String AFM)
    {
        String name;

        // επικοινωνία με το πληροφοριακό σύστημα
        // και κλήση της κατάλληλης business function

        return name;
    }
}

```

Εικόνα 1.2.8 Κώδικας Java για την δημιουργία επεκτάσεων

Πλεονεκτήματα συστήματος:

1. Ευέλικτο: μπορεί να χρησιμοποιηθεί από πολλές διαφορετικές πλατφόρμες υλικού / λογισμικού, επειδή είναι γραμμένο σε «Java», και παίζει μέσα από τον Web Browser.
2. Εύχρηστο: με απλό τρόπο μπορούν να προκύψουν Web services δίχως οι χρήστες να είναι απαραίτητο να καταφύγουν στη συγγραφή κώδικα. Ο κώδικας απαιτείται μόνο σε περιπτώσεις υλοποίησης πολυσύνθετων custom connectors.
3. Δυνατότητα χρήσης τεχνολογιών XML, XSD για τον ορισμό του σχήματος επικοινωνίας με το web service.
4. Δεν απαιτείται η ύπαρξη πολλών επιπλέον λογισμικών προκειμένου να λειτουργήσει το σύστημα και να επικοινωνήσει με άλλα web services.

Μειονεκτήματα συστήματος:

1. Ανάγκη για εγκατάσταση ειδικού JDBC driver για να μπορέσει να συνδεθεί στη βάση [11].
2. Όλες οι συνδέσεις γίνονται μέσω του driver που εγκαθίσταται, και συγκεκριμένα μέσω ODBC συνδέσεων, και όχι απευθείας με την εκάστοτε βάση δεδομένων.
3. Ανάγκη γνώσης από την μεριά του χρήστη καλής SQL προκειμένου να διαμορφωθούν πλήρως και ορθά τα ερωτήματα που θα εκτελεστούν [11].
4. Δυνατότητα επέκτασης του λογισμικού μέσα από την δημιουργία ειδικού κώδικα και συγκεκριμένα κλάσεων σε γλώσσα JMX. Εκεί δηλώνεται το Connection και η όλη επεξεργασία που θέλει ο χρήστης να γίνει σε ένα πεδίο για να επιστραφεί η σωστή τιμή πίσω [11].
5. Δεν παρέχεται εξαγωγή δεδομένων σε xml ή είσοδος από xml (xml-schema).

6. Ανάγκη εγκατάστασης ενός δωρεάν Application Server με όνομα JBoss Application Server.
7. Έχει ειδικές απαιτήσεις προς χρήστη – διαχειριστή:
 - Γνώση SQL για τον ορισμό web services που βασίζονται σε απευθείας πρόσβαση σε βάσεις δεδομένων.
 - Γνώση Java για περιπτώσεις χρήσης επεκτάσεων μέσω της επιλογής developer όταν έχουμε web services.
8. Δεν παρέχεται η δυνατότητα σύνθεσης δεδομένων και αντιμετώπισής τους ως μια αδιαίρετη και ενιαία πηγή (π.χ. join 2 πινάκων και εξαγωγή μόνο της επιθυμητής πληροφορίας)

2 Σχεδίαση και Ανάπτυξη

Τα συνήθη προβλήματα που έχουμε από τα σημερινά προγράμματα (όπως αναφέρθηκαν στην ενότητα *Βιβλιογραφική Ανασκόπηση*), είναι τα ακόλουθα:

1. Αδυναμία υποστήριξης συνδέσεων σε βάσεις δεδομένων και σε παράλληλη εισαγωγή ή εξαγωγή XML.
2. Δεν παρέχεται δυνατότητα σύνθεσης δεδομένων και η αντιμετώπισή τους ως μία ενιαία πηγή πληροφορίας, όπου μπορούν να συλλεχθούν, να επιλεγούν και να επεξεργαστούν οι πληροφορίες ενδιαφέροντος,
3. Στις περισσότερες αν όχι σε όλες τις περιπτώσεις είναι αναγκαίο να υπάρχουν γνώσεις προγραμματισμού προκειμένου να δημιουργηθεί η σύνδεση με την βάση και η λήψη του σχήματός της.
4. Δεν παρέχεται η δυνατότητα αυτοματοποιημένης ενημέρωσης των πεδίων όταν αλλάξει το σχήμα της βάσης πηγής. Μάλιστα πολλές φορές χρειάζεται να ξαναγραφτεί κώδικας για την σύνδεση που να λαμβάνει υπόψη το νέο σχήμα βάσης.
5. Είναι εμπορικά πακέτα τα οποία περιλαμβάνουν πολλές διαδικασίες που τις περισσότερες φορές δεν είναι αναγκαίες από την επιχείρηση ή τον οργανισμό.
6. Μπορεί να χρειαστεί επιπλέον προσωπικό για την εκμάθηση και χρήση του προγράμματος (π.χ. προγραμματιστές),
7. Πολλά λογισμικά χρησιμοποιούνται ως απλοί μεσολαβητές, οι οποίοι δεν παρέχουν διαδικασίες επεξεργασίας των δεδομένων πριν την αποστολή τους.
8. Δεν παρέχεται πάντα η δυνατότητα μέσα από την εφαρμογή η αποστολή και λήψη xml δεδομένων σε web services.

Το πρόγραμμα Ultimate Mediator έρχεται να καλύψει αυτές τις ελλείψεις / ανάγκες των άλλων εμπορικών προγραμμάτων, δίνοντας μία λύση ελεύθερου λογισμικού (free source), που βοηθάει κυρίως εκείνους τους χρήστες που χρειάζονται ένα πρόγραμμα για να μεταφέρουν δεδομένα από κάποιο/ κάποια συστήματα σε κάποιο άλλο, γρήγορα και εύκολα, χωρίς να περιλαμβάνει πολλές πρόσθετες λειτουργίες οι οποίες δεν είναι πάντα επιθυμητές.

Οι επόμενες ενότητες θα παρουσιάσουν το σχεδιασμό του προγράμματος αλλά και τον τρόπο υλοποίησής του, έτσι ώστε να γίνει κατανοητή η κατασκευή ενός τέτοιου συστήματος.

2.1 Σχεδίαση Συστήματος

Το σύστημα έχει σχεδιαστεί έτσι που να περιλαμβάνει όλες τις πιθανές περιπτώσεις μεταφοράς δεδομένων, από και προς βάσεις, xml αρχεία και web services. Οι βάσεις δεδομένων που υποστηρίζονται είναι οι ακόλουθες:

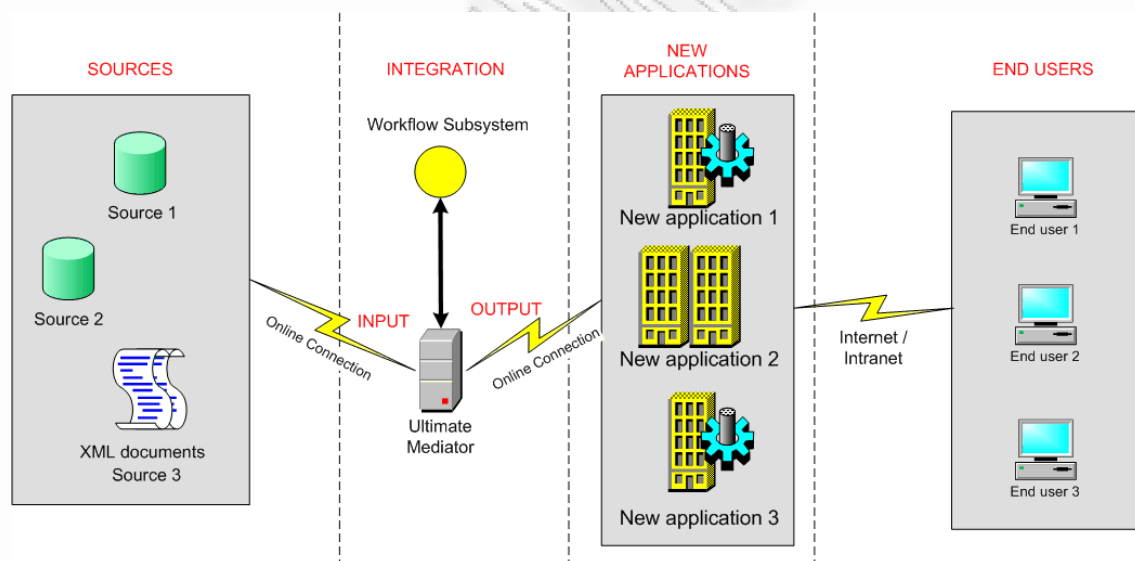
1. Microsoft SQL Server,
2. Microsoft Access,
3. Oracle,
4. MySQL

Στις επόμενες ενότητες θα παρουσιαστεί η σχεδίαση του συστήματος και θα αναφερθούν παραδείγματα επικοινωνίας μεταξύ των διαφόρων συνδέσεων και του Ultimate Mediator. Έτσι θα μπορέσει ο αναγνώστης να κατανοήσει τον τρόπο λειτουργίας της εφαρμογής σε υψηλότερο επίπεδο.

2.1.1 Γενική περιγραφή λογισμικού

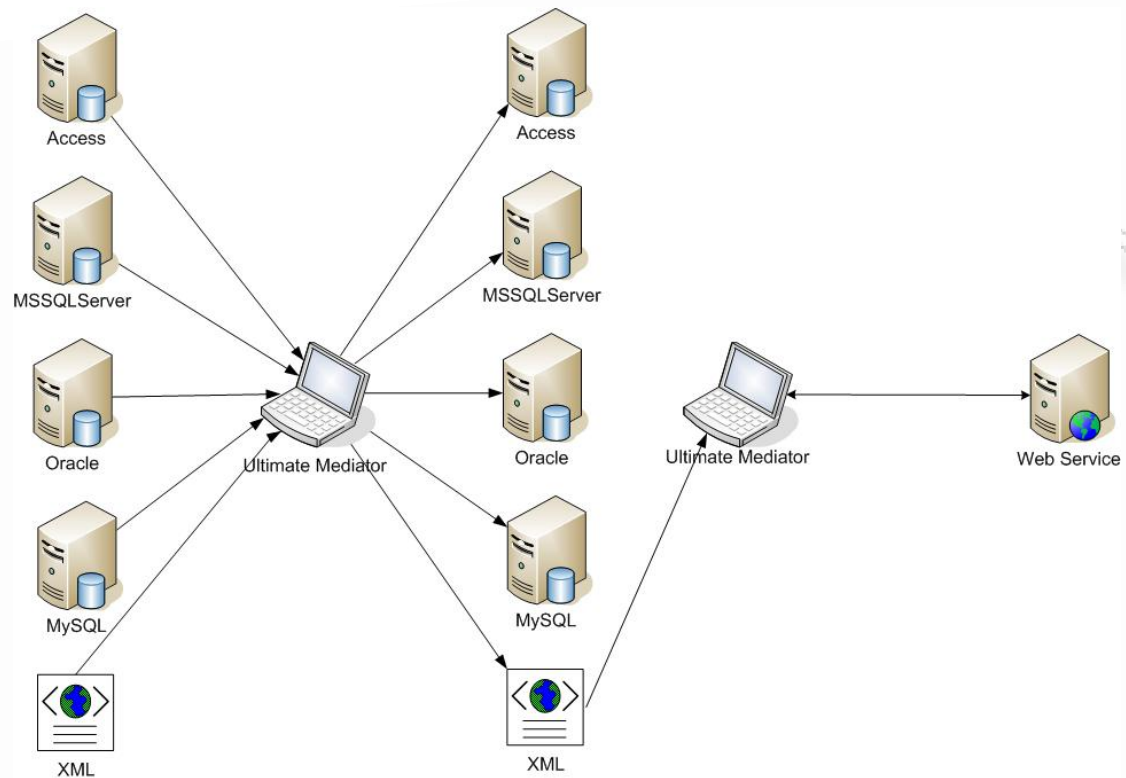
Η επικοινωνία μεταξύ των βάσεων δεδομένων προϋποθέτει ότι ο υπολογιστής στον οποίο είναι εγκατεστημένο το πρόγραμμα, θα έχει δικαιώματα να επικοινωνήσει με αυτές (τις βάσεις δεδομένων) που του ορίζονται. Ο χρήστης μπορεί να πραγματοποιήσει πολλαπλές συνδέσεις, έτσι ώστε να μπορεί να έχει όλες τις δυνατότητες μεταφοράς δεδομένων, από και προς αυτές. Οι συνδέσεις δεν είναι απαραίτητο να περιορίζονται μόνο σε οντότητες βάσεων δεδομένων. Μπορεί να είναι και η λήψη ή η αποστολή δεδομένων σε xml αρχεία ή σε web services οι οποίες υπάρχουν σε κάποιους απομακρυσμένους υπολογιστές που με τη σειρά τους παρέχουν, συγκεκριμένα, ορισμένες υπηρεσίες.

Τα παραπάνω όμως γίνονται ως εσωτερικές διαδικασίες μέσα στο πρόγραμμα Ultimate Mediator και αποτελούν τη βάση για την επικοινωνία με άλλες εφαρμογές στον «έξω κόσμο». Έτσι μέσα από ένα ενσωματωμένο σύστημα αποφάσεων, μπορούμε και ορίζουμε Workflow διαδικασίες, οι οποίες θα καθοδηγούν τη ροή των δεδομένων σε εξωτερικές εφαρμογές σε πραγματικό χρόνο. Οι εφαρμογές θα ενημερώνονται και θα παρέχουν έτσι πλήρως ενημερωμένη πληροφορία στους χρήστες. Η Εικόνα 2.1.1 παρέχει το γενικότερο μοντέλο σχεδίασης του προγράμματος, το οποίο και θα αναλυθεί εκτενέστερα στην ενότητα *Μεταφορά δεδομένων σε πραγματικό χρόνο (Online transactions)*.



Εικόνα 2.1.1 Το γενικότερο μοντέλο λειτουργίας του Ultimate Mediator

Στα πλαίσια όμως αυτής της διπλωματικής δεν επεκτάθηκε το πρόγραμμα στο σημείο της αυτόματης ενημέρωσης τρίτων συστημάτων μέσω διαδικασιών Workflow. Επεκτάθηκε εκτενώς μέχρι την «on demand» διαλογή, επεξεργασία και μεταφορά δεδομένων. Στην παρακάτω Εικόνα 2.1.2, περιγράφεται σχηματικά η επικοινωνία μεταξύ όλων των πιθανών οντοτήτων σύνδεσης με το πρόγραμμα. Ο Ultimate Mediator αναφέρεται δύο φορές, απλά για να δείξει όλες τις δυνατότητες σύνδεσης που παρέχονται.



Εικόνα 2.1.2 Επικοινωνία οντοτήτων με το πρόγραμμα

Βλέπουμε λοιπόν από την Εικόνα 2.1.2, ότι η επικοινωνία δεν πραγματοποιείται απευθείας μεταξύ των οντοτήτων βάσης ή xml. Περνάει μέσα από το λογισμικό (Ultimate Mediator), το οποίο, εφόσον πάρει και επεξεργαστεί τα δεδομένα, θα εφαρμόσει όλους τους κανόνες που έχουν οριστεί από τον χρήστη και θα κρατήσει μόνο τα επιθυμητά πεδία στην απαιτούμενη μορφή τους. Εφόσον τελειώσει με αυτήν τη διαδικασία τότε θα τα εξάγει σε κάποια οντότητα. Η μεταφορά δεδομένων μπορεί να γίνει από και προς την ίδια βάση, χωρίς να υπάρχει κάποιος περιορισμός. Επιπλέον δίνεται η δυνατότητα να γίνει η χρήση πολλαπλών οντοτήτων για την είσοδο των δεδομένων, συνδυάζοντας και βάσεις και xml για την παραγωγή του επιθυμητού αποτελέσματος.

Οι οντότητες δεν είναι απαραίτητο να βρίσκονται όλες στο ίδιο μηχάνημα. Μπορεί να υπάρχουν κάπου συνδεδεμένες μέσω δικτύου ή μέσω του internet. Το μόνο που χρειάζεται στη δεύτερη περίπτωση είναι να είναι σίγουρο ότι μπορεί να υπάρξει επικοινωνία, μεταξύ του προγράμματος και της βάσης.

Επειδή οι εταιρίες βγάζουν κατά διαστήματα νέες εκδόσεις των βάσεων δεδομένων τους, πολλές φορές δεν είναι δυνατό να γίνει γνωστό αν ο τρόπος ανάκτησης, για παράδειγμα των πινάκων, θα παραμείνει πάντα ίδιος. Ο λόγος είναι γιατί το ερώτημα που θα πρέπει να εκτελεστεί στον SQL Server 2000, με αυτό του SQL Server 2005 είναι διαφορετικό. Μία λύση που χρησιμοποιείται είναι η αποθήκευση των ρυθμίσεων του χρήστη σε κάποιο κεντρικό σημείο, το οποίο να είναι εύκολα προσπελάσιμο από την εφαρμογή, και εύκολα διαχειρίσιμο από αυτόν. Η αποθήκευση γίνεται σε μία βάση δεδομένων που εγκαθίσταται μαζί με το πρόγραμμα και είναι σε Access. Στην ενότητα *Μοντελοποίηση Βάσης Δεδομένων* θα

γίνει αναλυτική περιγραφή της, και θα παρουσιαστεί ο τρόπος που μπορεί να γίνει η παραμετροποίησή της, σύμφωνα με τις εκάστοτε ανάγκες.

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΡΡΑΙΑ

2.1.2 Στόχοι και Κανόνες του Ultimate Mediator

Ο στόχος του Ultimate Mediator είναι να παρέχει ένα ολοκληρωμένο λογισμικό, το οποίο να παρέχει υπηρεσίες διαλειτουργικότητας, για τη επιλογή, επεξεργασία και μεταφορά δεδομένων από και προς συνδέσεις σε συνδυασμό, με όσο το δυνατόν πιο εύχρηστες και πιο αυτοματοποιημένες διαδικασίες. Τελικός σκοπός του είναι να υπάρχει ένα πρόγραμμα το οποίο να παρέχει μόνο εκείνες τις δυνατότητες που χρειάζεται τελικά μία επιχείρηση για να περάσει σωστά και με βάση κανόνων τα δεδομένα που μπορεί να έχει σε μία βάση ή σε ένα αρχείο xml. Όλα αυτά ορίζονται σε ένα πακέτο λογισμικού με την επιπλέον δυνατότητα αποστολής δεδομένων (όπως xml), σε κάποιο web service.

Για την υλοποίηση των παραπάνω, δημιουργήθηκαν κάποιοι κανόνες που θα πρέπει να ακολουθεί το πρόγραμμα, για να μπορέσει να φέρει εις πέρας τους στόχους του. Οι κανόνες αυτοί παρουσιάζονται παρακάτω:

1. Οι συνδέσεις στο πρόγραμμα, θα πρέπει να είναι αυτοματοποιημένες χωρίς να χρειάζεται να γράψει ο χρήστης κάποιον κώδικα σε κάποια συγκεκριμένη γλώσσα (για την λήψη των δεδομένων και την ανάγνωση του σχήματος). Για αυτόν το λόγο θα πρέπει να χρησιμοποιηθούν κάποια ερωτήματα, τα οποία να είναι αρκετά γενικά, για να υπάρχει η δυνατότητα από το λογισμικό, να φορτώσει όλες τις απαραίτητες πληροφορίες αναφορικά με την επιλεγμένη σύνδεση. Αν τελικά όμως χρειαστεί να γίνει η αλλαγή κάποιων ερωτημάτων, θα πρέπει να μπορεί να γίνει απευθείας από το χρήστη του συστήματος, χωρίς να είναι αναγκαία η παραγωγή καινούργιας έκδοσης λογισμικού. Τα ερωτήματα αυτά, θα πρέπει επιπλέον να είναι σε ευρέως γνωστή γλώσσα όπως η SQL.
2. Το πρόγραμμα θα παρέχει (μέσω των συνδέσεων που επιλέγονται), την ικανότητα να απεικονίζονται μέσα σε πλαίσια, που θα ονομάζονται σχήματα, οι διαθέσιμες δομές δεδομένων. Αντίστοιχα θα μπορεί να ορίζει ο χρήστης την τελική σύνδεση, και ένα τελικό σχήμα με τη δομή που επιθυμεί να εξαχθούν τα δεδομένα.
3. Ο χρήστης θα μπορεί να εισάγει κανόνες για την επιλογή των δεδομένων. Οι κανόνες θα ορίζονται επάνω στα επιλεγμένα πεδία που θα θέτει στο τελικό σχήμα. Θα πρέπει να μπορεί να φτιάχνει ένα δέντρο αποφάσεων με την χρήση τελεστών «AND» και «OR», για την σωστή επιλογή των δεδομένων που θα εξάγει. Τα πεδία δεν θα είναι αναγκαίο να εμφανίζονται όλα, και θα μπορούν να αλλάζουν τη σειρά εμφάνισής τους.
4. Για να μπορέσουν να συνενωθούν τα δεδομένα μεταξύ τους, θα πρέπει να υπάρχει η δυνατότητα να ορίζονται συσχετίσεις μεταξύ των κύριων πεδίων τους. Οι συσχετίσεις αυτές για να είναι πιο κατανοητές στον χρήστη θα έχουν την μορφή και τις δυνατότητες των συνδέσεων μεταξύ πινάκων με την χρήση SQL ερωτημάτων. Η ίδια λογική που χρησιμοποιείται στην SQL θα χρησιμοποιείται και εδώ. Θα έχουμε συνδέσεις τύπου Full, Left και Right Join. Οι συνδέσεις επίσης θα πρέπει να μπορούν να οριστούν και στο τελικό

σχήμα, γιατί η τελική δομή μπορεί να είναι διαφορετική από την αρχική, και οι συνδέσεις να πρέπει να πραγματοποιηθούν με διαφορετική λογική.

5. Τέλος θα υπάρχει εύκολη διαδικασία αλλαγής και αποθήκευσης των τελικών δεδομένων, όταν αυτά είναι τύπου xml. Σαν επιπλέον λειτουργίες θα πρέπει να ορίζονται η αποστολή δεδομένων σε κάποιο web service, και η δυνατότητα επεξεργασίας της απάντησης του web service σαν νέα xml δεδομένα.
6. Όλα τα παραπάνω θα υποστηρίζονται μέσω κατάλληλα διαμορφωμένων διαδραστικών φορμών, που να διευκολύνουν το χρήστη στην επίτευξη του τελικού του στόχου.

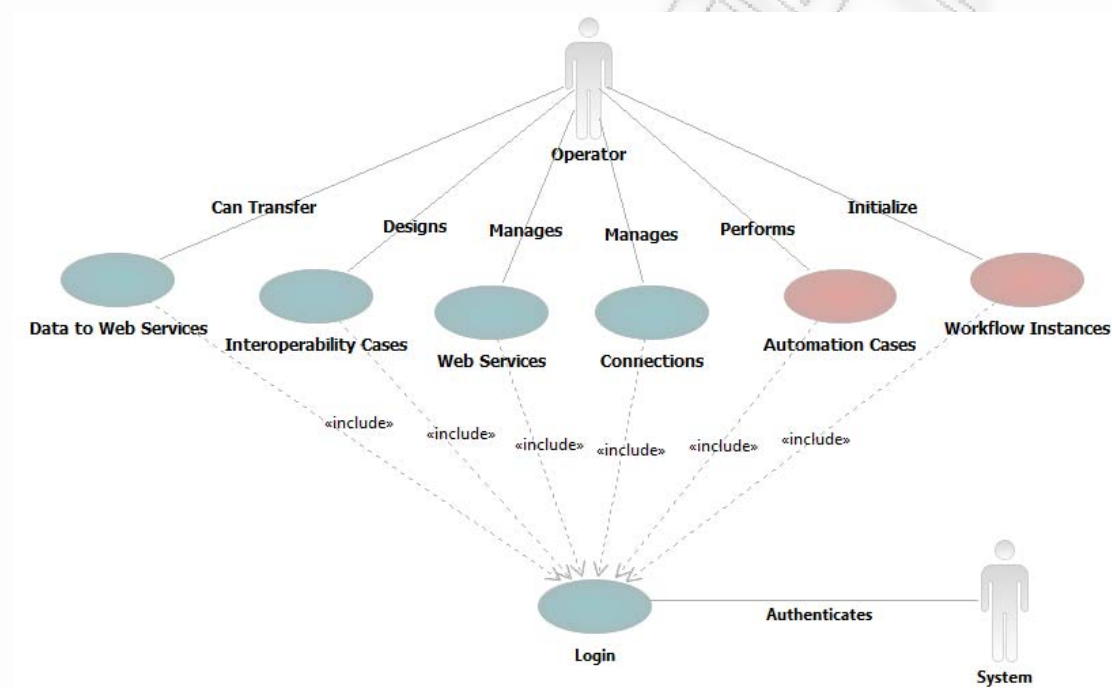
2.1.3 Διαγράμματα και παραδείγματα επικοινωνίας

Σε αυτήν την ενότητα θα παρουσιαστούν UML διαγράμματα, διαγράμματα κλάσεων και παραδείγματα επικοινωνίας του προγράμματος με τις εξωτερικές οντότητες σύνδεσης (μέσω sequence diagrams). Θα δοθούν δύο περιπτώσεις εξαγωγής δεδομένων σε xml και σε βάση, για την καλύτερη κατανόηση του αναγνώστη στον τρόπο σχεδίασης του λογισμικού.

2.1.3.1 UML Διαγράμματα

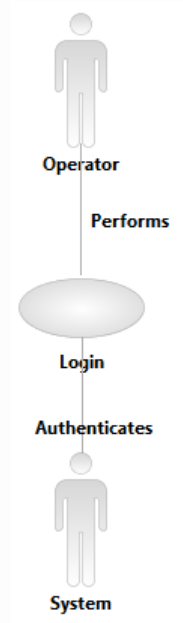
Παρακάτω θα παρουσιαστούν και θα περιγραφθούν τα uml διαγράμματα, για την καλύτερη κατανόηση του αναγνώστη, του τρόπου σχεδίασης της εφαρμογής. Τα use cases με χρώμα μπλε περιέχουν και άλλες λειτουργίες που περιγράφονται σε δεύτερο επίπεδο. Στα κόκκινα use cases έχουν αποτυπωθεί οι λειτουργίες του αυτοματισμού και οι λειτουργίες που χρησιμοποιούν το κέντρο αποφάσεων workflow. Περιγράφονται με κόκκινο γιατί δεν έχουν υλοποιηθεί από το πρόγραμμα Ultimate Mediator στην παρούσα φάση αυτής της διπλωματικής.

Διάγραμμα πρώτου επιπέδου - Λειτουργίες προγράμματος



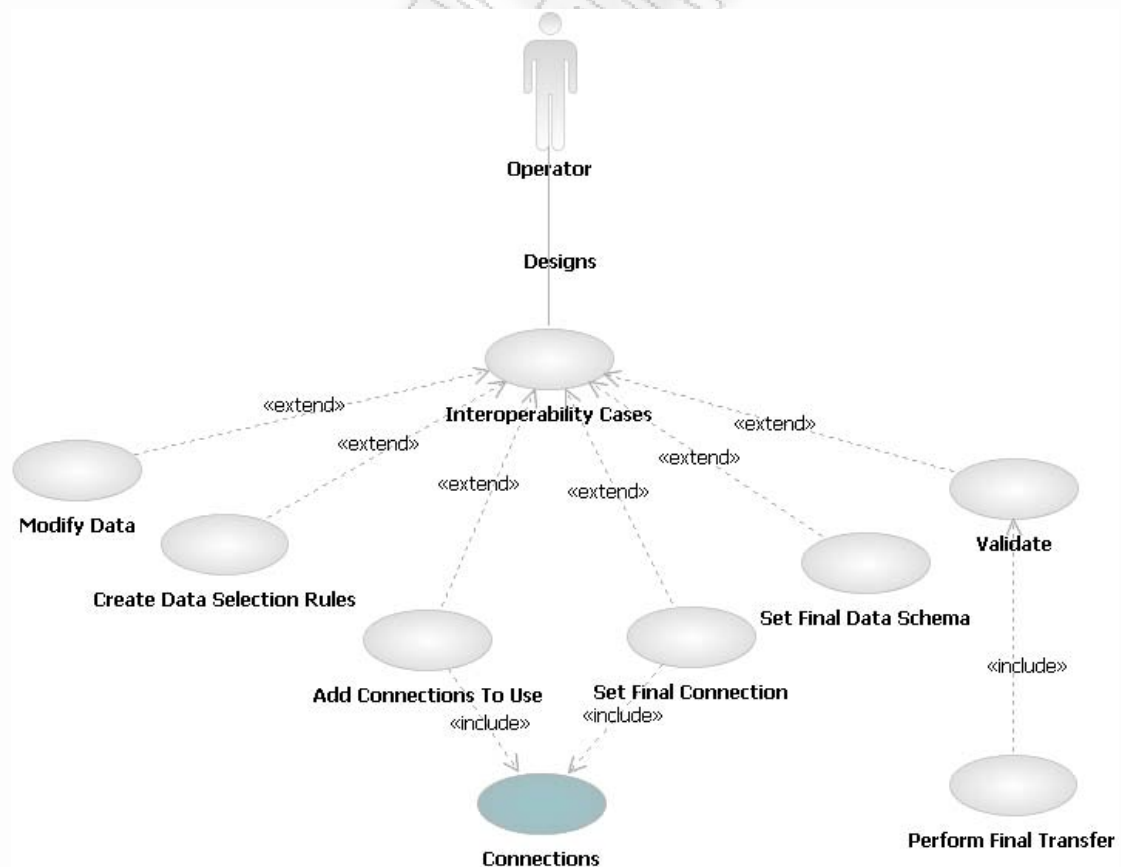
Εικόνα 2.1.3 Διάγραμμα λειτουργίας προγράμματος (1^ο επίπεδο)

Διάγραμμα δευτέρου επιπέδου – Είσοδος στο σύστημα «Login»



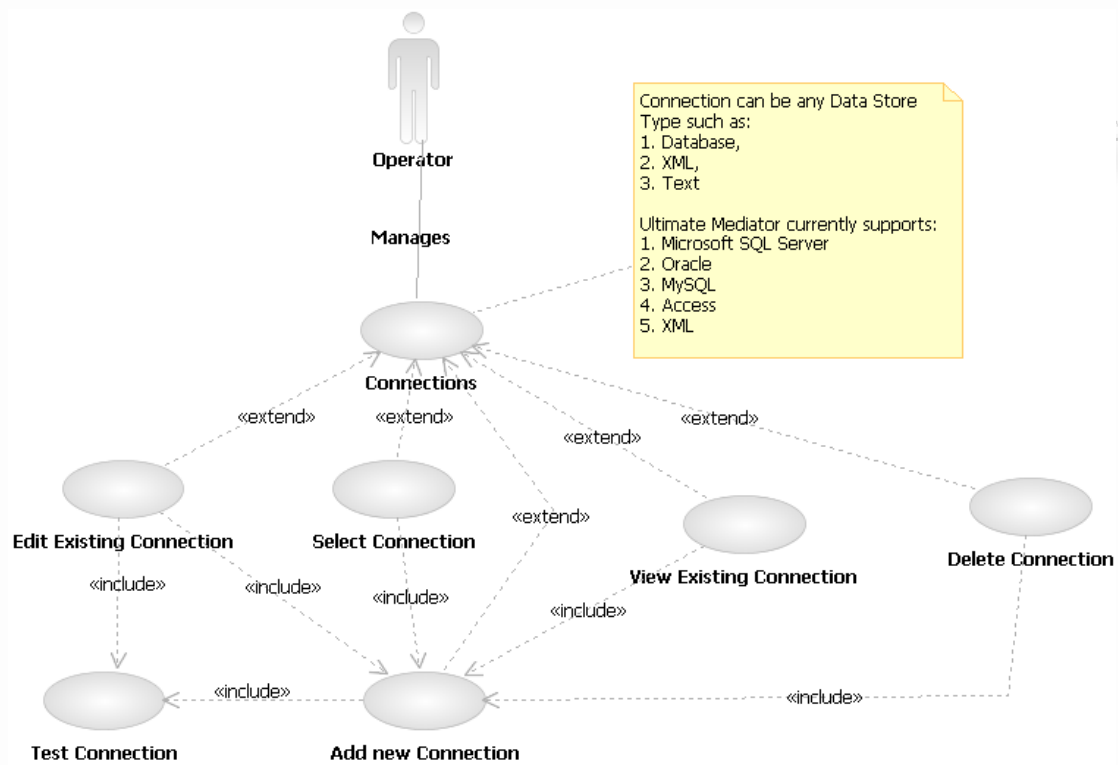
Εικόνα 2.1.4 Διάγραμμα εισόδου στο σύστημα

Διάγραμμα δευτέρου επιπέδου - «Interoperability Cases»



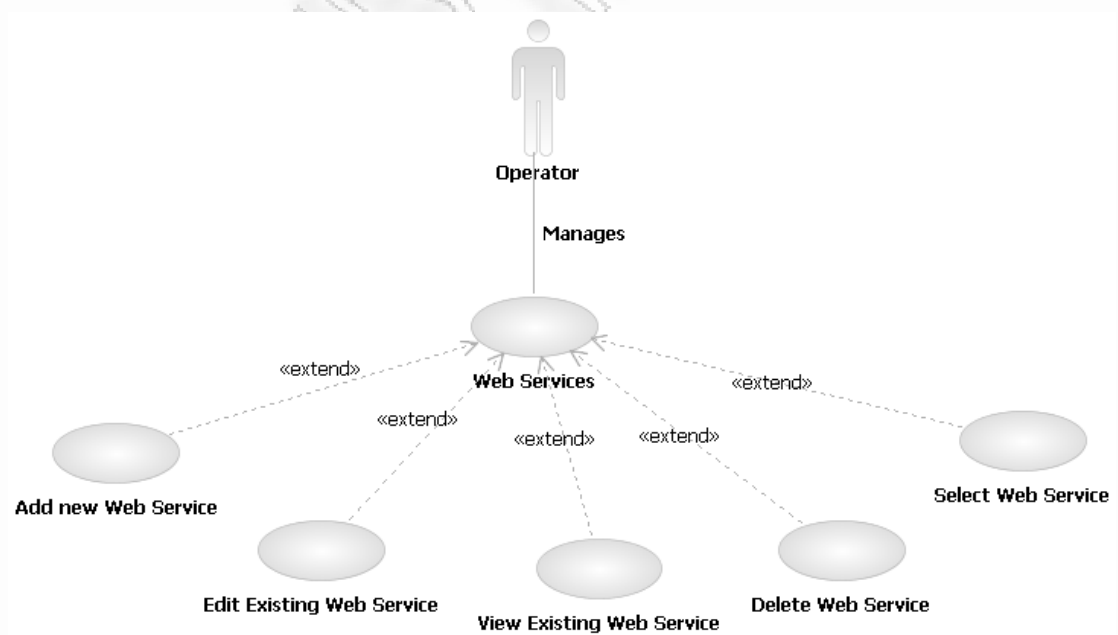
Εικόνα 2.1.5 Διάγραμμα σχεδίασης περιπτώσεων διαλειτουργικότητας

Διάγραμμα δευτέρου επιπέδου – «Διαχείριση Connections»



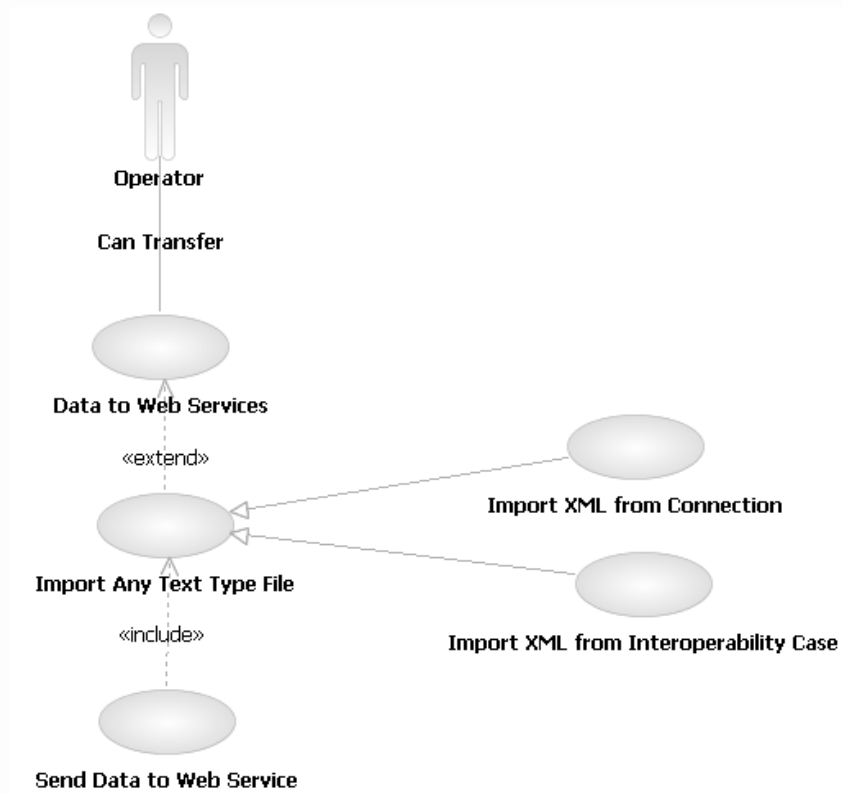
Εικόνα 2.1.6 Διάγραμμα διαχείρισης συνδέσεων

Διάγραμμα δευτέρου επιπέδου – «Διαχείριση Web Services»



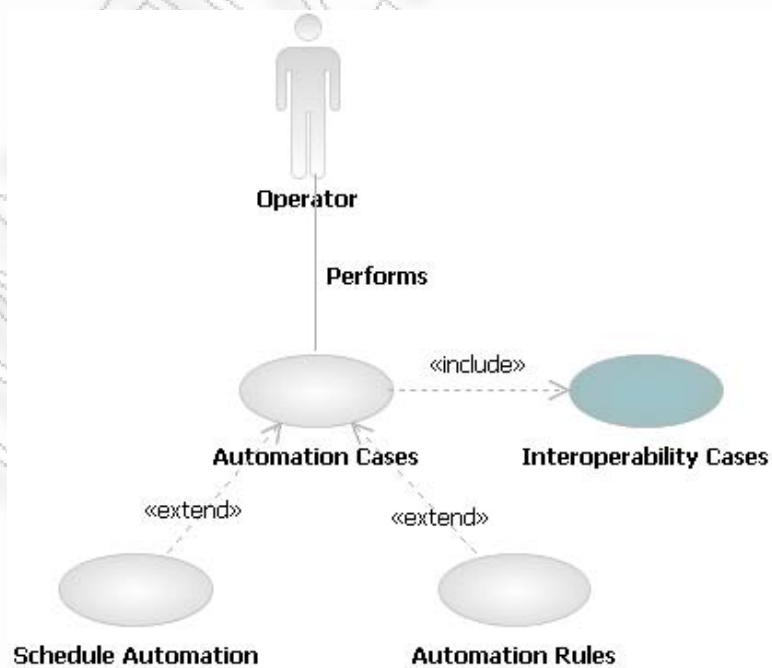
Εικόνα 2.1.7 Διάγραμμα διαχείρισης Web Services

Διάγραμμα δευτέρου επιπέδου – «Data to Web Services»



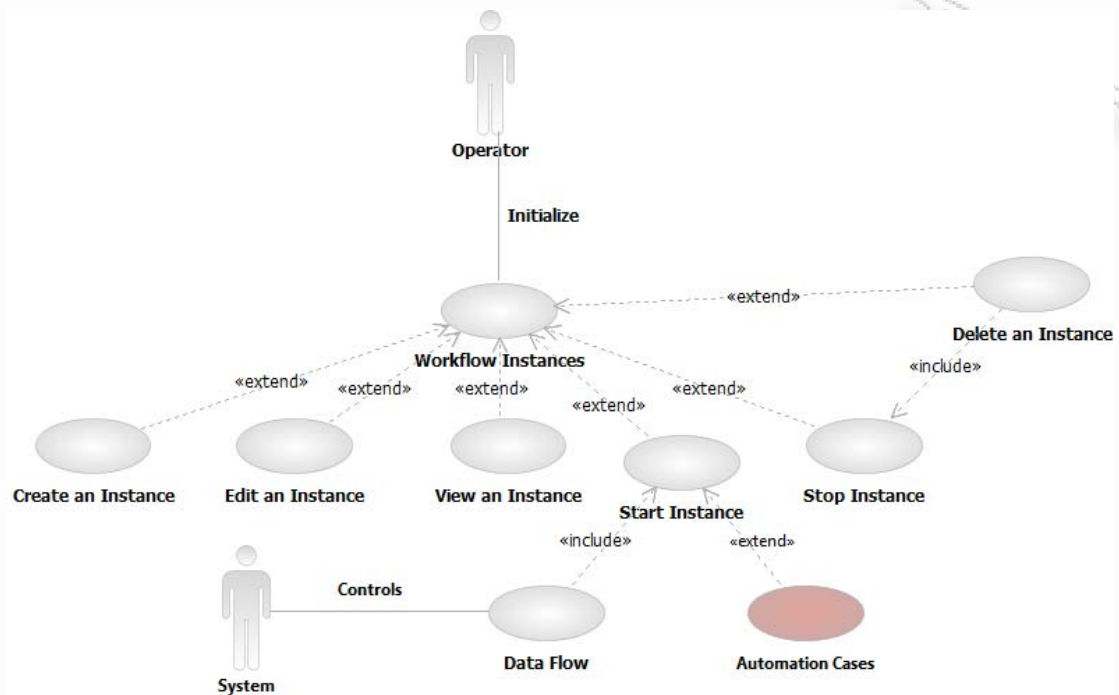
Εικόνα 2.1.8 Διάγραμμα αποστολής δεδομένων σε Web Services

Διάγραμμα δευτέρου επιπέδου – «Περιπτώσεις αυτοματισμού»



Εικόνα 2.1.9 Διάγραμμα περιπτώσεων αυτοματισμού

Διάγραμμα δευτέρου επιπέδου – «Workflow Instances»



Εικόνα 2.1.10 Διάγραμμα χρήσης workflow υποσυστήματος

Περιγραφή περιπτώσεων χρήσεων

Παρακάτω θα γίνει η περιγραφή όλων των περιπτώσεων χρήσεων που φάνηκαν από τα διαγράμματα των παραπάνω εικόνων. Επειδή έχουμε μόνο έναν χρήστη του συστήματος (που αναφέρθηκε ως «Operator») και το ίδιο το σύστημα (που αναφέρθηκε ως «System») δε θα αναφερθούν άλλοι κατά την περιγραφή.

Περιγραφή της περίπτωσης χρήσης «Login»

Περιγραφή	Ένας εγγεγραμμένος χρήστης του συστήματος μπορεί να εισαχθεί στο σύστημα και ύστερα να εκτελέσει διάφορες λειτουργίες σε αυτό
Βασική ροή	Το σύστημα του εμφανίζει να δώσει το Password. Ο χρήστης το συμπληρώνει και το στέλνει στο σύστημα, το οποίο θα τον αναγνωρίσει και θα τον εισάγει στην κεντρική οθόνη του προγράμματος.
Εναλλακτική ροή	Λανθασμένος κωδικός. Το σύστημα ζητάει την επανάληψη της εισαγωγής του κωδικού του χρήστη.

Περιγραφή της περίπτωσης χρήσης «Interoperability Cases»

Περιγραφή	<p>Ο χρήστης μπορεί να συνδυάσει διαφόρων ειδών συνδέσεων, για να ορίσει τα δεδομένα τα οποία θέλει από αυτές. Αν η σύνδεση είναι τύπου XML, τότε επιστρέφονται όλα τα δεδομένα του xml στη μορφή που έχουν. Αν είναι βάση τότε επιστρέφονται όλοι οι πίνακες και όλα τα πεδία του κάθε πίνακα.</p> <p>Εφόσον έχει τις συνδέσεις, τότε μπορεί να ορίσει τις συσχετίσεις μεταξύ τους και να επιλέξει την εξαγωγή των πεδίων που θέλει σε συνδυασμό. Μπορεί να ορίσει κανόνες για τη μερική λήψη των πεδίων όπως και να τα μετασχηματίσει, σύμφωνα με τις εκάστοτε ανάγκες του. Τέλος ορίζει την έξοδο των δεδομένων σαν μία καινούργια σύνδεση.</p>
Βασική ροή	<p>Το σύστημα εμφανίζει μία φόρμα όπου μπορεί να γίνει η εισαγωγή συνδέσεων μέσα από στατικά (στο πάνω μέρος της φόρμας) και δυναμικά (μέσα από το δεξί πλήκτρο του ποντικού) μενού. Με την εισαγωγή θα εμφανίζονται κάποια πλαίσια που θα προβάλλουν τα δεδομένα που έχουν διαθέσιμα. Θα εμφανίζεται το τελικό σχήμα όπου θα ορίσει τα πεδία που θέλει να εξάγει καθώς και το είδος της εξαγόμενης σύνδεσης. Η φόρμα θα είναι έτσι σχεδιασμένη ώστε να επιτρέπει τη μετακίνηση και την αλλαγή μεγέθους του τελικού σχήματος αλλά και των αρχικών πλαισίων. Τέλος θα υπάρχουν δύο κουμπιά για την επικύρωση και την εκτέλεση της διαδικασίας.</p>
Εναλλακτική ροή	<p>Μηνύματα λάθους μέσα από την επικύρωση και από εμφανιζόμενα παράθυρα διαλόγου, κατά τον λανθασμένο ορισμό του τελικού σχήματος ή τον ελλιπή ορισμό των συσχετίσεων μεταξύ των συνδέσεων.</p>
Προ-συνθήκες	<p>Να έχει γίνει η εισαγωγή στο σύστημα (Login)</p>
Μετά-συνθήκες	<p>Η εμφάνιση παραθύρου για την αποθήκευση ή την αποστολή των δεδομένων σε web service, όταν η εξαγωγή τους έγινε σε xml τύπου σύνδεσης.</p>
Extends	<p>Modify Data, Create Data Selection Rules, Add Connections To Use, Set Final Connection, Set Final Data Schema, Validate</p>
Includes	<p>Connections</p>
Ειδικές απαιτήσεις	<p>Να είναι σωστά ορισμένο το τελικό σχήμα. Δε θα πρέπει να περιέχει παραπάνω από δύο επίπεδα όταν είναι σχήμα βάσης.</p>
Σημειώσεις	<p>-</p>

Περιγραφή της περίπτωσης χρήσης «Διαχείριση Connections»

Περιγραφή	<p>Ο χρήστης μπορεί να διαχειριστεί τις συνδέσεις σε βάσεις δεδομένων ή σε xml αρχεία.</p> <p>Έχει τη δυνατότητα να προσθέσει μία νέα σύνδεση, να τη</p>
------------------	--

	δοκιμάσει για να δει αν τα στοιχεία που έδωσε είναι σωστά (στην περίπτωση της βάσης δεδομένων), να την επεξεργαστεί, να την προβάλλει (χωρίς δυνατότητα επεξεργασίας) και να τη διαγράψει.
Βασική ροή	Το σύστημα εμφανίζει μία διαχειριστική φόρμα η οποία περιλαμβάνει σε γραμμές όλες τις διαθέσιμες συνδέσεις. Από εκεί μπορεί να επιλέξει μία από αυτές και να τις επεξεργαστεί, να τις προβάλλει και να τις διαγράψει. Επίσης δίνεται η δυνατότητα μέσω ενός κουμπιού προσθήκης να εισάγει στο σύστημα μία νέα σύνδεση. Από εκεί αν είναι τύπου βάσης, μπορεί να επιλέξει τον τύπο της σύνδεσης και να δοκιμάσει αν την όρισε σωστά. Επίσης εφόσον έχει ορίσει έστω και μία σύνδεση, τότε μπορεί να την επιλέξει από τη διαχειριστική φόρμα και μέσω ενός κουμπιού «Select» να την εισάγει στη φόρμα που περιγράφηκε στην περίπτωση « <i>Interoperability Cases</i> ».
Εναλλακτική ροή	Στην περίπτωση που δεν δώσει σωστά credentials για τη βάση δεδομένων, θα του εμφανιστεί ειδικό μήνυμα λάθους. Δεν έχουν οριστεί όλα τα απαραίτητα πεδία κατά τη διαχείριση μίας σύνδεσης.
Προ-συνθήκες	Να έχει γίνει η εισαγωγή στο σύστημα (Login)
Μετά-συνθήκες	-
Extends	Edit Existing Connection, Select Connection, View Existing Connection, Delete Connection, Add New Connection
Includes	-
Ειδικές απαιτήσεις	Να μπορεί το πρόγραμμα να έχει αρκετά permissions για να κάνει ερωτήματα και να διαχειριστεί τις συνδέσεις που ορίζει ο χρήστης.
Σημειώσεις	Συνολικά χρησιμοποιούνται δύο φόρμες. Μία που περιέχει σε γραμμές όλες τις συνδέσεις και μία που λειτουργεί είτε για προσθήκη, είτε για επεξεργασία, είτε για προβολή.

Περιγραφή της περίπτωσης χρήσης «Διαχείριση Web Services»

Περιγραφή	Ο χρήστης μπορεί να διαχειριστεί τα Web Services με τα οποία μπορεί να επικοινωνήσει το σύστημα. Έχει τη δυνατότητα να προσθέσει πληροφορίες για τη σύνδεση σε ένα Web Service, όπως επίσης και να επεξεργαστεί / προβάλλει και διαγράψει ένα ήδη υπάρχων.
Βασική ροή	Το σύστημα εμφανίζει μία διαχειριστική φόρμα η οποία περιλαμβάνει σε γραμμές όλες τις διαθέσιμες συνδέσεις για web services. Από εκεί μπορεί να επιλέξει μία από αυτές και να τις επεξεργαστεί, να τις προβάλλει και να τις διαγράψει. Επίσης δίνεται η δυνατότητα μέσω ενός κουμπιού προσθήκης να εισάγει στο σύστημα μία νέα σύνδεση. Επίσης εφόσον έχει ορίσει έστω και μία σύνδεση σε web service, τότε μπορεί να την επιλέξει από τη διαχειριστική φόρμα και μέσω ενός κουμπιού «Select» να την εισάγει στη

	φόρμα που θα περιγραφθεί στην περίπτωση «Data to Web Services».
Εναλλακτική ροή	Δεν έχουν οριστεί όλα τα απαραίτητα πεδία κατά τη διαχείριση ενός web service.
Προ-συνθήκες	Να έχει γίνει η εισαγωγή στο σύστημα (Login)
Μετά-συνθήκες	-
Extends	Add new Web Service, Edit Existing Web Service, View Existing Web Service, Delete Web Service, Select Web Service.
Includes	-
Ειδικές απαιτήσεις	-
Σημειώσεις	Συνολικά χρησιμοποιούνται δύο φόρμες. Μία που περιέχει σε γραμμές όλες τις συνδέσεις σε web services και μία που λειτουργεί είτε για προσθήκη, είτε για επεξεργασία, είτε για προβολή.

Περιγραφή της περίπτωσης χρήσης «Data to Web Services»

Περιγραφή	Ο χρήστης μπορεί να στείλει δεδομένα που βρίσκονται σε κάποιο αρχείο στο σκληρό του δίσκο σε κάποιο web service.
Βασική ροή	Δίνεται μια φόρμα μέσα από την οποία ο χρήστης μπορεί να ορίσει τη διαδρομή του δίσκου από την οποία θα διαβάσει το αρχείο με τα δεδομένα. Τα δεδομένα μπορεί να είναι είτε απλό κείμενο, είτε xml. Επίσης παρέχεται και ένα ειδικό κουμπί για την εισαγωγή του xml αρχείου που έχει οριστεί σε κάποια σύνδεση τύπου xml. Σε αυτό το σημείο ανοίγει η διαχειριστική φόρμα για να επιλεγεί το xml εξόδου που έχει οριστεί σε αυτή. Έπειτα μέσω ειδικού κουμπιού επιλέγει μέσα από τη διαχειριστική φόρμα που αναφέρθηκε στο διάγραμμα Διαχείριση Web Services.
Εναλλακτική ροή	Μηνύματα λάθους σε περίπτωση που δεν έχουν οριστεί και το αρχείο και το Web Service. Άλλο μήνυμα σφάλματος εμφανίζεται όταν δεν πραγματοποιείται επιτυχής σύνδεση του προγράμματος με το web service. Τέλος μήνυμα σφάλματος εμφανίζεται και όταν το αρχείο που ορίστηκε στη διαδρομή δίσκου δεν υπάρχει ή δεν υπάρχουν δικαιώματα ανάγνωσης.
Προ-συνθήκες	Να έχει γίνει η εισαγωγή στο σύστημα (Login)
Μετά-συνθήκες	-
Extends	Import Any Text Type File
Includes	-
Ειδικές απαιτήσεις	Να υπάρχει σύνδεση του υπολογιστή στον οποίο τρέχει το πρόγραμμα και του υπολογιστή που φιλοξενεί το web service.
Σημειώσεις	-

Περιγραφή της περίπτωσης χρήσης «Περιπτώσεις αυτοματισμού»

Περιγραφή	Δημιουργία αυτόματων διαδικασιών οι οποίες εκτελούν τα Interoperability Cases, ανά χρονικά διαστήματα.
Βασική ροή	Ο χρήστης δημιουργεί ένα Interoperability Case, και ορίζει κανόνες πάνω σε αυτό. Οι κανόνες ορίζονται επάνω στον τρόπο και στον αριθμό εξαγωγής των ίδιων δεδομένων. Ο χρήστης ορίζει τις διάφορες παράλληλες εξόδους που επιθυμεί και έπειτα τις κάνει schedule / τις προγραμματίζει να εκτελεστούν ανά κάποιο χρονικό διάστημα.
Εναλλακτική ροή	Μήνυμα σφάλματος όταν ο χρήστης δεν ορίσει από τη φόρμα ποιο Interoperability Case θα χρησιμοποιήσει.
Προ-συνθήκες	Να έχει γίνει η εισαγωγή στο σύστημα (Login) Να έχουν φτιαχτεί Interoperability Cases
Μετά-συνθήκες	-
Extends	Schedule Automation, Automation Rules
Includes	Interoperability Cases
Ειδικές απαιτήσεις	Να έχει ήδη οριστεί κάποιο Interoperability Case
Σημειώσεις	-

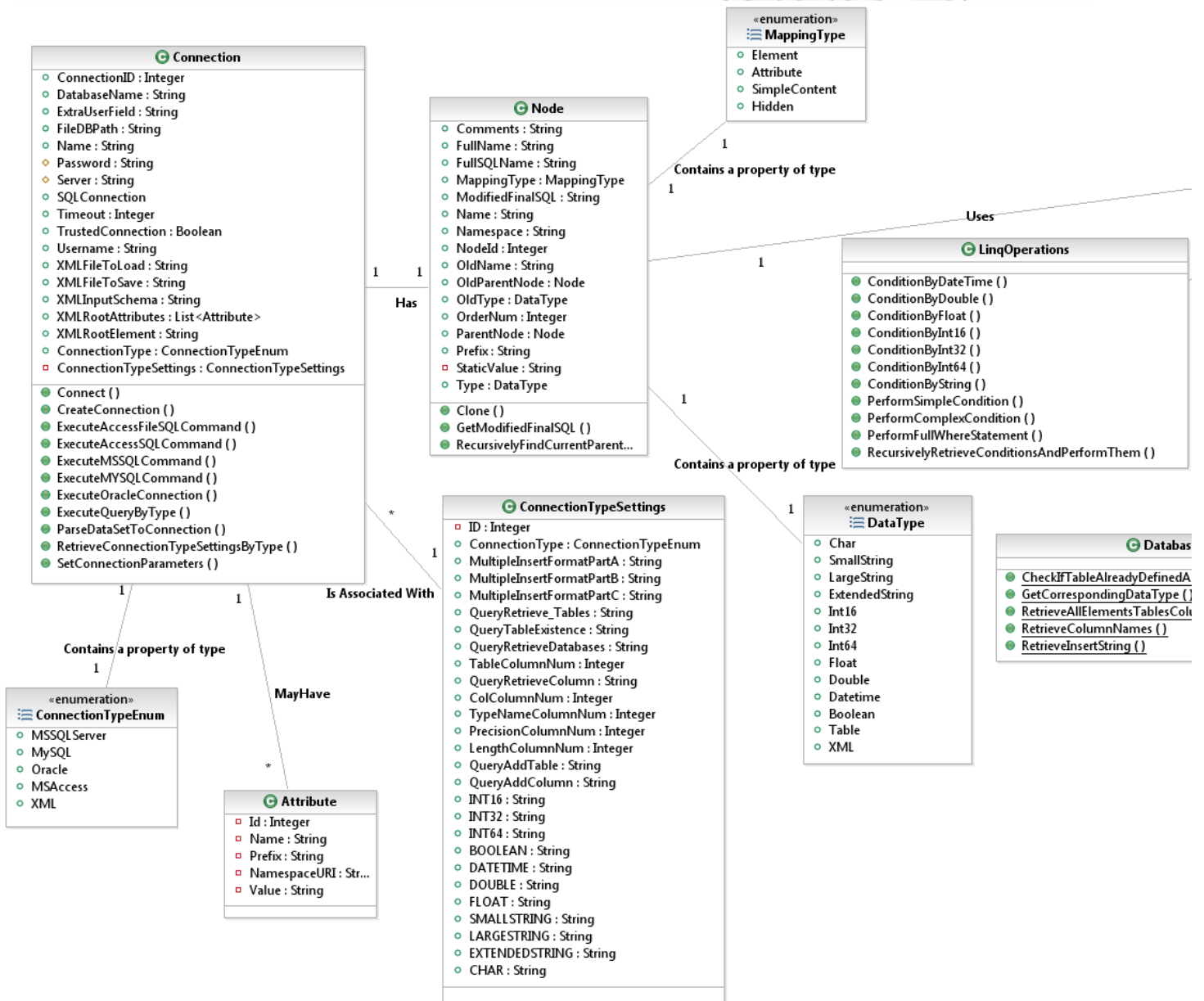
Περιγραφή της περίπτωσης χρήσης «Workflow Instances»

Περιγραφή	Διαχείριση workflow instances για τη δημιουργία αποφάσεων έτσι ώστε να κατευθυνθεί σωστά η ροή των δεδομένων ανάλογα με την τρέχουσα κατάσταση του συστήματος.
Βασική ροή	Υπάρχει μία διαχειριστική φόρμα στην οποία προβάλλονται όλα τα workflow instances που έχει δημιουργήσει ο χρήστης. Αυτά και άλλα μπορεί να τα διαχειριστεί πλήρως μέσα από τις λειτουργίες προσθήκης / επεξεργασίας / προβολής και διαγραφής. Αυτές οι λειτουργίες πραγματοποιούνται από μία φόρμα στην οποία ορίζεται η ροή των δεδομένων. Συγκεκριμένα υπάρχει μηχανισμός όπου μέσα από τα Connections και από τα Web Service Connections, μπορεί να ελέγχεται το output που παρέχουν αυτά, και έτσι να ορίζονται αποφάσεις μέσα από το συνδυασμό τέτοιας εξόδου. Η τελική απόφαση θα είναι της μορφής «SUCCESS» ή «FAILURE». Αν είναι η πρώτη τότε θα γίνει η μεταφορά όπως έχει προδιαγραφθεί στο Automation Cases. Κατά την εκτέλεση του Interoperability Case το σύστημα ελέγχει για σφάλματα και κρατάει log σε αρχείο για αυτά.
Εναλλακτική ροή	Περιπτώσεις που δε μπορεί να επικοινωνήσει με κάποια βάση ή να διαβάσει τα δεδομένα από κάποιο xml ή ακόμα και να μη μπορεί να επικοινωνήσει με κάποιο web service τότε επιστρέφει το «FAILURE», και κρατιέται ειδικό αρχείο με logs που να περιγράφει τα δεδομένα λάθους.

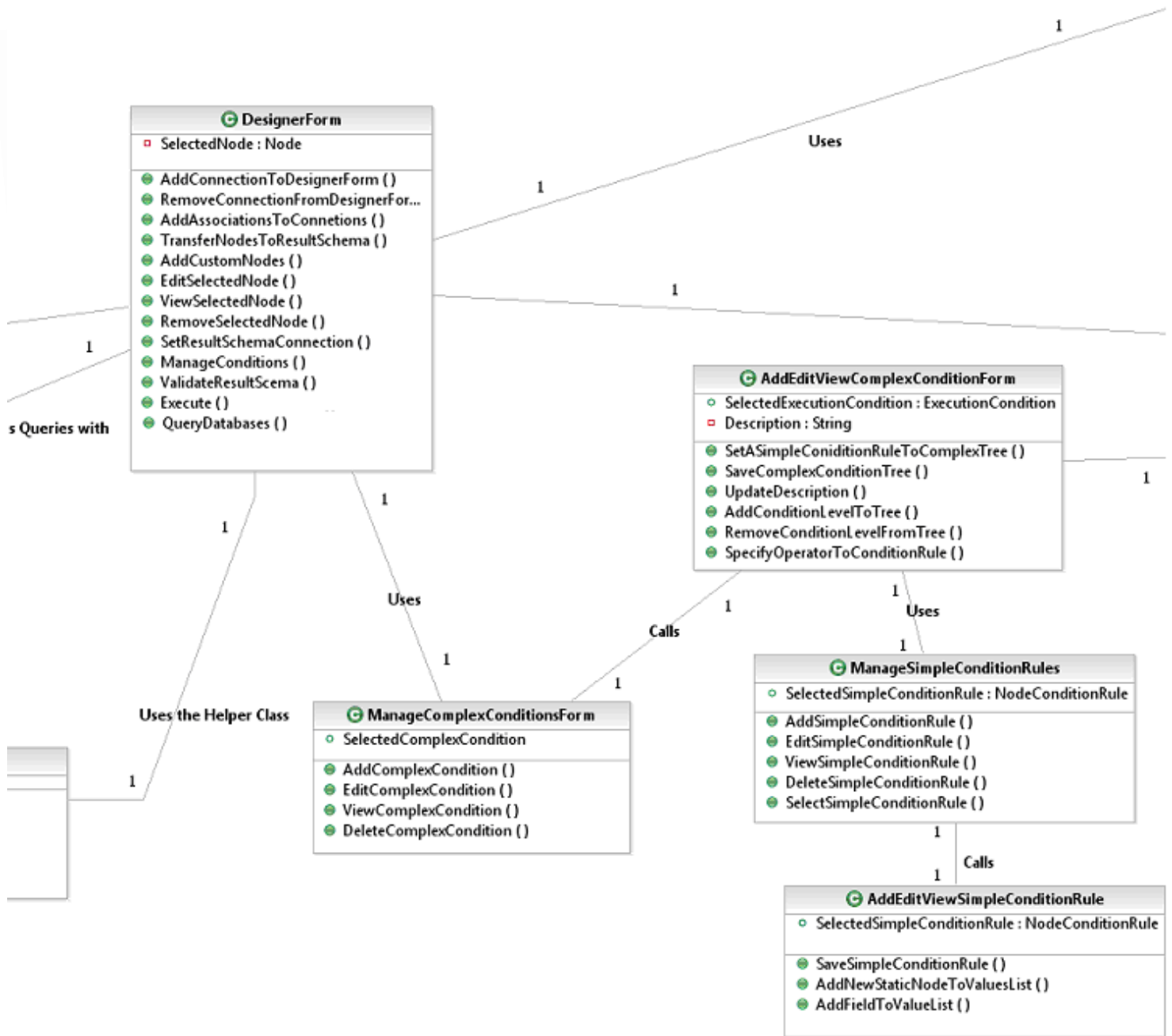
Προ-συνθήκες	Να έχει γίνει η εισαγωγή στο σύστημα (Login) Να έχουν φτιαχτεί Automation Cases Να έχουν φτιαχτεί Interoperability Cases
Μετά-συνθήκες	-
Extends	Create an Instance, Edit an Instance, View an Instance, Start Instance, Stop Instance, Delete Instance
Includes	-
Ειδικές απαιτήσεις	-
Σημειώσεις	-

2.1.3.2 Διαγράμματα κλάσεων

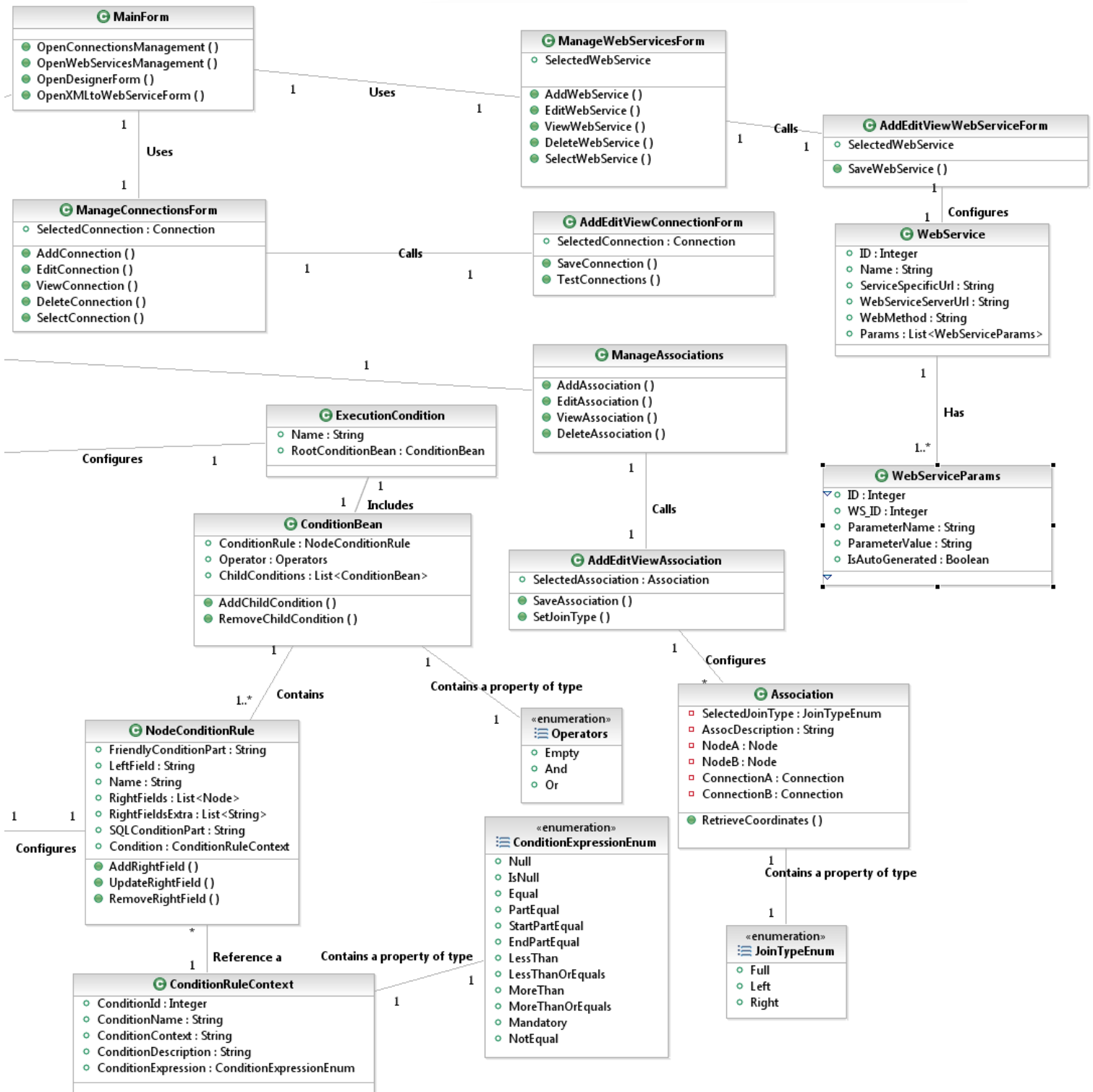
Η παρακάτω τρεις εικόνες δείχνουν το διάγραμμα κλάσεων που χρησιμοποιήθηκε για να φτιαχτεί ο Ultimate Mediator. Επάνω σε αυτό θα γίνει μία προσπάθεια περιγραφής των βασικών κλάσεων και των τρόπων χρησιμοποίησής τους από την εφαρμογή. Επίσης θα περιγραφτούν κάποια βασικά για τη λειτουργία του συστήματος πεδία. Σχεδόν όλα τα πεδία μπορούν να κατανοηθούν από τον αναγνώστη λόγω του ονόματός τους. Παρόλα αυτά θα δοθεί σημασία σε κάποια που παίζουν σημαντικό ρόλο στη λειτουργία του προγράμματος.



Εικόνα 2.1.11 Διάγραμμα κλάσεων υλοποίησης του Ultimate Mediator (α)



Εικόνα 2.1.12 Διάγραμμα κλάσεων υλοποίησης του Ultimate Mediator (β)



Εικόνα 2.1.13 Διάγραμμα κλάσεων υλοποίησης του Ultimate Mediator (γ)

Μία από τις βασικότερες κλάσεις που θα πρέπει να λάβει υπόψη του ο αναγνώστης είναι η κλάση Node (εφεξής κόμβος). Η κλάση αυτή αναπαριστά ένα πεδίο το οποίο μπορεί να έχει δεδομένα, ή να περιλαμβάνει άλλα πεδία τα οποία να έχουν δεδομένα, ή και εκείνα με τη σειρά τους να περιλαμβάνουν άλλα κ.λπ..

Το πεδίο αυτό λοιπόν, μπορεί να είναι μία στήλη στις βάσεις δεδομένων ή μία οντότητα (XML Element) σε μία δομή xml. Έτσι αν για παράδειγμα έχουμε τον πίνακα Customers ο οποίος έχει δύο πεδία το cu_id και το cu_name, τότε η αναπαράσταση από το πρόγραμμα θα γίνει ως εξής. Θα δημιουργηθούν τρεις κόμβοι από τους οποίους ο ένας θα είναι ο customers, και οι άλλοι δύο θα είναι τα πεδία cu_id και cu_name. Τα πεδία αυτά όμως δεν έχουν απλά μία τιμή. Είναι πολλά τα χαρακτηριστικά που θα πρέπει να οριστούν για αυτά. Συγκεκριμένα μπορούμε να αναφέρουμε το ModifiedFinalSQL, το οποίο αναπαριστά το τελικό sql που θα πρέπει να οριστεί από το χρήστη ή που δημιουργείται αυτόματα από το σύστημα. Το πεδίο αυτό μπορεί να περιέχει απλό κώδικα για μετατροπή δεδομένων. Δηλαδή θα μπορούσε να παίρνει τιμή από ένα άλλο πεδίο και να το προσθέτει σε αυτό αφού πρώτα το διαιρέσει με κάποιο αριθμό.

Αντίστοιχα υπάρχουν και άλλα χαρακτηριστικά όπως το MappingType το οποίο ορίζει αν το πεδίο είναι κρυφό ή κανονικό ή (στην περίπτωση που έχουμε xml έξοδο) είναι ιδιότητα.

Η κλάση Connection ορίζει τη σύνδεση του προγράμματος με κάποια βάση δεδομένων ή κάποιο αρχείο xml. Παρέχει λοιπόν όλα τα πεδία και τις λειτουργίες (functions), για να μπορέσει να πραγματοποιηθεί μία σύνδεση. Με τη βοήθεια της κλάσης Database μπορεί να γίνει η λήψη πληροφοριών βάσης. Συγκεκριμένα η κλάση Database περιέχει στατικές μεθόδους για την ανάγνωση των πινάκων που παρέχει η σύνδεση και την ανάγνωση των πεδίων κάθε πίνακα («RetrieveAllElementsTablesColumns»). Επίσης το πρόγραμμα μπορεί και λαμβάνει πληροφορίες για τον τύπο δεδομένων των κάθε στηλών της βάσης.

Η κλάση Connection μεταξύ άλλων μεταβλητών οι οποίες είναι αυτές που ορίζει ο χρήστης από τη φόρμα (όπως server, username, password κλπ.), περιέχει και την ConnectionTypeSettings. Αυτή η μεταβλητή είναι ένα instance, δηλαδή ένα αντικείμενο της ομώνυμης κλάσης «ConnectionTypeSettings». Το αντικείμενο αυτής της κλάσης περιέχει όλες τις ρυθμίσεις για την εκτέλεση ερωτημάτων όπως αυτό της λήψης των πινάκων. Επίσης περιέχεται πληροφορία για την αντιστοίχιση των τύπων δεδομένων που ορίζονται στο enumeration «DataType». Από το διάγραμμα κλάσεων βλέπουμε ότι η συγκεκριμένη κλάση περιέχει το πεδίο «INT16», «BOOLEAN», «LARGESTRING» κλπ. Σε αυτά τα πεδία μπαίνουν εγγραφές όπως *short*, *bit*, *varchar(500)* αντίστοιχα. Έτσι αν ορίσουμε έναν πίνακα που δεν υπάρχει, ή ένα πεδίο που δεν υπάρχει στον ήδη ορισμένο πίνακα, τότε δημιουργείται ένα ερώτημα προσθήκης πίνακα ή στήλης (ανάλογα με την περίπτωση) και το αντίστοιχο πεδίο αν είναι τύπου Boolean τότε θα πάρει οριστεί στη βάση με την τιμή bit.

Τέλος η Connection κλάση μαζί με την κλάση Attributes ορίζει τις ιδιότητες που θα έχει το ριζικό – root πεδίο στο τελικό xml που θα εξαχθεί. Περιέχει πληροφορίες για το Prefix, το Namespace, το Name και το Value της ιδιότητας. Η σχέση είναι ένα προς πολλά όπως μπορούμε να δούμε. Δηλαδή μπορεί να οριστούν για μία σύνδεση πολλά Attributes. Στην περίπτωση που η σύνδεση δεν είναι xml τότε δεν μπορεί να υπάρχει association μεταξύ τους.

Η κλάση «LinqOperations» περιέχει μεθόδους με τις οποίες μπορούν να γίνουν τα ερωτήματα σε Linq για την επιλογή των δεδομένων. Για παράδειγμα έχουμε την «ConditionByIn16» μέθοδο. Αυτή παίρνει τα δεδομένα και εκτελεί ένα ερώτημα επάνω σε αυτά, με βάση κάποιο πεδίο που γνωρίζει ότι έχει Int16 DataType. Έτσι μπορεί για παράδειγμα να ελέγξει αν η ηλικία ενός ατόμου είναι μεγαλύτερη από 30. Το πρόγραμμα περνάει αυτά τα χαρακτηριστικά για να μπορέσει να εκτελέσει

το ερώτημα. Ο λόγος που έχουν χωριστεί ανάλογα με το DataType τους είναι γιατί η Linq όπως θα αναφέρουμε στην περιγραφή της στην ενότητα *LINQ (Language Integrated Query)*, έχει κάποια μειονεκτήματα όπως η αδυναμία εκτέλεσης συνθήκης χωρίς να είναι γνωστός ο τύπος δεδομένων του.

Η κλάση «ExecutionCondition» περιέχει τον ορισμό μίας σύνθετης συνθήκης. Περιλαμβάνει το όνομα της σύνθετης συνθήκης (το οποίο εισάγεται καθαρά για να την καλύτερη κατανόηση του χρήστη), και το ριζικό – root RootConditionBean αντικείμενο (της κλάσης ConditionBean) του δέντρου συνθηκών. Έτσι το πρόγραμμα μπορεί να ξεκινήσει από το ριζικό και μέσω της κλάσης ConditionBean, να πάρει από το πεδίο NodeConditionRule, την πραγματική τιμή της απλής συνθήκης. Επιπλέον η κλάση ConditionBean περιέχει και την πληροφορία για το αν η ίδια, με αυτές που έχει για παιδιά, θα συνδέονται με Operator AND ή OR. Τα παιδιά της συνθήκης ορίζονται στο ChildConditions, και αυτά αποτελούν ουσιαστικά μία λίστα με αντικείμενα της κλάσης ConditionBean. Αυτά με τη σειρά του μπορεί να έχουν μία λίστα από παιδιά με απλές συνθήκες κλπ.

Η κλάση NodeConditionRule ορίζει ουσιαστικά μία απλή συνθήκη. Ορίζει δηλαδή το πεδίο που θα ελεγχθεί, ένα όνομα για την συνθήκη και μία τιμή από το ConditionRuleContext enumeration (πεδίο Condition) που ορίζει τον τρόπο σύγκρισης. Επίσης τη λίστα με τις τιμές που θα συγκρίνει τις κρατάει στο RightFields ενώ το LeftField είναι το πεδίο προς σύγκριση.

Η κλάση WebService και η WebServiceParams ορίζουν τη σύνδεση σε ένα web service. Η σύνδεση μπορεί να περιλαμβάνει στοιχεία όπως τη διεύθυνση του service, τη μέθοδο που θα καλέσει και τις παραμέτρους που θα περάσει στη μέθοδο. Η συσχέτιση εδώ είναι 1 προ 1..*. Ο λόγος είναι γιατί κάθε Web Service που ορίζεται θα πρέπει υποχρεωτικά να έχει μία παράμετρο η οποία θα παίρνει τιμή από τη δυναμική εισαγωγή του xml αρχείου. Θα παίρνει δηλαδή τα δεδομένα του αρχείου αυτού, είτε αυτό είναι xml, είτε απλό αρχείο κειμένου. Το αν θα είναι η παράμετρος δυναμική ή όχι, ορίζεται από το πεδίο «IsAutoGenerated». Οπότε ο αριθμός των παραμέτρων μπορεί να είναι μόνο η δυναμική ή και κάποια άλλη στατική που θα πρέπει να τη γνωρίζει ο χρήστης για να την εισάγει στο σύστημα μέσω κατάλληλων διαμορφωμένων φορμών.

Τις παραπάνω κλάσεις τις χρησιμοποιούν οι φόρμες που έχουν οριστεί με την κατάληξη «...Form». Οι φόρμες περιέχουν όλα τα πεδία και τα Events για να ελέγχουν την κάθε είσοδο και έξοδο, έτσι ώστε να προβάλλουν σωστά μηνύματα στο χρήστη σε κάθε διαδικασία. Επειδή όλες οι διαδικασίες που χρησιμοποιούν οι φόρμες είναι standard (add/edit/view/delete), αξίζει απλά να αναφέρουμε τη φόρμα «DesignerForm», στην οποία ο χρήστης κάνει όλο τον σχεδιασμό και τη διαχείριση, έτσι ώστε να μπορέσει να εκτελέσει μία περίπτωση διαλειτουργικότητας.

Η κλάση «DesignerForm» περιέχει μεθόδους όπως «AddCustomNodes», «EditSelectedNode», «RemoveSelectedNode» με τις οποίες μπορούν να διαχειριστούν τους κόμβους που έχουν οριστεί στο τελικό σχήμα. Αυτές οι μέθοδοι μπορούν να καλεστούν μέσα από διάφορα events που ενεργοποιούνται κατά τη χρήση της φόρμας. Η μέθοδος «SetResultSchemaConnection», εμφανίζει τη διαχειριστική φόρμα των συνδέσεων στο χρήστη για να μπορέσει να επιλέξει τη σύνδεση που θα έχει το τελικό σχήμα. Με τον ίδιο περίπου τρόπο λειτουργεί και η «AddConnectionToDesignerForm», όπου εμφανίζει τη διαχειριστική φόρμα έτσι

ώστε να γίνει η εισαγωγή μίας σύνδεσης από όπου μετέπειτα ο χρήστης θα μπορεί να επιλέξει τα δεδομένα που θα εξάγει στο τελικό σχήμα.

Η μέθοδος «AddAssociationToConnections», προβάλλει τη διαχειριστική φόρμα εισαγωγής / επεξεργασίας / προβολής / διαγραφής των συσχετίσεων μεταξύ των συνδέσεων. Από εκεί μπορούν να οριστούν τα πεδία που συνδέονται με τα πεδία της άλλης σύνδεσης ή ακόμα και της ίδιας κάποιου συγκεκριμένου πίνακα. Ορίζει δηλαδή συνήθως τις συνδέσεις μεταξύ πρωτεύων και ξένων κλειδιών. Η μέθοδος «TransferNodesToResultSchema», έχει σαν λειτουργία να δημιουργεί κλώνους των αρχικών κόμβων και να τα μεταφέρει στο τελικό σχήμα.

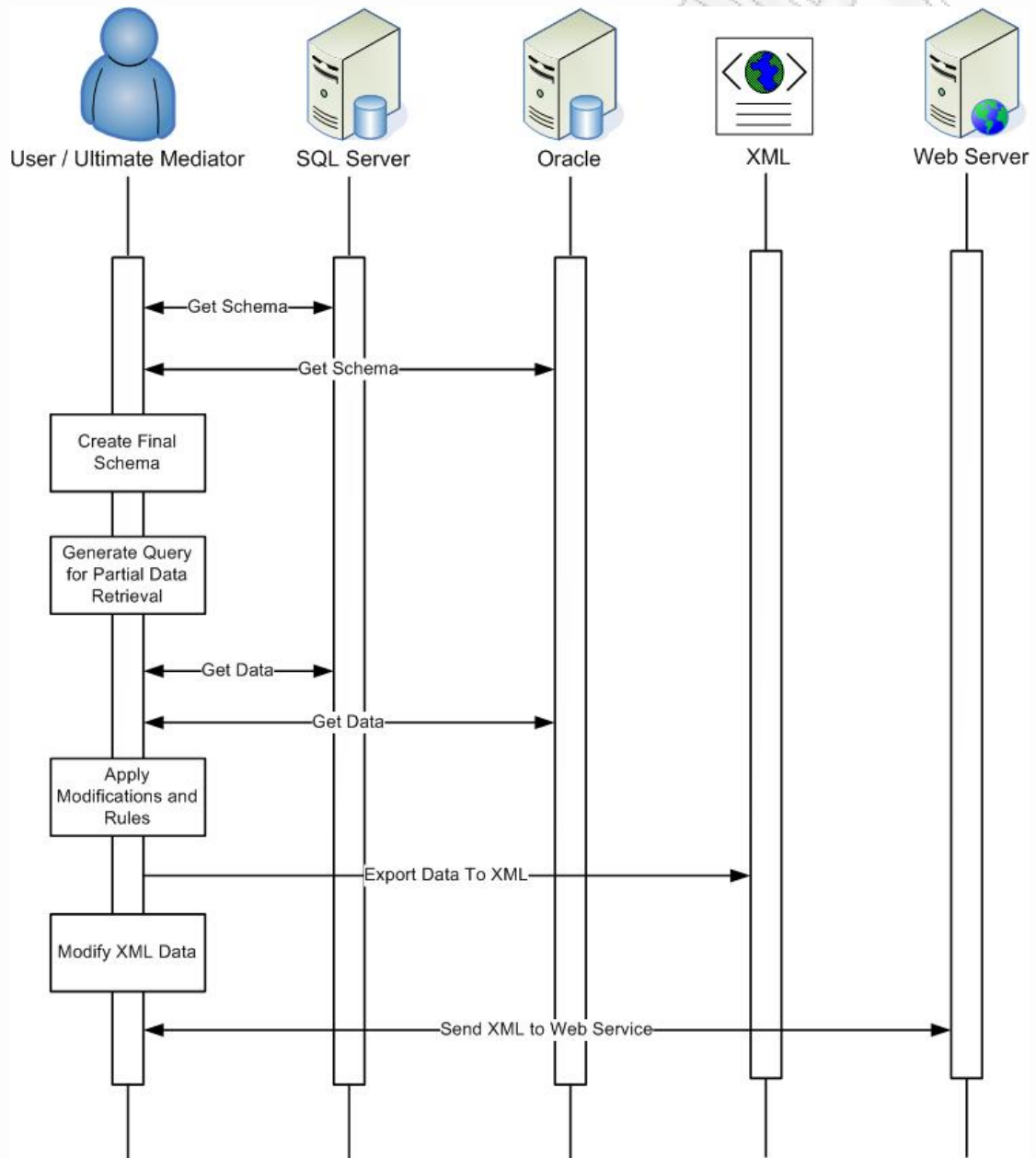
Επίσης υπάρχουν και δύο ακόμα μέθοδοι, μία για την επικύρωση και μια για την εκτέλεση της διαδικασίας, οι οποίες έχουν ως στόχο την εκτέλεση της ορισμένης εργασίας. Η επικύρωση ελέγχει αν το τελικό σχήμα έχει οριστεί σωστά και αν οι συσχετίσεις μεταξύ των συνδέσεων είναι ορισμένες πλήρως. Η διαδικασία «Execute» εκτελεί όλη την εργασία, από την επιλογή των δεδομένων, μέχρι την επεξεργασία τους και την τελική μεταφορά τους (σύμφωνα πάντα με ότι έχουν οριστεί ως κανόνες και πεδία).

Η τελευταία μέθοδος QueryDatabases είναι βοηθητική των μεθόδων εκτέλεσης και επικύρωσης. Έχει σαν στόχο να πάρει τα δεδομένα από τη στατική μέθοδο «RetrieveAllElementsTablesColumns» της κλάσης «Databases», και να τα προβάλλει σε ένα πλαίσιο όπου να φαίνονται καθαρά οι διαθέσιμοι πίνακες και τα πεδία τους με τον τύπο δεδομένων τους.

2.1.3.3 Παραδείγματα επικοινωνίας (sequence diagrams)

Σε αυτήν την ενότητα θα παρουσιαστεί μέσω κυρίως sequence diagrams η διαδικασία επικοινωνίας του λογισμικού με τις διαφορετικές οντότητες σύνδεσης του προγράμματος.

Έστω ότι έχουμε μία περίπτωση όπου θα πάρουμε δεδομένα από δύο συνδέσεις, θα τις εξάγουμε σε xml, έπειτα θα το επεξεργαστούμε και τελικά θα το στείλουμε σε ένα web service. Η Εικόνα 2.1.14 δείχνει αυτήν την περίπτωση μαζί με όλα τα βήματα που θα ακολουθηθούν μέχρι και την τελική αποστολή.



Εικόνα 2.1.14 Sequence Diagram για την εξαγωγή και αποστολή δεδομένων σε Web Service

Αρχικά η εφαρμογή επικοινωνεί με τις δύο βάσεις για να πάρει τα διαθέσιμα σχήματα, που περιλαμβάνουν τους πίνακες με τις στήλες τους και τους ορισμούς τους. Εφόσον έχει και τα δύο ο χρήστης επιλέγει τα πεδία ενδιαφέροντος, με ότι αλλαγές μπορεί να θέλει σε αυτά, και ορίζει, αν το επιθυμεί, συνθήκες για την επιλογή των τελικών δεδομένων μεταφοράς.

Αφού έφτιαξε το τελικό σχήμα, έπειτα μπορεί να εκτελέσει τη μεταφορά. Το πρώτο στάδιο της διαδικασίας της μεταφοράς είναι να φτιαχτεί το σωστό ερώτημα SQL, το οποίο θα πρέπει να εκτελεστεί στη βάση, για να πάρει σαν είσοδο τα δεδομένα και να τα επεξεργαστεί σύμφωνα με τις ανάγκες που έχουν προδιαγραφθεί. Για να μην διαβάσει όλα τα δεδομένα που έχουν οι πίνακες που επέλεξε, διαμορφώνει το SQL ερώτημα έτσι ώστε να τα παίρνει σύμφωνα με τις συνθήκες (Conditions) που έχει ορίσει.

Επειδή όμως οι συνθήκες μπορεί να περιλαμβάνουν πεδία που δεν ορίζονται στο συγκεκριμένο πίνακα, το λογισμικό ξεχωρίζει τις συνθήκες ενδιαφέροντος και εκτελεί μόνο εκείνες, εφόσον δεν επηρεάζουν τα υπόλοιπα δεδομένα. Έστω για παράδειγμα ότι έχουμε τις συνθήκες να ορίζονται ως εξής:

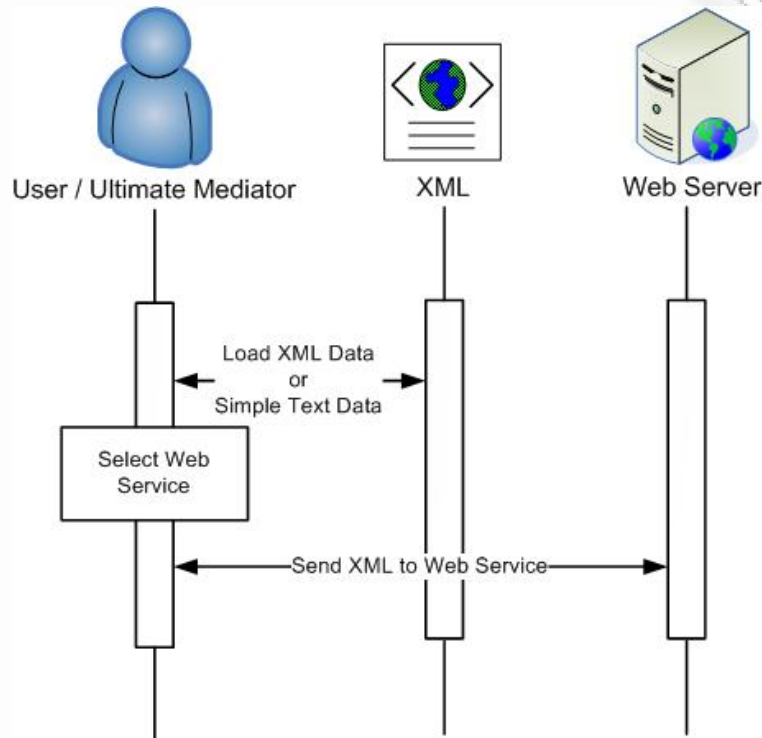
Συνθήκη_A AND Συνθήκη_B AND Συνθήκη_Γ OR (Συνθήκη_Δ AND Συνθήκη_Ε) AND Συνθήκη_Z

Έστω επίσης ότι οι Συνθήκες A, Δ και Z, περιέχουν πεδία τα οποία είναι στον πρώτο πίνακα που διαβάζουμε. Για να γεμίσουμε λοιπόν τον πίνακα με δεδομένα, στο ερώτημα SQL που εκτελούμε, ορίζουμε ότι θα πάρει δεδομένα με βάση την Συνθήκη_A και την Συνθήκη_Z. Οι υπόλοιπες ανήκουν σε άλλο πίνακα, γι' αυτό και δεν τις λαμβάνουμε υπόψη σε αυτόν. Επίσης, η Συνθήκη_Δ μπορεί να ανήκει στον πίνακα αυτό, αλλά βρίσκεται εμφωλευμένη μέσα σε OR που περιλαμβάνει συνθήκη με πεδίο από άλλο πίνακα, και επιπλέον το OR συσχετίζεται και με την Συνθήκη_Γ που και αυτή ανήκει σε άλλο πίνακα. Ακόμα και αν η Συνθήκη_Γ άνηκε στον προαναφερθέντα πίνακα, δεν θα μπορούσαμε να την χρησιμοποιήσουμε γιατί το δεύτερο σκέλος του OR δεν περιλαμβάνει συνθήκες με πεδία του πίνακα ενδιαφέροντος.

Με τον τρόπο αυτό, μπορούμε να μειώσουμε ουσιαστικά το χρόνο τον οποίο η εφαρμογή θα δαπανούσε για να εκτελέσει τη μεταφορά. Αυτό γίνεται γιατί το λογισμικό, αν δεν εκτελούσε την παραπάνω διαδικασία, θα διάβαζε όλα τα δεδομένα που έχουν οι πίνακες, έπειτα θα τα επεξεργαζόταν και σαν τελευταία διαδικασία θα εφάρμοζε τις συνθήκες. Έτσι το σύστημα επιβαρύνεται αρκετά και από θέμα χρόνου διεκπεραίωσης, αλλά και από θέμα μνήμης.

Εφόσον λοιπόν έχει τα δεδομένα από τις δύο συνδέσεις, εφαρμόζει τις συνθήκες και τους κανόνες που όρισε ο χρήστης, στο τελικό σχήμα. Τα δεδομένα που τελικά θα εξάγει, θα τα αποθηκεύσει σε ένα xml με το όνομα και την διαδρομή του δίσκου που έχουν αποθηκευτεί κατά τη δημιουργία της σύνδεσης που επιλέχτηκε για την εξαγωγή αυτών. Εφόσον ολοκληρωθεί η διαδικασία, υπάρχει η δυνατότητα να γίνει μία επεξεργασία επάνω σε αυτά (το στάδιο αυτό είναι προαιρετικό). Τα τελικά δεδομένα αποστέλλονται σε κάποιο web service (προαιρετικό στάδιο), και αυτό με τη σειρά του αποστέλλει πίσω την απάντηση. Εφόσον έχει την απάντηση ο χρήστης μπορεί να τη χρησιμοποιήσει (αν αυτή περιέχει κάποια δεδομένα ενδιαφέροντος) και να τα αποθηκεύσει, ή να τα επεξεργαστεί, ή και να τα αποστείλει σε κάποιο πάλι web service.

Μία έξτρα λειτουργία που υποστηρίζεται, είναι ότι το λογισμικό δίνει τη δυνατότητα να αποστέλλονται δεδομένα απευθείας σε Web Service, χωρίς να χρειαστεί να εκτελέσει όλη την προηγούμενη διαδικασία επιλογής και εξαγωγής δεδομένων. Η Εικόνα 2.1.15 δείχνει ένα άλλο παράδειγμα επικοινωνίας μεταξύ του λογισμικού και κάποιου web service.



Εικόνα 2.1.15: Sequence Diagram για την απευθείας αποστολή δεδομένων σε web service

Η διαδικασία είναι ουσιαστικά ένα κομμάτι της προηγούμενης διαδικασίας που ήδη περιγράφηκε από τη στιγμή που έχει τα xml δεδομένα. Συγκεκριμένα, ο χρήστης επιλέγει το αρχείο με τα δεδομένα (xml ή απλά δεδομένα κειμένου), το οποίο υπάρχει στη διαδρομή δίσκου που ορίζει. Αν τα δεδομένα που θέλει να στείλει τα έχει ήδη ορίσει σε κάποια σύνδεση τύπου xml, τότε μπορεί να επιλέξει το αρχείο από εκεί (δίνεται ειδική λειτουργία όπως θα περιγραφτεί στην ενότητα *XML to Web Service Transporter*). Έπειτα επιλέγει το web service στο οποίο θέλει να αποστείλει τα δεδομένα και τα αποστέλλει.

Μετά την αποστολή το web service στέλνει πίσω μία απάντηση για την οποία ο χρήστης έχει δυνατότητες επεξεργασίας, αποθήκευσης ή και αποστολής σε κάποιο web service. Η διαδικασία αυτή μπορεί να συνεχιστεί όσο χρειάζεται και κρίνεται αναγκαία από τον χρήστη, χωρίς περιορισμός στον αριθμό αποστολής xml.

2.1.4 Μεταφορά δεδομένων σε πραγματικό χρόνο (Online transactions)

Η έως τώρα περιγραφή επικεντρώθηκε κυρίως στις παρεχόμενες λειτουργίες της εφαρμογής Ultimate Mediator, τους τρόπους διασύνδεσης με τρίτες πηγές καθώς και τις διαφορετικές μορφές έκδοσης των αποτελεσμάτων. Η προσέγγιση στηρίχθηκε στη θεώρηση ενός ανεξάρτητου μοντέλου white box, που τροφοδοτείται με δεδομένα, εκτελεί μια σειρά από επεξεργαστικές ενέργειες και παράγει ένα τελικό αποτέλεσμα.

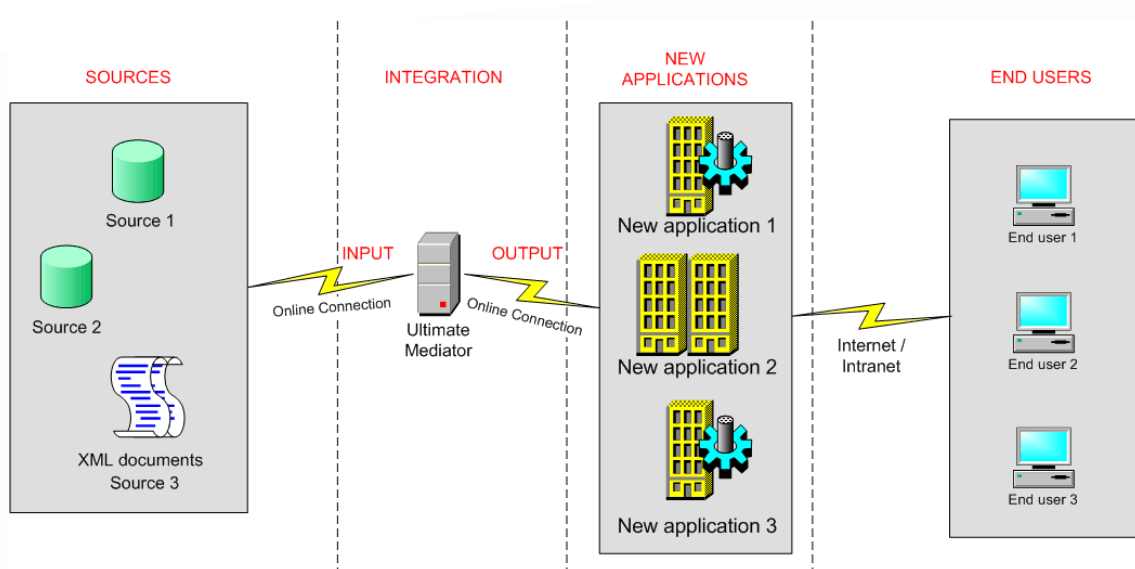
Στο κεφάλαιο *Μεταφορά δεδομένων σε πραγματικό χρόνο (Online transactions)*, ο συγγραφέας επιδιώκει να περιγράψει ένα μοντέλο λειτουργίας της εφαρμογής Ultimate Mediator, που βασίζεται σε ένα πιο διαδραστικό περιβάλλον. Στη συνέχεια η εφαρμογή Ultimate Mediator θα θεωρηθεί ως ένα black box. Οι περιγραφές και αναλύσεις που θα πραγματοποιηθούν, δε θα αναφέρονται στις διαδικασίες που εκτελούνται εσωτερικά, αντιθέτως θα δοθεί έμφαση στις εξωτερικές συνδέσεις εισόδου και εξόδου πληροφορίας από και προς την εφαρμογή.

Προκειμένου να γίνει κατανοητή η ανάγκη λειτουργίας της εφαρμογής Ultimate Mediator σε συνθήκες ενός διαδραστικού περιβάλλοντος, ο αναγνώστης καλείται να θεωρήσει ως παράδειγμα, την ανάγκη αναβάθμισης των συστημάτων και υποσυστημάτων μιας τράπεζας.

Οι περισσότερες τράπεζες συνήθως διαθέτουν πληθώρα συστημάτων και υποσυστημάτων που εκτελούν κρίσιμες, για την απρόσκοπτη λειτουργία τους, ενέργειες. Μετά από χρόνια ελεγχόμενης λειτουργίας, τα πολύπλοκα συστήματα των τραπεζών φθάνουν σε ένα επίπεδο ωριμότητας, που καθιστά απαγορευτική την αντικατάστασή τους από νέα συστήματα που θα υλοποιηθούν από την αρχή και σίγουρα θα παρουσιάσουν προβλήματα.

Ο αναγνώστης καλείται να αναλογιστεί τις επιπτώσεις που θα είχε μια αστοχία λογισμικού σε οργανισμούς όπως τράπεζες, πυρηνικά εργαστήρια, κατασκευαστές αεροπλάνων, κατασκευαστές διαστημικών αεροσκαφών κ.λπ. προκειμένου να γίνει κατανοητό το μέγεθος της κρίσης που θα προκληθεί.

Λύση υπόσχεται να δώσει η εφαρμογή Ultimate Mediator. Η αρχιτεκτονική διασύνδεσης της εφαρμογής παρουσιάζεται στην εικόνα που ακολουθεί (Εικόνα 2.1.16).



Εικόνα 2.1.16 Αρχιτεκτονική διασύνδεσης Ultimate Mediator

Ο Ultimate Mediator δρα ως ενδιάμεσο επίπεδο επικοινωνίας και η χρήση του αποσκοπεί στη διασύνδεση του επιπέδου των «SOURCES», με το επίπεδο των «NEW APPLICATIONS». Οι τεχνικοί της τράπεζας μπορούν πλέον να αναπτύξουν - ή να προσλάβουν ανάδοχο εταιρεία για την ανάπτυξη- νέες(ων) εφαρμογές(ών), χρησιμοποιώντας οποιαδήποτε από τις τεχνολογίες αιχμής (state of the art), χωρίς να ανησυχούν για τη δυνατότητα διασύνδεσης των νέων τεχνολογικών επιλογών, με τα υφιστάμενα συστήματα.

Η διασύνδεση του Ultimate Mediator πραγματοποιείται αυτόματα και δυναμικά με τις πηγές σε πραγματικό χρόνο. Παράλληλα παρέχονται δυναμικοί και ευέλικτοι τρόποι εξόδου των αποτελεσμάτων προς τις νέες εφαρμογές. Ο Ultimate Mediator παρέχει δύο τρόπους δυναμικής διασύνδεσης εξόδου αποτελεσμάτων:

- **WebService:** Το Webservice τροφοδοτεί τις νέες εφαρμογές με την απαραίτητη πληροφορία, που είναι πάντα επίκαιρη (online προσέγγιση) με τη μορφή XML. Οι νέες εφαρμογές που αναπτύσσονται δεν έχουν παρά να «καλέσουν» το Webservice του Ultimate Mediator και να επεξεργαστούν τη δομημένη πληροφορία που επιστρέφεται.
- **Βάση Δεδομένων:** Ο δεύτερος τρόπος που παρέχεται για τη δυναμική διασύνδεση εξόδου αποτελεσμάτων, στηρίζεται σε μια σχεσιακή βάση δεδομένων. Οι νέες εφαρμογές καλούνται να συνδεθούν με την προσωρινή βάση που δημιουργείται για τις ανάγκες διασύνδεσης και να αντλήσουν / επεξεργαστούν την πληροφορία που παρέχεται.

Επιπλέον των παραπάνω, ο Ultimate Mediator διαθέτει ένα σύνολο κανόνων που δύναται να εφαρμοστούν στην πληροφορία που διακινείται από και προς τις πηγές. Οι κανόνες που παρέχονται, ορίζουν ένα επιπλέον επίπεδο προστασίας, διασφαλίζοντας την ακεραιότητα των δεδομένων που αποθηκεύονται στις πηγές. Για να επανέλθουμε στο παράδειγμα της τράπεζας και προκειμένου να γίνει κατανοητή η έννοια των κανόνων, αναφέρεται το ακόλουθο σενάριο.

Ο διαχειριστής συστημάτων της τράπεζας μπορεί να ορίσει κανόνες:

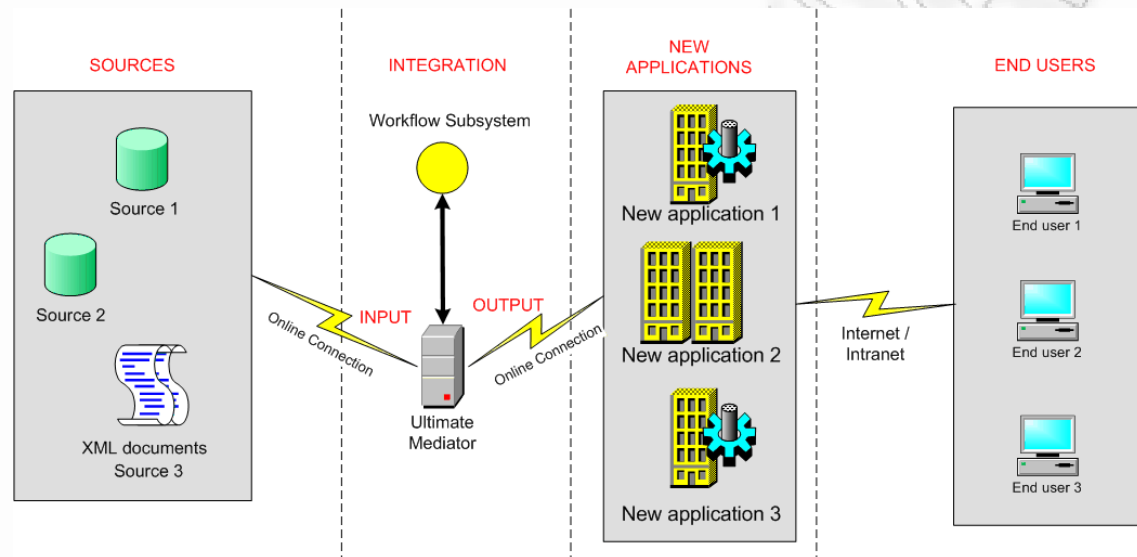
- για τους τύπους των δεδομένων (π.χ. αριθμοί –ακέραιοι, δεκαδικοί κτλ-, κείμενο μικρού μήκους, κείμενο μεγάλου μήκους, ημ/νια, κτλ)
- για τον τρόπο αποθήκευσης των δεδομένων (π.χ. κεφαλαία, πεζά κτλ)
- για τα σύνολα ή υποσύνολα των δεδομένων που επιτρέπεται ή απαγορεύεται να αντληθούν / επεξεργαστούν
- για τις απαραίτητες προϋποθέσεις που πρέπει να ισχύουν πριν ολοκληρωθούν συγκεκριμένες ενέργειες (π.χ. για την αποθήκευση ενός υποσυνόλου δεδομένων (1), απαιτείται πρώτα η αποθήκευση του υποσυνόλου δεδομένων (2))
- και άλλοι κανόνες που μπορεί να απορρέουν από τον τρόπο λειτουργίας της επιχειρήσεις ή και από τα ήθη / έθιμά της.

Οι έλεγχοι / κανόνες που επιβάλλονται από το ενδιάμεσο στάδιο του Ultimate Mediator καθοδηγούν τους «υλοποιητές» των νέων εφαρμογών και πρακτικά ελαχιστοποιούν την πιθανότητα:

- λάθους αποθήκευσης της πληροφορίας από αστοχία της νέας εφαρμογής
- μη εξουσιοδοτημένης πρόσβασης σε σύνολα / υποσύνολα δεδομένων που προέρχονται στις πηγές
- μη εξουσιοδοτημένης επεξεργασίας συνόλων / υποσυνόλων δεδομένων
- αποθήκευσης πληροφορίας σε λάθος μορφή / δομή
- μη ολοκληρωμένης αποθήκευσης, στις περιπτώσεις που υφίστανται προαπαιτούμενες ενέργειες
- και άλλα προβλήματα που μπορεί να προκύψουν από δυσκολίες ή λάθη σχεδιασμού και σύνδεσης.

Ολοκληρώνοντας την περιγραφή του μοντέλου online λειτουργίας σημειώνεται ότι οι τελικοί χρήστες αγνοούν την ύπαρξη ενδιάμεσου, έχουν την αίσθηση ότι αλληλεπιδρούν με ένα νέο σύστημα που παρέχεται από την τράπεζα που εμπιστεύονται και μπορούν να απολαύσουν τις νέες υπηρεσίες που παρέχονται. Στην πράξη βέβαια είναι προφανές ότι η βασική υποδομή (back-end) έχει παραμείνει η ίδια και χάρη στην ύπαρξη του ενδιάμεσου, στον τελικό χρήστη παρουσιάζεται ένα αφαιρετικό μοντέλο της πραγματικής αρχιτεκτονικής.

Προκειμένου η παραπάνω προσέγγιση να γίνει ακόμα πιο αληθοφανής, αξίζει να προστεθεί ακόμα μια λειτουργικότητα, αυτή του Workflow. Με τον όρο Workflow, στα πλαίσια της μελέτης, ορίζεται ένα υποσύστημα που ελέγχει τις ροές δεδομένων από τον Ultimate Mediator προς τις νέες εφαρμογές και αντίστροφα. Το Workflow Subsystem επιτρέπει στους διαχειριστές του Ultimate Mediator να ελέγχουν όλες τις ροές δεδομένων και να τις κατευθύνουν ανάλογα με τις επιθυμίες τους, βάσει κανόνων που εφαρμόζονται κατά περίπτωση. Στη συνέχεια αναφέρεται ένα απλό παράδειγμα (Εικόνα 2.1.17) που θα βοηθήσει στην καλύτερη κατανόηση του Workflow Subsystem.



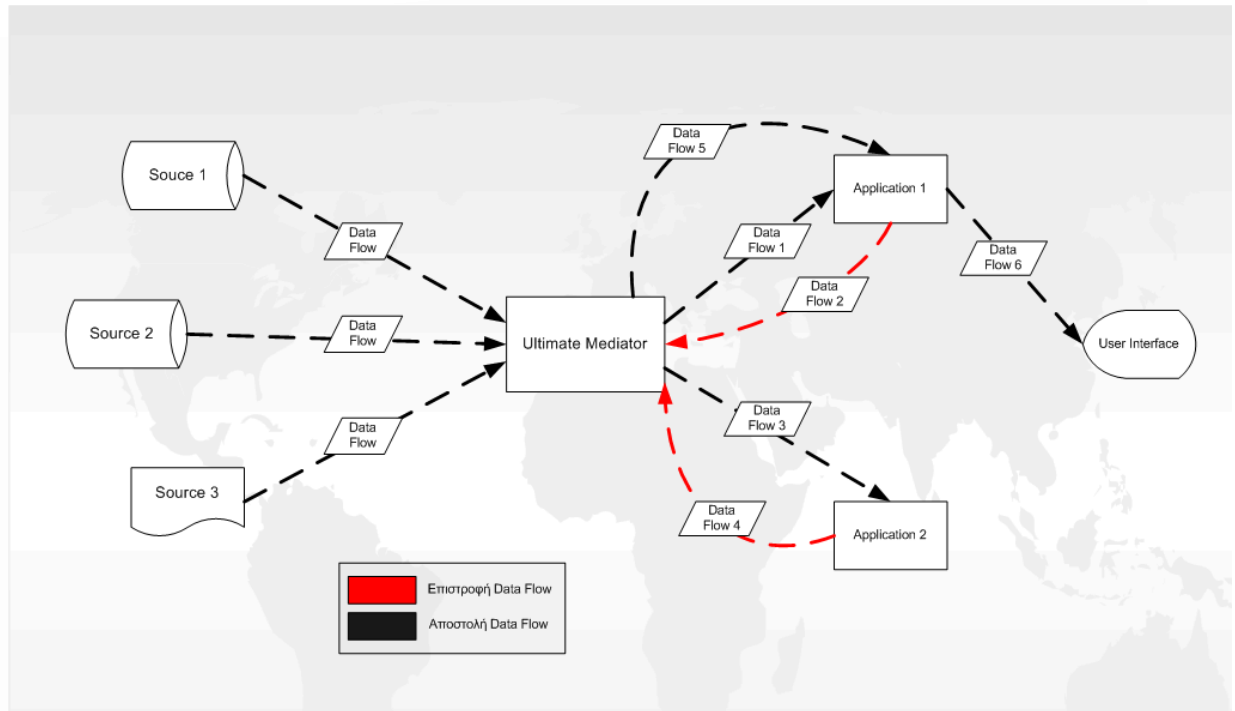
Εικόνα 2.1.17 Ultimate Mediator με Workflow Subsystem

Παράδειγμα

Ας θεωρηθεί μια ροή δεδομένων που κατευθύνεται σε κάποια εφαρμογή του επιπέδου «NEW APPLICATIONS». Ας θεωρηθεί επίσης ότι για να είναι πλήρης η εν λόγω ροή, απαιτείται η ύπαρξη συγκεκριμένων παραμέτρων.

Αν κατά τη διάρκεια του κύκλου ζωής της παραπάνω αρχιτεκτονικής τύχει να μην υπάρχουν όλες οι απαιτούμενες παράμετροι, τότε η ροή δεδομένων οδηγείται σε αποτυχία. Ο ρόλος του Workflow καλύπτει περιπτώσεις σαν και αυτές, «παγώνοντας» τη ροή δεδομένων μέχρι να καλυφθούν οι απαραίτητες απαιτήσεις.

Πέραν των προαναφερθέντων και ανάλογα με την εκάστοτε περίπτωση, θα μπορούσε το Workflow να ανακατευθύνει τη ροή σε εξωτερική εφαρμογή, η έξοδος της οποίας θα ανακατευθυνόταν σε κάποια άλλη εξωτερική εφαρμογή κ.λ.π. μέχρις ότου καλυφθούν οι απαιτήσεις. Τότε και μόνο τότε η ροή θα ανακατευθυνόταν στον αρχικό προορισμό της. Το παράδειγμα που περιγράφηκε αναπαρίσταται σχηματικά στο διάγραμμα που ακολουθεί.



Εικόνα 2.1.18 Ultimate Mediator με Workflow στον πραγματικό κόσμο

2.2 Επιλογές Υλοποίησης

Για τη δημιουργία του Ultimate Mediator χρησιμοποιήθηκαν διάφορες τεχνολογίες σε συνδυασμό, για την επίτευξη του τελικού στόχου του. Αυτές θα παρουσιαστούν και θα αναλυθούν, όπως και θα εξηγηθούν οι λόγοι για τους οποίους χρησιμοποιήθηκαν στις επόμενες ενότητες. Έπειτα θα παρουσιαστεί η βάση που χρησιμοποιεί το λογισμικό και θα αναλυθούν τα πεδία της.

Επιγραμματικά οι τεχνολογίες που χρησιμοποιήθηκαν είναι:

1. Visual Studio 2008,
2. LINQ,
3. XML και XML Schema
4. SOAP και HTTP.

Για τη χρήση της εφαρμογής αρκεί να δουλέψουν αρμονικά οι παραπάνω τεχνολογίες. Το τελικό αποτέλεσμα θα είναι ένα πρόγραμμα που θα μπορεί να δουλέψει σαν ενδιάμεσος διαφόρων βάσεων δεδομένων και xml αρχείων, παρέχοντας τη δυνατότητα του συνδυασμού και της επιλογή των δεδομένων.

2.2.1 Visual Studio 2008

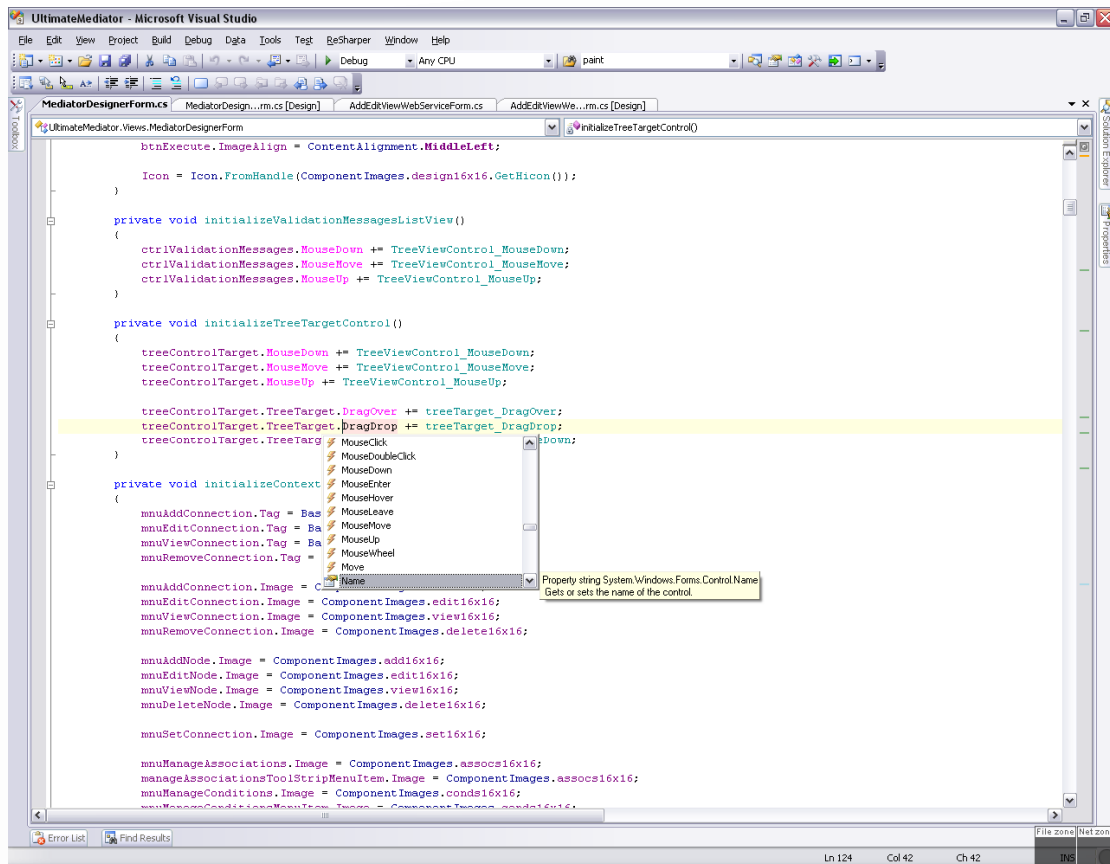
Το Visual Studio της Microsoft είναι το βασικό περιβάλλον ανάπτυξης προγραμμάτων στις γλώσσες προγραμματισμού που παρέχει η Microsoft [1]. Μπορεί να χρησιμοποιηθεί για να αναπτύξει ένας προγραμματιστής προγράμματα κονσόλας, διαδραστικά προγράμματα με γραφικές εφαρμογές ενδιάμεσων, ιστοσελίδες, εφαρμογές και υπηρεσίες του Παγκόσμιου Ιστού, όπως και άλλες εφαρμογές οι οποίες μπορούν να χρησιμοποιηθούν σε όλες τις πλατφόρμες της Microsoft όπως Microsoft Windows, Windows Mobile, όπου είναι δυνατή η εγκατάσταση του .Net Framework ή / και Microsoft Silverlight [1][2].

Το .Net Framework είναι ο πυρήνας επάνω στον οποίο στηρίζεται και λειτουργεί το Visual Studio αλλά και τα προγράμματα που δημιουργήθηκαν σε μία από τις γλώσσες που υποστηρίζει. Οι γλώσσες αυτές είναι οι Visual Basic, Visual J#, Visual C#, and Visual C++[1] [2].

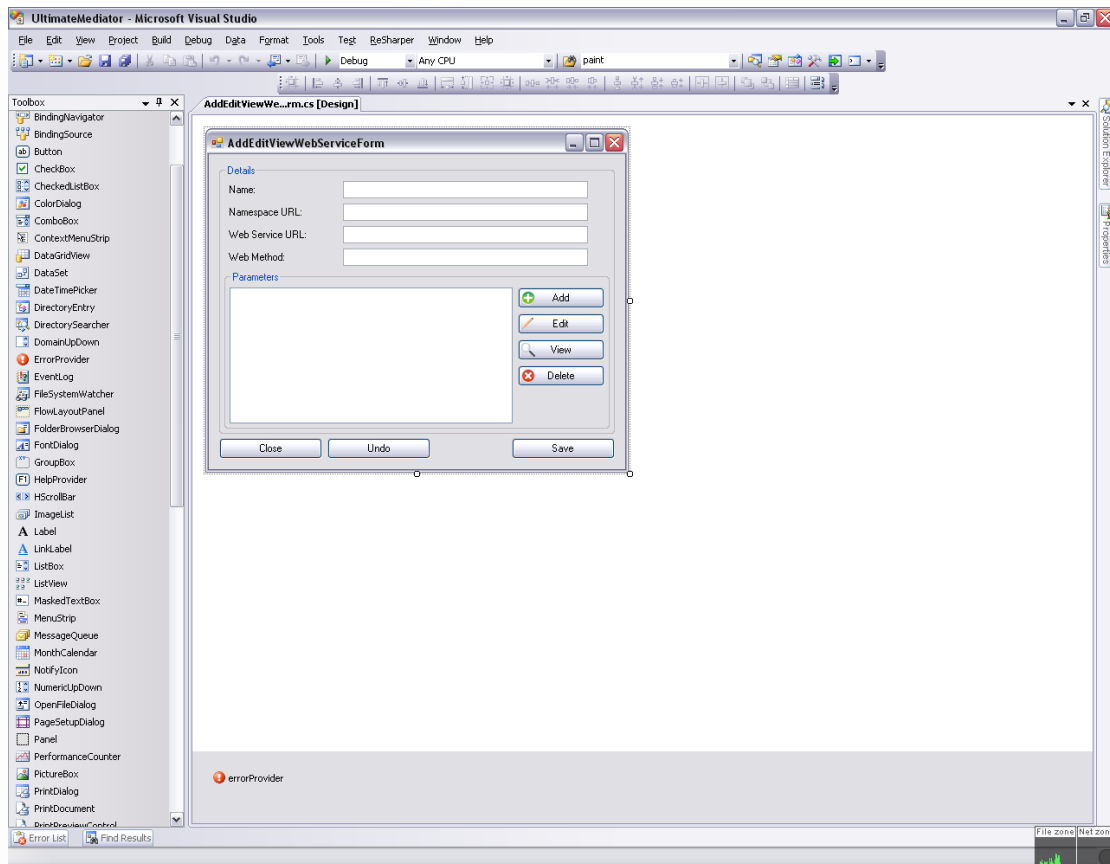
Το Silverlight είναι ένα πρόσθετο πρόγραμμα συνδεδεμένο στους φυλλομετρητές των χρηστών και το οποίο προσφέρει τη δυνατότητα ενεργοποίησης πολυμέσων και άλλων πιο εξελιγμένων συστατικών για τον εμπλουτισμό μίας ιστοσελίδας [1] [2].

Το Visual Studio περιλαμβάνει έναν συντάκτη κώδικα, που υποστηρίζει IntelliSense (τεχνολογία με την οποία ενώ ο προγραμματιστής γράφει κώδικα, το πρόγραμμα προβλέπει τι γράφει και του παρέχει συντομεύσεις κάνοντας auto - complete), καθώς επίσης και έτοιμα κομμάτια κώδικα [1][2]. Υπάρχει ενσωματωμένος compiler, ο οποίος βρίσκει και αναφέρει συντακτικά και λογικά λάθη του προγραμματιστή [1] [2]. Επίσης υπάρχουν και άλλα εργαλεία που βοηθάνε στην σχεδίαση παραθύρων και στην εύκολη και γρήγορη εισαγωγή διαδραστικών αντικειμένων επάνω στα παράθυρα σχεδίασης [1] [2].

Μέσα από την Εικόνα 2.2.1 και Εικόνα 2.2.2, μπορούμε να δούμε ένα μικρό μέρος από τις δυνατότητες που παρέχει το Visual Studio, και σε επίπεδο κώδικα αλλά και σε επίπεδο σχεδίασης παραθύρων.



Εικόνα 2.2.1 Χρωματικές διαφοροποιήσεις μεταξύ λειτουργία auto-complete



Εικόνα 2.2.2 Οθόνη σχεδίασης παραθύρου μέσα από το Visual Studio

Χρησιμοποιώντας λοιπόν την τεχνολογία της Microsoft μπορούμε αρκετά γρήγορα, να φτιάξουμε πολλές φόρμες αλληλεπίδρασης, για να πετύχουμε όσο το δυνατόν πιο ευνόητες και πιο διευκρινιστικές οθόνες. Κάτι τέτοιο θα καταστήσει πιο εύκολη και γρήγορη τη χρήση του προγράμματος από τους τελικούς χρήστες. Στην επίτευξη του παραπάνω χρησιμοποιούμε τη γλώσσα προγραμματισμού C# (C Sharp) version 3.0, η οποία βασίζεται επάνω στο .Net Framework για να εκτελέσει τον κώδικα του χρήστη.

2.2.2 LINQ (Language Integrated Query)

Η Linq είναι ένα έξτρα συστατικό που παρέχεται από το Visual Studio 2008 με χρήση του .Net Framework 3.5 [1][2]. Δίνει τη δυνατότητα να πραγματοποιούνται ερωτήματα επάνω σε όλα τα είδη πληροφοριών, όπως μεταβλητές, πίνακες, xml δεδομένα [1][2]. Το συντακτικό που χρησιμοποιείται είναι παραπλήσιο της SQL², και ορίζει μία ομάδα από τελεστές με τους οποίους εκτελούνται τα ερωτήματα επάνω στα αντικείμενα με τα δεδομένα [1][2].

Ένα περιορισμό που εισάγει η Linq, είναι το γεγονός ότι αναγκάζει τον προγραμματιστή να χρησιμοποιεί μόνο λίστες αντικειμένων με δεδομένα, ή αντικείμενο που περιέχει πεδία τύπου λίστας. Αυτό μπορεί να λυθεί μετατρέποντας \ μεταφέροντας όλη την πληροφορία σε ένα αποδεκτό τύπο δεδομένων [1][2].

Ένας άλλος περιορισμός είναι η αδυναμία της Linq να παρέχει τη δυνατότητα να εκτελούνται ερωτήματα επάνω σε βάσεις δεδομένων, στις οποίες δεν επιθυμείται ο ορισμός τους στατικά μέσα στον κώδικα. Σε τέτοιες περιπτώσεις δεν μπορεί να γίνουν ερωτήματα αν δεν οριστεί επακριβώς που θα είναι η βάση και ποιο θα είναι το σχήμα της. Για την αποφυγή τέτοιων προβλημάτων μπορεί να γίνει η λήψη των δεδομένων από εξωτερικές συναρτήσεις, οι οποίες θα γεμίσουν συγκεκριμένους πίνακες, από όπου μετά θα εκτελεστούν τα ερωτήματα μέσω Linq [2].

Ο λόγος για τον οποίο χρησιμοποιήθηκε στο Ultimate Mediator η Linq, είναι γιατί μπορούν να γίνουν ερωτήματα επάνω σε μεταβλητές και έτσι να φιλτράρεται κάθε φορά το επιθυμητό τελικό αποτέλεσμα. Μπορούν να οριστούν ερωτήματα, μέσα από τα οποία θα γίνεται η σύνδεση των πινάκων μεταξύ τους και να οριστούν κριτήρια επιλογής σε αυτά. Έτσι υπάρχει η ευκαιρία συνδυασμού και επιλογής των δεδομένων που θα χρησιμοποιηθούν. Έχοντας τους πίνακες μπορεί να γίνει η εξαγωγή τους, είτε σε βάση δεδομένων, είτε σε xml αρχεία παρέχοντας το επιθυμητό αποτέλεσμα.

Παραδείγματα κώδικα που δείχνουν τη χρήση της Linq μέσα από το πρόγραμμα Ultimate Mediator δίνονται παρακάτω. Η Εικόνα 2.2.3 δείχνει ένα παράδειγμα ανάγνωσης των δεδομένων που παρέχονται από xml αρχείο, και συνδυασμό αυτών, με δεδομένα από ένα dataset³. Το τελικό αποτέλεσμα αποθηκεύεται στην μεταβλητή *query* την οποία μπορούμε να την επεξεργαστούμε μετά, ή να εξάγουμε τα δεδομένα της, ή να την χρησιμοποιήσουμε σε άλλο Linq ερώτημα.

² Η SQL (Structured Query Language) είναι η γλώσσα με την οποία μπορεί να εκτελέσεις ερωτήματα σε βάσεις δεδομένων [1].

³ Το dataset είναι μία συλλογή από δεδομένα τα οποία είναι οργανωμένα στη μορφή των πινάκων μίας βάσης δεδομένων. Περιέχει πίνακες που μπορεί να έχουν πολλές γραμμές για τις ορισμένες στήλες [1][2].

```

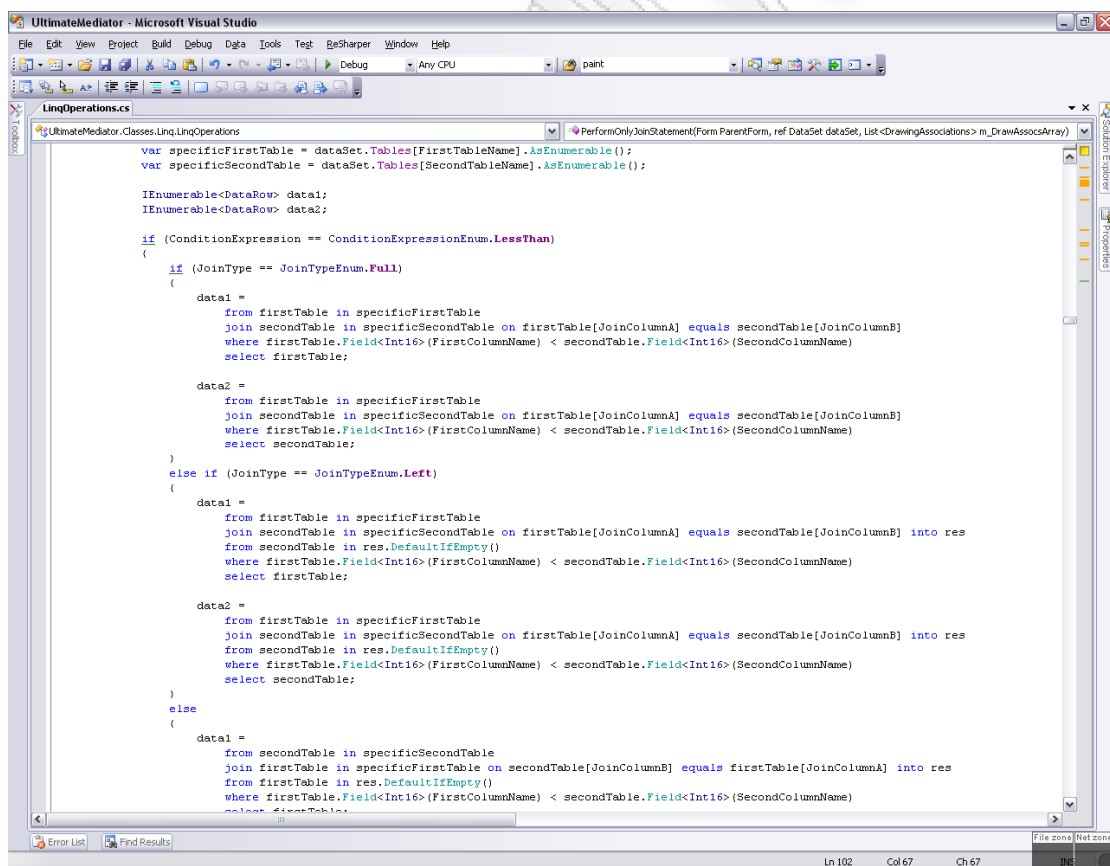
if (JoinType == JoinTypeEnum.Full)
{
    var emptyElement = XElement.EmptySequence;

    query =
        from t1 in table1.Descendants("results").Descendants(Static.DefaultDummyTableName)
        join t2 in
            table2.Descendants(ds.DataSetName).Take(1).Descendants(
                ds.Tables[i].TableName)
            on (string) t1.Element(col1) equals (string) t2.Element(col2)
        select
            new XElement(Static.DefaultDummyTableName,
                t1 == null ? emptyElement.Nodes() : t1.Nodes(),
                t2 == null ? emptyElement.Nodes() : t2.Nodes());
}

```

Εικόνα 2.2.3 Παράδειγμα χρήσης της Linq για την πραγματοποίηση ερωτημάτων σε xml δεδομένα

Συνεχίζοντας η Εικόνα 2.2.4 δείχνει ένα ακόμα παράδειγμα όπου πραγματοποιούνται πολλά ερωτήματα από το πρόγραμμα για να γίνει η ένωση όλων των πινάκων, με βάση τον τύπο της ένωσης (Full Join, Left Join, Right Join).



Εικόνα 2.2.4 Παράδειγμα πολλών Linq ερωτημάτων για τον συνδυασμό όλων των διαφορετικών πινάκων

2.2.3 XML και XML Schema

Η xml είναι μία γλώσσα περιγραφής δεδομένων [3][4][19]. Μέσω της περιγραφής αυτής ουσιαστικά δημιουργείται μία νέα γλώσσα με βάση τις προδιαγραφές που ορίζονται [3][4][19]. Οι προδιαγραφές αυτές ορίζονται με την χρήση μίας άλλης γλώσσας προγραμματισμού που είναι η XML Schema [3][4][19]. Μέσω του σχήματος μπορούν να οριστούν κανόνες για τον τρόπο καταχώρησης των δεδομένων [3][4][19]. Αυτοί οι κανόνες χρησιμοποιούνται για να επαληθεύσουν ότι το xml είναι σωστό με βάση τον ορισμό του σχήματος [3][4][19].

Τα αρχικά του xml είναι Extensible Markup Language [3][4][19]. Το xml έχει πολλές εφαρμογές και μάλιστα εφαρμογές που έχουν σαν κύριο σκοπό την αποστολή δεδομένων με κάποια συγκεκριμένη μορφή [3][4][19]. Ένα παράδειγμα ενός xml αρχείου είναι το ακόλουθο (Εικόνα 2.2.5):

```
<?xml version="1.0" encoding="UTF-8"?>
<Μάθημα>
  <Τίτλος_Μαθήματος>Μάθημα1</Τίτλος_Μαθήματος>
  <Κωδικός_Μαθήματος>52146</Κωδικός_Μαθήματος>
  <Πανεπιστήμιο>Πανεπιστήμιο Πειραιώς</Πανεπιστήμιο>
  <Τμήμα>Διδακτική της Τεχνολογίας και Ψηφιακών Συστημάτων</Τμήμα>
  <Καθηγητής>
    <Όνοματεπώνυμο>Καθηγητής1</Όνοματεπώνυμο>
    <email>teacher@sth.gr</email>
  </Καθηγητής>
  <Έτος>4</Έτος>
  <Εξάμηνο>8</Εξάμηνο>
  <Βιβλίο>
    <Τίτλος>Οδηγός της XML</Τίτλος>
    <Συγγραφέας>Συγγραφέας1</Συγγραφέας>
    <Εκδότης>Εκδότης1</Εκδότης>
    <Έτος_Έκδοσης>2006</Έτος_Έκδοσης>
  </Βιβλίο>
  <Βιβλίο>
    <Τίτλος>Ασκήσεις XML</Τίτλος>
    <Συγγραφέας>Συγγραφέας2</Συγγραφέας>
    <Εκδότης>Εκδότης2</Εκδότης>
    <Έτος_Έκδοσης>2006</Έτος_Έκδοσης>
  </Βιβλίο>
</Μάθημα>
```

Εικόνα 2.2.5 Παράδειγμα ενός XML

Πολλά συστήματα εξάγουν τις πληροφορίες που παρέχουν σε xml αρχεία. Τέτοια αρχεία μπορούν να χρησιμοποιηθούν από την εφαρμογή για να σταλούν σε κάποιο web service. Έτσι, συστήματα που βγάζουν έξοδο xml ή που λαμβάνουν τέτοια είσοδο, μπορούν να χρησιμοποιήσουν το πρόγραμμα αυτό για να πάρουν τα δεδομένα τροποποιημένα, όπως είναι οι προδιαγραφές τους [19].

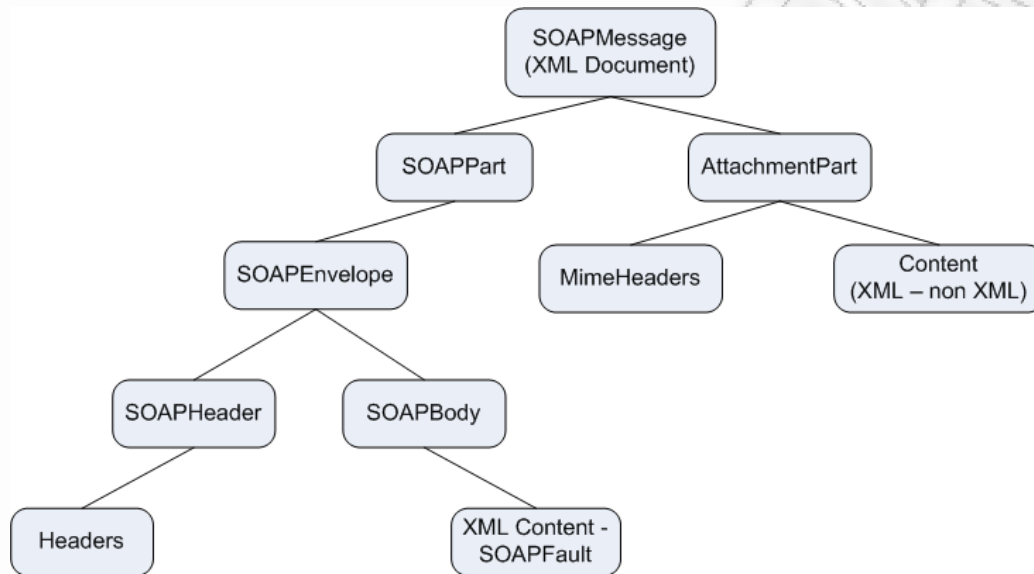
Ο Ultimate Mediator έχει την δυνατότητα να λάβει σαν είσοδο και xml schema. Σε αυτήν την περίπτωση θα διαβάσει τη δομή του σχήματος και θα την αποτυπώσει στην αντίστοιχη σχεδιαστική φόρμα (όπως θα δούμε και σε επόμενα κεφάλαια). Όταν χρειαστεί, θα χρησιμοποιήσει για είσοδο τα xml δεδομένα που του δίνει ο χρήστης. Ο λόγος για τον οποίο χρησιμοποιούνται και οι δύο τεχνολογίες όπως περιγράφηκαν, είναι γιατί, όταν το xml αρχείο είναι μεγάλο σε μέγεθος (επειδή

περιέχει έναν μεγάλο αριθμό από δεδομένα), τότε το σύστημα μέχρι να διαβάσει όλο το xml για να αποτυπώσει τα πεδία που διαθέτει, θα καθυστερήσει περισσότερο.

Χρήζει προσοχή το γεγονός, ότι, θα πρέπει να είναι σωστά ορισμένα τα δεδομένα στο xml έτσι ώστε να συμφωνούν με την περιγραφή του xml schema, διαφορετικά θα προκύψει σφάλμα κατά την τροποποίηση και μεταφορά τους.

2.2.4 SOAP και HTTP

Το πρωτόκολλο SOAP, χρησιμοποιεί δικές του επικεφαλίδες και σύνταξη, για τον ορισμό ονομάτων – namespaces, και το διαχωρισμό των πεδίων του XML που έχουν απλό κείμενο, από εκείνων που έχουν επισυνάψει κάποιο αρχείο, όπως για παράδειγμα μία εικόνα [1][5]. Πιο συγκεκριμένα η αρχιτεκτονική του πρωτοκόλλου SOAP φαίνεται στην Εικόνα 2.2.6.



Εικόνα 2.2.6 Δομή ενός SOAP μηνύματος με ένα επισυναπτόμενο αρχείο

Μπορούμε να ξεχωρίσουμε σε ένα SOAP μήνυμα (SOAPMessage) τα εξής πεδία, όπως αυτά παρουσιάστηκαν στην Εικόνα 2.2.6 [1][6]:

1. **SOAPMessage:** Είναι ολόκληρο το XML αρχείο που στέλνεται.
 - a. **SOAPPart:** Το κομμάτι της SOAP που δεν περιλαμβάνει επισυναπτόμενα αρχεία.
 - i. **SOAPEnvelope:** Το πεδίο που περικλείει σαν φάκελος την επικεφαλίδα και το σώμα των πληροφοριών του XML που στέλνεται.
 1. **SOAPHeader:** Προαιρετικό πεδίο που περιέχει επικεφαλίδες.
 - a. **Headers:** Επικεφαλίδες εφόσον υπάρχουν.
 2. **SOAPBody:** Περιέχει το κύριο σώμα – τις πληροφορίες και τα δεδομένα XML του SOAP μηνύματος.
 - a. **XML Content / SOAPFault:** Το κείμενο XML ή το SOAP μήνυμα λάθους που θα σταλεί στο δίκτυο.
 - b. **AttachmentPart:** Περιέχει επισυναπτόμενο αρχείο το οποίο μπορεί να είναι για παράδειγμα ένα XML ή μία εικόνα. AttachmentPart πεδία, μπορεί να υπάρχουν παραπάνω από ένα, σε ένα SOAPMessage.
 - i. **MIME Headers:** Επικεφαλίδες του Attachment

- ii. **Content:** Το επισυναπτόμενο αρχείο το οποίο μπορεί να είναι για παράδειγμα ακόμα και ένα κείμενο.

Παράδειγμα ενός xml αρχείου παρουσιάζεται στην Εικόνα 2.2.7, όπου το xml περιέχει την λειτουργία της πρόσθεσης, και ορίζει τους δύο αριθμούς που θα προστεθούν.

```
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <Request>
      <Math>
        <Add>
          <x>10</x>
          <y>12</y>
        </Add>
      </Math>
    </Request>
  </soap:Body>
</soap:Envelope>
```

Εικόνα 2.2.7 Παράδειγμα ενός xml με τα πεδία ορισμένο με τις ετικέτες του soap

Το πρωτόκολλο soap χρησιμοποιείται για να υπάρχει μία κοινή δομή αποστολής δεδομένων. Επειδή πολλά Web Services απαιτούν την ύπαρξη του πρωτοκόλλου soap για τη λήψη δεδομένων από άλλα προγράμματα, γι αυτόν το λόγο έχει ενσωματωθεί στον Ultimate Mediator. Κάθε φορά που χρειάζεται να γίνει η αποστολή των δεδομένων από την εφαρμογή, τα δεδομένα xml ενσωματώνονται μέσα στο πεδίο SOAPBody του soap μηνύματος και αποστέλλονται στο Internet με τη βοήθεια του HTTP πρωτοκόλλου.

Το HTTP πρωτόκολλο (Hypertext Transfer Protocol), είναι ένα επικοινωνιακό πρωτόκολλο, για τη μεταφορά δεδομένων μέσω του Internet. Χρησιμοποιείται στο πρόγραμμα για να πραγματοποιηθεί η αποστολή των δεδομένων προς κάποιο Web Service που βρίσκεται στο Internet. Δεν υπάρχουν συγκεκριμένες διαδικασίες που γίνονται κατά τη μεταφορά. Ένα παράδειγμα που δείχνει τις επικεφαλίδες που στέλνονται από το HTTP πρωτόκολλο φαίνεται στην Εικόνα 2.2.8.

```

josh@blackbox: ~
File Edit View Terminal Tabs Help
josh@blackbox:~$ telnet en.wikipedia.org 80
Trying 208.80.152.2...
Connected to rr.pmtpa.wikimedia.org.
Escape character is '^]'.
GET /wiki/Main_Page http/1.1
Host: en.wikipedia.org

HTTP/1.0 200 OK
Date: Thu, 03 Jul 2008 11:12:06 GMT
Server: Apache
X-Powered-By: PHP/5.2.5
Cache-Control: private, s-maxage=0, max-age=0, must-revalidate
Content-Language: en
Vary: Accept-Encoding, Cookie
X-Vary-Options: Accept-Encoding,list-contains=gzip, Cookie;string-contains=enwikiToken;string-contains=enwikiLoggedOut;string-contains=enwiki_session;string-contains=centralauth_Token;string-contains=centralauth_Session;string-contains=centralauth_LoggedOut
Last-Modified: Thu, 03 Jul 2008 10:44:34 GMT
Content-Length: 54218
Content-Type: text/html; charset=utf-8
X-Cache: HIT from sq39.wikimedia.org
X-Cache-Lookup: HIT from sq39.wikimedia.org:3128
Age: 3
X-Cache: HIT from sq38.wikimedia.org
X-Cache-Lookup: HIT from sq38.wikimedia.org:80
Via: 1.0 sq39.wikimedia.org:3128 (squid/2.6.STABLE18), 1.0 sq38.wikimedia.org:80 (squid/2.6.STABLE18)
Connection: close

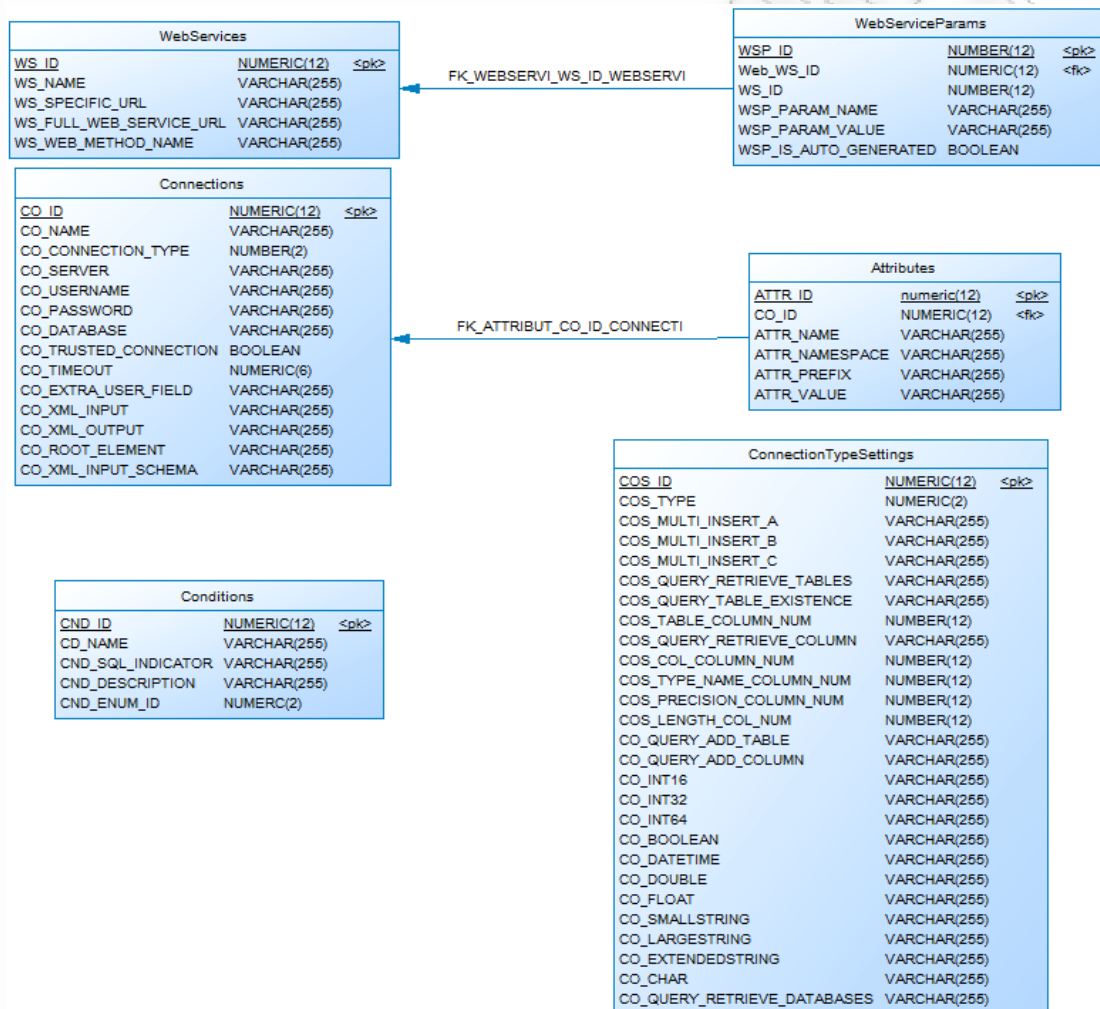
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en" dir="ltr">
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
    <meta name="keywords" content="Main Page,1778,1844,1863,1938,1980 Summer Olympics,2008,2008 Guizhou riot,2008 Jerusal
    ...
    ... This content has been removed to save space
    ...
    <li id="privacy"><a href="http://wikimediafoundation.org/wiki/Privacy_policy" title="wikimedia:Privacy policy">Privac
    y policy</a></li>
    <li id="about"><a href="/wiki/Wikipedia:About" title="Wikipedia:About">About Wikipedia</a></li>
    <li id="disclaimer"><a href="/wiki/Wikipedia:General_disclaimer" title="Wikipedia:General disclaimer">Disclaimers</a>
  </li>
  </ul>
</div>
<script type="text/javascript">if (window.runOnloadHook) runOnloadHook();</script>
<!-- Served by srv93 in 0.050 secs. --></body></html>
Connection closed by foreign host.
josh@blackbox:~$

```

Εικόνα 2.2.8 Οι επικεφαλίδες αποστολής του HTTP πρωτοκόλλου

2.2.5 Μοντελοποίηση Βάσης Δεδομένων

Το λογισμικό, όπως έχει προαναφερθεί, χρησιμοποιεί μία βάση δεδομένων σε Access, για να κρατάει ρυθμίσεις που κάνει ο χρήστης από το πρόγραμμα, αλλά και άλλες επιλογές που ορίζονται μόνο στη βάση. Τέτοιες επιλογές είναι ο ορισμός των ερωτημάτων που θα πρέπει να εκτελούνται για να πάρει το πρόγραμμα πληροφορίες για τους διαθέσιμους πίνακες της βάσης, στατικές ρυθμίσεις για την τελική αναπαράσταση των τύπων δεδομένων, κ.α.. Το σχήμα της βάσης παρουσιάζεται παρακάτω (Εικόνα 2.2.9).



Εικόνα 2.2.9 Σχήμα Βάσης Δεδομένων του Ultimate Mediator

Όπως μπορούμε να δούμε και από την παραπάνω Εικόνα, υπάρχουν οι πίνακες **WebServices**, **WebServiceParams**, **Connections**, **Attributes**, **Conditions** και **ConnectionTypeSettings**. Οι δύο πρώτοι κρατάνε πληροφορίες σχετικά με τα web services που έχει ορίσει ο χρήστης στο σύστημα. Ο πίνακας **WebServiceParams**, κρατάει τον ορισμό όλων των παραμέτρων που πρέπει να αποστέλλονται στο συγκεκριμένο web service. Η εισαγωγή και επεξεργασία δεδομένων σε αυτούς τους δύο πίνακες δε θα πρέπει να γίνει από την βάση, επειδή υπάρχουν ήδη οθόνες

διαχείρισης από το λογισμικό που μπορεί εύκολα ο χρήστης να χρησιμοποιήσει (*Manage Web Services*). Ο πίνακας *WebServices* παρουσιάζεται παρακάτω:

WebServices : Table		
	Field Name	Data Type
	WS_ID	AutoNumber
	WS_NAME	Text
	WS_SPECIFIC_URL	Text
	WS_FULL_WEB_SERVICE_URL	Text
	WS_WEB_METHOD_NAME	Text

Εικόνα 2.2.10 Τα πεδία του πίνακα *WebServices*

1. Το πεδίο *WS_ID* είναι το μοναδικό πρωτεύον κλειδί του πίνακα. Οι τιμές του έχουν οριστεί να μπαίνουν αυτόματα όταν εισέρχεται μία νέα εγγραφή.
2. Το πεδίο *WS_NAME* περιέχει το όνομα του web service. Χρησιμοποιείται καθαρά για να διευκολύνεται ο χρήστης στην αναγνώρισή του μεταξύ των υπόλοιπων.
3. Το πεδίο *WS_SPECIFIC_URL* περιέχει ολόκληρη τη διεύθυνση με την οποία μπορεί το πρόγραμμα να επικοινωνήσει με το συγκεκριμένο web service. Ένα παράδειγμα είναι: «<http://localhost:3906/Service1.asmx>»
4. Το πεδίο *WS_FULL_WEB_SERVICE_URL*, αποθηκεύει τη διεύθυνση του server που περιέχει το αντίστοιχο web service. Ένα παράδειγμα είναι: «<http://localhost:3906/>».
5. Το πεδίο *WS_WEB_METHOD_NAME*, περιέχει το όνομα της μεθόδου που θα πρέπει να καλέσει το πρόγραμμα στο web service για να του στείλει τα δεδομένα.

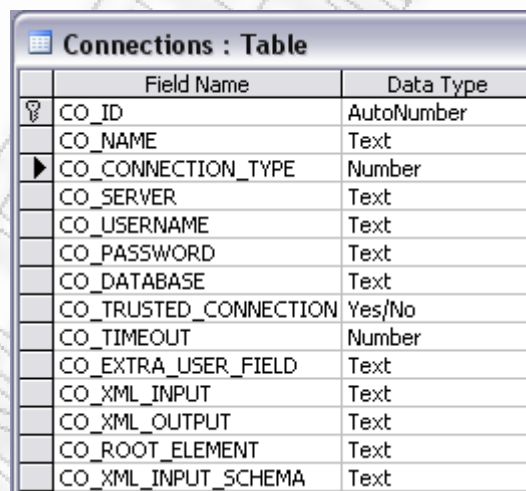
Ο πίνακας *WebServiceParams* περιέχει πληροφορίες σχετικά με τις παραμέτρους που θα πρέπει να αποσταλούν στο web service. Γι' αυτόν το λόγο συνδέεται μέσω του ξένου κλειδιού *WS_ID*, για να δείξει για ποιο web service αναφέρονται οι συγκεκριμένες παράμετροι. Παρακάτω παρουσιάζονται τα πεδία του πίνακα και η περιγραφή τους.

WebServiceParams : Table		
	Field Name	Data Type
	WSP_ID	AutoNumber
	WS_ID	Number
	WSP_PARAM_NAME	Text
	WSP_PARAM_VALUE	Text
	WSP_IS_AUTO_GENERATED	Yes/No

Εικόνα 2.2.11 Τα πεδία του πίνακα *WebServiceParams*

1. Το WSP_ID είναι το πρωτεύον κλειδί του πίνακα το οποίο επιπλέον έχει οριστεί να παίρνει τιμές αυτόματα, χωρίς να τις βάζει το πρόγραμμα.
2. Το WS_ID είναι το ξένο κλειδί του πίνακα WebServices, και χρησιμοποιείται για να συνδέσει την παράμετρο με το web service.
3. Το WSP_PARAM_NAME κρατάει το όνομα της παραμέτρου στο οποίο θα βάλει την τιμή το πρόγραμμα, όταν θα καλέσει τη μέθοδο του web service.
4. Το WSP_PARAM_VALUE έχει την τιμή που θα στείλει το πρόγραμμα στο web service. Αν η τιμή είναι δυναμική και όχι στατική, τότε θα πρέπει αυτό το πεδίο να είναι κενό και το επόμενο πεδίο WSP_IS_AUTO_GENERATED να είναι αληθές (Yes, ή 1 ή True).
5. Το WSP_IS_AUTO_GENERATED δείχνει ότι το πεδίο παίρνει δυναμική τιμή και όχι κάποια στατική που ορίζεται στο WSP_PARAM_VALUE.

Παρόμοια με τους προηγούμενες πίνακες έτσι και οι Connections και Attributes, δεν θα πρέπει να διαχειριστούν από το χρήστη μέσα από τη βάση, αλλά από τις ειδικές οθόνες της εφαρμογής (*Manage Connections*). Αυτοί κρατάνε πληροφορίες για τις συνδέσεις όπως τη διεύθυνση του server, το όνομα χρήστη, τον κωδικό πρόσβασης κ.λ.π.. Σε περίπτωση που η σύνδεση είναι xml τότε μπορεί να ορίσει τα Attributes του αρχικού πεδίου κόμβου ή αλλιώς όπως θα δούμε του Root Element. Ο πίνακας Connections παρουσιάζεται παρακάτω.



	Field Name	Data Type
PK	CO_ID	AutoNumber
	CO_NAME	Text
	CO_CONNECTION_TYPE	Number
	CO_SERVER	Text
	CO_USERNAME	Text
	CO_PASSWORD	Text
	CO_DATABASE	Text
	CO_TRUSTED_CONNECTION	Yes/No
	CO_TIMEOUT	Number
	CO_EXTRA_USER_FIELD	Text
	CO_XML_INPUT	Text
	CO_XML_OUTPUT	Text
	CO_ROOT_ELEMENT	Text
	CO_XML_INPUT_SCHEMA	Text

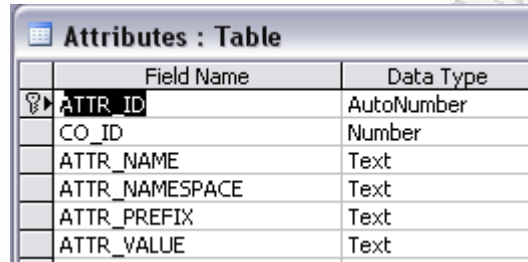
Εικόνα 2.2.12 Τα πεδία του πίνακα Connections

1. Το CO_ID είναι το πρωτεύον κλειδί του πίνακα Connections και έχει οριστεί να παίρνει τις τιμές αυτόματα, κατά την εισαγωγή μίας νέας εγγραφής.
2. Το CO_NAME περιέχει το όνομα της σύνδεσης που χρησιμοποιείται μόνο σαν χαρακτηριστικό διάκρισης μεταξύ των άλλων συνδέσεων.

3. Το CO_CONNECTION_TYPE είναι το πεδίο που ορίζει τον τύπο της σύνδεσης. Ο τύπος αυτός μπορεί να έχει μία από τις παρακάτω τιμές:
 - i. MSSQLServer = 1,
 - ii. Oracle = 2,
 - iii. MSAccess = 3,
 - iv. MySQL = 4
4. Το CO_SERVER περιέχει το όνομα (διεύθυνση) του server που έχει τη βάση δεδομένων.
5. Το CO_USERNAME περιέχει το όνομα χρήστη για τη σύνδεση στη βάση δεδομένων.
6. Το CO_PASSWORD περιέχει τον κωδικό / συνθηματικό που χρησιμοποιείται για τη σύνδεση στη βάση δεδομένων.
7. Το CO_DATABASE περιέχει το όνομα της βάσης δεδομένων που θα χρησιμοποιηθεί από τις διαθέσιμες του server.
8. Το CO_TRUSTED_CONNECTION περιέχει μία ένδειξη αληθείας (True/False ή Yes/No ή 1/0), για να χρησιμοποιηθεί κατά τη σύνδεση με τη βάση δεδομένων.
9. Το CO_TIMEOUT περιέχει τον αριθμό σε δευτερόλεπτα για τον οποίο θα γίνει προσπάθεια σύνδεσης με τη βάση δεδομένων πριν λήξει (timeout) η προσπάθεια.
10. Το CO_EXTRA_USER_FIELD περιέχει μία επιπλέον πληροφορία που χρειάζεται όταν γίνει η κλήση κάποιου πίνακα. Για παράδειγμα στον SQL Server μπορούμε να ορίσουμε την κλήση ενός πίνακα CUSTOMERS μίας βάσης MARKET ως εξής: «MARKET.dbo.CUSTOMERS». Όπου το dbo είναι αυτό το χαρακτηριστικό και συμβολίζει στην προκειμένη περίπτωση το χρήστη συστήματος μέσω του οποίου θα βρει τον συγκεκριμένο πίνακα.
11. Το CO_XML_INPUT περιέχει τη διαδρομή του δίσκου με το αρχείο εισόδου για την ανάγνωση ενός xml αρχείου. Χρησιμοποιείται μόνο σε xml τύπου σύνδεσης.
12. Το CO_XML_OUTPUT περιέχει τη διαδρομή του δίσκου με το αρχείο εξόδου για την εγγραφή ενός xml αρχείου. Χρησιμοποιείται μόνο σε xml τύπου σύνδεσης.
13. Το CO_ROOT_ELEMENT περιέχει το όνομα του κεντρικού αρχικού πεδίου που θα περιέχει όλα τα δεδομένα xml. Χρησιμοποιείται μόνο σε xml τύπου σύνδεσης.
14. Το CO_XML_INPUT_SCHEMA, περιέχει τη διαδρομή για το αρχείο εισόδου που έχει πληροφορίες για το σχήμα του xml. Αυτό βοηθάει στην πιο γρήγορη

ανάγνωση των πινάκων που βρίσκονται στο αρχείο. Χρησιμοποιείται μόνο σε xml τύπου σύνδεσης.

Ο πίνακας Attributes περιέχει τις ιδιότητες που μπορούν να οριστούν επάνω στο Root Element. Οι ιδιότητες αυτές μπορεί να έχουν κάποια τιμή και κάποιο πρόθεμα. Το πρόθεμα όμως ορίζεται μοναδικά μέσα από κάποιο namespace όπως θα δούμε και από τα πεδία παρακάτω. Ο πίνακας αυτός γεμίζει μόνο στην περίπτωση που η σύνδεση είναι τύπου xml και όταν χρειαζόμαστε να ορίσουμε ιδιότητες για αυτά.

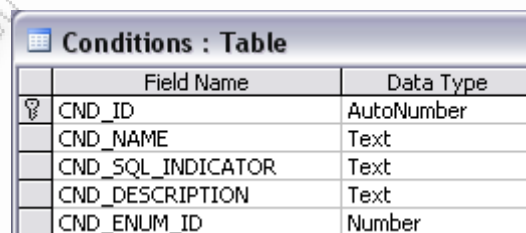


	Field Name	Data Type
?	ATTR_ID	AutoNumber
	CO_ID	Number
	ATTR_NAME	Text
	ATTR_NAMESPACE	Text
	ATTR_PREFIX	Text
	ATTR_VALUE	Text

Εικόνα 2.2.13 Τα πεδία του πίνακα Attributes

1. Το πεδίο ATTR_ID κρατάει το πρωτεύον κλειδί του πίνακα. Και αυτό όπως και τα άλλα αντίστοιχα έχει οριστεί να παίρνει αυτόματη αύξουσα αρίθμηση.
2. Το CO_ID είναι το ξένο κλειδί του πίνακα Connections και χρησιμοποιείται για να αναγνωρίζεται η σύνδεση μεταξύ τους.
3. Το ATTR_NAME είναι το όνομα της ιδιότητας, που θα βγει στο xml αρχείο.
4. Το ATTR_NAMESPACE είναι το url συνήθως που αναγνωρίζει / περιγράφει κάποιο πρόθεμα της ιδιότητας.
5. Το ATTR_PREFIX είναι το πρόθεμα της ιδιότητας.
6. Το ATTR_VALUE είναι η τιμή της ιδιότητας.

Οι πίνακες που μπορεί να χρειαστούν να επεξεργαστούν από το χρήστη είναι, ουσιαστικά, οι δύο τελευταίοι (Conditions, ConnectionTypeSettings). Ο πίνακας Conditions παρουσιάζεται παρακάτω στην Εικόνα 2.2.14.



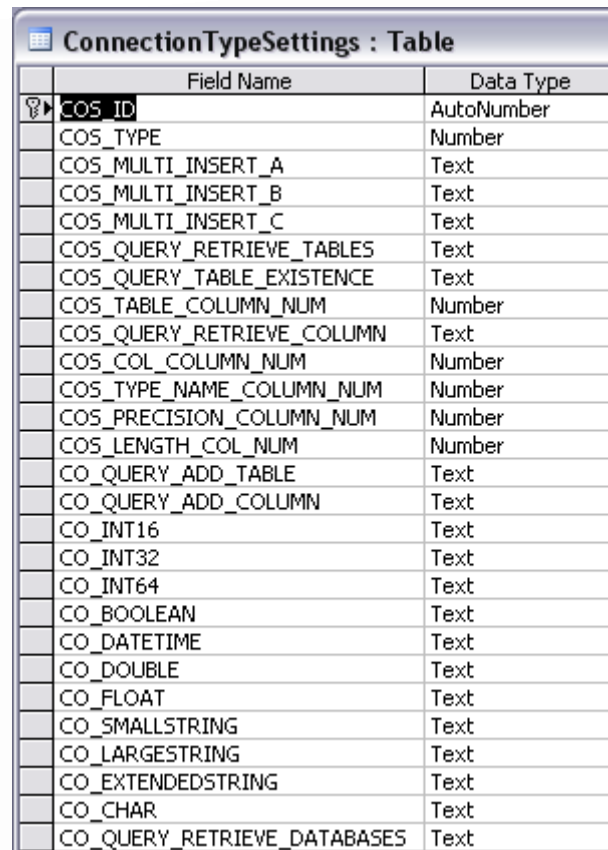
	Field Name	Data Type
?	CND_ID	AutoNumber
	CND_NAME	Text
	CND_SQL_INDICATOR	Text
	CND_DESCRIPTION	Text
	CND_ENUM_ID	Number

Εικόνα 2.2.14 Τα πεδία του πίνακα Conditions

1. Το CND_ID αντιπροσωπεύει το μοναδικό κλειδί της κάθε συνθήκης. Επειδή έχει οριστεί να παίρνει αυτόματα τιμές δεν θα πρέπει να ορίζεται από το χρήστη.
2. Το CND_NAME περιέχει το όνομα της συνθήκης. Το όνομα αυτό θα εμφανιστεί στην οθόνη δημιουργίας νέας συνθήκης (Εικόνα 3.7.18), και θα πρέπει να είναι αντιπροσωπευτικό έτσι ώστε να μπορεί να γίνει κατανοητή η λειτουργία του. Ένα παράδειγμα είναι το «less than or equals».
3. Το CND_SQL_INDICATOR χρησιμοποιείται για να αντιπροσωπεύσει τον κώδικα που θα πρέπει να εκτελεστεί για να πραγματοποιηθεί αυτή η συνθήκη. Μία περίπτωση του «less than or equals» είναι η «{0} <= {1}». Μπορούμε να δούμε ότι υπάρχει το {0} και το {1}. Αυτά αντιπροσωπεύουν τα πεδία που θα πρέπει να μπουν αριστερά και δεξιά της συνθήκης. Το {0} υποδηλώνει το πεδίο που θα πρέπει να μπει αριστερά της συνθήκης ενώ το {1} το πεδίο που θα μπει δεξιά της.
4. Το CND_DESCRIPTION έχει μία περιγραφή για τη συνθήκη που να είναι πιο κατανοητή από το όνομα αυτής.
5. Το CND_ENUM_ID περιέχει το μοναδικό κλειδί, που για τα υπάρχοντα, θα πρέπει να είναι ένα από τα επόμενα, ενώ για νέα θα πρέπει να είναι κλειδιά διαφορετικά από τα παρακάτω:
 - i. Null = 0,
 - ii. IsNull = 1,
 - iii. Equal = 2,
 - iv. PartEqual = 3,
 - v. StartPartEqual = 4,
 - vi. EndPartEqual = 5,
 - vii. LessThan = 6,
 - viii. LessThanOrEquals = 7,
 - ix. MoreThan = 8,
 - x. MoreThanOrEquals = 9,
 - xi. Mandatory = 10,
 - xii. NotEqual = 11,

Ο πίνακας ConnectionTypeSettings περιέχει πληροφορίες που σχετίζονται με το κάθε είδος σύνδεσης τύπου βάσης δεδομένων. Για τα τέσσερα υποστηριζόμενα είδη σύνδεσης υπάρχουν ήδη εγγραφές. Ο χρήστης δεν χρειάζεται να εισάγει νέες παρά μόνο να επεξεργαστεί τις ήδη υπάρχουσες αν και εφόσον το επιθυμεί.

Ο λόγος για τον οποίο μπορεί να χρειαστεί να κάνει επεξεργασία των εγγραφών είναι γιατί τα ερωτήματα μπορεί να αλλάξουν ανάλογα με την έκδοση της βάσης που χρησιμοποιεί ο εκάστοτε χρήστης. Αυτό θα έχει σαν αποτέλεσμα να πρέπει να ενημερωθεί ο πίνακας για την εκάστοτε έκδοση της βάσης. Τα πεδία του εμφανίζονται στην Εικόνα 2.2.15 και αναλύονται πιο κάτω.



Field Name	Data Type
COS_ID	AutoNumber
COS_TYPE	Number
COS_MULTI_INSERT_A	Text
COS_MULTI_INSERT_B	Text
COS_MULTI_INSERT_C	Text
COS_QUERY_RETRIEVE_TABLES	Text
COS_QUERY_TABLE_EXISTENCE	Text
COS_TABLE_COLUMN_NUM	Number
COS_QUERY_RETRIEVE_COLUMN	Text
COS_COL_COLUMN_NUM	Number
COS_TYPE_NAME_COLUMN_NUM	Number
COS_PRECISION_COLUMN_NUM	Number
COS_LENGTH_COL_NUM	Number
CO_QUERY_ADD_TABLE	Text
CO_QUERY_ADD_COLUMN	Text
CO_INT16	Text
CO_INT32	Text
CO_INT64	Text
CO_BOOLEAN	Text
CO_DATETIME	Text
CO_DOUBLE	Text
CO_FLOAT	Text
CO_SMALLSTRING	Text
CO_LARGESTRING	Text
CO_EXTENDEDSTRING	Text
CO_CHAR	Text
CO_QUERY_RETRIEVE_DATABASES	Text

Εικόνα 2.2.15 Τα πεδία του πίνακα ConnectionTypeSettings

Στα πεδία του πίνακα που θα περιγραφτούν παρακάτω θα πρέπει να γραφτούν με τέτοιο τρόπο όπου το πρόγραμμα να μπορεί να καταλάβει που να βάλει κάποιες πληροφορίες που συνήθως χρειάζονται σε τέτοια ερωτήματα. Συγκεκριμένα, υπάρχουν πεδία που περιέχουν αγκύλες με αριθμούς. Όταν το πρόγραμμα συναντήσει μία αγκύλη τέτοια, ανάλογα με τον αριθμό που θα οριστεί μέσα, θα την αντικαταστήσει με την κατάλληλη πληροφορία. Ειδικότερα οι πιθανές αγκύλες μαζί με περιγραφή των πιθανών πληροφοριών που θα αντικαταστήσουν είναι οι ακόλουθες:

- {0}: Το όνομα της βάσης δεδομένων.
- {1}: Το πεδίο που υποδηλώνει την επιπλέον πληροφορία που θα πρέπει να χρησιμοποιηθεί μεταξύ της βάσης δεδομένων και του πίνακα. Ένα τέτοιο παράδειγμα είναι το «dbo», που υποδηλώνει ότι το πρόγραμμα θα βρει τον πίνακα μέσω του χρήστη «dbo» (ο συγκεκριμένος είναι super user στον SQL Server).
- {2}: Το όνομα του πίνακα που θα εκτελεστεί το αντίστοιχο ερώτημα.
- {4}: Τα πεδία στα οποία θα εισαχθούν πληροφορίες.
- {5}: Στατικές τιμές που θα αντιστοιχούν στα πεδία που ορίστηκαν στην αγκύλη {4}.

1. Το πεδίο COS_ID είναι το μοναδικό κλειδί για την εκάστοτε ρύθμιση σύνδεσης, και δεν θα πρέπει να αλλάχθει από το χρήστη.
2. Το πεδίο COS_TYPE είναι ένας κωδικός με τον οποίο το πρόγραμμα αναγνωρίζει το είδος της βάσης δεδομένων η οποία θα χρησιμοποιήσει αυτές τις ρυθμίσεις. Οι πιθανές τιμές που ορίζονται σε αυτό είναι οι εξής:
 - i. MSSQLServer = 1,
 - ii. Oracle = 2,
 - iii. MSAccess = 3,
 - iv. MySQL = 4
3. Το πεδίο COS_MULTI_INSERT_A περιέχει το πρώτο σκέλος του ερωτήματος που χρησιμοποιείται για την εισαγωγή δεδομένων στη βάση (που ορίζεται από το πεδίο COS_TYPE). Ένα παράδειγμα ενός τέτοιου σκέλους είναι το «INSERT INTO {0}{1}{2} ({4})». Οι αγκύλες με τα νούμερα περιλαμβάνουν τις πληροφορίες όπως αυτές παρουσιάστηκαν προηγουμένως.
4. Το πεδίο COS_MULTI_INSERT_B περιέχει το δεύτερο μέρος του ερωτήματος εισαγωγής δεδομένων. Ένα τέτοιο παράδειγμα είναι το «SELECT {5} UNION ALL».
5. Το πεδίο COS_MULTI_INSERT_C περιέχει τον αριθμό των χαρακτήρων που θα πρέπει το πρόγραμμα να κόψει πριν εκτελέσει το ερώτημα. Όταν για παράδειγμα έχουμε (με βάση τα πεδία COS_MULTI_INSERT_A και COS_MULTI_INSERT_B), το ερώτημα που φτιάχνεται δυναμικά και είναι «INSERT INTO BANK_db.dbo.TRANSACTIONS (id, name, price) SELECT 1, 'Trans1', 500 UNION ALL SELECT 2, 'Trans2', 230 UNION ALL», τότε το δεύτερο UNION ALL, θα πρέπει να σβηστεί. Γι' αυτόν το λόγο το πρόγραμμα χρειάζεται αν γνωρίζει πόσοι θα πρέπει να είναι οι χαρακτήρες που θα κόψει στο τέλος. Στην περίπτωση μας θα είναι 9 και έτσι το τελικό ερώτημα θα είναι: «INSERT INTO BANK_db.dbo.TRANSACTIONS (id, name, price) SELECT 1, 'Trans1', 500 UNION ALL SELECT 2, 'Trans2', 230».
6. Το πεδίο COS_QUERY_RETRIEVE_TABLES περιέχει το ερώτημα που θα πρέπει να εκτελεστεί, για να διαβάσει η εφαρμογή τους πίνακες που περιέχει η αντίστοιχη βάση. Ένα παράδειγμα είναι το ακόλουθο: «SHOW TABLES FROM {0}».
7. Το πεδίο COS_QUERY_TABLE_EXISTENCE περιέχει ένα ερώτημα με το οποίο θα μπορεί η εφαρμογή να καταλάβει αν ο πίνακας υπάρχει στην επιλεγμένη βάση. Ένα τέτοιο παράδειγμα είναι το ακόλουθο: «SELECT * FROM information_schema.tables t WHERE t.TABLE_NAME = '{2}' AND t.TABLE_SCHEMA='{0}'». Το ερώτημα θα πρέπει να επιστρέφει μία εγγραφή που να δείχνει τον πίνακα για να επιβεβαιώσει η εφαρμογή ότι

υπάρχει. Το ποιο πεδίο θα πρέπει να ελέγξει θα οριστεί στα επόμενα πεδία όπως θα δούμε.

8. Το πεδίο `COS_TABLE_COLUMN_NUM` περιέχει έναν αριθμό που δείχνει ποια στήλη θα πρέπει να κοιτάζει που θα περιέχει το όνομα του πίνακα.
9. Το πεδίο `COS_QUERY_RETRIEVE_COLUMN` ορίζει το ερώτημα που είναι απαραίτητο να εκτελεστεί για να πάρει τις στήλες που έχει ο πίνακας που θα περάσει στην αγκύλη {2} το πρόγραμμα. Ένα τέτοιο παράδειγμα είναι το ακόλουθο: «EXEC SP_COLUMNS '{2}'». Το ερώτημα θα πρέπει να επιστρέφει μία εγγραφή για κάθε στήλη που θα περιέχει πληροφορίες που θα ζητηθούν από τα πεδία:
 - i. `COS_COL_COLUMN_NUM`,
 - ii. `COS_TYPE_NAME_COLUMN_NUM`,
 - iii. `COS_PRECISION_COLUMN_NUM`,
 - iv. `COS_LENGTH_COL_NUM`
10. Το πεδίο `COS_COL_COLUMN_NUM` περιέχει τη θέση στην οποία υπάρχει το όνομα της στήλης από το ερώτημα που εκτελέστηκε του πεδίου `COS_QUERY_RETRIEVE_COLUMN`.
11. Το πεδίο `COS_TYPE_NAME_COLUMN_NUM` περιέχει τη θέση στην οποία υπάρχει το όνομα του τύπου δεδομένων (data type) της συγκεκριμένης στήλης.
12. Το πεδίο `COS_PRECISION_COLUMN_NUM` περιέχει τη θέση της στήλης που δείχνει τον αριθμό των επιτρεπτών χαρακτήρων του πεδίου. Μετράει δηλαδή την ακρίβεια του πεδίου έτσι που αν για παράδειγμα έχουμε `varchar(255)`, τότε η τιμή του πεδίου θα είναι 255.
13. Το πεδίο `COS_LENGTH_COL_NUM` περιέχει τη θέση της στήλης που δείχνει πόσα δεκαδικά επιτρέπονται στο συγκεκριμένο πεδίο. Όπως μπορεί κανείς εύκολα να καταλάβει στην περίπτωση των πεδίων τύπου χαρακτήρα το `COS_LENGTH_COL_NUM` θα δείχνει σε μία στήλη που δεν θα έχει τιμή.
14. Το πεδίο `CO_QUERY_ADD_TABLE` κρατάει το ερώτημα που θα πρέπει να εκτελεστεί για την δημιουργία ενός νέου πίνακα όταν αυτός δεν υπάρχει (Ένα τέτοιο παράδειγμα παρουσιάζεται στην Εικόνα 3.7.30). Ένα παράδειγμα ενός τέτοιου ερωτήματος είναι το ακόλουθο: «CREATE TABLE {1} {2} ({4})». Σε αυτό το σημείο στο {4} δεν ορίζονται απλά τα πεδία, αλλά και ο τύπος δεδομένων τους με βάση τον τύπο που έχει οριστεί στο τελικό σχήμα.
15. Το πεδίο `CO_QUERY_ADD_COLUMN` περιέχει το ερώτημα με το οποίο θα μπορεί το πρόγραμμα να προσθέσει κάποιο έξτρα πεδίο αν αυτό λείπει από τη βάση αλλά έχει οριστεί στο τελικό σχήμα. Και αυτό θα γίνει πρώτα μέσα από την επιβεβαίωση του χρήστη. Ένα παράδειγμα είναι το: «ALTER TABLE {1} {2} ADD {4}».

16. Το πεδίο CO_INT16 αντιπροσωπεύει το μικρό ακέραιο που ορίζεται από το πρόγραμμα. Αν θα πρέπει να αλλάξει τότε θα οριστεί το αντιπροσωπευτικό κείμενο του συγκεκριμένου πεδίου. Ένα τέτοιο παράδειγμα είναι το «smallint».
17. Το CO_INT32 αντιπροσωπεύει τον ακέραιο μεγέθους 32 bit. Ένα παράδειγμα είναι το «int».
18. Το CO_INT64 αντιπροσωπεύει τον ακέραιο μεγέθους 64 bit. Ένα παράδειγμα είναι το «bigint».
19. Το CO_BOOLEAN αντιπροσωπεύει τον τύπο δεδομένων που μπορεί να πάρει δύο μόνο τιμές δηλαδή μηδέν ή ένα. Διαφορετικά ορίζεται ως ο τύπου που παίρνει τιμές «true» ή «false». Ένα παράδειγμα είναι το «bit».
20. Το CO_DATETIME αντιπροσωπεύει το πεδίο ημερομηνίας. Ένα παράδειγμα είναι το «datetime».
21. Το CO_DOUBLE αντιπροσωπεύει το δεκαδικό διπλής ακρίβειας. Ένα παράδειγμα είναι το «decimal(18,7)».
22. Το CO_FLOAT αντιπροσωπεύει το δεκαδικό απλής ακρίβειας. Ένα παράδειγμα είναι το «float».
23. Το CO_SMALLSTRING αντιπροσωπεύει τη μικρή συμβολοσειρά λίγων σχετικά χαρακτήρων. Ένα παράδειγμα είναι το «varchar(255)».
24. Το CO_LARGESTRING αντιπροσωπεύει τη συμβολοσειρά σχετικά πολλών χαρακτήρων. Ένα παράδειγμα είναι το «varchar(2000)».
25. Το CO_EXTENDEDSTRING αντιπροσωπεύει τη συμβολοσειρά πολλών χαρακτήρων. Ένα παράδειγμα είναι το «varchar(8000)».
26. Το CO_CHAR αντιπροσωπεύει τον τύπο δεδομένων που έχει μόνο ένα χαρακτήρα. Ένα παράδειγμα είναι το «char(1)».
27. Το πεδίο CO_QUERY_RETRIEVE_DATABASES περιέχει το ερώτημα που επιστρέφει τις διαθέσιμες βάσεις δεδομένων από το σύστημα. Ένα τέτοιο παράδειγμα είναι το εξής: «SELECT name FROM SYS.DATABASES».

3 Εγχειρίδιο χρήσης

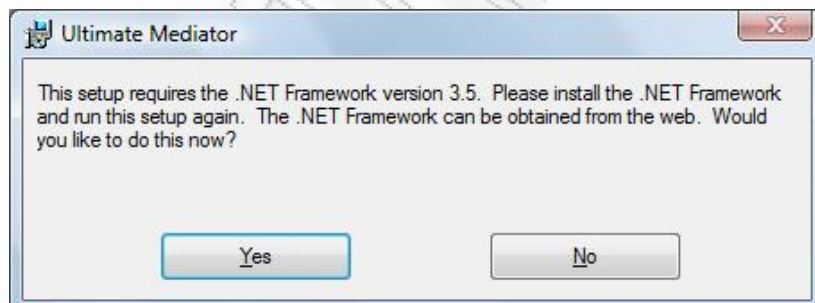
Στην συγκεκριμένη ενότητα θα γίνει αναλυτική παρουσίαση του προγράμματος με όλα τα πιθανά σενάρια χρήσης, από τη διαδικασία εγκατάστασης, μέχρι και την τελική του διαχείριση. Θα δοθούν αρκετά παραδείγματα με στόχο να αποσαφηνιστούν όλα εκείνα τα σημεία που μπορεί να προκαλέσουν σύγχυση κατά τη χρήση, όπως επίσης και όλες οι περιπτώσεις σφαλμάτων που μπορεί να προκύψουν. Αποτέλεσμα όλων αυτών θα είναι η διευκόλυνση στην εκμάθηση του αναγνώστη.

3.1 Απαιτούμενο λογισμικό

Για την εγκατάσταση του προγράμματος είναι απαραίτητη η ύπαρξη του *.Net Framework 3.5.*, και η χρήση Windows 2000 ή μεταγενέστερου λειτουργικού συστήματος Windows.

Αναφορικά με το *.Net Framework 3.5.*, ο χρήστης μπορεί να το κατεβάσει από Update της Microsoft ή πηγαίνοντας κατευθείαν μέσω browser στη σελίδα της Microsoft <http://www.microsoft.com/> και από εκεί να το κατεβάσει από τα διαθέσιμα downloads (<http://www.microsoft.com/downloads/>).

Αν ο χρήστης δεν έχει τη συγκεκριμένη έκδοση και προσπεράσει αυτό το βήμα τότε κατά την εκτέλεση του setup (*UMediatorSetup.msi*) θα λάβει το ακόλουθο μήνυμα (Εικόνα 3.1.1).



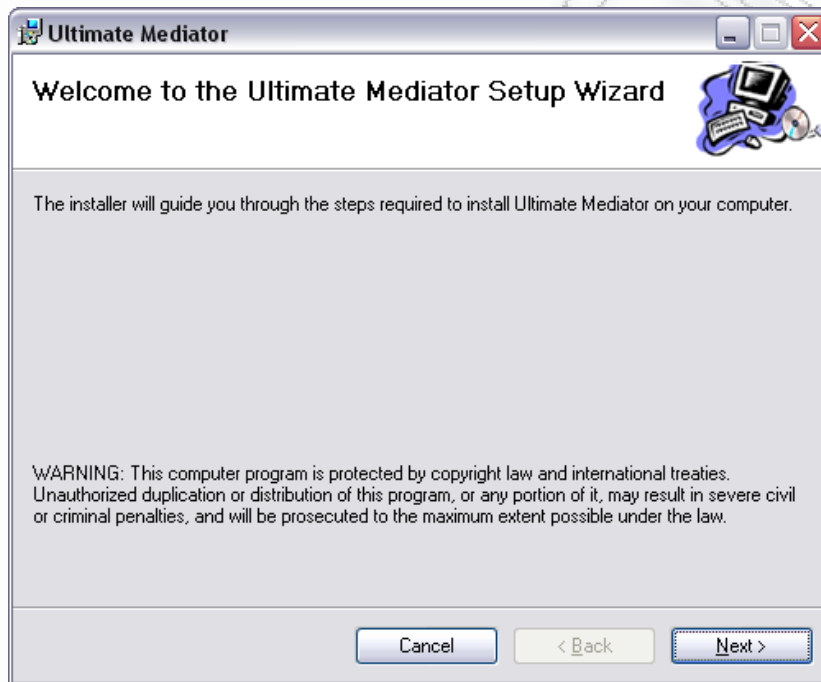
Εικόνα 3.1.1 Μήνυμα της εφαρμογής όταν δεν είναι εγκατεστημένη η 3.5 έκδοση λογισμικού του *.Net Framework*

Σε αυτήν την περίπτωση αν ο χρήστης επιλέξει το «Yes», τότε θα ανοίξει αυτόματα η σελίδα εγκατάστασης του *.Net 3.5.*

3.2 Εγκατάσταση εφαρμογής

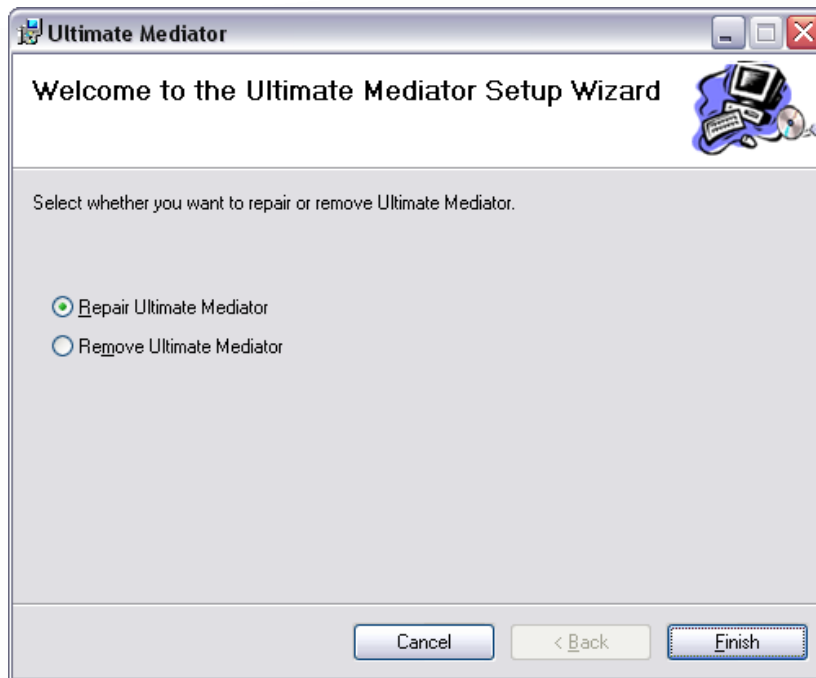
Το πρόγραμμα παραδίδεται στον τελικό χρήστη μέσω ειδικής εγκατάστασης που παρέχεται από το αρχείο *UMediatorSetup.msi*. Κατά την εκτέλεση του αρχείου θα εμφανιστούν διάφορες διαδοχικές οθόνες στις οποίες χρήστης θα πρέπει να συμπληρώσει σε κάποιες από αυτές στοιχεία της επιλογής του, ενώ κάποιες άλλες είναι απλά πληροφοριακές.

Ξεκινώντας η πρώτη οθόνη πληροφορεί ότι πρόκειται να γίνει η εγκατάσταση του συγκεκριμένου λογισμικού (Εικόνα 3.2.1).



Εικόνα 3.2.1 Πρώτη οθόνη εγκατάστασης του λογισμικού

Αν η εγκατάσταση έχει ήδη γίνει, τότε κατά την εκτέλεση του setup θα εμφανιστεί το παράθυρο που φαίνεται στην Εικόνα 3.2.2. Θα πρέπει να γίνει πρώτα η απ-εγκατάσταση του προγράμματος από τον υπολογιστή επιλέγοντας «*Remove Ultimate Mediator*» ή μπορεί να γίνει μέσα από το Control Panel των Windows, πηγαίνοντας στην προσθαφαίρεση προγραμμάτων και από εκεί στην απ-εγκατάσταση του «Ultimate Mediator».



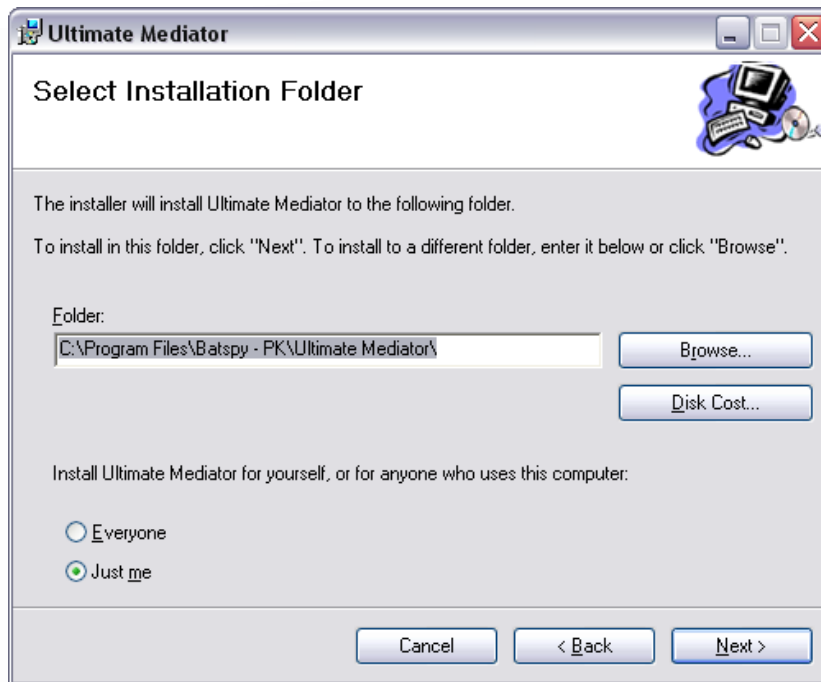
Εικόνα 3.2.2 Περίπτωση ήδη εγκατεστημένης εφαρμογής

Στην περίπτωση που ο χρήστης δεν λάβει ούτε την παραπάνω οθόνη (Εικόνα 3.2.2), αλλά του εμφανιστεί μήνυμα που παρουσιάζεται παρακάτω (Εικόνα 3.2.3), τότε θα πρέπει αναγκαστικά να ακολουθήσει τη διαδικασία που ήδη προαναφέρθηκε, για την προσθαφαίρεση προγραμμάτων. Κάτι τέτοιο παρουσιάζεται όταν γίνεται εγκατάσταση νέας έκδοσης στον υπολογιστή. Επειδή το νέο setup δεν περιέχει τις πληροφορίες για την παλιότερα εγκατεστημένη έκδοση, δε δίνεται η δυνατότητα της απευθείας απ-εγκατάστασης από εκεί.



Εικόνα 3.2.3 Μήνυμα που δηλώνει ότι υπάρχει ήδη μία εγκατεστημένη έκδοση του προγράμματος.

Συνεχίζοντας από την πρώτη οθόνη, με το κουμπί Next προχωράμε στην επόμενη όπου και εκεί ορίζουμε τη διαδρομή στο δίσκο στην οποία θα γίνει η εγκατάσταση (Εικόνα 3.2.4).



Εικόνα 3.2.4 Οθόνη επιλογής διαδρομής και χρηστών συστήματος

Σε αυτήν την οθόνη η εγκατάσταση έχει προεπιλεγεί να γίνει στο φάκελο:
C:\Program Files\Batspy - PK\Ultimate Mediator

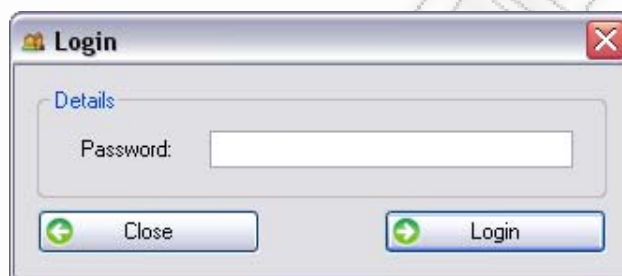
Αν όμως επιθυμείται κάποια άλλη διαδρομή τότε αυτή μπορεί να αλλάξει. Στη συγκεκριμένη οθόνη ορίζεται επίσης και αν θα υπάρχει δυνατότητα να τρέξουν όλοι οι χρήστες του συστήματος το λογισμικό ή μόνο ο τωρινός (που πραγματοποιεί την εγκατάσταση).

Μετά την ολοκλήρωση των προηγούμενων βημάτων το λογισμικό είναι έτοιμο για εγκατάσταση. Κατά το τέλος της ολοκλήρωσης της εγκατάστασης θα έχουν δημιουργηθεί όλα τα απαραίτητα αρχεία που χρειάζονται για να εκτελεστεί η εφαρμογή, όπως επίσης και ένα εικονίδιο στο Start Up Programs των Windows, για γρήγορη πρόσβαση στο πρόγραμμα.

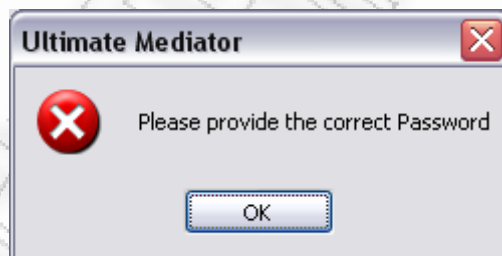
3.3 Είσοδος στο πρόγραμμα

Κατά την εκτέλεση του προγράμματος ζητείται από το χρήστη να δώσει το συνθηματικό / password για να συνδεθεί στο σύστημα. Αυτό είναι ουσιαστικά ο κωδικός με τον οποίο συνδέεται στη βάση του λογισμικού. Αν ο κωδικός αυτός είναι σωστός, τότε ο χρήστης μπορεί να κάνει χρήση του προγράμματος. Με αυτή τη δυνατότητα κάθε φορά που επιθυμείται να αλλάξει ο κωδικός απλά χρειάζεται να αλλάξει ο κωδικός της MSAccess βάσης.

Η Εικόνα 3.3.1 δείχνει την οθόνη εισόδου κωδικού πρόσβασης. Αν ο χρήστης πληκτρολογήσει λάθος κωδικό, τότε ένα μήνυμα λάθους επιστρέφεται από την εφαρμογή (Εικόνα 3.3.2). Ο χρήστης θα μπορεί να δοκιμάσει ξανά και ξανά για την εισαγωγή του κωδικού μέχρι να κλείσει τη φόρμα ή να πατήσει το κουμπί «Close».



Εικόνα 3.3.1 Οθόνη εισόδου κωδικού πρόσβασης



Εικόνα 3.3.2 Μήνυμα σφάλματος σε περίπτωση λάθους κωδικού

Αν ο χρήστης επιλέξει να κλείσει την Login, τότε η εφαρμογή τερματίζει μη δίνοντας τη δυνατότητα εισόδου στο χρήστη. Αν ο κωδικός πρόσβασης είναι όμως σωστός, τότε η εφαρμογή δείχνει την κύρια οθόνη της (Εικόνα 3.3.3).



Εικόνα 3.3.3 Κύρια οθόνη εφαρμογής

3.4 Περιγραφή των menu της βασικής Φόρμας

Μέσα από την κεντρική φόρμα παρέχονται σε menu όλες οι διαθέσιμες λειτουργίες της εφαρμογής. Συγκεκριμένα υπάρχει η ακόλουθη δομή φορμών:

- **File**
 - *Exit*
- **Management**
 - *Manage Connections*
 - *Manage Web Services*
- **Mediator**
 - *Designer*
- **Transporter**
 - *XML to Web Service*
- **Help**
 - *Help Topics*
 - *About*

Παρακάτω παρουσιάζεται μία συνοπτική περιγραφή των menu για τις λειτουργίες που περιέχουν.

File:

Στο μενού του File υπάρχει μόνο η λειτουργία εξόδου της εφαρμογής. Η έξοδος μπορεί να γίνει και από το standard κουμπί «X» εξόδου της φόρμας.

Management:

Στο μενού του management βρίσκονται οι φόρμες οι οποίες διαχειρίζονται τις πληροφορίες που σώζει ο χρήστης κατά τη χρήση του προγράμματος. Στην τρέχων έκδοση του λογισμικού (ver. 1.0.0), η διαχείριση εστιάζεται σε επίπεδο Connections και Web Services.

Η διαχείριση των connections όπως θα παρουσιαστεί και σε επόμενη ενότητα, δίνει την ευκαιρία στο χρήστη να προσθέσει, να αφαιρέσει, και να επεξεργαστεί όλα τα Connections που έχει ορίσει μεταξύ των βάσεων και XML αρχείων. Η διαχείριση αυτή είναι σημαντική και είναι η βάση όλου του συστήματος αφού μέσω των Connections γίνεται η μεταφορά δεδομένων από το ένα στο άλλο.

Η διαχείριση των Web Services είναι μία έξτρα λειτουργία μέσω της οποίας μπορεί να προσθέσει, να επεξεργαστεί και να αφαιρέσει ο χρήστης ρυθμίσεις σχετικές με Web Services, οι οποίες θα χρησιμοποιηθούν αργότερα για την αποστολή των δεδομένων. Η χρήση των Web Services δεν έχει άμεση σύνδεση με τα Connections. Ο χρήστης όμως μπορεί μέσα από την έξοδο κάποιων δεδομένων σε XML να τα αποστείλει σε κάποιο Web Service της επιλογής του.

Mediator Designer:

Μέσα από το μενού αυτό παρέχεται η πρόσβαση στη φόρμα σχεδίασης των συνδέσεων μεταξύ βάσεων και της μετατροπής και επιλογής των δεδομένων προς μεταφορά. Πιο συγκεκριμένα ο χρήστης μπορεί να επιλέξει Source Connections (Βάση ή XML), από όπου θα γίνει η λήψη των δεδομένων, και να τα στείλει σε ένα Target Connection (Βάση ή XML). Κατά την αποστολή μπορεί να ορίσει κριτήρια επιλογής δεδομένων (Conditions) και να ορίσει τις τελικές μετατροπές δεδομένων που επιθυμεί. Τέλος, δίνεται η δυνατότητα για έλεγχο σφαλμάτων πριν την τελική μεταφορά των δεδομένων.

Transporter:

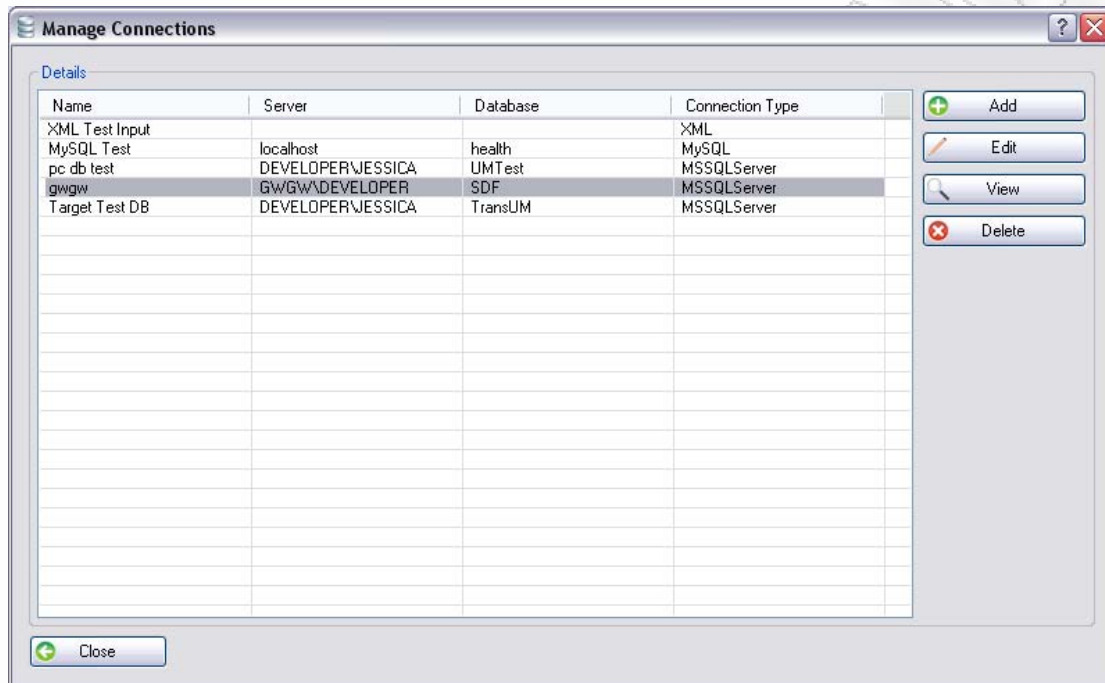
Μέσα από αυτό το μενού ο χρήστης μπορεί επίσης να αποστείλει, κατευθείαν σε κάποιο Web Service, δεδομένα που έχει ήδη κάνει εξαγωγή σε XML, ή άλλα δεδομένα που μπορεί να στείλει. Αυτή η φόρμα χρησιμοποιεί τις δύο managerial που αναφέραμε στο menu Management, για να πάρει όλες τις απαραίτητες πληροφορίες για την αποστολή.

Help:

Σε αυτό το μενού ο χρήστης μπορεί να δει πληροφορίες / βοήθεια για το πρόγραμμα. Η βοήθεια υπάρχει και στο φάκελο της εφαρμογής κάτω από τον «HTML Files» σε html σελίδες. Μέσα από αυτήν ο χρήστης μπορεί να λάβει πληροφορίες για τη λειτουργία κάποιων δυσνόητων λειτουργιών του προγράμματος. Επίσης παρέχεται και το About στο οποίο εμφανίζονται οι πληροφορίες για το ίδιο το πρόγραμμα (Έκδοση, Προγραμματιστής, κ.λ.π.).

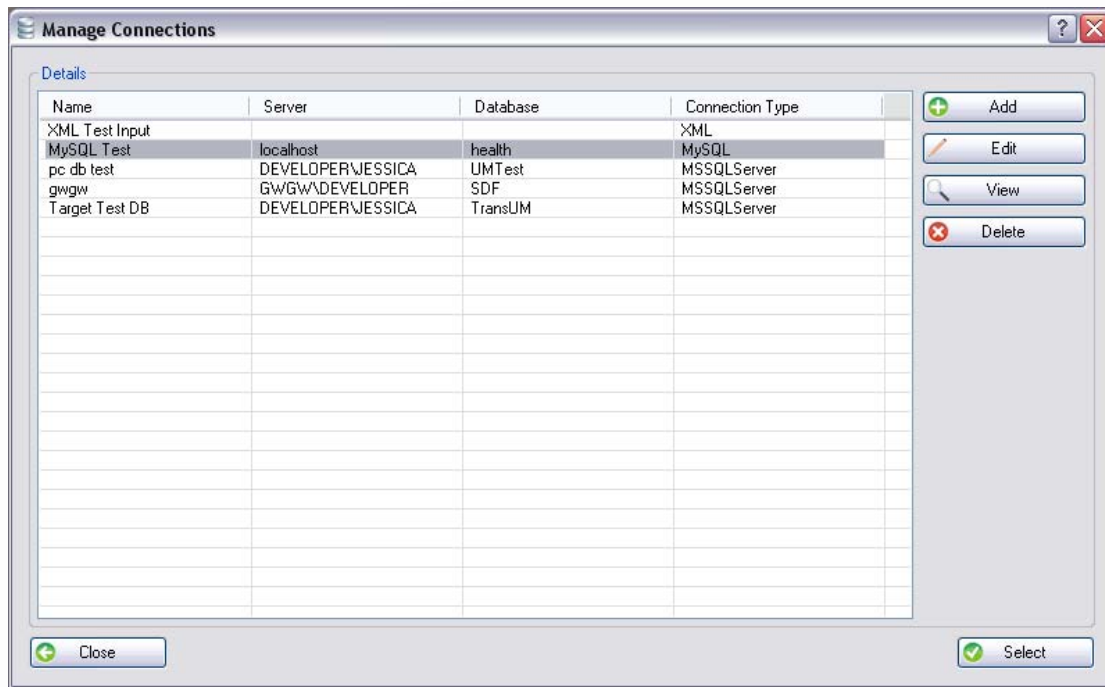
3.5 Manage Connections

Σε αυτήν την ενότητα παρουσιάζεται η διαχείριση των συνδέσεων (βάσης / xml). Η Εικόνα 3.5.1, παρουσιάζει μία πρώτη οθόνη της διαχείρισης των συνδέσεων.



Εικόνα 3.5.1 Οθόνη διαχείρισης Συνδέσεων

Επειδή η φόρμα αυτή χρησιμοποιείται και σε άλλα σημεία, σαν φόρμα επιλογής σύνδεσης, σε αυτές τις περιπτώσεις ενεργοποιείται ένα έξτρα κουμπί «Select» που δίνει τη δυνατότητα επιλογής κάποιου Connection. Η Εικόνα 3.5.2, δείχνει ένα τέτοιο παράδειγμα.



Εικόνα 3.5.2 Οθόνη διαχείρισης συνδέσεων με κουμπί επιλογής

Στην Εικόνα 3.5.1 και Εικόνα 3.5.2, βλέπουμε τις διαθέσιμες συνδέσεις που έχουμε ορίσει στη βάση δεδομένων του προγράμματος. Οι στήλες στη λίστα της Εικόνα 3.5.2 παρουσιάζουν (με την σειρά που αναφέρονται):

- i. το όνομα που έχει ορίσει ο χρήστης για την εκάστοτε σύνδεση,
- ii. το όνομα του υπολογιστή που παρέχεται η βάση δεδομένων,
- iii. το όνομα της επιλεγμένης βάσης που δίνει ο server,
- iv. το είδος του connection.

Από την προηγούμενη Εικόνα 3.5.2, μπορούμε να δούμε επίσης και τις διαθέσιμες λειτουργίες που έχει ο χρήστης. Οι λειτουργίες αυτές είναι:

- i. Προσθήκη νέας σύνδεσης,
- ii. Επεξεργασία υπάρχουσας,
- iii. Εμφάνιση Σύνδεσης χωρίς δυνατότητα επεξεργασίας,
- iv. Διαγραφή Σύνδεσης .

i. Προσθήκη νέας Σύνδεσης

Στην περίπτωση προσθήκης νέας σύνδεσης (Εικόνα 3.5.3), δίνονται συγκεκριμένα πεδία τα οποία την ορίζουν μοναδικά για τη μετέπειτα χρήση της. Τα πεδία αυτά όπως φαίνονται και στην εικόνα διαχωρίζονται ανάλογα με τον τύπο της σύνδεσης που θα επιλέξει ο χρήστης. Σε νέα σύνδεση σαν προσωρινή προεπιλογή επιλέγεται η MSAccess. Αυτός είναι και ο λόγος που ενεργοποιούνται τα πεδία εισαγωγής πληροφοριών σύνδεσης τύπου βάσης και όχι τα πεδία για την εισαγωγή XML (Εικόνα 3.5.4).

Add Connection

Details

Name :

Connection Type : MSAccess

DB Connection XML Connection

Server :

Username :

Password :

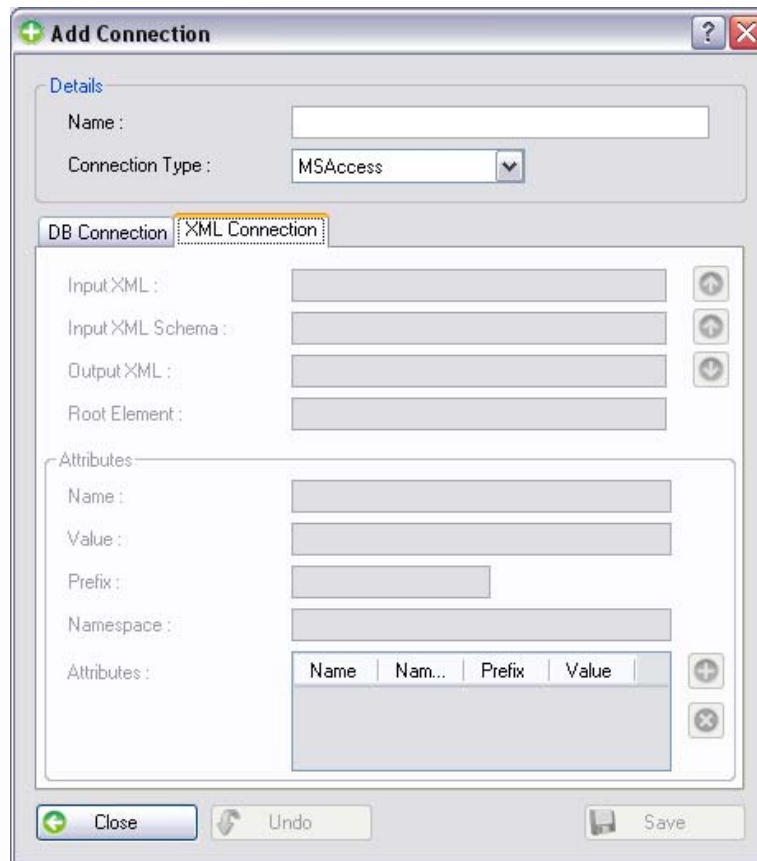
Default Database : Not Available

Extra User Field : (e.g. dbo)

Trusted Connection : No

Timeout : 30 seconds

Εικόνα 3.5.3 Οθόνη προσθήκης νέας σύνδεσης



Εικόνα 3.5.4 Απόκρυψη δεδομένων εισαγωγής XML λόγω επιλογής σύνδεσης τύπου Βάσης

Τα διαθέσιμα είδη σύνδεσης είναι τα ακόλουθα:

1. MSSQLServer,
2. Oracle,
3. MSAccess,
4. MySQL,
5. XML

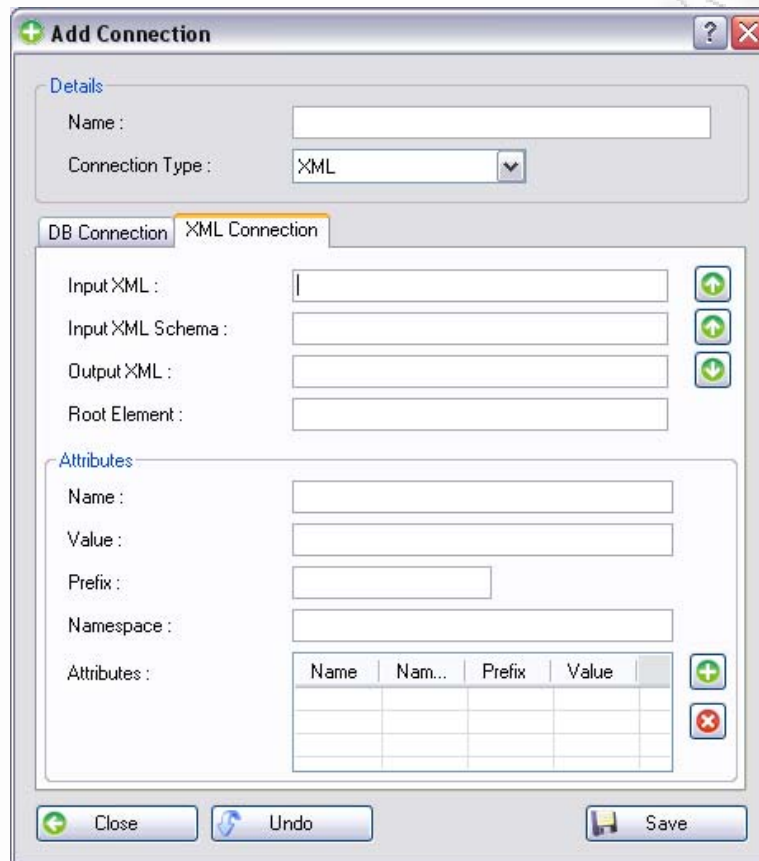
Σύνδεση τύπου βάσης

Επιλέγοντας σύνδεση τύπου βάσης και συμπληρώνοντας τα τρία πρώτα διαθέσιμα πεδία, *Server*, *Username*, *Password*, ο χρήστης μπορεί πατώντας τη λίστα για την επιλογή της βάσης να κάνει σύνδεση με βάση τα δεδομένα που ήδη έδωσε στη βάση και να λάβει όλες τις διαθέσιμες βάσεις δεδομένων. Σε περίπτωση που έχει κάνει λάθος και θέλει να αλλάξει τη σύνδεση και να ξαναπροσπαθήσει μπορεί να χρησιμοποιήσει το Refresh κουμπί.

Εφόσον εισάγει όλα τα δεδομένα μπορεί να κάνει ένα Test Connection χρησιμοποιώντας το αντίστοιχο κουμπί, για να δει αν η σύνδεση πραγματοποιείται επιτυχώς. Αν η σύνδεση είναι επιτυχής, τότε μπορεί να γίνει η αποθήκευση των πληροφοριών σύνδεσης στη βάση.

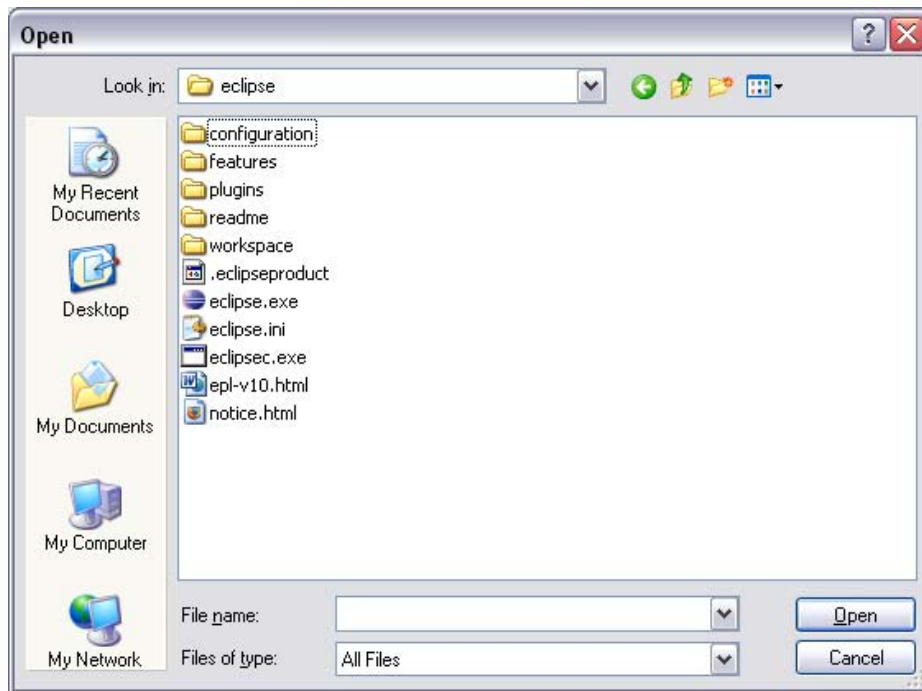
Σύνδεση τύπου αρχείου XML

Αν ο χρήστης επιλέξει την σύνδεση τύπου αρχείου XML, τότε μπορεί να δώσει πληροφορίες για το XML που θα παράγεται. Η Εικόνα 3.5.5 δείχνει ένα τέτοιο παράδειγμα.



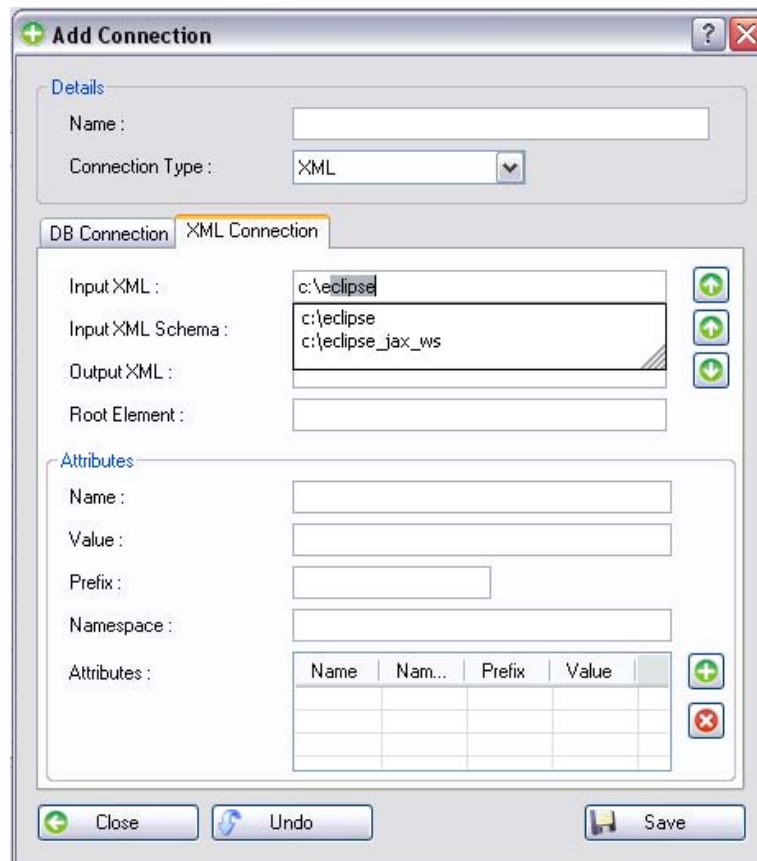
Εικόνα 3.5.5 Δεδομένα εισαγωγής σε σύνδεση τύπου XML αρχείου

Στα πεδία εισαγωγής Input XML, Input XML Schema και Output XML, δίνεται η δυνατότητα στο χρήστη να δώσει τα αρχεία επιλέγοντάς τα από τα διαθέσιμα κουμπιά. Κάθε ένα από αυτά ανοίγει μία νέα φόρμα επιλογής αρχείου (Εικόνα 3.5.6).



Εικόνα 3.5.6 Παράθυρο διαλόγου επιλογής αρχείου

Ο χρήστης, πέρα όμως αυτού του τρόπου εισόδου, μπορεί να γράψει ο ίδιος τη διαδρομή στο δίσκο όπου βρίσκεται το αρχείο. Σε αυτήν την περίπτωση σε μία μικρή έξτρα βοήθεια του χρήστη, παρέχονται οι διαδρομές (paths) του συστήματος (Εικόνα 3.5.7).



Εικόνα 3.5.7 Εμφάνιση διαδρομής δίσκου κατά τη χειροκίνητη εισαγωγή αρχείου

Αν ο χρήστης συμπληρώσει το πεδίο Input XML και το πεδίο Input XML Schema τότε, όταν θα επιλέξει να πάρει τους διαθέσιμους πίνακες από τη συγκεκριμένη σύνδεση, θα διαβάσει το σχήμα των δεδομένων του XML από το Schema. Το Input XML θα χρησιμοποιηθεί αργότερα, όταν θα χρειαστεί να φορτώσει τα δεδομένα. Αν δεν ορίσει το schema, τότε θα διαβάσει τα δεδομένα και μέσα από αυτά, θα παράγει το Schema που θα εμφανίσει στο χρήστη.

Το Output XML χρησιμοποιείται για να ορίσει τη διαδρομή στο δίσκο όπου θα αποθηκευτεί το XML αρχείο όταν ο χρήστης θέσει σαν έξοδο σύνδεση τύπου XML. Το πεδίο αυτό είναι υποχρεωτικό όπως και το Input XML. Αν η σύνδεση χρησιμοποιηθεί μόνο για είσοδο ή μόνο για έξοδο, μπορεί να οριστεί ένα αρχείο το οποίο δεν υπάρχει. Στην περίπτωση όμως που χρησιμοποιηθεί σύνδεση και για τον δεύτερο μη-ορισμένο σωστά σκοπό, θα πρέπει να δοθούν σωστά δεδομένα, γιατί το πρόγραμμα θα χρησιμοποιήσει τα δοθέντα.

Μαζί με την είσοδο των αρχείων εισόδου και εξόδου, απαιτείται και η εισαγωγή του αρχικού πεδίου (Root Element). Κάτω από αυτό το πεδίο θα μπουν όλα τα εξαγόμενα δεδομένα. Αν για παράδειγμα τα δεδομένα που θα εξάγουμε είναι Customers, τότε το Root Element (π.χ. έστω ότι ο χρήστης όρισε «Results») θα οριστεί πάνω από τους Customers, όπως δείχνει η Εικόνα 3.5.8.

```

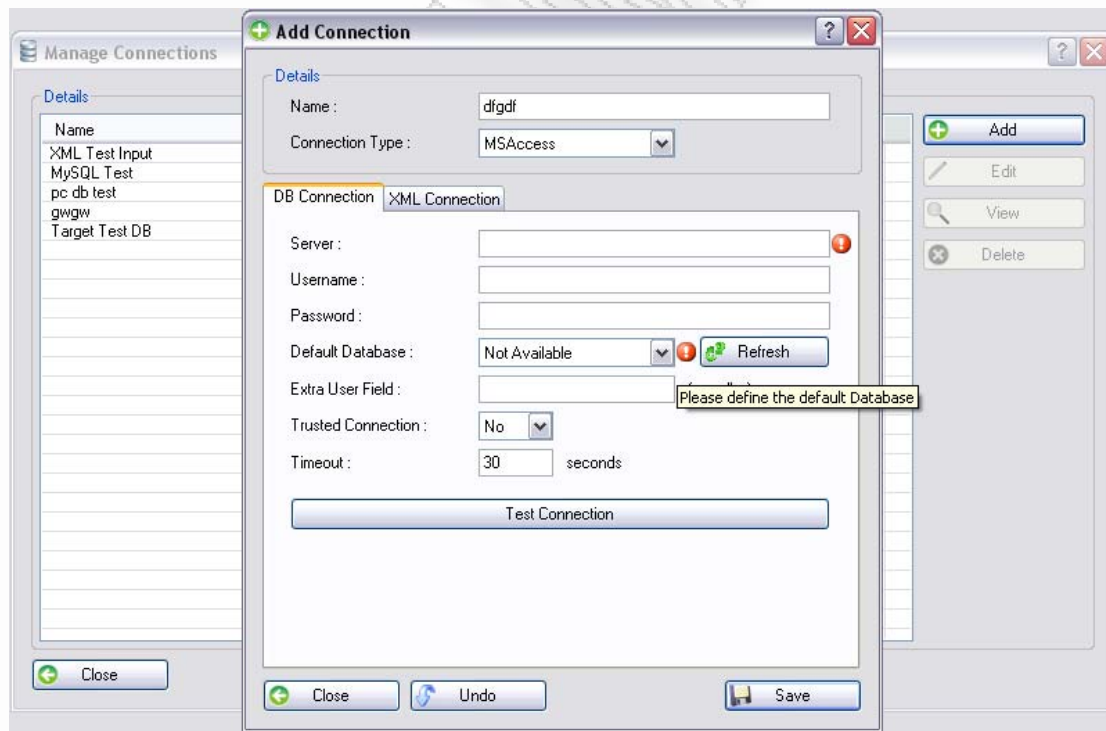
<Results>
  <Customers>
    <CU_ID>1</CU_ID>
    <CU_NAME>John</CU_NAME>
  </Customers>
  <Customers>
    <CU_ID>5</CU_ID>
    <CU_NAME>Catrine</CU_NAME>
  </Customers>
</Results>

```

Εικόνα 3.5.8 Παράδειγμα εμφάνισης Root Element

Από την Εικόνα 3.5.7, βλέπουμε ότι μαζί με τα άλλα δεδομένα εισαγωγής για το XML δίνεται και η δυνατότητα εισαγωγής Attributes. Τα Attributes ή αλλιώς γνωρίσματα, ορίζονται επάνω στο Root Element. Όντας ιδιότητες χαρακτηρίζουν το Root Element με κάποια τιμή. Οι συγκεκριμένες ιδιότητες μπορούν προαιρετικά να πάρουν prefixes και namespaces πάνω σε αυτά. Με τη χρήση των 2 κουμπιών, προσθήκης και αφαίρεσης attributes, μπορεί να αλλάζει συνεχώς η λίστα.

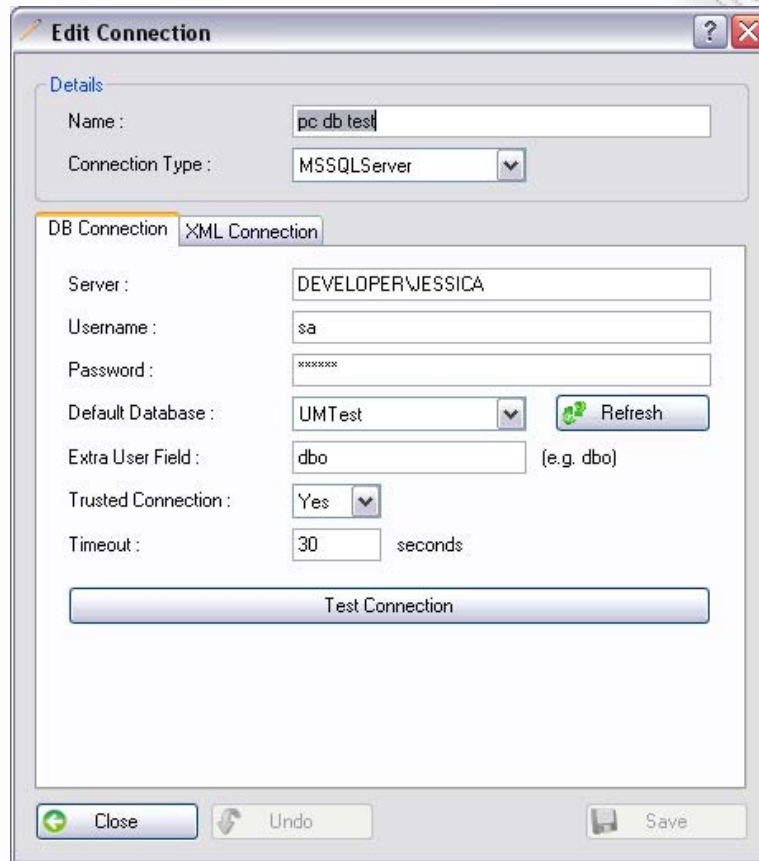
Σε περίπτωση που ο χρήστης πατήσει Save χωρίς να έχει δώσει όλες τις απαραίτητες πληροφορίες που χρειάζεται το σύστημα, τότε θα εμφανιστούν κάποια εικονίδια, στα αντίστοιχα σημεία εισόδου όπου θα αναφέρουν μία μικρή περιγραφή του λάθους (Εικόνα 3.5.9).



Εικόνα 3.5.9 Οθόνη εμφάνισης λάθους κατά την αποθήκευση των στοιχείων

ii. Επεξεργασία υπάρχουσας Σύνδεσης

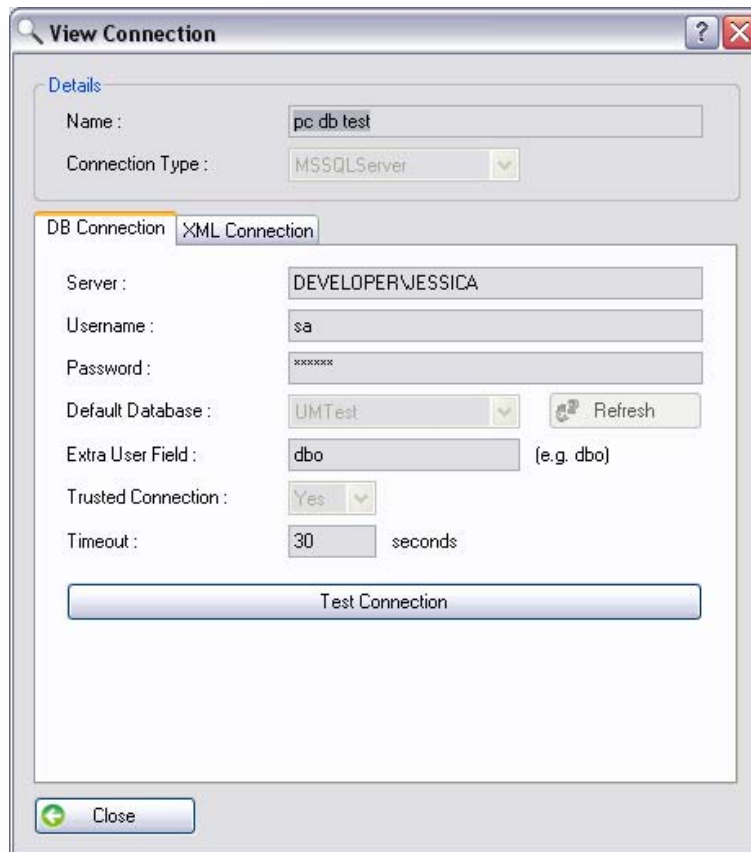
Η παραπάνω διαδικασία αναφέρεται στην περίπτωση της προσθήκης νέας σύνδεσης. Σε περίπτωση που ο χρήστης επιλέξει, από τη λίστα με τις συνδέσεις (Εικόνα 3.5.1), να επεξεργαστεί μία από αυτές, τότε ανοίγει η ίδια οθόνη, όπως είδαμε και στη διαδικασία προσθήκης, μόνο που υπάρχουν ήδη συμπληρωμένα όλα τα στοιχεία της σύνδεσης. Η Εικόνα 3.5.10, δείχνει μία τέτοια περίπτωση.



Εικόνα 3.5.10 Οθόνη εμφάνισης στοιχείων επεξεργασίας Σύνδεσης

iii. Εμφάνιση Σύνδεσης χωρίς δυνατότητα επεξεργασίας

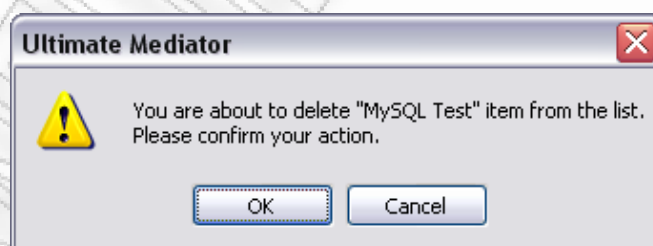
Αυτή η περίπτωση είναι μία έξτρα λειτουργία με την οποία μπορεί να δει τις πληροφορίες ο χρήστης χωρίς να έχει τη δυνατότητα να αλλάξει κάποιες από αυτές (Εικόνα 3.5.11). Όπως φαίνεται και στην εικόνα τα κουμπιά αποθήκευσης (Save) και ακύρωσης (Undo), δεν υπάρχουν.



Εικόνα 3.5.11 Οθόνη προβολής Σύνδεσης

iv. Διαγραφή Σύνδεσης

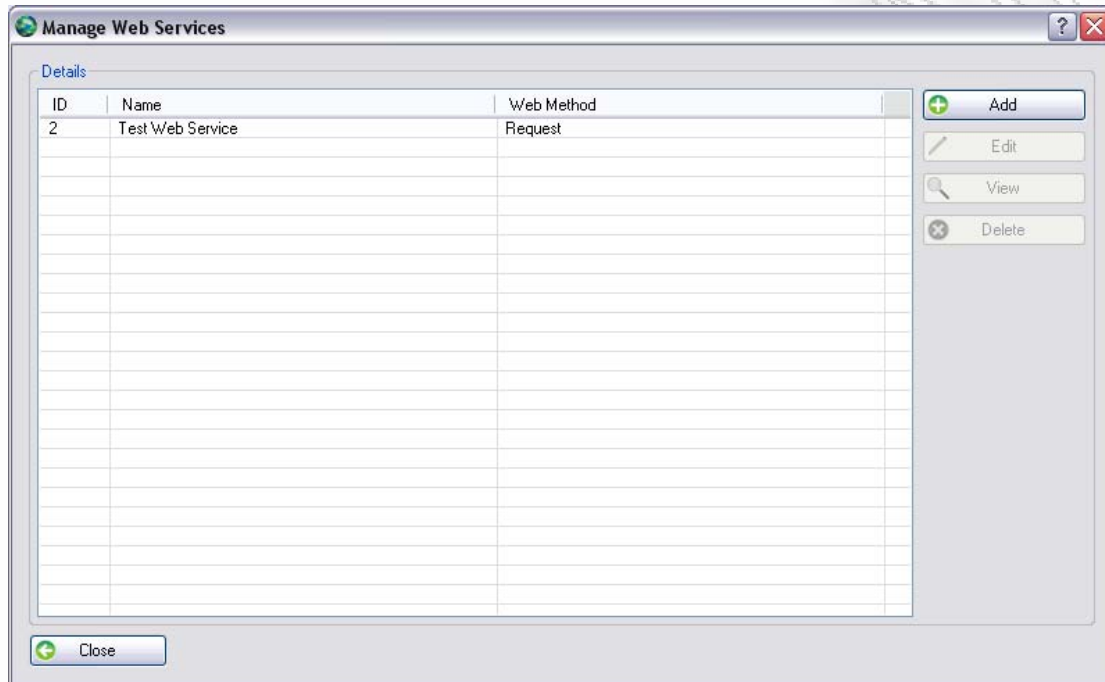
Στην περίπτωση της διαγραφής της σύνδεσης λαμβάνεται ένα μήνυμα για την επικύρωση της απάντησης του χρήστη (Εικόνα 3.5.12).



Εικόνα 3.5.12 Μήνυμα επικύρωσης διαγραφής Σύνδεσης

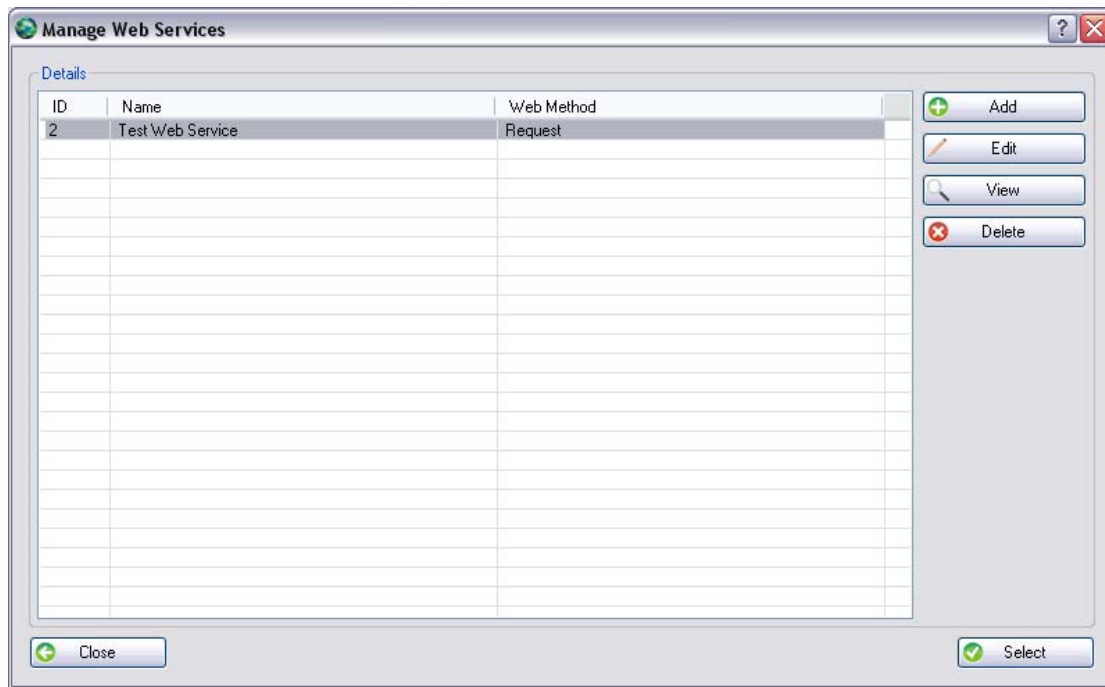
3.6 Manage Web Services

Στη φόρμα αυτή γίνεται η διαχείριση των Web Services που έχει ορίσει ο χρήστης. Όπως βλέπουμε και στην Εικόνα 3.6.1, έχει τη δυνατότητα προσθήκης, επεξεργασίας, εμφάνισης και διαγραφής ενός Web Service.



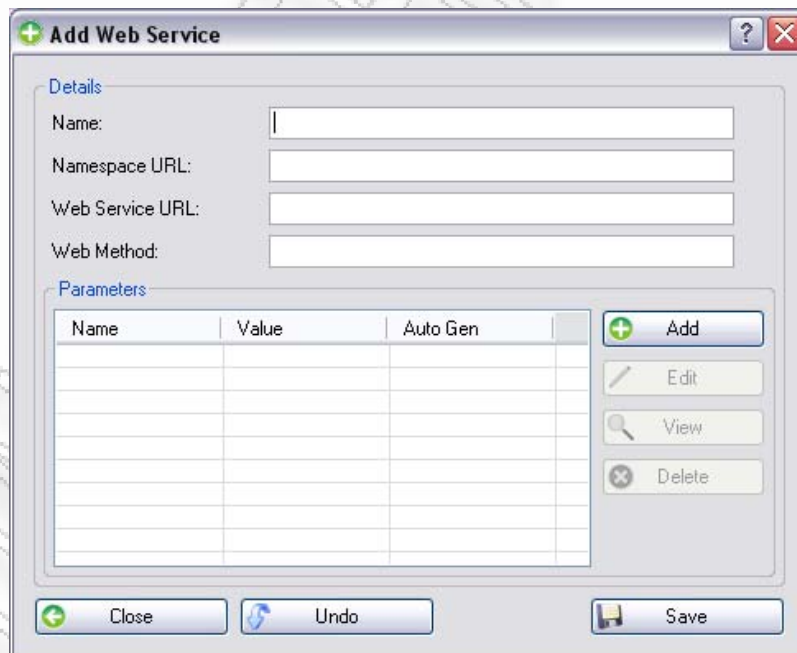
Εικόνα 3.6.1 Φόρμα Manage Web Services

Η συγκεκριμένη φόρμα, όπως και η «Manage Connections», χρησιμοποιείται όπως θα δούμε και σε επόμενη ενότητα για την επιλογή ενός από τα Web Services. Όταν χρησιμοποιηθεί για το σκοπό αυτό, τότε ένα έξτρα κουμπί ενεργοποιείται (Εικόνα 3.6.2), το οποίο προσφέρει τη δυνατότητα επιλογής.



Εικόνα 3.6.2 Φόρμα Manage Web Services με κουμπί επιλογής

Έστω ότι ο χρήστης επιλέξει να κάνει προσθήκη ενός Web Service. Σε αυτήν την περίπτωση του εμφανίζεται η Εικόνα 3.6.3.



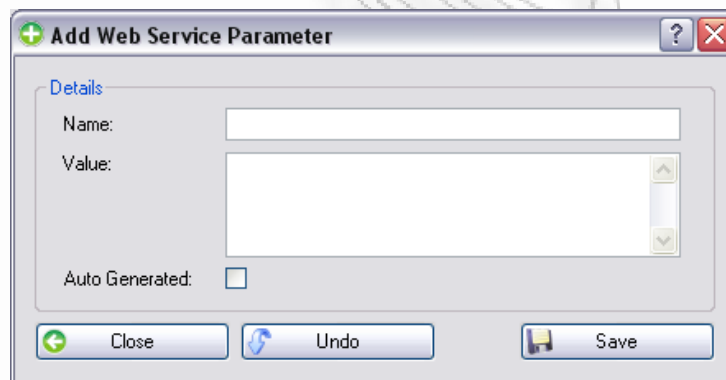
Εικόνα 3.6.3 Φόρμα προσθήκης νέου Web Service

Σε αυτήν τη φόρμα μπορεί να συμπληρώσει το όνομα του Web Service το οποίο είναι καθαρά για να το βλέπει ο ίδιος και όχι για να χρησιμοποιηθεί κατά τη σύνδεση. Επίσης ο χρήστης δίνει και το Namespace URL, το οποίο είναι ουσιαστικά

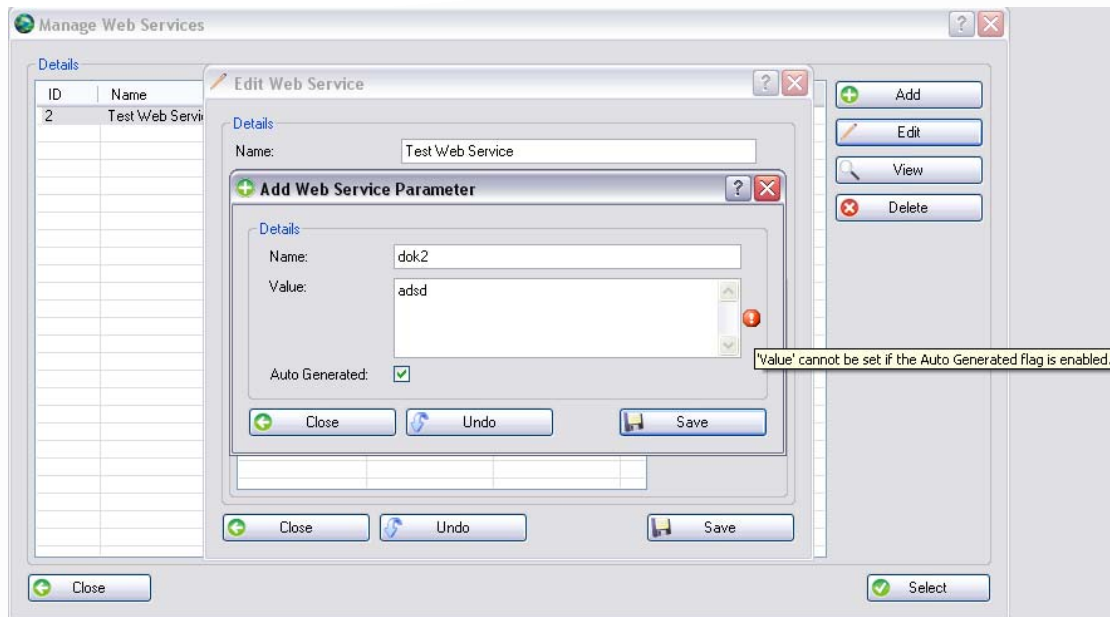
το URL του Server που φιλοξενεί το web service. Ακόμα δίνει και το Web Service URL που είναι ολόκληρη η διαδρομή του URL για τη συγκεκριμένη υπηρεσία. Το Web Method είναι η μέθοδος που θα καλέσει η εφαρμογή στην απομακρυσμένη υπηρεσία, και όπου θα περάσει τις παραμέτρους που παρατίθενται για εισαγωγή στο κάτω μέρος της φόρμας.

Η διαχείριση των παραμέτρων μπορεί να γίνει από τη λίστα που παρέχεται στο κάτω μέρος της φόρμας, όπου υπάρχουν ειδικά κουμπιά προσθήκης, επεξεργασίας, προβολής και διαγραφής. Οι προαναφερθέντες λειτουργίες μπορούν εν μέρη να εκτελεστούν και μέσα από το πληκτρολόγιο (Enter και Delete εφόσον έχει επιλέξει μία παράμετρο), και με τη χρήση του ποντικιού (κάνοντας διπλό κλικ σε κάποια, αυτόματα γίνεται η επεξεργασία της).

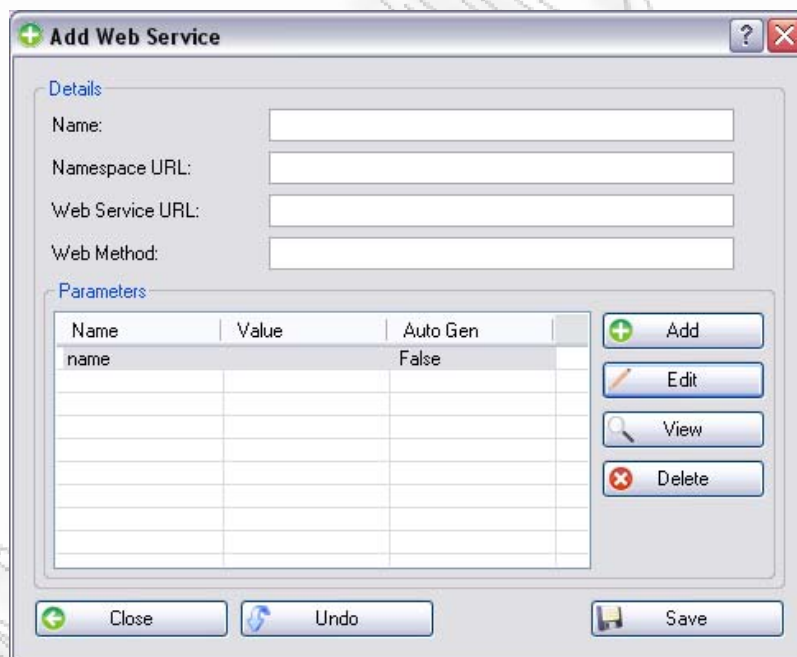
Κατά την επιλογή προσθήκης εμφανίζεται η Εικόνα 3.6.4. Εκεί ο χρήστης μπορεί να ορίσει το όνομα, την τιμή της ή να τσεκάρει ότι η τιμή θα συμπληρώνεται δυναμικά. Αν επιλέξει να συμπληρώνεται δυναμικά, τότε δεν μπορεί να ορίσει στατική τιμή. Ο χρήστης μπορεί όμως να μη ορίσει στατική τιμή, ενώ δεν έχει τσεκάρει τη δυναμική συμπλήρωσή του, γιατί μπορεί να χρειάζεται να είναι κενή η τιμή της παραμέτρου. Αν συμβούν οι δύο αναφερθέντες περιπτώσεις, τότε το πρόγραμμα θα παράγει μήνυμα λάθους για την πρώτη, ενώ θα επιτρέψει την εισαγωγή της παραμέτρου για τη δεύτερη. Αυτές οι περιπτώσεις φαίνονται στην Εικόνα 3.6.5, και στην Εικόνα 3.6.6.



Εικόνα 3.6.4 Οθόνη προσθήκης παραμέτρου σε Web Service

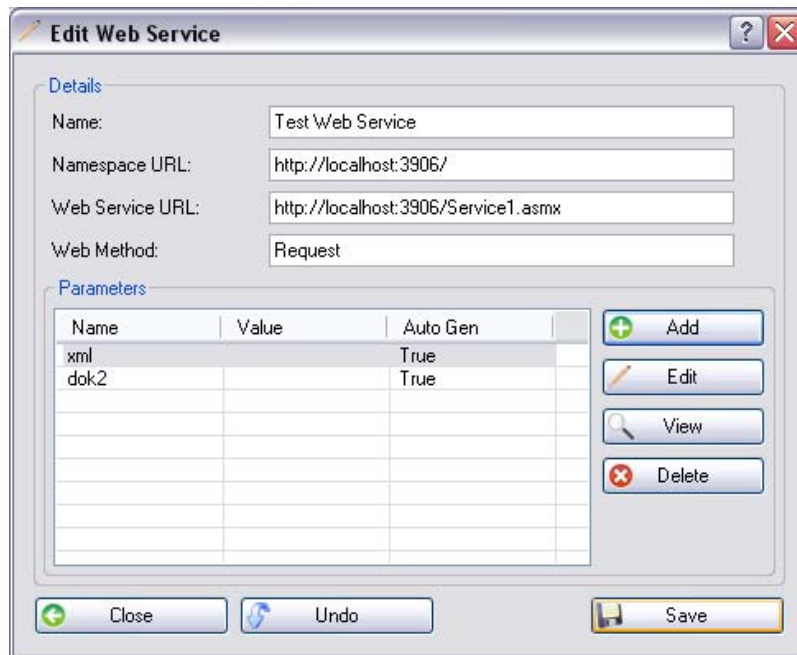


Εικόνα 3.6.5 Περίπτωση που δηλώνεται και η στατική και η δυναμική τιμή

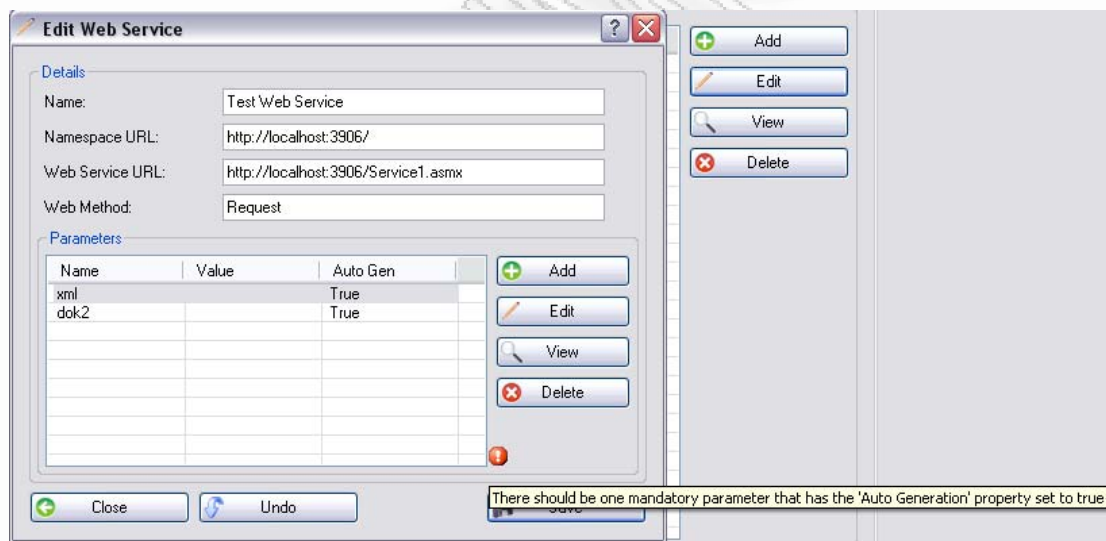


Εικόνα 3.6.6 Περίπτωση που δεν έχει οριστεί δυναμική ή στατική τιμή

Οι παράμετροι μπορεί να είναι όσοι χρειάζονται για να καλεστεί το Web Service. Δεν δίνεται κανένας περιορισμός, παρά μόνο το γεγονός ότι θα παίρνουν στατικές τιμές όπως αναφέρθηκε προηγουμένως. Μπορούμε δηλαδή να επιλέξουμε μόνο μία που θα είναι «Auto Generated». Αυτή θα παίρνει ως τιμή τα δεδομένα του XML αρχείου, που στέλνει ο χρήστης. Αν προσπαθήσουμε να βάλουμε παραπάνω από μία παράμετρο που να παίρνει δυναμική τιμή (Εικόνα 3.6.7), τότε θα λάβουμε μήνυμα λάθους από το πρόγραμμα (Εικόνα 3.6.8).

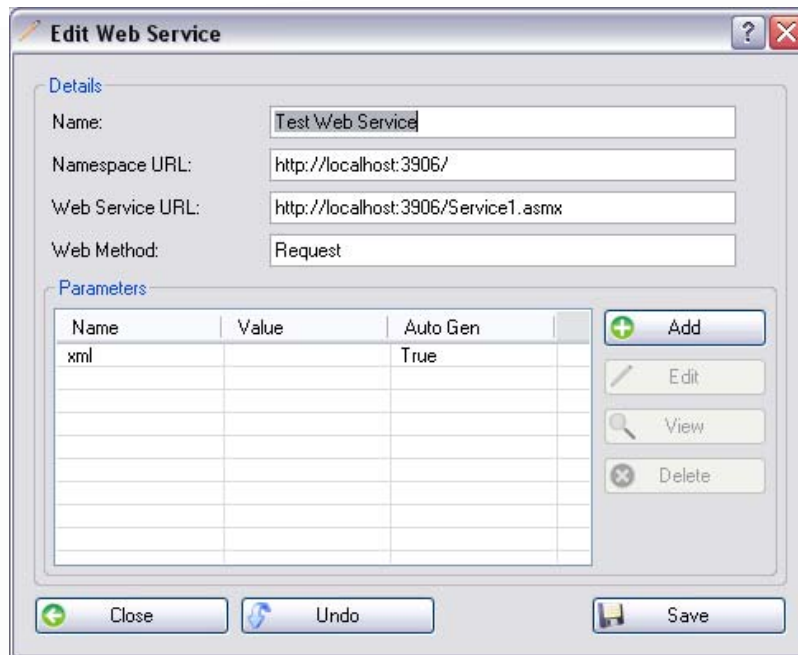


Εικόνα 3.6.7 Ο χρήστης έχει βάλει παραπάνω από μία παραμέτρους με δυναμική τιμή

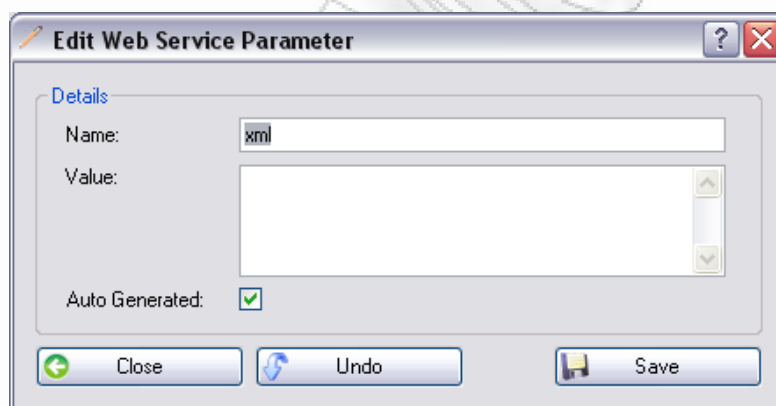


Εικόνα 3.6.8 Περίπτωση που δεν έχει οριστεί δυναμική ή στατική τιμή

Όπως και στη διαχειριστική φόρμα των συνδέσεων έτσι και εδώ μπορεί να γίνει η επεξεργασία ενός Web Service. Κατά την επεξεργασία οι τιμές που έχουν οριστεί σε όλα τα πεδία φορτώνονται από τη βάση του προγράμματος. Ο χρήστης έχει πλήρη ελευθερία στο να αλλάξει ολοκληρωτικά το Web Service (Εικόνα 3.6.9), περιλαμβάνοντας και αλλαγή στις παραμέτρους του (Εικόνα 3.6.10).

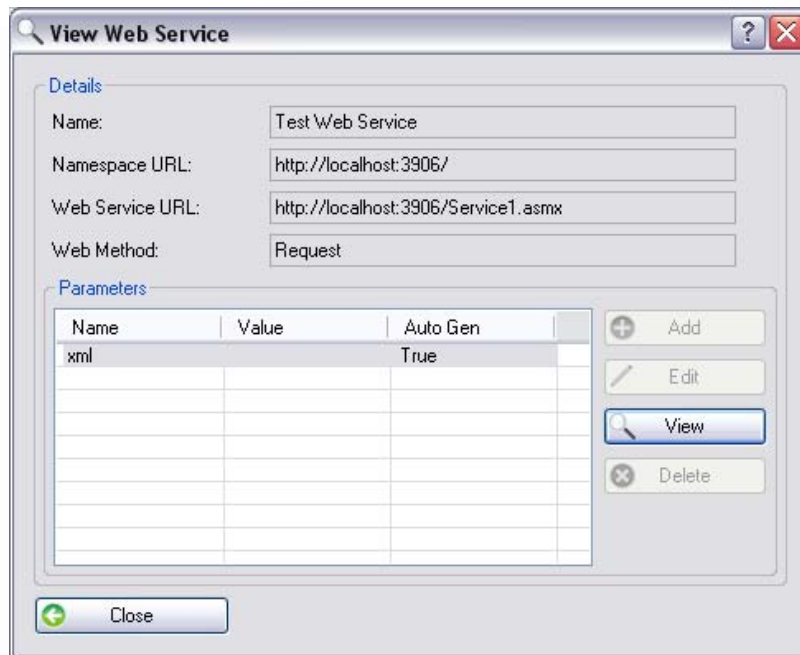


Εικόνα 3.6.9 Επεξεργασία ενός Web Service

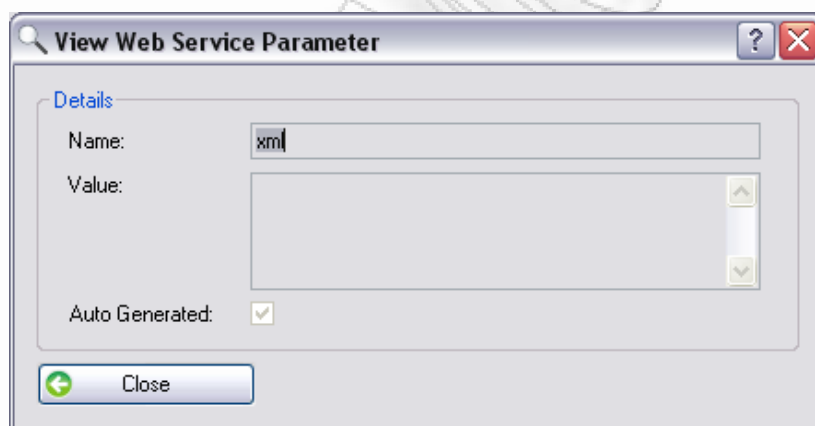


Εικόνα 3.6.10 Επεξεργασία παραμέτρου ενός Web Service

Όπως η επεξεργασία έτσι και η λειτουργία της προβολής υποστηρίζεται από το πρόγραμμα για κάποιο Web Service (Εικόνα 3.6.11). Στη διαδικασία της προβολής όλα τα πεδία είναι απενεργοποιημένα έτσι που ο χρήστης να μην μπορεί να αλλάξει την τιμή τους. Το μόνο ενεργό κουμπί εκτός του «Close», είναι το «View» στις παραμέτρους. Αυτό συμβαίνει για να μπορέσει να ανοίξει τη φόρμα και να δει την παράμετρο καλύτερα (Εικόνα 3.6.12).



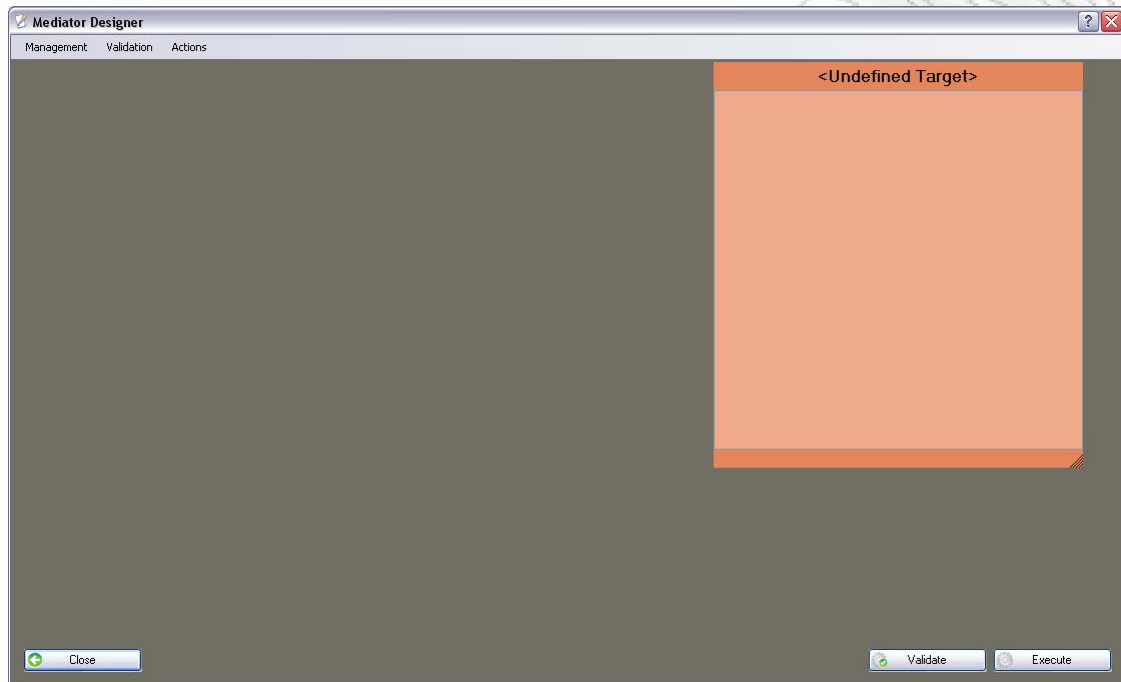
Εικόνα 3.6.11 Προβολή Web Service



Εικόνα 3.6.12 Προβολή παραμέτρου Web Service

3.7 Mediator Designer

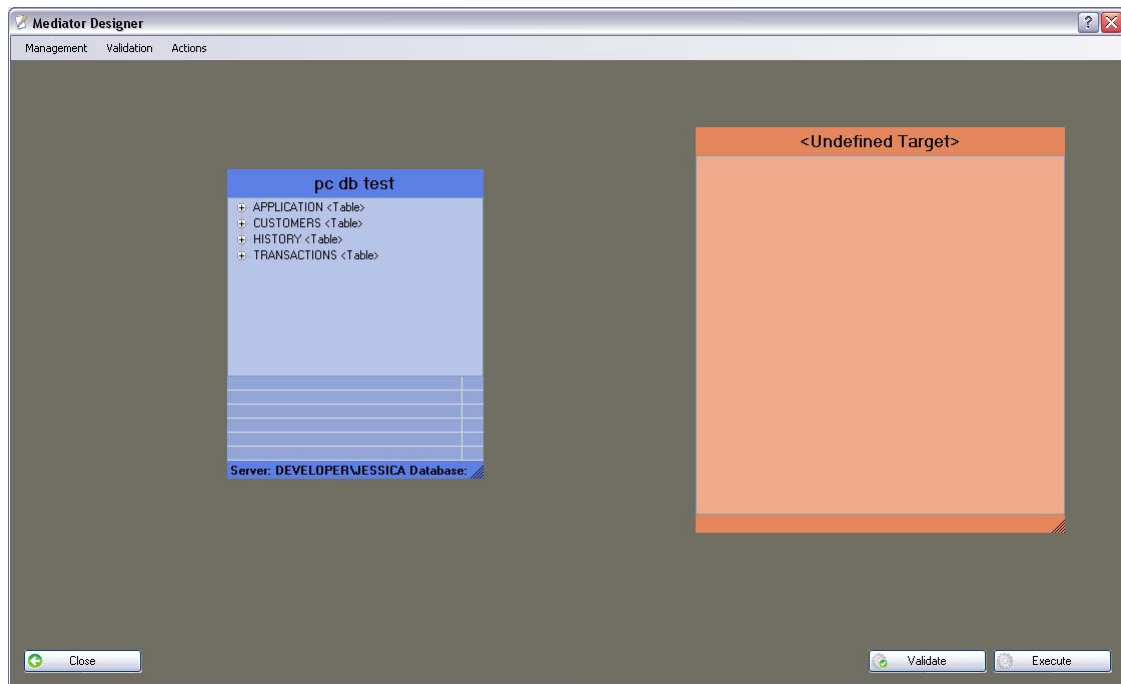
Το μενού αυτό, όπως αναφέραμε, παρέχει μια σχεδιαστική υλοποίηση των συνδέσεων μεταξύ βάσεων και XML αρχείων, με το οποίο μπορούμε να κάνουμε μεταφορά δεδομένων από ένα σχήμα σε άλλο με τη χρήση κριτηρίων. Στην Εικόνα 3.7.1, βλέπουμε την αρχική οθόνη του Mediator Designer.



Εικόνα 3.7.1 Αρχική οθόνη Mediator Designer

Στη συγκεκριμένη οθόνη, όπως φαίνεται, υπάρχει ένα μενού (το οποίο είναι το βασικό μενού επιλογών), κουμπιά στο κάτω μέρος της φόρμας, και ένα πορτοκαλί πλαίσιο στα δεξιά το οποίο αντιπροσωπεύει το τελικό σχήμα που θα ακολουθηθεί κατά τη μεταφορά των δεδομένων. Το τελικό αυτό σχήμα όπως θα δούμε και παρακάτω το ορίζει ο χρήστης και μπορεί να θέσει κριτήρια πάνω στα δεδομένα που θα επιλέξει.

Ο κενός χώρος αντιπροσωπεύει το μέρος στο οποίο μπορεί να ορίσει τα connections του και να πάρει από εκεί δεδομένα, για το τελικό του σχήμα. Τα Connections και ο τρόπος χρήσης τους θα παρουσιαστεί σε επόμενη ενότητα (*Add/Set/Edit/View/Remove Connection*). Ένα παράδειγμα ενός connection φαίνεται στην Εικόνα 3.7.2. Στο connection αυτό ο χρήστης έχει κάποιους πίνακες τους οποίους μπορεί να χρησιμοποιήσει για να πραγματοποιήσει τη μεταφορά δεδομένων.

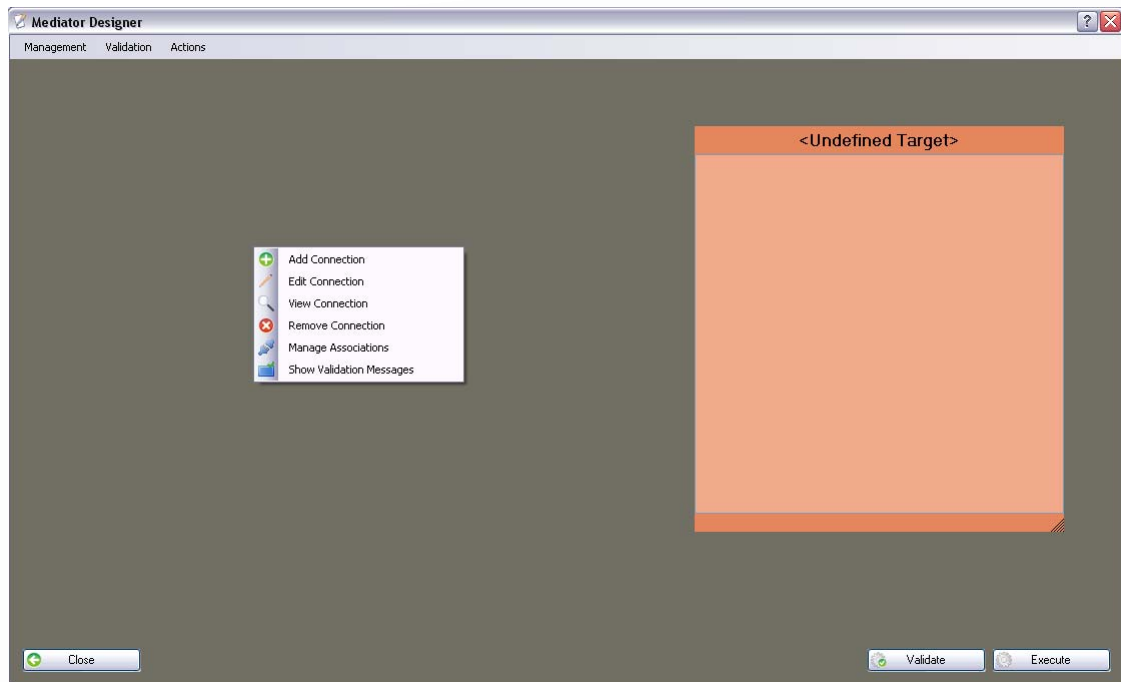


Εικόνα 3.7.2 Παράδειγμα εισαγωγής connection στη σχεδιαστική φόρμα

Η κεντρική οθόνη επίσης περιέχει το βασικό μενού του Designer στο πάνω μέρος της. Το μενού αυτό περιέχει τις ακόλουθες λειτουργίες:

- **Management**
 - Manage Associations
 - Manage Conditions
- **Validation**
 - Validate
 - Show Window
- **Actions**
 - Execute

Οι παραπάνω λειτουργίες είναι διαθέσιμες και μέσα από το μενού επιλογών που εμφανίζεται όταν ο χρήστης πατήσει το δεξί πλήκτρο του Mouse. Ανάλογα όμως με το που θα πατήσει το δεξί πλήκτρο, θα εμφανιστούν και άλλες διαφορετικές επιλογές μαζί με κάποιες από τις προηγούμενες για γρήγορη προσπέλαση. Συγκεκριμένα αν το πατήσει στον κενό χώρο τότε θα εμφανιστούν οι επιλογές που φαίνονται στην Εικόνα 3.7.3.

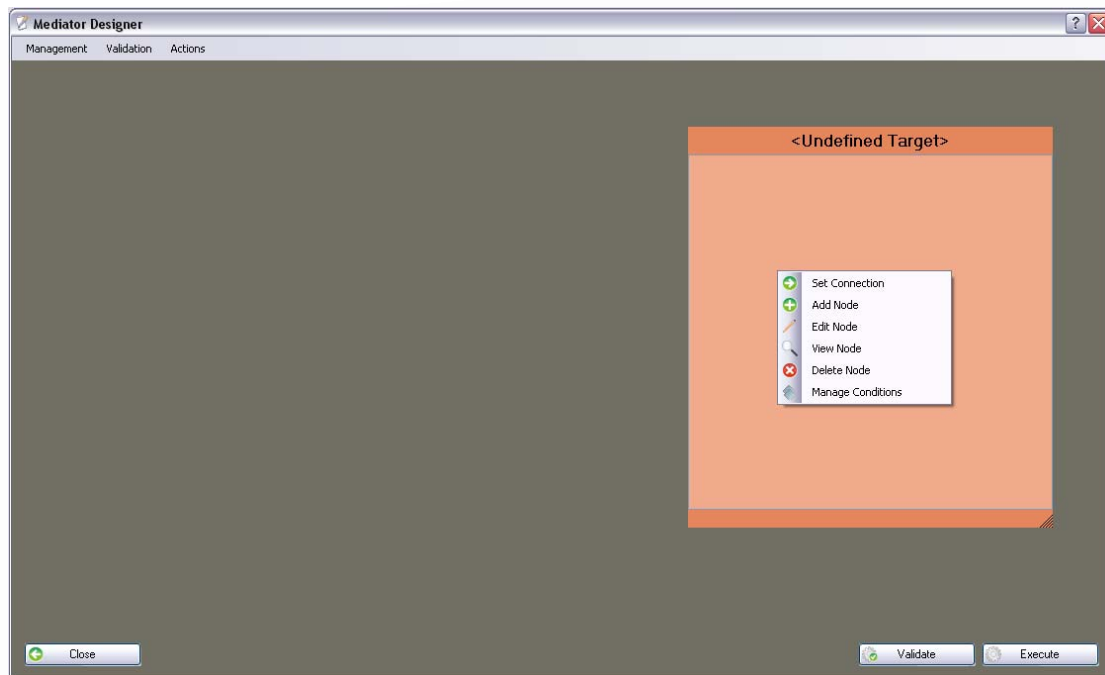


Εικόνα 3.7.3 Επιλογές πατώντας στον κενό χώρο

Οι επιλογές που εμφανίζονται είναι οι παρακάτω:

- Add Connection
- Edit Connection
- View Connection
- Remove Connection
- Manage Associations
- Show Validation Messages

Στην περίπτωση όμως που πατήσει επάνω στο πορτοκαλί χώρο με τίτλο <Undefined Target> (όπως θα δούμε στο χώρο αυτό ορίζεται το τελικό σχήμα το οποίο θα ακολουθήσουν τα δεδομένα που θα μεταφερθούν), εκεί θα έχει ένα άλλο μενού το οποίο παρουσιάζεται στην Εικόνα 3.7.4.



Εικόνα 3.7.4 Επιλογές που εμφανίζονται επάνω στο τελικό σχήμα μεταφοράς

Οι επιλογές που εμφανίζονται είναι οι παρακάτω:

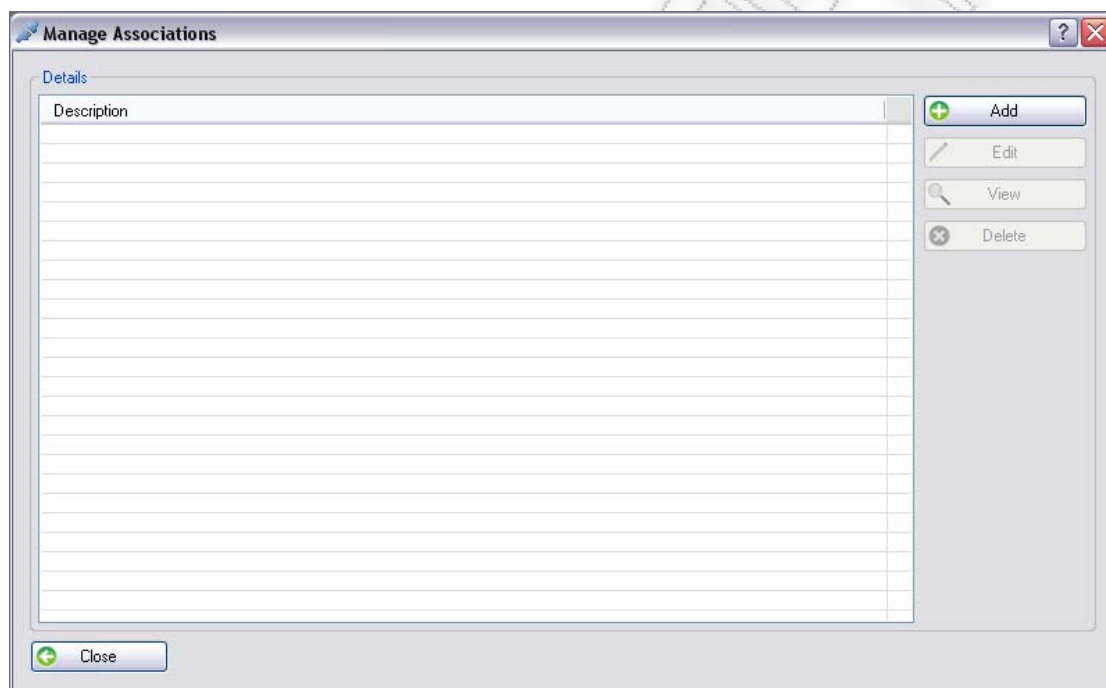
- Set Connection
- Add Node
- Edit Node
- View Node
- Delete Node
- Manage Conditions

Όλες οι λειτουργίες που αναφέρθηκαν παραπάνω θα παρουσιαστούν στις επόμενες ενότητες.

3.7.1 Manage Associations

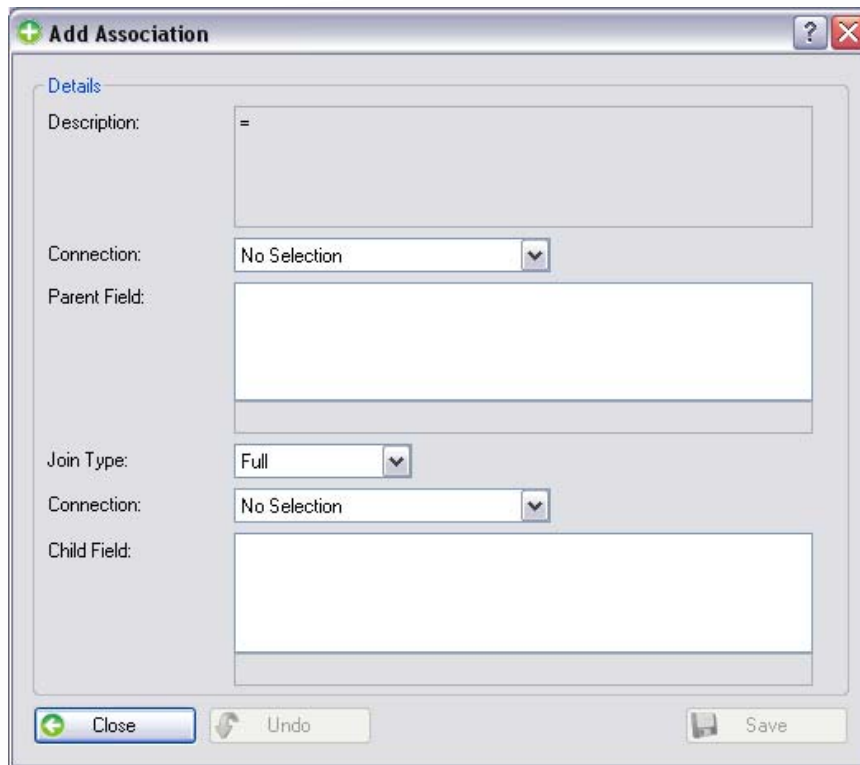
Σε αυτήν τη λειτουργία ο χρήστης μπορεί να ορίσει μία λίστα από συσχετίσεις (associations), μεταξύ των πινάκων που έχει ορίσει στην αρχική φόρμα σχεδιασμού (Εικόνα 3.7.2). Οι συσχετίσεις αυτές θα χρησιμοποιηθούν αργότερα από το πρόγραμμα για να μπορέσει να συνδέσει τους πίνακες μεταξύ τους. Συνήθως αυτή η σύνδεση γίνεται πάνω στο πρωτεύον κλειδί του εκάστοτε πίνακα.

Για να κρατήσει το πρόγραμμα τις συνδέσεις του χρήστη χρησιμοποιεί το ίδιο μοντέλο με τη διαχείριση των Συνδέσεων και των Web Services που είδαμε σε προηγούμενη ενότητα. Η διαχειριστική φόρμα, μαζί με όλες τις προκαθορισμένες ενέργειες (προσθήκη, επεξεργασία, προβολή, διαγραφή) παρουσιάζεται στην Εικόνα 3.7.5.



Εικόνα 3.7.5 Διαχειριστική φόρμα Συσχετίσεων

Κατά την προσθήκη μίας νέας συσχέτισης, ο χρήστης αντιμετωπίζει την οθόνη στην Εικόνα 3.7.6. Εδώ καλείται να επιλέξει ένα από τα connections που έχει ορίσει ήδη στη φόρμα σχεδιασμού, και μέσα από τα 2 αυτά να επιλέξει τον πίνακα και το πεδίο το οποίο θα συσχετιστεί. Επίσης δίνεται η δυνατότητα να επιλέξει και το Join Type. Το Join Type (είδος συσχέτισης) είναι ένα πεδίο επιλογής, στο οποίο επιλέγεται η συσχέτιση για το αν θα είναι «Full», «Left», ή «Right».



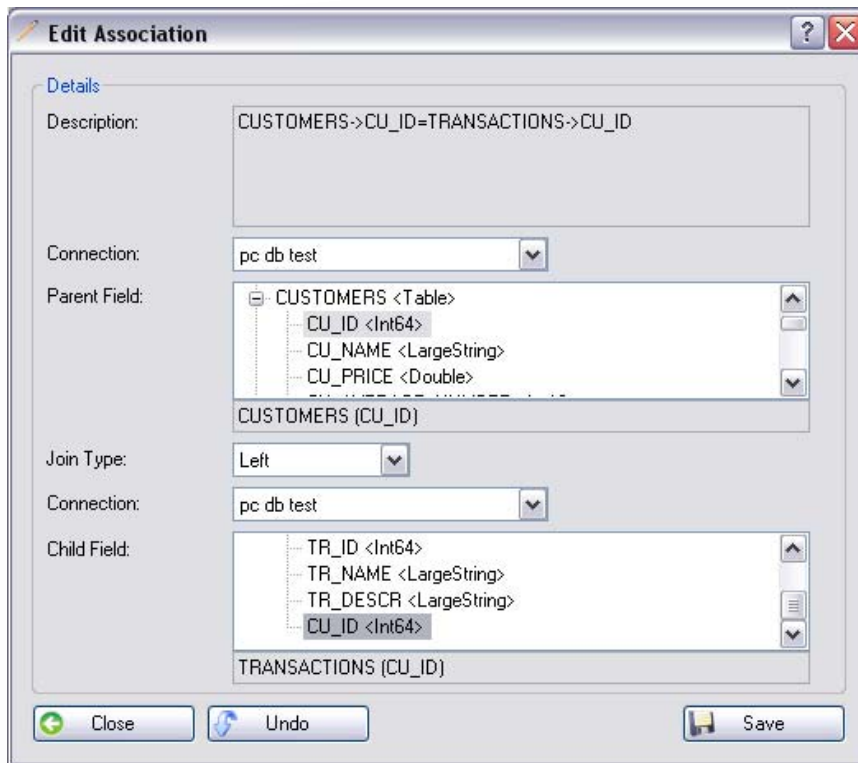
Εικόνα 3.7.6 Προσθήκη νέας συσχέτισης

Αν η συσχέτιση είναι «Full», τότε ισχύουν οι ίδιοι κανόνες με αυτούς στην SQL. Δηλαδή τα πεδία που θα συσχετιστούν θα πρέπει να υπάρχουν και στους δύο πίνακες, για να γίνει εκεί η σύνδεση μεταξύ τους. Αν ο ένας έχει μία εγγραφή επάνω στο συσχετιζόμενο πεδίο, την οποία δεν την έχει ο άλλος, τότε δεν συμπεριληφθεί αυτή στην τελική έξοδο.

Στην περίπτωση όμως που έχουμε «Left» Join, τα δεδομένα που θα προκύψουν από αυτήν τη σύνδεση, θα περιέχουν όλα εκείνα τα πεδία από την πρώτη σύνδεση πατέρα, και όσα από αυτά συνδέονται με τα πεδία του δεύτερου πίνακα, θα επιλεχθούν. Αν όμως δεν υπάρχει κάποιο στον δεύτερο τότε θα χρησιμοποιηθούν κενές τιμές.

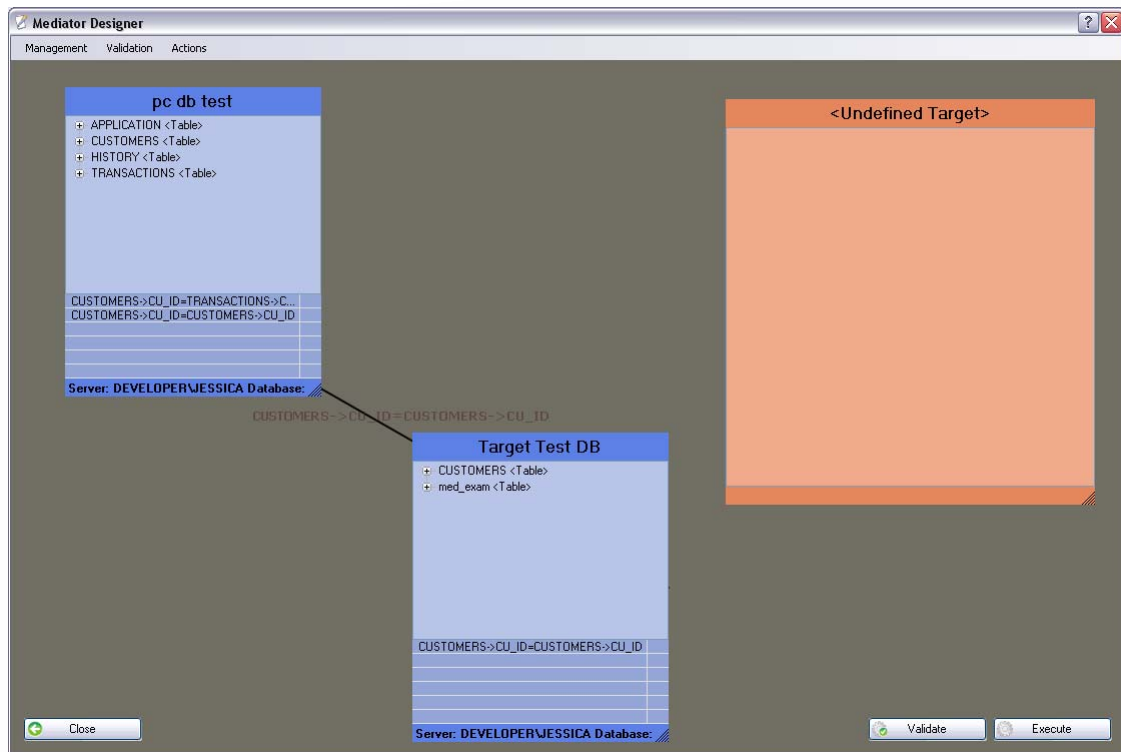
Το ίδιο ισχύει και για το «Right Join», μόνο που εδώ τα υποχρεωτικά πεδία που θα χρησιμοποιηθούν είναι τα πεδία που ορίζονται στο δεύτερο - child connection. Όσο ο χρήστης ορίζει τις επιλογές του, το «Description» το οποίο βρίσκεται στο πάνω μέρος της φόρμας ενημερώνεται, δείχνοντας την τελική συσχέτιση.

Η Εικόνα 3.7.7, δείχνει μία συσχέτιση στο στάδιο της επεξεργασίας όπου έχουν γίνει όλοι οι επιθυμητοί πορισμοί στη φόρμα.



Εικόνα 3.7.7 Επεξεργασία υπάρχουσας συσχέτισης

Εφόσον προστεθούν οι συσχέτισεις, αυτές φαίνονται μετά στη φόρμα σχεδίασης στις συνδέσεις που έχουν εισαχθεί. Αν μία συσχέτιση ενώνει δύο διαφορετικά Connections, τότε εκτυπώνεται επιπλέον και μία γραμμή που συνδέει τις συνδέσεις αυτές. Η Εικόνα 3.7.8, δείχνει τις δύο αυτές περιπτώσεις. Τα «μπλε κουτιά» αντιπροσωπεύουν, όπως θα δούμε και σε επόμενη ενότητα, τις συνδέσεις που έχει επιλέξει ο χρήστης να χρησιμοποιήσει.

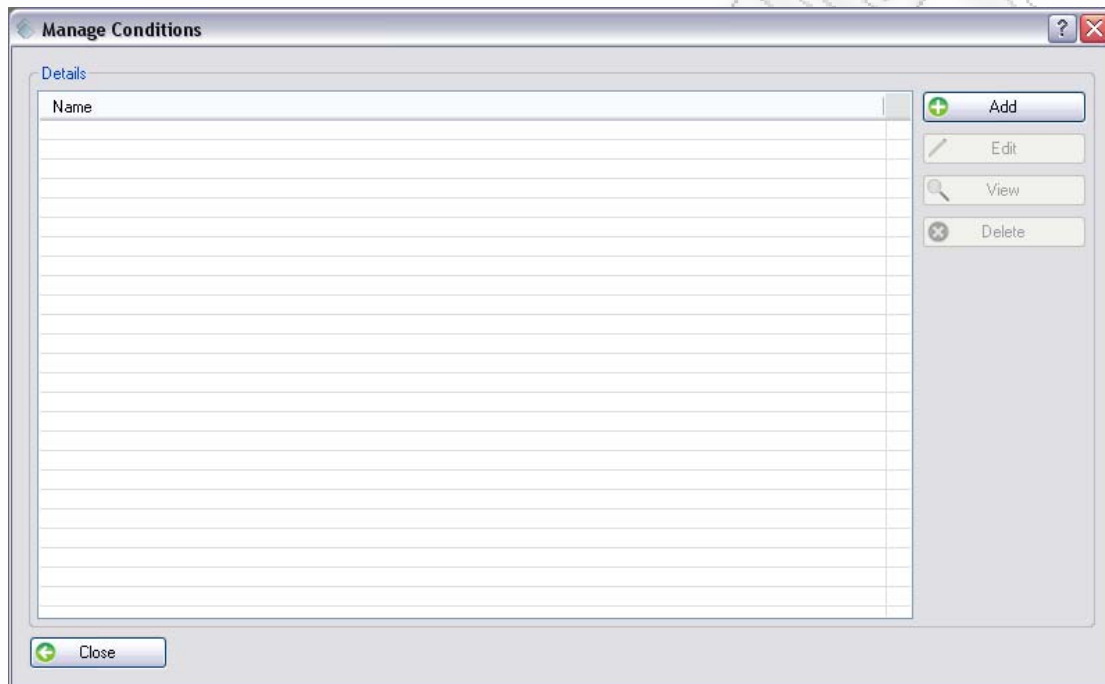


Εικόνα 3.7.8 Εμφάνιση συσχετίσεων στις συνδέσεις της σχεδιαστικής φόρμας

3.7.2 Manage Conditions

Η διαχείριση των Συνθηκών – Conditions βοηθάει στην επιλογή κάποιων δεδομένων ενδιαφέροντος με βάση τα κριτήρια που θα δώσει στη συνθήκη. Ένα παράδειγμα μίας πιθανής συνθήκης είναι να θέλει ο χρήστης να πάρει τις εγγραφές οι οποίες στο πεδίο όνομα ξεκινάνε με το γράμμα «Π». Παραδείγματα τέτοια θα παρουσιαστούν παρακάτω για να γίνει πιο κατανοητή η χρήση της φόρμας.

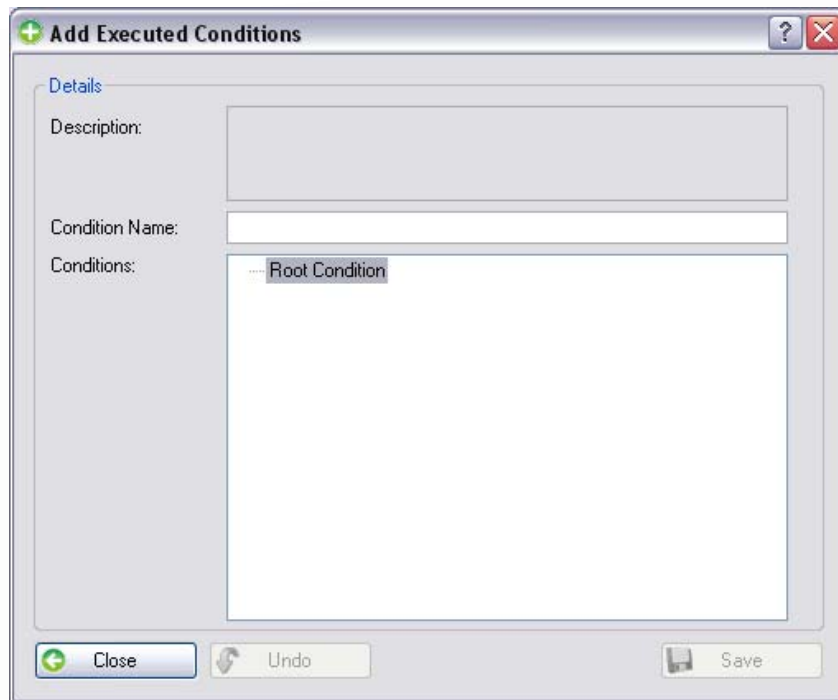
Η Εικόνα 3.7.9, δείχνει τη διαχειριστική φόρμα με την οποία ο χρήστης μπορεί να προσθέσει, να αφαιρέσει, να προβάλει και να διαγράψει κάποια συνθήκη.



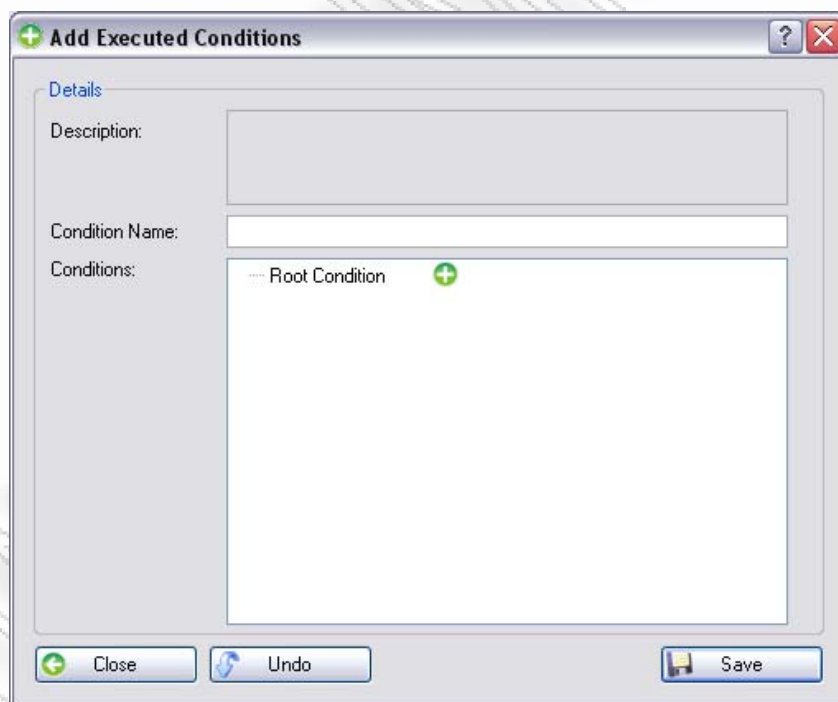
Εικόνα 3.7.9 Φόρμα διαχείρισης συνθηκών

Κατά τη λειτουργία της προσθήκης (Εικόνα 3.7.10), ο χρήστης καλείται να εισάγει ένα όνομα το οποίο θα αντιπροσωπεύει αυτήν τη συνθήκη. Ουσιαστικά, στη διαθέσιμη φόρμα δεν ορίζονται οι συνθήκες καθαυτές, αλλά ένα δέντρο ιεραρχίας μεταξύ απλών συνθηκών. Το δέντρο ιεραρχίας παρέχει τις δυνατότητες προσθήκης των τελεστών «AND» και «OR», ενώ το «Empty» ορίζει ότι η συνθήκη στο τρέχων επίπεδο κλείνει σε εκείνο το σημείο. Εκτενέστερη ανάλυση του τρόπου δημιουργίας της δεντρικής ιεραρχίας θα γίνει παρακάτω.

Έχοντας ορίσει το όνομα του δέντρου ιεραρχίας, ο χρήστης μπορεί να χτίσει τις συνθήκες όπως αυτός τις θέλει. Αυτό για να το κάνει θα πρέπει αρχικά να πατήσει επάνω στο Root Condition. Με την ενέργεια αυτή θα εμφανιστούν κάποια έξτρα κουμπιά λειτουργιών, τα οποία για το root πεδίο είναι μόνο ένα (Εικόνα 3.7.11).



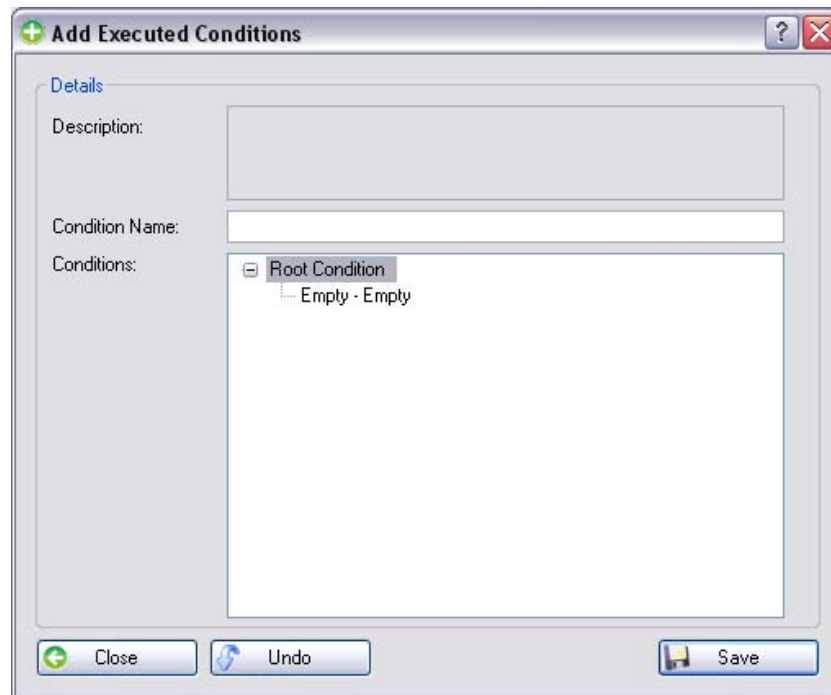
Εικόνα 3.7.10 Φόρμα προσθήκης συνθήκης



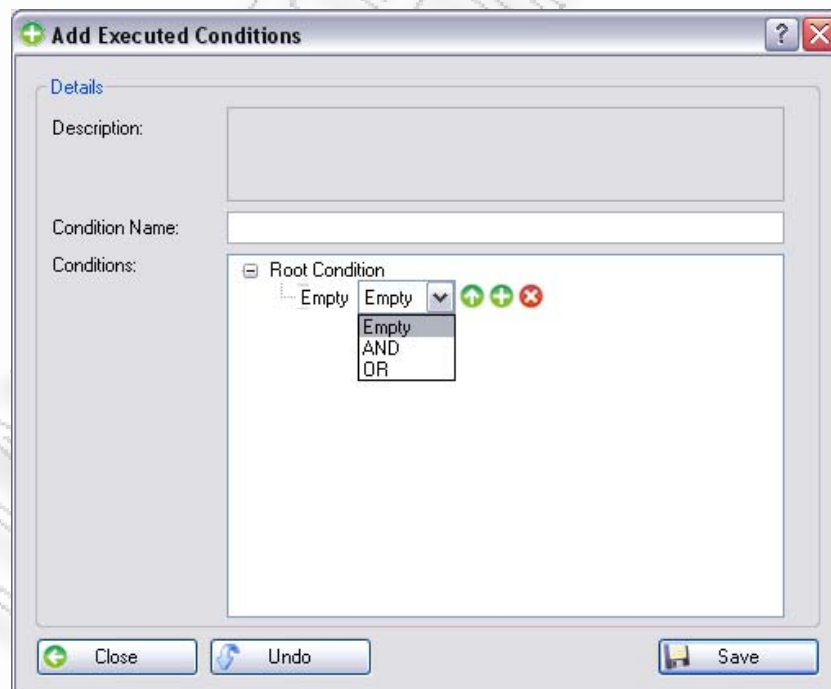
Εικόνα 3.7.11 Πατώντας στο «Root Condition» το κουμπί προσθήκης ενεργοποιείται

Μέσα από το κουμπί προσθήκης ο χρήστης μπορεί να ορίσει όλη τη δομή των συνθηκών με τη χρήση Operators (Τελεστές), And και Or. Αρχικά δημιουργείται ένα κενό Condition, το οποίο δεν έχει ούτε ορισμένη συνθήκη, ούτε τελεστή ορισμένο (Εικόνα 3.7.12). Για να οριστούν αυτά τα πεδία θα πρέπει να επιλέξει την κενή

συνθήκη για να ενεργοποιηθούν και πάλι οι έξτρα λειτουργίες, μόνο που τώρα θα είναι και οι τρεις μαζί και όχι μόνο η προσθήκη (Εικόνα 3.7.13).



Εικόνα 3.7.12 Εισαγωγή κενής συνθήκης



Εικόνα 3.7.13 Έξτρα λειτουργίες της κενής συνθήκης

Οι έξτρα λειτουργίες είναι:

1. Επιλογή τελεστή,
2. Επιλογή συνθήκης (βέλος με φορά προς τα πάνω),
3. Προσθήκη νέας κενής συνθήκης (σύμβολο «+»),
4. Διαγραφή της συνθήκης (σύμβολο «x»)

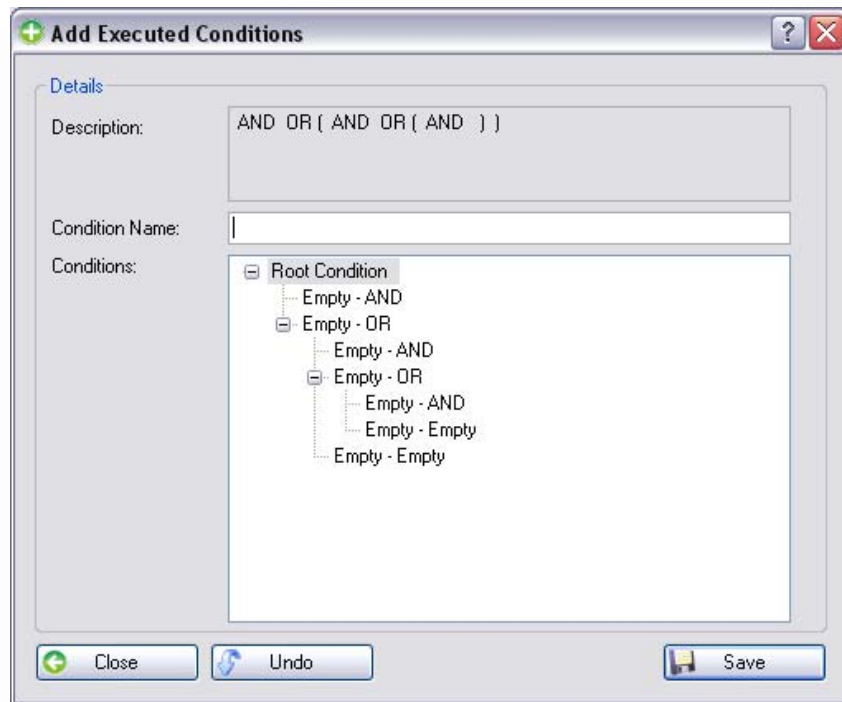
Επιλέγοντας ένα από τους τελεστές «AND» ή «OR», ο χρήστης μπορεί να ορίσει κι άλλες συνθήκες οι οποίες θα συνδέονται με αυτούς τους τελεστές. Ο τρόπος αναπαράστασης στο δέντρο με τις συνθήκες πραγματοποιείται ως εξής. Όταν ο τελεστής είναι ο «AND», τότε δημιουργείται στο δέντρο μία καινούργια κενή συνθήκη, η οποία βρίσκεται στο ίδιο επίπεδο βάθους με την πατρική. Αν όμως ο τελεστής είναι ο «OR», τότε δημιουργείται μία κενή συνθήκη ως παρακλάδι σε, κατά ένα, μεγαλύτερο επίπεδο. Αν δηλαδή η πατρική συνθήκη βρίσκεται στο επίπεδο 1, τότε με τη χρήση του τελεστή «OR», η νέα θα προστεθεί κάτω από την πατρική σε επίπεδο 2 (Εικόνα 3.7.14).

```

Συνθήκη(α) AND - Επίπεδο 1
Συνθήκη(β) OR - Επίπεδο 1
    Συνθήκη(γ) AND - Επίπεδο 2
    Συνθήκη(δ) OR - Επίπεδο 2
        Συνθήκη(ε) Empty - Επίπεδο 3
  
```

Εικόνα 3.7.14 Επίπεδα ιεραρχίας με βάση τους τελεστές

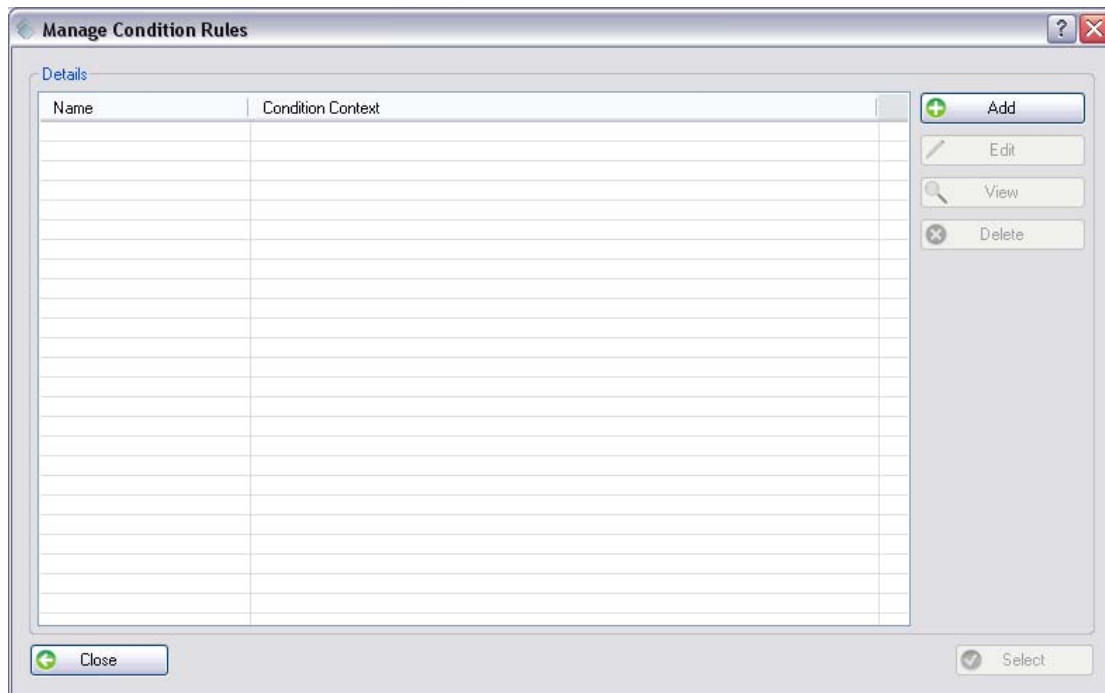
Η χρήση του κενού τελεστή «Empty», πραγματοποιείται όταν πρέπει να υποδείξουμε ότι σε αυτό το επίπεδο στο συγκεκριμένο κλαδί ιεραρχίας οι συνθήκες τερματίζουν. Έτσι και στην Εικόνα 3.7.14, η Συνθήκη (ε), δεν έχει άλλη από κάτω της και για αυτό το λόγο τερματίζει με τελεστή «Empty». Έτσι, για παράδειγμα, θα μπορούσαμε να φτιάξουμε το ακόλουθο δέντρο ιεραρχίας με συνθήκες, το οποίο παρουσιάζεται στην Εικόνα 3.7.15. Για μεγαλύτερη διευκόλυνση υπάρχει και το πεδίο «Description» στο πάνω μέρος της φόρμας, έτσι ώστε να φαίνεται η τελική μορφή σύνδεσης των συνθηκών με παρενθέσεις.



Εικόνα 3.7.15 Παράδειγμα δέντρου ιεραρχίας συνθηκών

Η χρήση του κουμπιού διαγραφής (Εικόνα 3.7.13), είναι προφανής με μοναδικό σκοπό να διαγράψει τη συνθήκη που μόλις δημιούργησε ο χρήστης. Αν η συνθήκη έχει και άλλα παρακλάδια – παιδιά τότε θα χαθούν και αυτά.

Η διαδικασία επιλογής συνθήκης, η οποία θα γεμίσει κάποια άδεια ή συμπληρωμένη στο δέντρο συνθηκών, γίνεται μέσα από το βέλος επιλογής που αναφέρθηκε νωρίτερα, το οποίο βρίσκεται στις έξτρα λειτουργίες. Κατά τη χρήση του δίνεται η δυνατότητα να επιλεγεί μία από τις ήδη ορισμένες απλές συνθήκες, που θα ορίσουμε ή που έχουμε ήδη ορίσει. Η διαδικασία αυτή περιλαμβάνει μία χειριστική φόρμα για την προσθήκη, επεξεργασία, προβολή, διαγραφή και επιλογή μία απλής συνθήκης (Εικόνα 3.7.16).

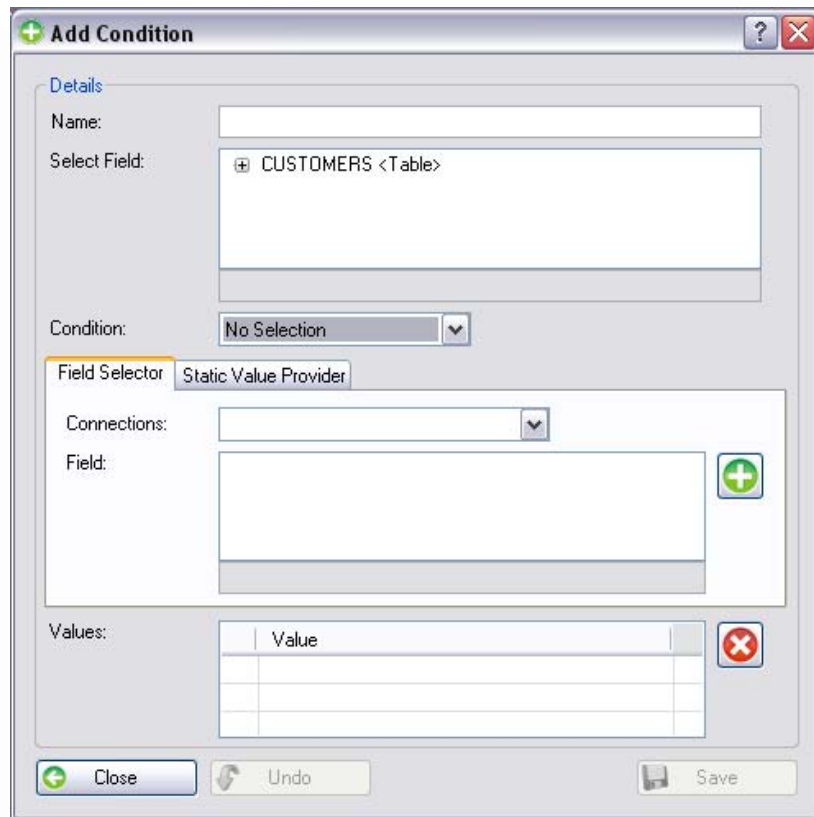


Εικόνα 3.7.16 Φόρμα διαχείρισης συνθηκών

Στην προσθήκη νέας συνθήκης το πρόγραμμα εμφανίζει τη φόρμα της Εικόνα 3.7.17. Εκεί μπορούν να οριστούν όλα τα απαραίτητα στοιχεία τα οποία θα συνθέσουν μία απλή συνθήκη.

Τα στοιχεία αυτά είναι:

1. Όνομα συνθήκης,
2. Επιλογή πεδίου πάνω στο οποίο θα εφαρμοστεί η συνθήκη,
3. Επιλογή είδος συνθήκης (π.χ. ισότητα, ή ισότητα μέρους χαρακτήρων, κ.λ.π.),
4. Επιλογή του πεδίου το οποίο θα ανήκει στο δεύτερο μέρος της συνθήκης ή
5. Εισαγωγή στατικής τιμής για τη συνθήκη.



Εικόνα 3.7.17 Προσθήκη νέας απλής συνθήκης

Τα διαθέσιμα είδη συνθηκών που υπάρχουν για χρήση από το σύστημα είναι τα ακόλουθα:

1. is null,
2. equal to,
3. equal to a part(middle),
4. equal to (defined start string),
5. equal to (defined end string),
6. less than,
7. less than or equals,
8. more than,
9. more than or equals,
10. is mandatory,
11. not equal to

is null

Η χρήση του πραγματοποιείται όταν θέλουμε να ελέγξουμε ότι το πεδίο δεν έχει τιμή και είναι κενό δηλ. «NULL».

equal to

Με αυτό το είδος ο χρήστης μπορεί να πραγματοποιήσει την απλή ισότητα μεταξύ ενός πεδίου και ενός άλλου ή μίας στατικής τιμής.

equal to a part(middle)

Εδώ ο χρήστης μπορεί να ελέγξει αν το συγκεκριμένο πεδίο περιέχει την τιμή που ορίζεται στατικά ή σε άλλο πεδίο. Αν για παράδειγμα υπάρχει ανάγκη να επιστραφούν οι εγγραφές που περιέχουν στο όνομα του χρήστη τον χαρακτήρα «ρ», τότε θα επιστραφούν δεδομένα (αν υποθέσουμε ότι υπάρχουν οι παρακάτω εγγραφές), με ονόματα, για παράδειγμα Χριστίνα, Κατερίνα, Χρήστος, Αφροδίτη, Γεωργία, αλλά όχι το όνομα Ευαγγελία, ή Θανάσης, γιατί αυτά δεν το περιέχουν.

equal to (defined start string)

Σε αυτήν την περίπτωση θα επιστραφούν όσα ξεκινάνε με την τιμή που θα ορίσει ο χρήστης. Για παράδειγμα έστω ότι δώσει το γράμμα «Χρ», τότε θα επιστραφούν τα ονόματα Χρήστος, Χριστίνα, Χρυσόστομος, κ.λπ., δεν θα επιστραφούν όλα τα υπόλοιπα ονόματα τα οποία δεν περιέχουν στην αρχή τους συγκεκριμένους χαρακτήρες.

equal to (defined end string)

Αυτή η περίπτωση είναι παρόμοια με την προηγούμενη, μόνο που τώρα ελέγχονται οι τελευταίοι χαρακτήρες και όχι οι αρχικοί.

less than

Εδώ ελέγχεται αν το αρχικό πεδίο είναι μικρότερο από το τελικό, για παράδειγμα $Price < 5$.

less than or equals

Παρόμοια με το προηγούμενο μόνο που τώρα ελέγχεται και η ισότητα. Δηλαδή $Price \leq 5$.

more than

Σε αυτήν την περίπτωση ελέγχεται αν το πεδίο είναι μεγαλύτερο από κάποια τιμή. Ένα παράδειγμα είναι το $Price > 10$.

more than or equals

Παρόμοια με την παραπάνω περίπτωση μόνο που τώρα ελέγχεται και η ισότητα. Δηλαδή έχουμε $Price \geq 10$.

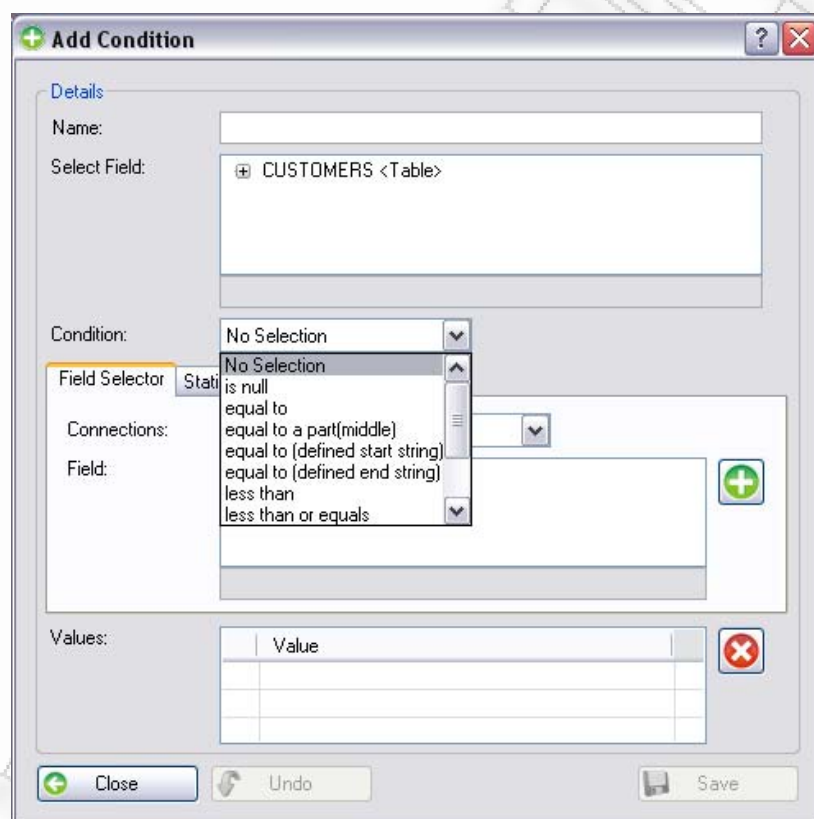
is mandatory

Εδώ ελέγχεται αν το πεδίο είναι υποχρεωτικό δηλ. αν έχει τιμή, και επιστρέφει την εγγραφή του.

not equal to

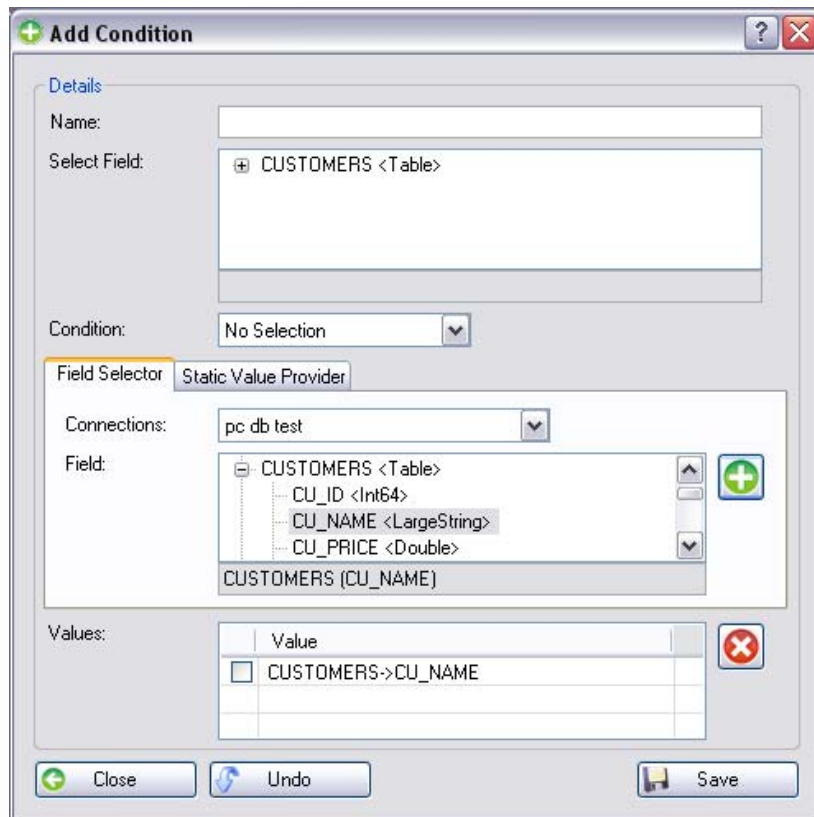
Εδώ ορίζεται το αντίθετο της ισότητας. Αν δηλαδή το πεδίο **δεν** είναι ίσο με την τιμή που δίνεται. Για παράδειγμα ID \neq 5.

Η Εικόνα 3.7.18, δείχνει το σημείο επιλογής του είδους του ελέγχου της συνθήκης.

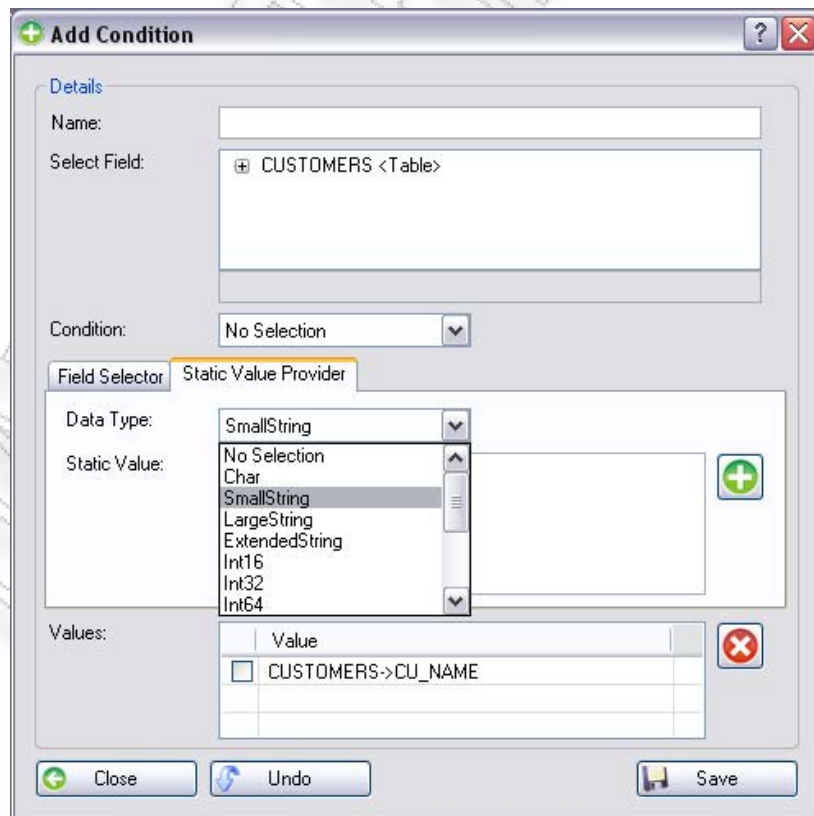


Εικόνα 3.7.18 Επιλογή είδους ελέγχου συνθήκης

Όπως ήδη αναφέρθηκε, η τιμή που μπορεί να χρησιμοποιηθεί μπορεί να είναι είτε κάποιο άλλο πεδίο από τους πίνακες (Εικόνα 3.7.19), είτε κάποια στατική τιμή (Εικόνα 3.7.20). Στη δεύτερη περίπτωση θα πρέπει να οριστεί και το είδος της στατικής τιμής. Το είδος της στατικής τιμής **θα πρέπει** να συμφωνεί με το είδος της τιμής του αρχικού πεδίου. Για παράδειγμα δε μπορεί ο χρήστης να ελέγξει ένα πεδίο που είναι νούμερο αν περιλαμβάνει τους χαρακτήρες «Πα» σαν τιμές. Σε αυτήν την περίπτωση θα προκύψει σφάλμα κατά την εκτέλεση της συνθήκης.



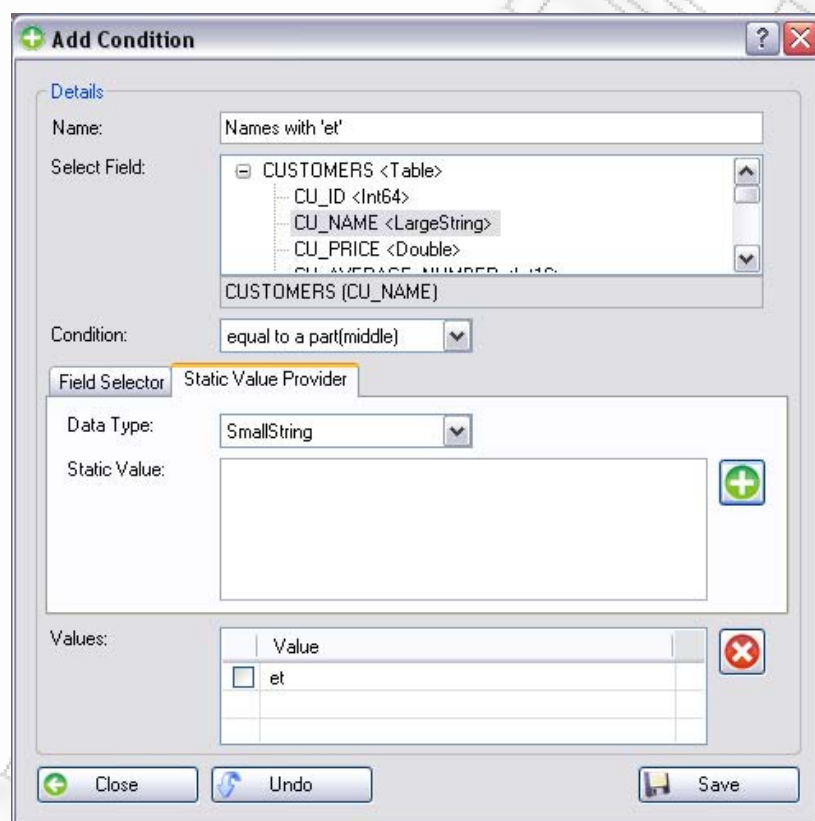
Εικόνα 3.7.19 Επιλογή πεδίου για τη σύγκρισή του με το αρχικό



Εικόνα 3.7.20 Επιλογή στατικής τιμής για τη σύγκρισή της με το αρχικό πεδίο

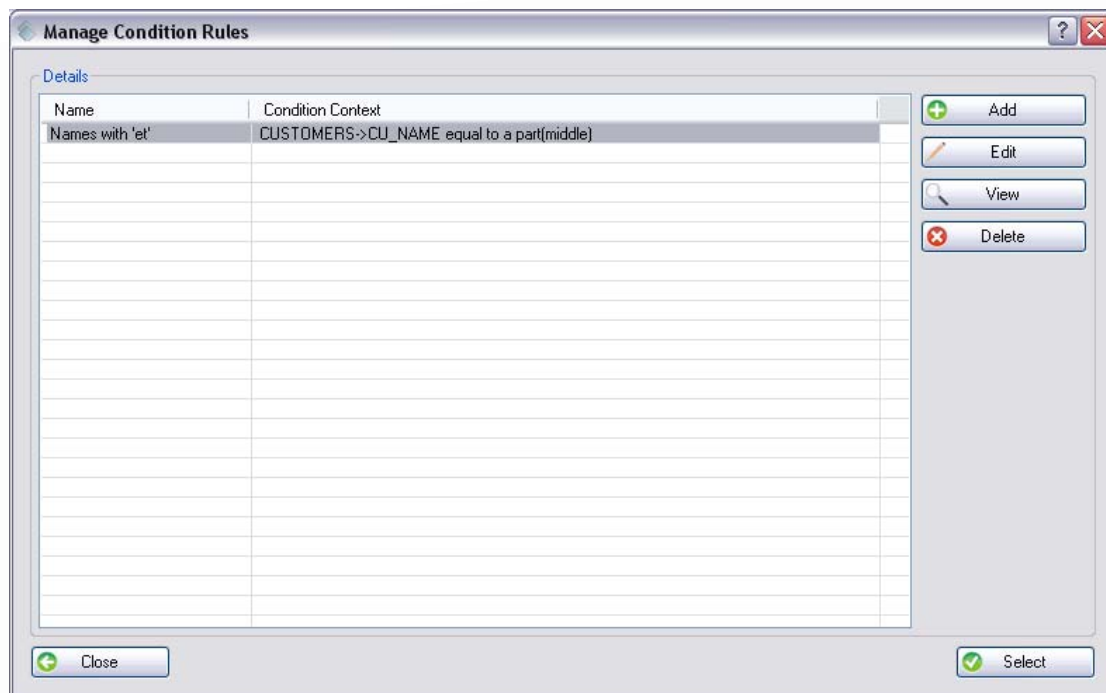
Εφόσον έχει επιλέξει όλα τα πεδία για να ορίσει τη συνθήκη, θα πρέπει να προσθέσει το δεύτερο πεδίο ή τη στατική τιμή, σαν επιλεγμένες τιμές. Για να γίνει αυτό είναι απαραίτητο να χρησιμοποιήσει το κουμπί προσθήκης που δίνεται κάτω από τον ορισμό της τιμής και στις δύο περιπτώσεις. Μετά την προσθήκη η επιλεγμένη τιμή θα εμφανιστεί στη λίστα που παρέχεται στο κάτω μέρος τη φόρμας (Εικόνα 3.7.21).

Ο χρήστης μπορεί να ορίσει παραπάνω από μία τιμές για το ίδιο πεδίο και για το ίδιο είδος συνθήκης. Όλες αυτές θα ελεγχθούν μεταξύ τους με τελεστή «OR». Αν δηλαδή ο χρήστης δώσει τη συνθήκη το CU_ID equals to 5 και 10, τότε αυτό θα μεταφραστεί σε δύο υπο-συνθήκες οι οποίες θα ελέγχουν το πεδίο CU_ID να είναι 5 ή να είναι 10.

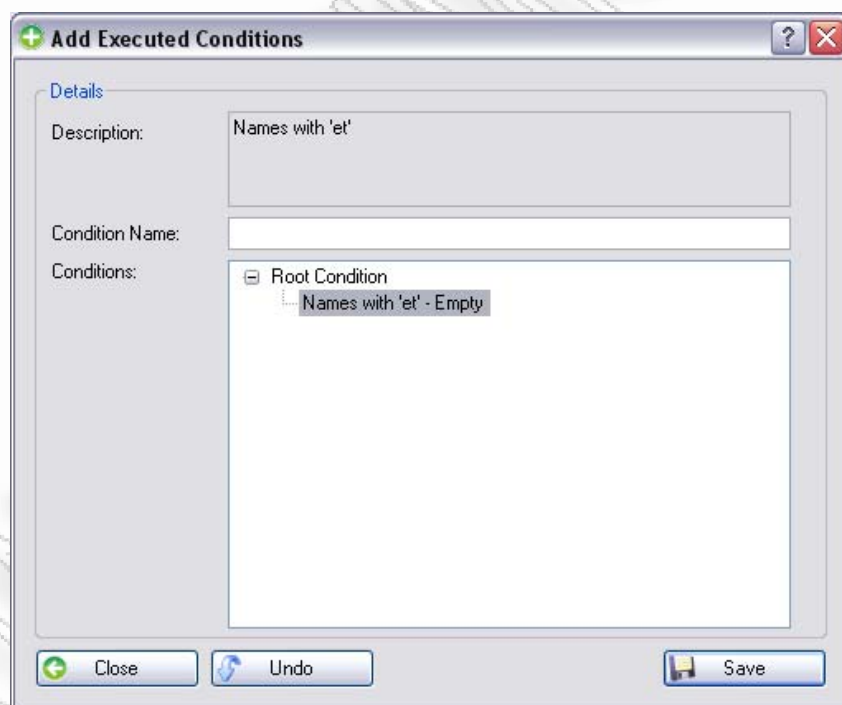


Εικόνα 3.7.21 Παράδειγμα ορισμού νέας συνθήκης

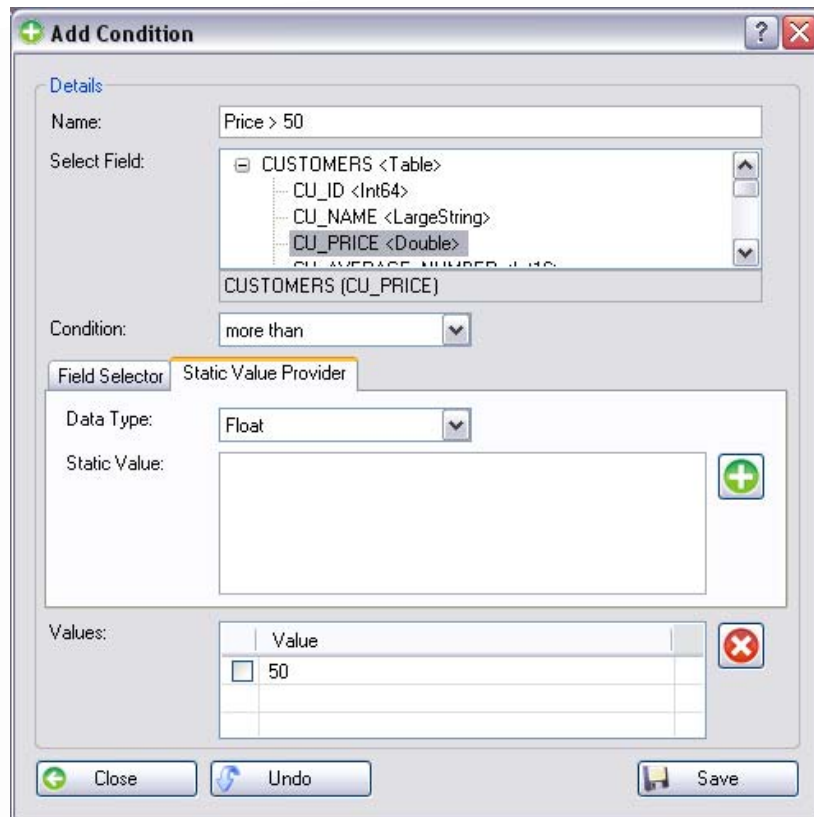
Εφόσον αποθηκεύσει τη συνθήκη αυτή, θα εμφανιστεί στη λίστα διαχείρισης (Εικόνα 3.7.22). Από εκεί μπορεί να την επιλέξει και να τη χρησιμοποιήσει στο δέντρο ιεραρχίας συνθηκών (Εικόνα 3.7.23). Έτσι μπορεί να ορίσει πολλές τέτοιες συνθήκες που θα προστίθενται στο δέντρο ιεραρχίας συνθηκών μέχρι να φτιάξει το τελικό σχήμα συνθήκης. Έστω για παράδειγμα ότι ο χρήστης συνεχίζει και ορίζει και άλλη συνθήκη (Εικόνα 3.7.24), αυτή την προσθέτει στο δέντρο, και έχει το αποτέλεσμα της Εικόνα 3.7.25.



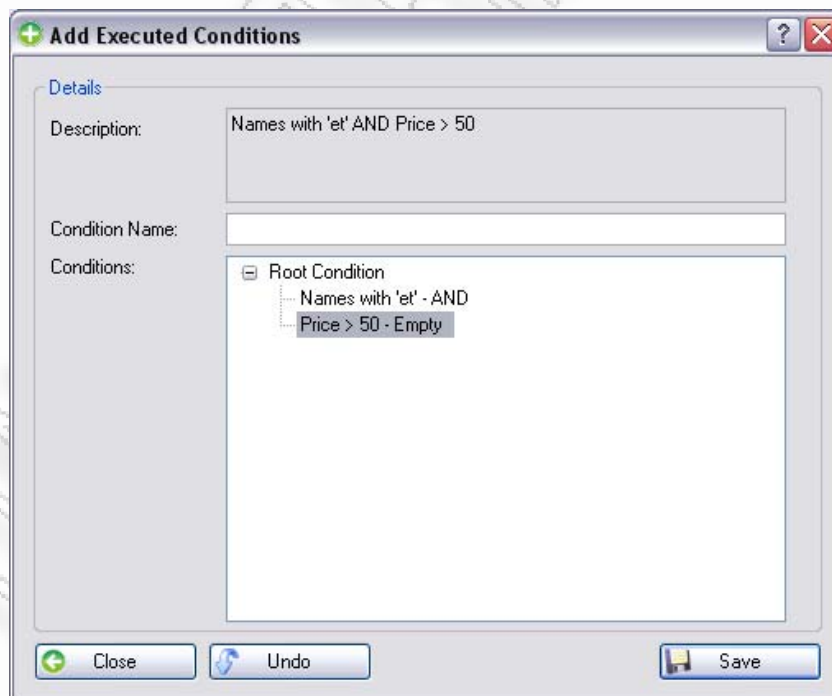
Εικόνα 3.7.22 Επιλογής της συνθήκης από τη λίστα διαχείρισης



Εικόνα 3.7.23 Χτίσιμο του δέντρου συνθηκών



Εικόνα 3.7.24 Προσθήκη δεύτερης συνθήκης



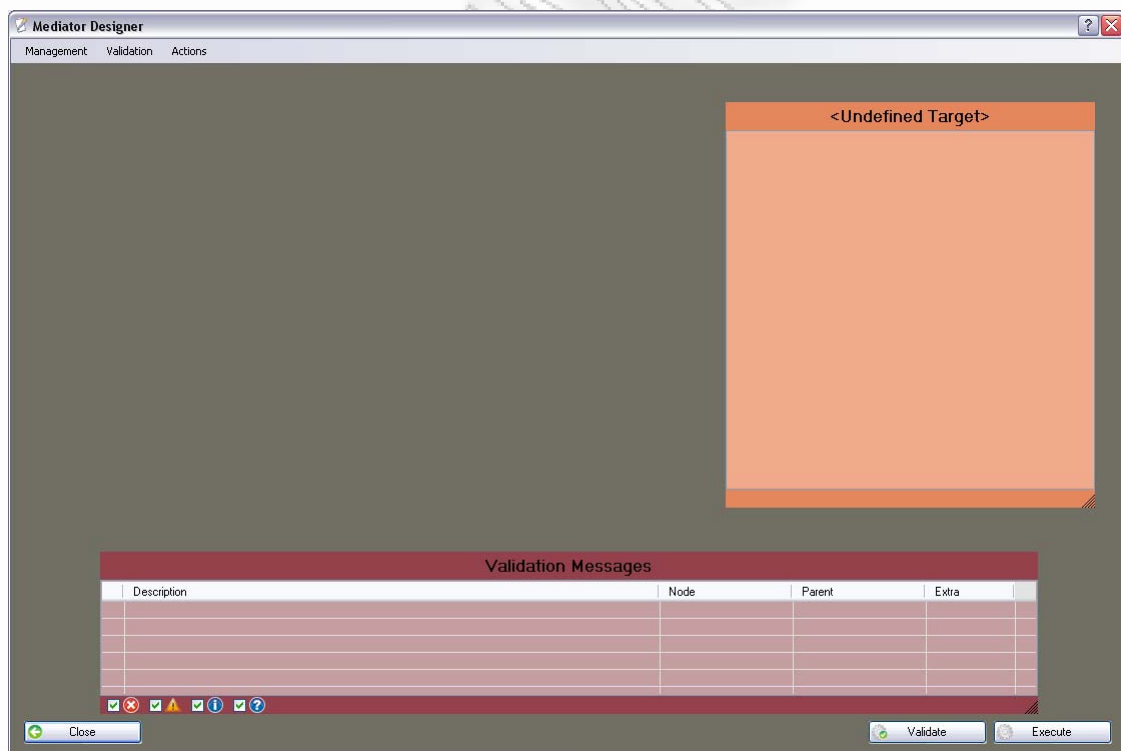
Εικόνα 3.7.25 Το δέντρο ιεραρχίας συνθηκών με δύο συνθήκες

3.7.3 Validation

Η λειτουργία του validation χρησιμοποιείται για να εμποδίσει το χρήστη να κάνει κάποια λάθη τα οποία μπορούν να ελεγχθούν από το πρόγραμμα, σε ένα βήμα πριν τη χρησιμοποίηση του τελικού σχήματος. Η λειτουργία όμως αυτή δεν δείχνει μόνο τα λάθη. Δείχνει επιπλέον όλες τις πληροφορίες που προκύπτουν μετά τον έλεγχο του τελικού σχήματος. Τα διαθέσιμα είδη πληροφοριών είναι τα ακόλουθα:

1. Info (απλή πληροφόρηση – για παράδειγμα),
2. Warning (προειδοποιητικό μήνυμα),
3. Error (μήνυμα λάθους),
4. Question (μήνυμα ερώτησης)

Όταν αυτά υπάρχουν και εμφανίζονται στην αντίστοιχη φόρμα, δίνεται η δυνατότητα να γίνει φιλτράρισμα αυτών. Μπορεί, δηλαδή, ο χρήστης να φιλτράρει τις πληροφορίες για να δει, για παράδειγμα, μόνο τα μηνύματα λάθους και τα μηνύματα προειδοποίησης. Αυτές οι δυνατότητες παρέχονται επάνω στη φόρμα επικύρωσης η οποία εμφανίζεται μαζί με τα αρχικά και τελικά σχήματα (Εικόνα 3.7.26, Εικόνα 3.7.27).



Εικόνα 3.7.26 Φόρμα επικύρωσης (α)

Description	Node	Parent	Extra

Εικόνα 3.7.27 Φόρμα επικύρωσης (β)

Από τη φόρμα αυτή μπορούμε να δούμε ότι οι πληροφορίες που εμφανίζονται είναι οι ακόλουθες:

1. Description – Το μήνυμα περιγράφεται σε αυτό το πεδίο,
2. Node – Ο κόμβος (π.χ. στήλη), στον οποίο εμφανίστηκε το πρόβλημα,
3. Parent – Ο πατέρας του κόμβου που περιέχει το πρόβλημα,
4. Extra – Έξτρα πληροφορίες που μπορεί να βοηθήσουν το χρήστη.

Τα μηνύματα που εμφανίζονται σε αυτήν τη στήλη είναι:

- Table Valid (Ο πίνακας υπάρχει και μπορεί να χρησιμοποιηθεί),
- Table Invalid (Ο πίνακας δεν υπάρχει και θα δημιουργηθεί αν αυτό είναι που επιθυμεί ο χρήστης),
- Column Valid (Η στήλη υπάρχει και έχει το σωστό τύπο δεδομένων π.χ. είναι τύπου ακεραίου),
- Column Invalid (Η στήλη δεν είναι έγκυρη γιατί στο τελικό σχήμα η ίδια στήλη έχει διαφορετικό τύπο δεδομένων),
- Column Extra (Η στήλη που όρισε ο χρήστης δεν υπάρχει στο τελικό σχήμα και θα δημιουργηθεί),
- Column Less (Υπάρχουν στήλες που υπάρχουν στη βάση δεδομένων αλλά δεν έχουν χρησιμοποιηθεί στο τελικό σχήμα).

Τα φίλτρα που προαναφέραμε, υπάρχουν στο κάτω μέρος της φόρμας, προσφέροντας στο χρήστη τη δυνατότητα του φιλτραρίσματος. Έτσι, αν για παράδειγμα έχουμε την επόμενη περίπτωση με σφάλματα (Εικόνα 3.7.28), επιλέγοντας από τα check boxes μόνο τα μηνύματα σφάλματος και προειδοποίησης, η φόρμα θα φιλτράρει και θα δείξει μόνο αυτά (Εικόνα 3.7.29).

Validation Messages			
Description	Node	Parent	Extra
Table does not exist and it will be created.			TableInvalid
Column CU_ID has Int64 data type but the source column has ExtendedString.	CU_ID	CUSTOMERS	ColumnInvalid
Column CU_NAME is Valid.	CU_NAME	CUSTOMERS	ColumnValid
Column CU_PRICE has Double data type but the source column has Extended...	CU_PRICE	CUSTOMERS	ColumnInvalid
Column CUSTOMERS_Id is an extra column in the Table CUSTOMERS. It will ...	CUSTOMERS_Id	CUSTOMERS	ColumnExtra
In a DB Connection you must set the root Field: 'THIRD_LEVEL' type of <Table>.			
In a DB Connection you must set the root Field: 'CUSTOMERS' type of <Table>.			

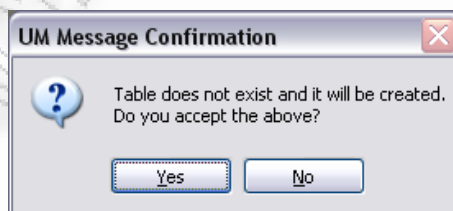
Εικόνα 3.7.28 Περίπτωση μηνυμάτων με όλα τα φίλτρα ενεργά

Validation Messages			
Description	Node	Parent	Extra
Column CU_ID has Int64 data type but the source column has ExtendedString.	CU_ID	CUSTOMERS	ColumnInvalid
Column CU_NAME has LargeString data type but the source column has Exten...	CU_NAME	CUSTOMERS	ColumnInvalid
Column CU_PRICE has Double data type but the source column has Extended...	CU_PRICE	CUSTOMERS	ColumnInvalid
In a DB Connection you must set the root Field: 'THIRD_LEVEL' type of <Table>.			
In a DB Connection you must set the root Field: 'CUSTOMERS' type of <Table>.			

Εικόνα 3.7.29 Περίπτωση μηνυμάτων με δύο ενεργά φίλτρα

Σε μηνύματα τύπου ερωτήματος, ο χρήστης καλείται να επικυρώσει το ερώτημα που του παρατίθεται στη φόρμα. Ένα παράδειγμα τέτοιου ερωτήματος είναι όταν δεν υπάρχει ο ορισμένος πίνακας στη βάση δεδομένων που χρησιμοποιείται και έτσι γίνεται η ερώτηση στο αν πρέπει να δημιουργηθεί. Ο λόγος που γίνεται αυτή η ερώτηση είναι γιατί μπορεί ο πίνακας να μην είναι συμπληρωμένος σωστά.. Έτσι για να αποφευχθούν τέτοια λάθη δημιουργείται αυτό το ερώτημα.

Αν ο χρήστης επιθυμεί τη δημιουργία του πίνακα, θα πρέπει να κάνει διπλό κλικ επάνω στο μήνυμα. Κατά το διπλό κλικ θα εμφανιστεί ένα μήνυμα όπως αυτό εμφανίζεται στην Εικόνα 3.7.30. Το μήνυμα αυτό θα τον ρωτάει για τελευταία φορά αν επιθυμεί τη δημιουργία του πίνακα. Αν πατήσει «OK», τότε θα δημιουργηθεί ο πίνακας με βάση τα πεδία που έχουν οριστεί στο τελικό σχήμα.



Εικόνα 3.7.30 Τελικό μήνυμα επιβεβαίωσης πριν τη δημιουργία του πίνακα

Με τη χρήση του Show / Hide Window κάτω από το μενού Validation, μπορεί να εμφανίσει ή να αποκρύψει, οποιαδήποτε στιγμή, τη φόρμα επικύρωσης. Αυτό

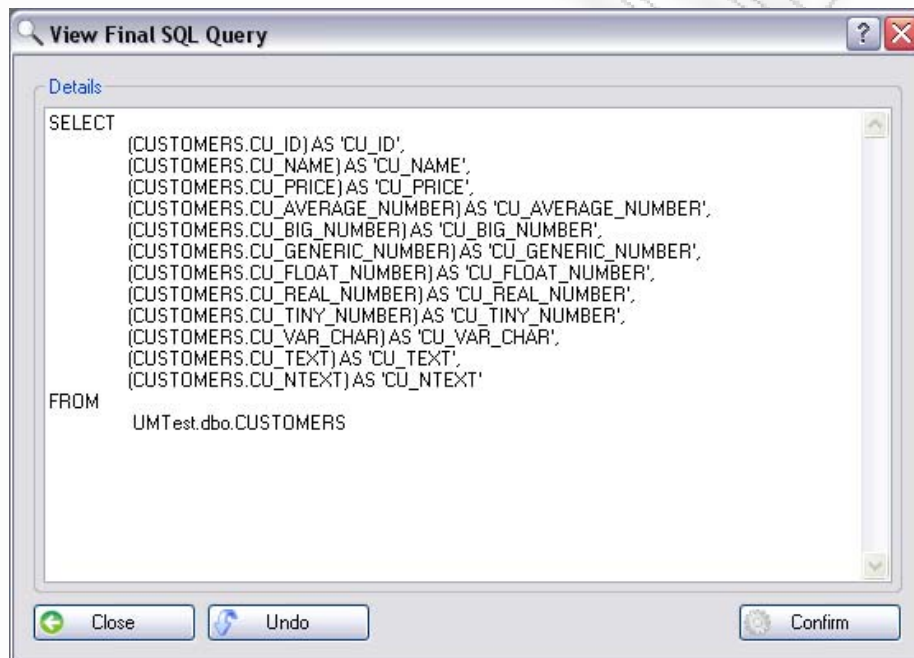
μπορεί να γίνει είτε με τη χρήση του μενού επικύρωσης, είτε μέσα από το δυναμικό μενού που εμφανίζεται πατώντας δεξί πλήκτρο στον κενό χώρο σχεδίασης.

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΡΑΙΑ

3.7.4 Execution

Η χρήση της εκτέλεσης – Execution, γίνεται όταν ο χρήστης έχει ορίσει το τελικό σχήμα και θέλει να πραγματοποιηθεί η μεταφορά. Αρχικά θα ελέγξει αυτόματα αν υπάρχουν μηνύματα τα οποία εμποδίζουν τη μεταφορά. Τα μηνύματα αυτά, είναι τα μηνύματα λάθους και ερωτήματος. Αν υπάρχουν τέτοιες περιπτώσεις, τότε σταματάει η εκτέλεση και εμφανίζονται τα μηνύματα αυτά.

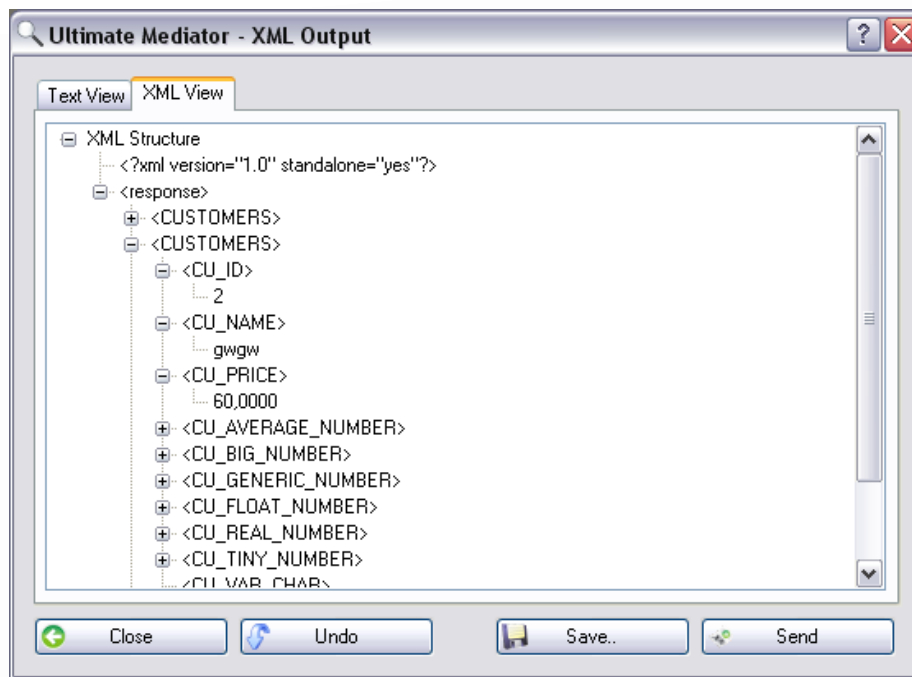
Εφόσον η επικύρωση ολοκληρωθεί επιτυχώς, τότε στην περίπτωση που τα δεδομένα προήλθαν από βάσεις δεδομένων, θα εμφανιστεί ένα παράθυρο με το τελικό sql ερώτημα το οποίο επρόκειτο να εκτελεστεί για να γίνει η λήψη των δεδομένων αυτών (Εικόνα 3.7.31). Αυτό δίνει την ευκαιρία για έναν έξτρα έλεγχο από το χρήστη, αν το ερώτημα το οποίο επρόκειτο να εκτελεστεί είναι αυτό που θα ήθελε ή όχι. Εκεί ο χρήστης μπορεί να κάνει αλλαγές, αλλάζοντας τελείως το ερώτημα χωρίς να δημιουργείται κάποιος περιορισμός.



Εικόνα 3.7.31 Εμφάνιση τελικού sql ερωτήματος

Αν η εκτέλεση περάσει από το πρώτο στάδιο της επικύρωσης και το δεύτερο του ελέγχου (αν είναι βάση δεδομένων αλλιώς μόνο από το πρώτο), τότε θα ξεκινήσει η διαδικασία της μεταφοράς των δεδομένων στο τελικό σχήμα της βάσης. Η διαδικασία αυτή μπορεί να έχει μεγάλη ή μικρή διάρκεια, ανάλογα με την ποσότητα δεδομένων που θα πρέπει να διαβάσει, τις ταχύτητες δικτύωσης (αν δεν βρίσκονται όλα στο ίδιο pc π.χ. δύο βάσεις και οι δύο σε απομακρυσμένους υπολογιστές), τις ορισμένες συνθήκες, και τον αριθμό των πινάκων που χρησιμοποιούνται.

Μετά την ολοκλήρωση της εκτέλεσης, θα εμφανιστεί ένα μήνυμα ότι η μεταφορά ολοκληρώθηκε επιτυχώς. Αν το τελικό σχήμα είναι XML, τότε θα εμφανιστεί μία φόρμα που θα δείχνει το xml αυτό, και σαν απλό κείμενο και σε δενδρική μορφή (Εικόνα 3.7.32).

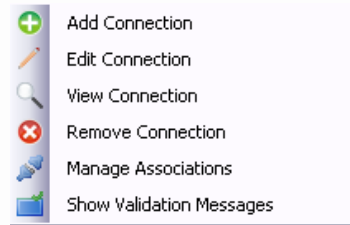


Εικόνα 3.7.32 Οθόνη εμφάνισης XML δεδομένων στη δενδρική τους μορφή

Λεπτομερής περιγραφή της φόρμας αυτής θα γίνει στην ενότητα *XML to Web Service Transporter*. Σε αυτό το σημείο αναφέρεται απλά ότι ο χρήστης έχει τη δυνατότητα αλλαγής των δεδομένων, αποθήκευσής τους σε αρχεία, και απευθείας αποστολής τους σε κάποιο Web Service.

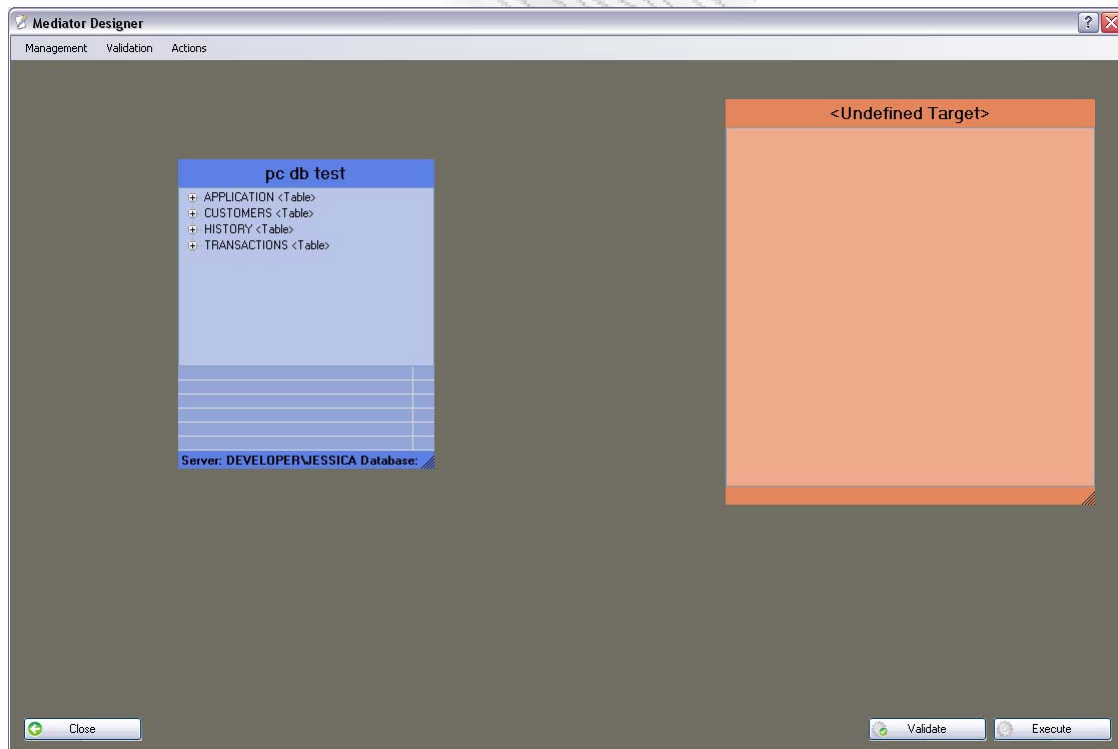
3.7.5 Add/Set/Edit/View/Remove Connection

Για την προσθήκη σύνδεσης στη σχεδιαστική φόρμα αρκεί να κάνουμε δεξί πλήκτρο επάνω στον κενό χώρο της φόρμας. Εκεί θα εμφανιστεί το μενού της Εικόνα 3.7.33.



Εικόνα 3.7.33 Μενού επιλογών που εμφανίζεται στον κενό χώρο

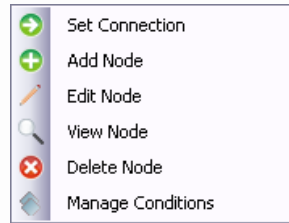
Μέσω της επιλογής Add Connection μπορεί να προσθέσει μία σύνδεση στη φόρμα σχεδιασμού. Για να γίνει αυτό, ανοίγει η διαχειριστική φόρμα επιλογής σύνδεσης που παρουσιάστηκε στην ενότητα *Manage Connections*. Από εκεί επιλέγει τη σύνδεση και αυτή εμφανίζεται στη φόρμα όπως δείχνει η Εικόνα 3.7.34.



Εικόνα 3.7.34 Εμφάνιση σύνδεσης στη φόρμα σχεδίασης

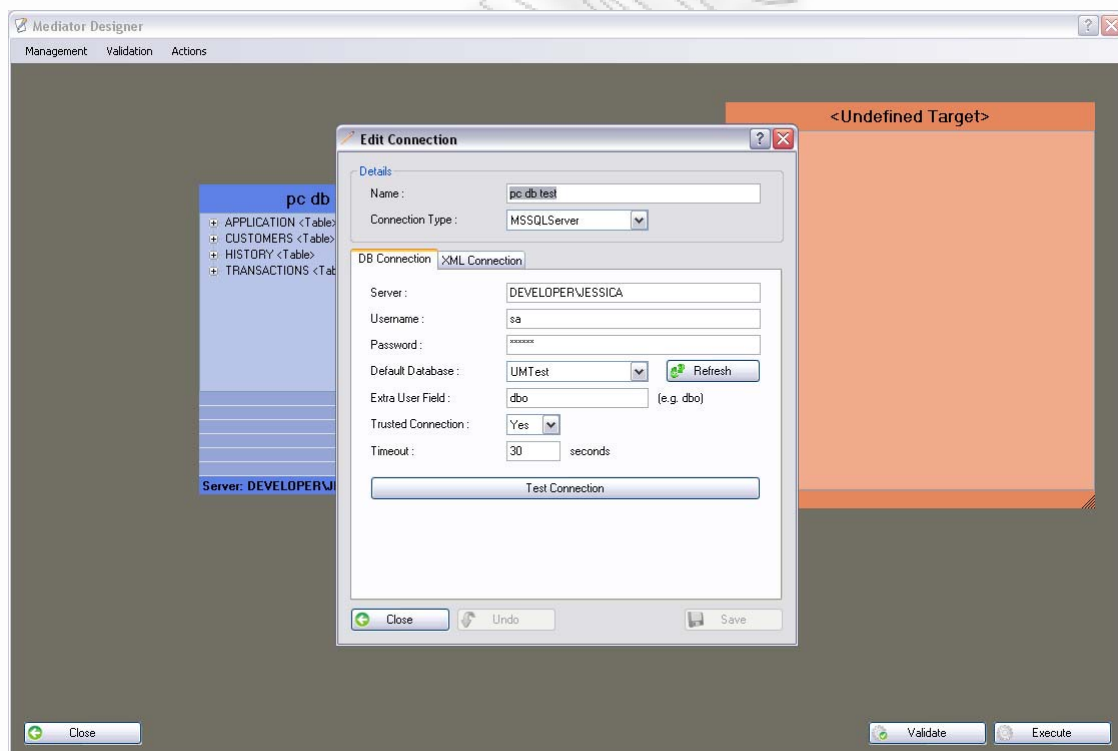
Παρόμοια με την Add Connection έχουμε και την Set Connection, έχοντας ακριβώς την ίδια λειτουργία. Η διαφορά τους είναι μόνο εννοιολογική. Η Set

Connection χρησιμοποιείται για να ορίσει στο τελικό σχήμα το Connection. Αν για παράδειγμα επιλέξει κάποιο connection τύπου XML, τότε το τελικό σχήμα που θα εξάγει η εφαρμογή θα είναι ένα XML. Η επιλογή Set Connection είναι προσπελάσιμη από το μενού και επιπλέον εμφανίζεται πατώντας το δεξί πλήκτρο του mouse επάνω στο πλαίσιο με το τελικό σχήμα (Εικόνα 3.7.35).

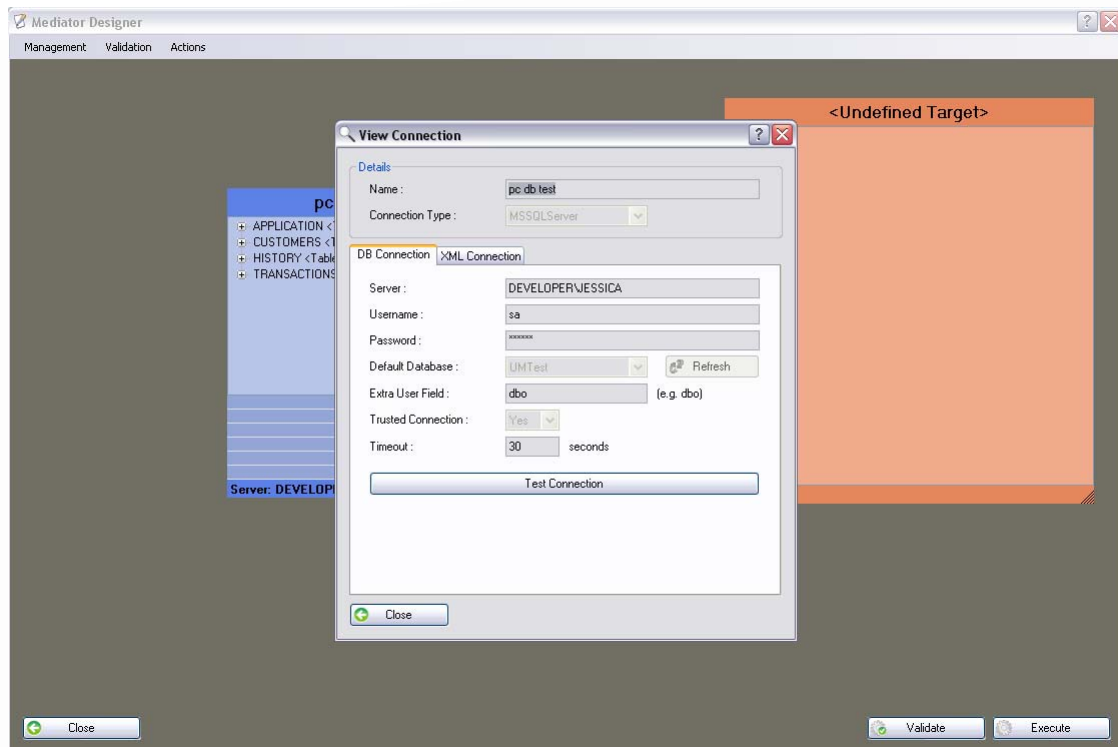


Εικόνα 3.7.35 Μενού που εμφανίζεται με το δεξί πλήκτρο του ποντικού επάνω στο τελικό σχήμα

Εκτός της προσθήκης μπορεί εκτελέσει και επεξεργασία μίας σύνδεσης, από το μενού που αναφέραμε στην Εικόνα 3.7.33. Σε αυτήν την περίπτωση ανοίγει απευθείας η φόρμα επεξεργασίας σύνδεσης για επεξεργασία (Εικόνα 3.7.36). Παρόμοια ανοίγει και η φόρμα σύνδεσης για απλή προβολή (Εικόνα 3.7.37).



Εικόνα 3.7.36 Επεξεργασία σύνδεσης από τη φόρμα σχεδίασης



Εικόνα 3.7.37 Απλή προβολή σύνδεσης από τη διαχειριστική φόρμα

Το «Remove Connection», δεν διαγράφει τελείως τη σύνδεση από το σύστημα, απλά την αφαιρεί από τη φόρμα σχεδίασης. Θα πρέπει όμως ο χρήστης να διαγράψει όλες τις αναφορές που έχουν γίνει για τη συγκεκριμένη σύνδεση, για να μην μείνουν κενές αναφορές.

3.7.6 Add/Edit/View/Delete Node

Όπως είδαμε και στην προηγούμενη ενότητα με το Set Connection (Add/Set/Edit/View/Remove Connection), αν ο χρήστης πατήσει το δεξί πλήκτρο του ποντικού επάνω στο τελικό σχήμα, θα εμφανιστεί το μενού της Εικόνας (Εικόνα 3.7.33). Το μενού αυτό έχει, μεταξύ άλλων, τη δυνατότητα της προσθήκης, επεξεργασίας, προβολής και διαγραφής ενός κόμβου (Node). Κόμβος ορίζεται οποιοδήποτε πεδίο μπορεί να μπει στο τελικό σχήμα, είτε αυτό είναι καινούργιο (manual), είτε αυτό υπάρχει ήδη από άλλο σχήμα (Existing).

Αν το τελικό σχήμα είναι xml, τότε ο κόμβος αντιπροσωπεύει την έννοια του Element όπως αυτό ορίζεται από την W3C [16]. Στην περίπτωση όμως που η σύνδεση είναι τύπου βάσης, τότε ο κόμβος αντιπροσωπεύει κάποια στήλη (column), ή πίνακα, της βάσης. Το τι ακριβώς θα είναι ορίζεται από τον τύπο που δηλώνουμε κατά την αρχικοποίηση και επεξεργασία του κόμβου.

Στις επόμενες υπό-ενότητες θα περιγραφούν οι φόρμες, προσθήκης, επεξεργασίας, προβολής και διαγραφής, όπως αυτές χρησιμοποιούνται στην εφαρμογή.

3.7.6.1 Προσθήκη κόμβου (Add Node)

Η προσθήκη ενός κόμβου είναι η διαδικασία με την οποία δημιουργούμε έναν καινούργιο κόμβο ο οποίος θα πάρει τιμή μέσα από μία παράσταση που θα δηλώσουμε στη φόρμα. Η φόρμα αυτή παρουσιάζεται στην Εικόνα 3.7.38.

Εικόνα 3.7.38 Φόρμα προσθήκης κόμβου

Η φόρμα έχει τρεις καρτέλες (ομάδα από διαφορετικές εννοιολογικά οθόνες - tabs), οι οποίες παρουσιάζουν τρεις διαφορετικές οθόνες στο χρήστη ανάλογα με την επιλογή του. Η οθόνη της πρώτη καρτέλας (Details) φαίνεται στην Εικόνα 3.7.38. Περιέχει κάποια μη προσπελάσιμα πεδία όπως το Connection, το Old Name και το Old Data Type. Αυτά υπάρχουν στην περίπτωση που γίνει επεξεργασία του κόμβου, έτσι ώστε ο χρήστη τη στιγμή που κάνει επεξεργασία να μπορεί να δει τι είχε αρχικά σαν ρυθμίσεις ο κόμβος.

Η αρχική οθόνη περιέχει επίσης το όνομα του κόμβου που θα δοθεί, κάποια σχόλια που χρησιμοποιούνται σαν σημειώσεις για το χρήστη και τον τύπο των δεδομένων. Ο τύπος των δεδομένων μπορεί να είναι ένα από τα ακόλουθα:

1. **Char** – Ένας μόνο χαρακτήρας,
2. **Small String** – Μικρό κείμενο μέχρι 20 χαρακτήρες,
3. **Large String** – Κείμενο 255 χαρακτήρων,
4. **Extended String** – Κείμενο 8000 χαρακτήρων,
5. **Int 16** – Μικρός Ακέραιος 2 Bytes,
6. **Int 32** – Ακέραιος 4 Bytes,
7. **Int 64** – Ακέραιος μεγάλου μεγέθους 8 Bytes,

8. **Float** – Δεκαδικός αριθμός,
9. **Double** – Δεκαδικός αριθμός διπλής ακρίβειας,
10. **Datetime** – Ημερομηνία,
11. **Boolean** – Περιλαμβάνει λογικές τιμές (0, 1) ή (true, false),
12. **Table** – Δηλώνει ότι είναι πίνακας ο κόμβος για XML ή Βάση,
13. **XML** – Δηλώνει ότι είναι κόμβος που περιέχει άλλους κόμβους (για XML),
14. **Unknown** – Χρησιμοποιείται όταν το σύστημα δεν αναγνωρίζει τον τύπο δεδομένων.

Εκτός του τύπου δεδομένων υπάρχει και το Mapping Type. Το πεδίο αυτό ορίζει το πως θα εμφανιστεί ο κόμβος στο τελικό σχήμα. Οι πιθανές τιμές είναι:

1. **Standard Column** – Καθιερωμένο πεδίο κόμβου όπως η στήλη (χρησιμοποιείται και στο XML σχήμα και στο σχήμα Βάσης),
2. **Attribute** – Το πεδίο αυτό ορίζεται μόνο σε XML σχήμα και αντιπροσωπεύει το γνώρισμα / την ιδιότητα, ενός «Standard Column». Αν το πεδίο αυτό χρησιμοποιηθεί σε σχήμα βάσης δεδομένων, τότε θα θεωρηθεί ότι είναι ένα «Standard Column»,
3. **Hidden** – Το πεδίο αυτό υποδηλώνει ότι ο κόμβος θα είναι κρυφός, και στην περίπτωση της βάσης δεδομένων και στην περίπτωση του XML.

Το επόμενο πεδίο που βλέπουμε στην Εικόνα 3.7.38, είναι το «Order number». Το order number έχει ουσιαστικά το ρόλο του πεδίου ταξινόμησης των κόμβων στο τελικό σχήμα. Με βάση αυτό μπορεί ο χρήστης να ξανα-ορίσει τη σειρά όπως εκείνος θέλει. Αυτό πολλές φορές τον βοηθάει να κρατάει τα πεδία στη λίστα όπως πιστεύει ότι είναι καλύτερα για τον ίδιο οπτικά. Αν ορίσει κάποιο πεδίο να έχει τιμή ταξινόμησης πολύ μεγαλύτερη από αυτήν που μπορεί να πάρει το σύστημα με βάση τον αριθμό των κόμβων που ήδη έχει, τότε θα οριστεί αυτομάτως στο μέγιστο των πεδίων του. Ο λόγος είναι ότι η σειρά που δηλώνεται στο τελικό σχήμα είναι και αυτή που τελικά θα ακολουθηθεί για την παραγωγή της τελικής εξόδου. Μπορεί να μην έχει μεγάλο νόημα αυτό όταν το τελικό σχήμα είναι η βάση δεδομένων, αλλά έχει νόημα κατά την προβολή του, όταν είναι XML.

Το πεδίο «Parent» ορίζει τον πατέρα του κόμβου. Όλοι οι κόμβοι έχουν κάποιον πατέρα, εκτός αυτού που βρίσκεται πρώτος – πάνω από όλους τους άλλους. Έτσι αν επιθυμείται να αλλαχτεί ο πατέρας κάποιου κόμβου, όταν για παράδειγμα ο χρήστης του όρισε λανθασμένα άλλο πατέρα, το πεδίο αυτό θα καλύψει την ανάγκη. Το πεδίο δείχνει το υπάρχον, μέχρι εκείνη τη στιγμή, σχήμα που έχει οριστεί για την τελική έξοδο.

Η οθόνη του δεύτερου Tab με όνομα «Column Modification» παρουσιάζεται στην Εικόνα 3.7.42. Αυτή περιέχει τέσσερα μόνο πεδία τα οποία συνδέονται άμεσα μεταξύ τους, και βοηθάνε στο να δημιουργηθεί το τελικό «query» / ερώτημα για τον συγκεκριμένο κόμβο. Τα πεδία αυτά είναι:

1. Connection

Το πεδίο αυτό είναι απλά βοηθητικό και δε χαρακτηρίζει καθόλου τον κόμβο. Χρησιμοποιείται για τον ορισμό του επόμενου πεδίου «Column Fields». Μέσα από τον ορισμό της σύνδεσης εμφανίζονται όλοι οι πίνακες μαζί με τα πεδία τους στο «Column Fields».

2. Column Fields

Το συγκεκριμένο πεδίο βοηθάει στην εισαγωγή του τρόπου με τον οποίο θα πρέπει ο χρήστης να χρησιμοποιήσει το όνομα κάποιου άλλου πεδίου για τον υπολογισμό του ίδιου. Αν για παράδειγμα το τελικό σχήμα είναι τύπου βάσης, τότε τα πεδία που πιθανώς θα χρησιμοποιήσει ο χρήστης, θα πρέπει να γραφτούν στο «Final Modification» (το επόμενο πεδίο στην οθόνη), με βάση το όνομα του πατέρα τους.

Αν όμως το τελικό σχήμα είναι XML, τότε είναι σωστότερο να γραφτούν απλά με το όνομα τους. Για να μην υπάρχουν τέτοια προβλήματα το Column Fields βοηθάει στην απευθείας εισαγωγή άλλου κόμβου με το σωστό ορισμό του ονόματος. Αυτό γίνεται εφόσον επιλεγθεί ο κόμβος από τη λίστα με τη χρήση του κουμπιού προσθήκης που υπάρχει στα δεξιά του πεδίου.

3. Predefined Statements

Το πεδίο Predefined Statements περιέχει κάποιες συγκεκριμένες εκφράσεις μετατροπής δεδομένων από έναν τύπο δεδομένων σε άλλο. Χρησιμοποιείται για να διευκολύνει τον χρήστη στις πιθανές μετατροπές που μπορεί να κάνει. Έχουν οριστεί κάποιες κλασσικές που είναι πιθανόν να χρησιμοποιήσει. Για όλες τις υπόλοιπες μπορεί αν δει την σχετική βοήθεια που παρέχεται από την εφαρμογή. Η μετατροπή δεδομένων παρέχεται και για xml τύπο εισαγωγής και για sql. Ο τρόπος διαχωρισμός τους γίνεται από τα δύο Groups που έχει η διαθέσιμη λίστα (Εικόνα 3.7.39 και Εικόνα 3.7.40).

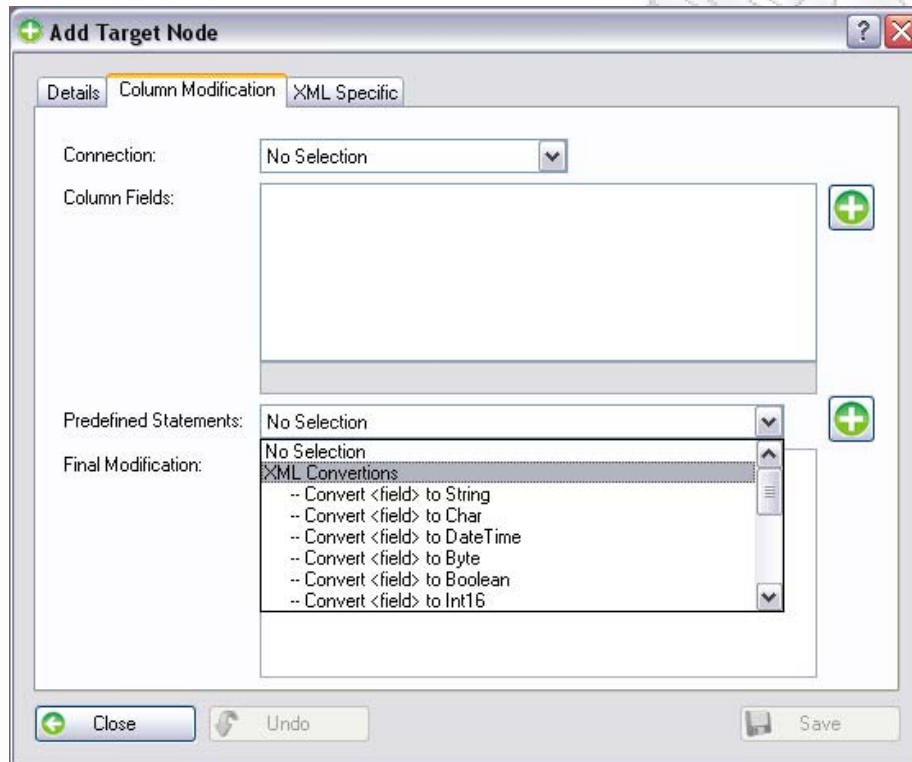
Εφόσον ο χρήστης επιλέξει κάποιο από τα διαθέσιμα, μπορεί να έπειτα να πατήσει το κουμπί προσθήκης, για την εισαγωγή του στο Final Modification πεδίο (Εικόνα 3.7.41). Εφόσον το προσθέσει, μπορεί να αλλάξει έπειτα το πεδίο που βρίσκεται μεταξύ των χαρακτήρων '<' και '>', με την αντίστοιχη τιμή. Για παράδειγμα αν έχουμε <field>, τότε θα πρέπει να γίνει αντικατάσταση με το πεδίο ενδιαφέροντος από το οποίο θα γίνει η μετατροπή. Αν για παράδειγμα έχουμε ένα πεδίο το οποίο θέλουμε να παίρνει την ακέραια τιμή του πεδίου «phone_with_chars» τότε θα γράψουμε το ακόλουθο:
Convert(phone_with_chars, 'System.Int64')

4. Final Modification

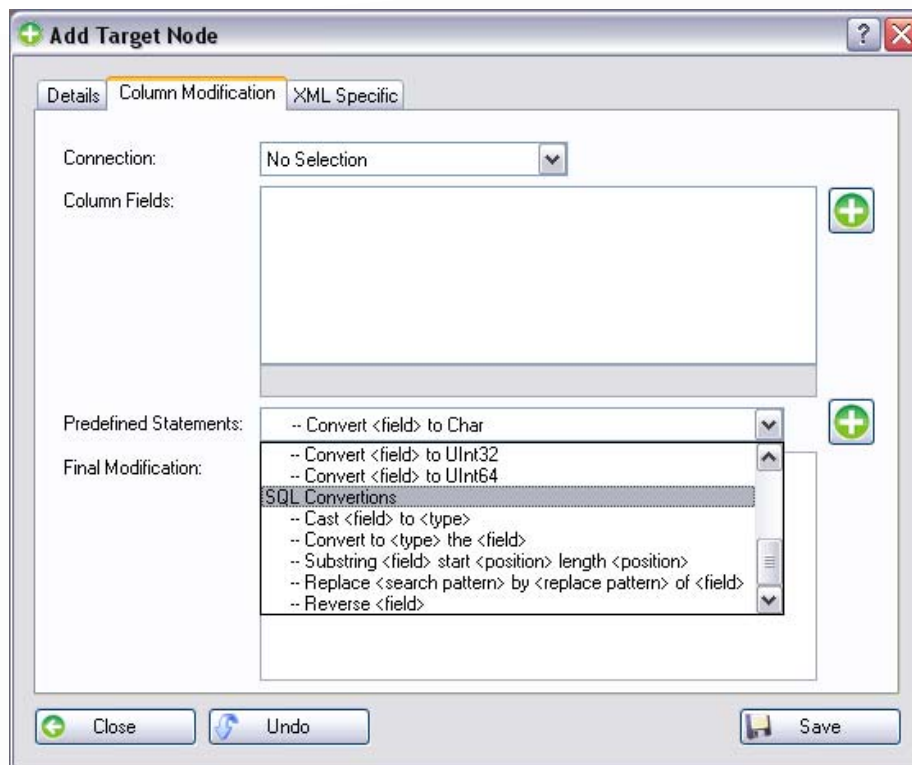
Εδώ ορίζεται όλη η τελική παράσταση που θα πρέπει να υπολογιστεί για να πάρει ο κόμβος κάποια τιμή. Η παράσταση αυτή μπορεί να είναι από τις πλέον απλές όπως να πάρει μία απλή στατική τιμή, μέχρι και σε πιο σύνθετες, όπως να υπολογίσει το άθροισμα ενός άλλου κόμβου εφόσον ισχύει κάποια συνθήκη.

Χρησιμοποιείται δηλαδή μία ψευδογλώσσα με την οποία μπορεί ο χρήστης να γράψει αυτές τις παραστάσεις. Στην περίπτωση που το τελικό σχήμα είναι XML, τον τρόπο χρήσης της ψευδογλώσσας μπορεί να τον βρει μέσα από τη βοήθεια που του παρέχεται από το ερωτηματικό στο πάνω δεξιά μέρος της οθόνης (Εικόνα 3.7.42). Όπως θα αναφέρουμε και σε επόμενη ενότητα (*Help και About*), η βοήθεια θα δώσει όλες τις πληροφορίες που χρειάζονται για να κατανοήσει και να δημιουργήσει ο χρήστης δικές του παραστάσεις.

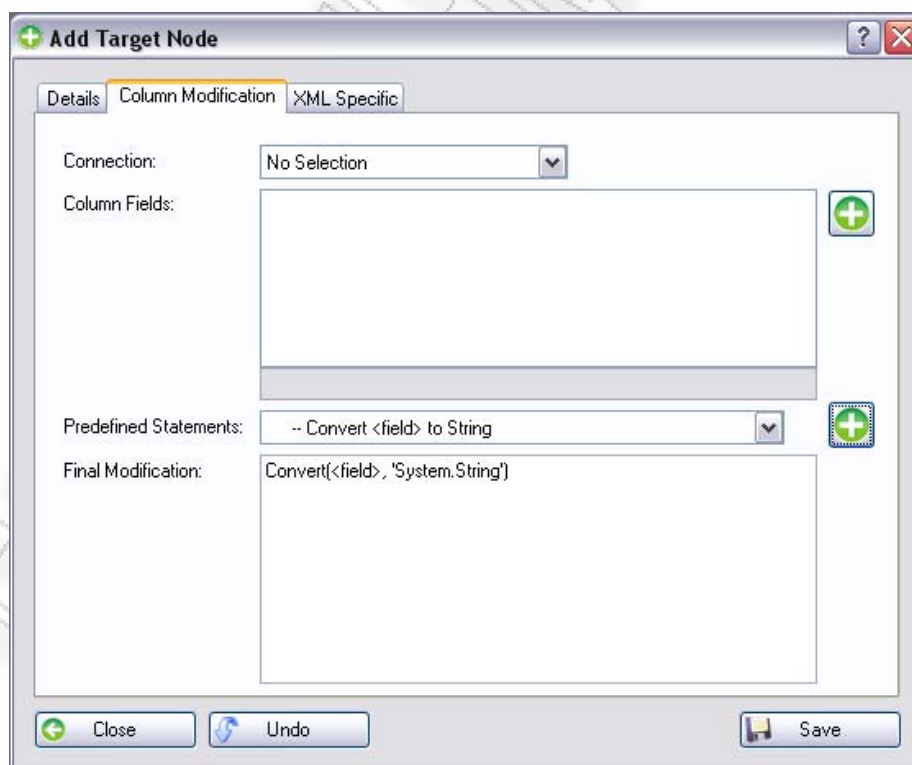
Έστω όμως ότι το τελικό σχήμα είναι μία βάση δεδομένων, τότε για την παράσταση θα πρέπει να χρησιμοποιηθεί η σύνταξη της sql, όπως αυτή ορίζεται στη βάση δεδομένων από την οποία προέρχεται το πεδίο.



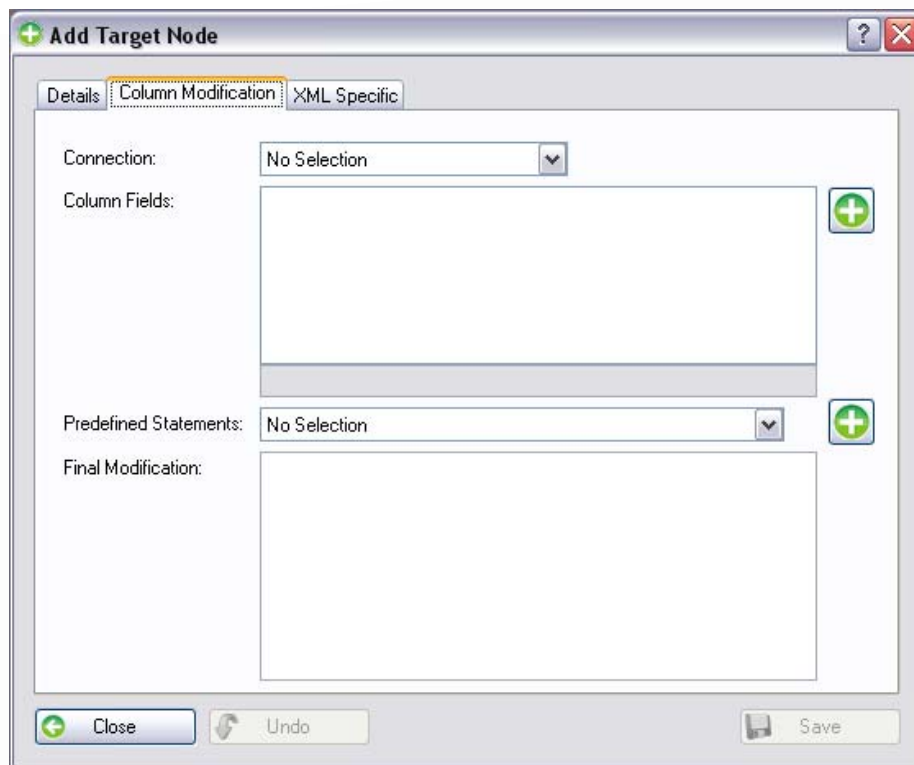
Εικόνα 3.7.39 Το group XML Conversions στα Predefined Statements



Εικόνα 3.7.40 To group SQL Conversions στα Predefined Statements

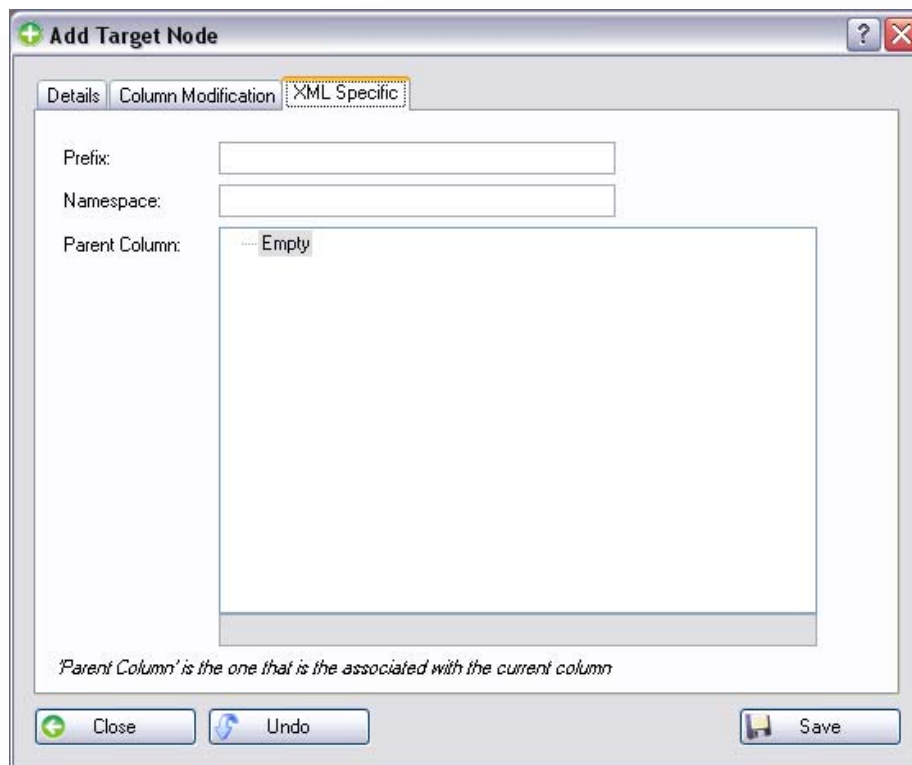


Εικόνα 3.7.41 Προσθήκη ενός Predefined Statement



Εικόνα 3.7.42 Οθόνη της καρτέλας Column Modification

Η τελευταία καρτέλα είναι η «XML Specific» (Εικόνα 3.7.42). Αυτό το Tab χρησιμοποιείται μόνο όταν έχουμε ως τελικό σχήμα μία σύνδεση τύπου XML. Αν γίνουν ορισμοί σε αυτό το σημείο ενώ το τελικό σχήμα είναι βάση δεδομένων, τότε αυτοί θα αγνοηθούν.

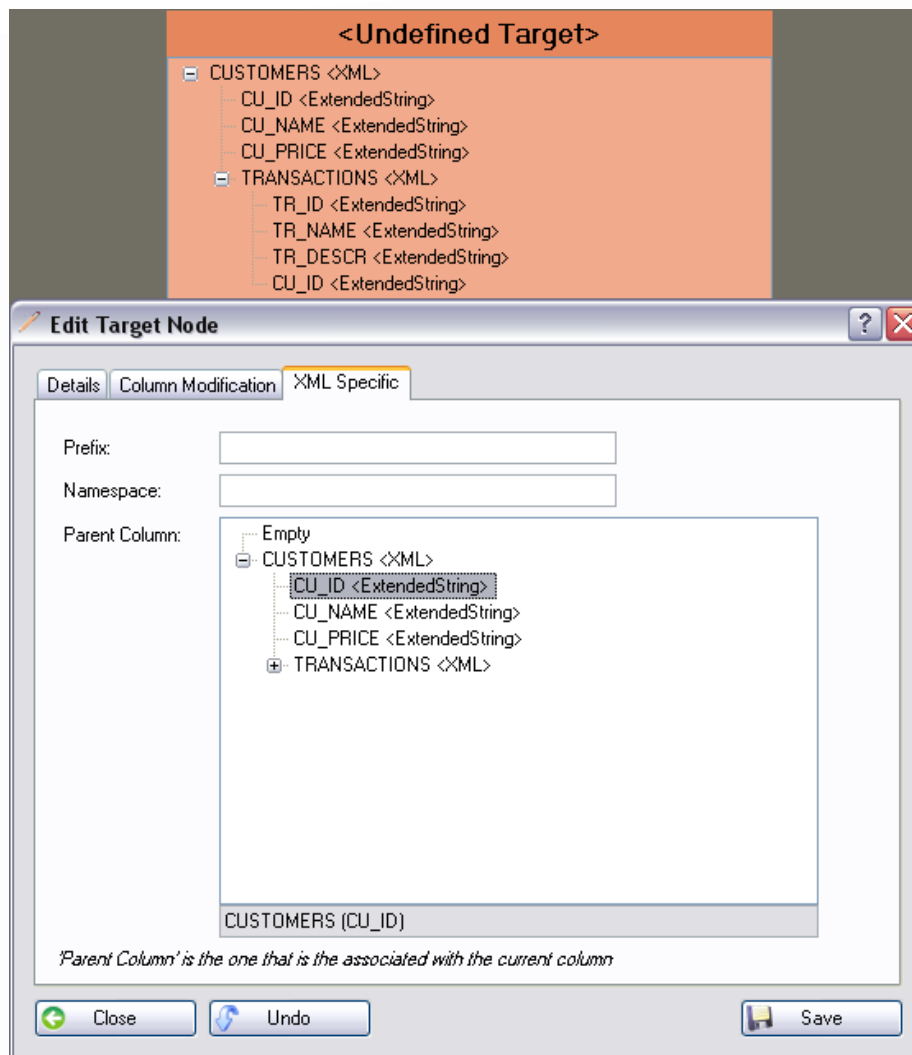


Εικόνα 3.7.43 Οθόνη της καρτέλας «XML Specific»

Στην τρίτη οθόνη υπάρχουν τα ακόλουθα πεδία, τα οποία χαρακτηρίζουν τον τρέχων κόμβο.

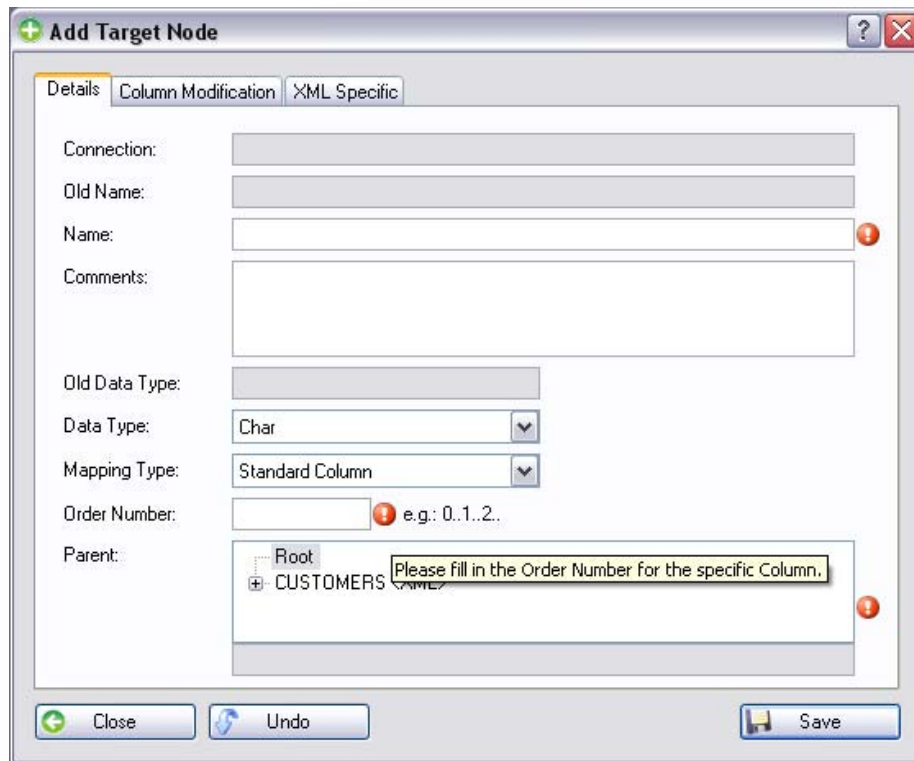
1. **Prefix** – Εδώ ορίζεται το πρόθεμα prefix για τον κόμβο αυτόν,
2. **Namespace** – Το namespace του προθέματος,
3. **Parent Column** – Εδώ ορίζεται η πιθανή συσχέτιση του κόμβου με κάποιον άλλον κόμβο. Είναι η συσχέτιση μεταξύ στοιχείων του σχήματος (XML Associations).

Η έννοια του πατέρα κόμβου χρησιμοποιείται όπως ακριβώς ορίζεται το ξένο κλειδί ενός πίνακα συσχετιζόμενο με το πρωτεύον κάποιου άλλου. Έτσι και εδώ ισχύει κάτι αντίστοιχο. Έστω ότι έχουμε το παράδειγμα της Εικόνα 3.7.44, όπου ο κόμβος που προσθέτουμε ανήκει στον πίνακα Transactions και είναι ο CU_ID. Ο κόμβος αυτός όμως συνδέεται με τον κόμβο CU_ID που ανήκει στον πίνακα Customers. Έτσι για να ορίσουμε τον πατέρα, ώστε να μπορεί το πρόγραμμα μετά να πάρει τα δεδομένα σωστά, φτιάχνοντας σωστές συνδέσεις μεταξύ των κόμβων του, θα πρέπει να ορίσουμε το Parent Column.



Εικόνα 3.7.44 Ορισμός XML Association (σύνδεσης στοιχείων σχήματος)

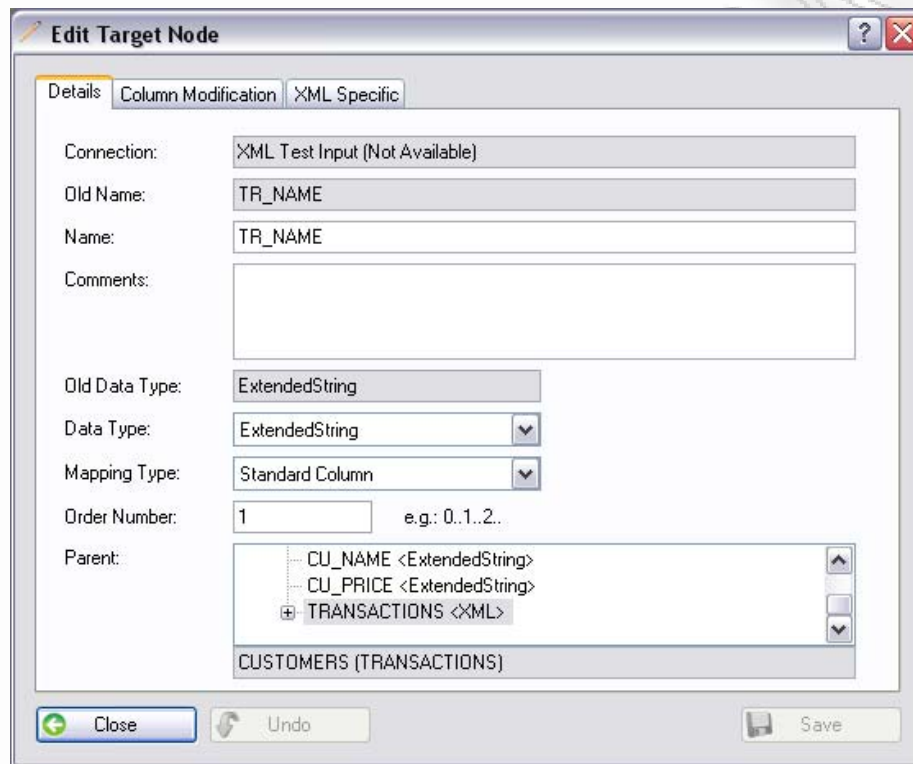
Κατά την προσθήκη ενός κόμβου, τα πεδία που είναι υποχρεωτικά είναι τα πεδία της πρώτης οθόνης (Tab - Details). Όλα τα συγκεκριμένα πεδία, εκτός του comments, θα πρέπει να οριστούν, διαφορετικά θα προκύψουν σφάλματα κατά την αποθήκευση (Εικόνα 3.7.45). Αν δεν προκύψει κάποιο σφάλμα τότε ο νέος κόμβος θα εμφανιστεί στο τελικό σχήμα.



Εικόνα 3.7.45 Λάθη μη εισαγωγής δεδομένων στη δημιουργία νέου κόμβου

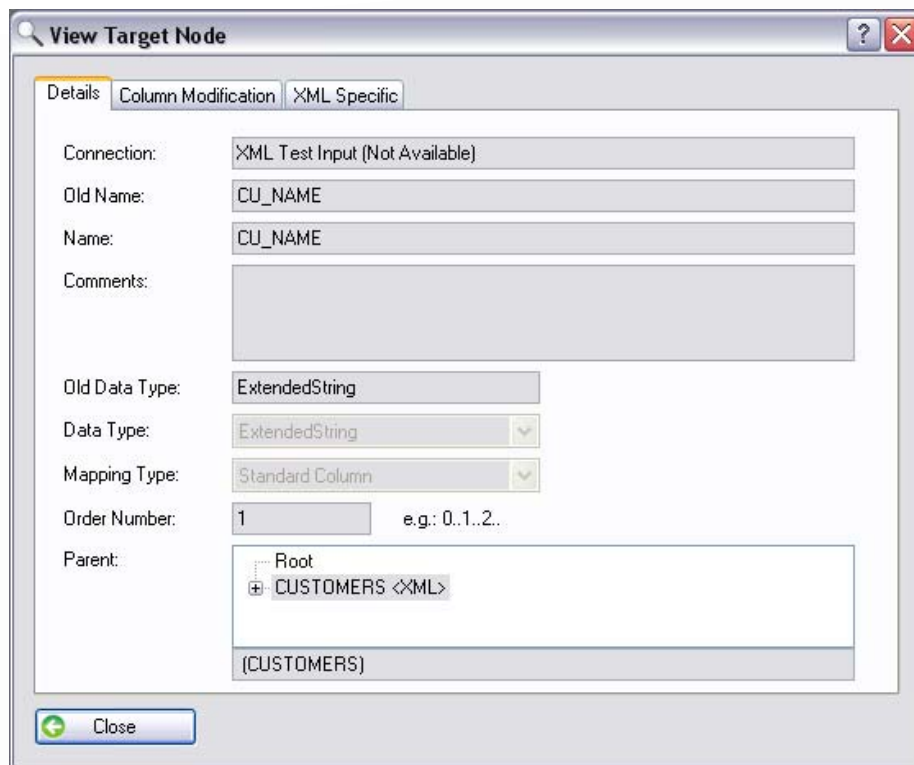
3.7.6.2 Επεξεργασία / Προβολή / Διαγραφή Κόμβου

Σε αυτήν την ενότητα θα περιγραφτούν και οι υπόλοιπες λειτουργίες που υπάρχουν για τους κόμβους. Κατά την επεξεργασία, όπως έχουμε αναφέρει και σε άλλες παρόμοιες λειτουργίες επεξεργασίας, εμφανίζεται η φόρμα με συμπληρωμένα όλα τα πεδία της (Εικόνα 3.7.46).



Εικόνα 3.7.46 Παράδειγμα επεξεργασίας κόμβου

Η προβολή αντίστοιχα εμφανίζει όλες τις πληροφορίες χωρίς να δίνει όμως τη δυνατότητα να γίνει επεξεργασία αυτών (Εικόνα 3.7.47).



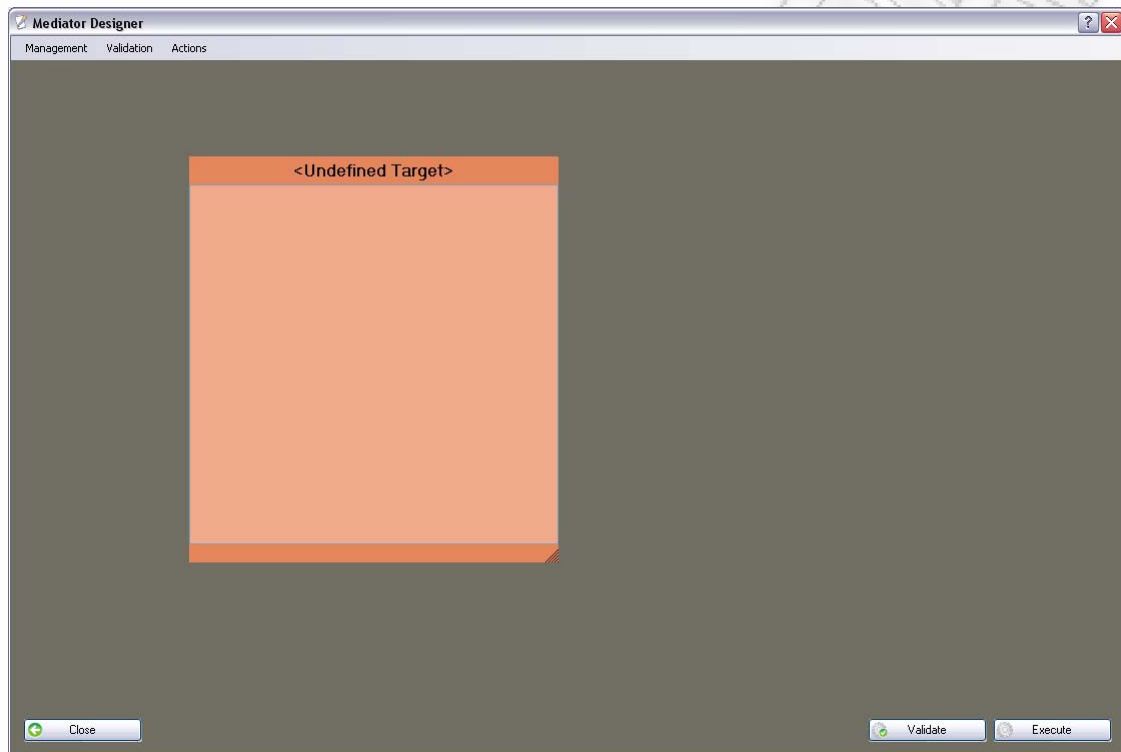
Εικόνα 3.7.47 Παράδειγμα προβολής κόμβου

Τέλος έχουμε τη διαγραφή του κόμβου. Η διαγραφή του κόμβου έχει σαν αποτέλεσμα να αλλάξει το Order Number των υπόλοιπων κόμβων. Ο λόγος είναι ότι αν ο κόμβος ήταν ο 5^{ος} από τους 12, τότε θα πρέπει οι υπόλοιποι ένδεκα να λάβουν καινούργιο Order Number σύμφωνα με τη νέα σειρά εμφάνισής τους.

3.7.7 Διαχείριση Σχημάτων στη φόρμα σχεδίασης

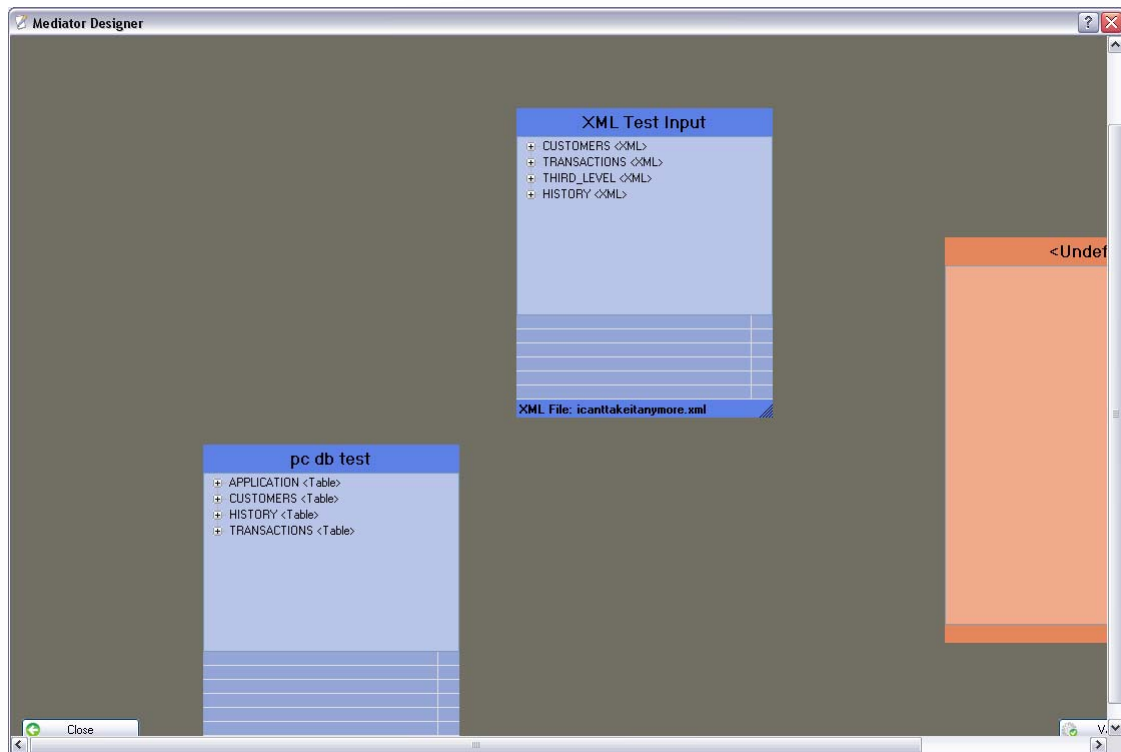
Η φόρμα σχεδίασης δίνει τη δυνατότητα να είναι εύκολα διαχειρίσιμη με το να προσφέρει την ευχέρεια της αλλαγής και μετακίνησης των πλαισίων, των αρχικών σχημάτων, αλλά και του τελικού.

Για τη μετακίνηση ενός πλαισίου αρκεί να πατήσουμε επάνω στη μπάρα και να σύρουμε το πλαίσιο σε άλλο σημείο (Εικόνα 3.7.48).



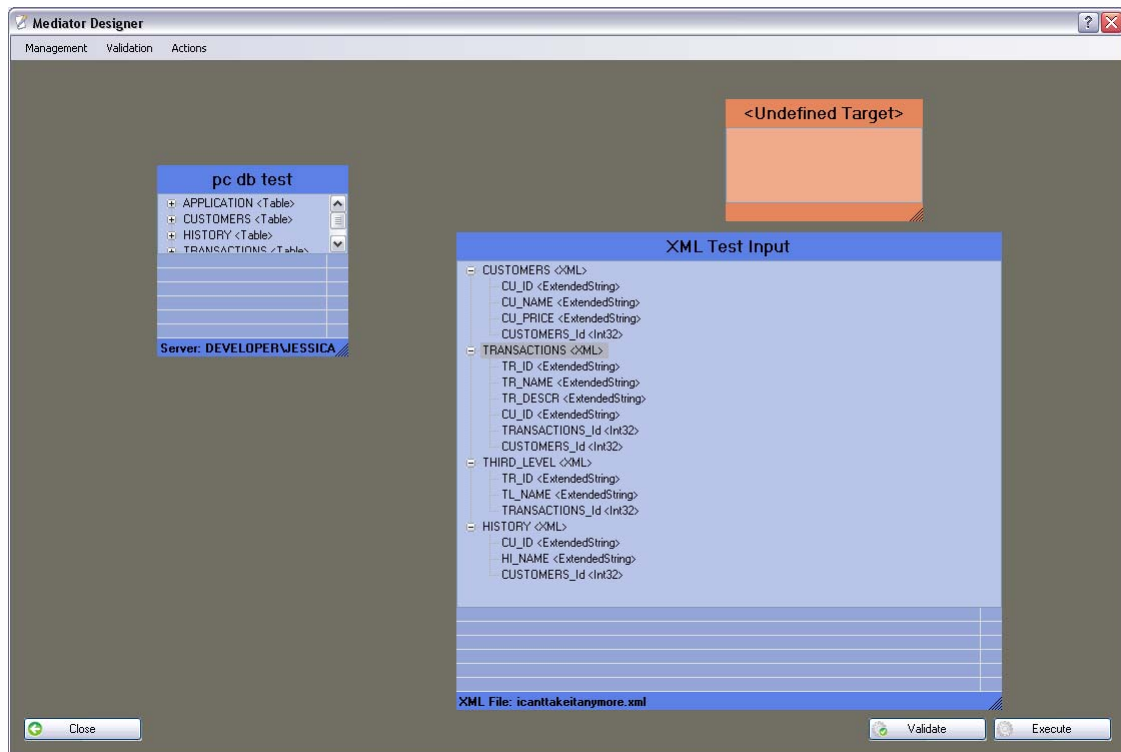
Εικόνα 3.7.48 Μετακίνηση πλαισίου προς τα αριστερά

Η μετακίνηση ενός πλαισίου μπορεί να βγει και εκτός ορίων του παραθύρου αρκεί να το μετακινήσουμε προς τα κάτω ή προς τα δεξιά (Εικόνα 3.7.49). Τότε εμφανίζεται η οριζόντια ή / και η κάθετη μπάρα για να γίνεται εφικτή η προσπέλαση των πλαισίων.



Εικόνα 3.7.49 Μετακίνηση πλαισίων εκτός ορίων παραθύρου

Τέλος, μία ακόμη ευκολία είναι και η αλλαγή μεγέθους των πλαισίων. Αυτό βοηθάει το χρήστη κυρίως στην περίπτωση που τα πεδία που ορίζονται στα σχήματα είναι μεγάλου μεγέθους. Έτσι μπορεί να μεγαλώσει το πλαίσιο για να δει όλα τα πεδία χωρίς την standard χρήση της μπάρας κύλισης. Για να μεγαλώσει το πλαίσιο αρκεί να «πιάσει» με το ποντίκι του την κάτω δεξιά γωνία του και να την τραβήξει προς τα έξω ή προς το κέντρο του. Η Εικόνα 3.7.50 δείχνει τέτοια παραδείγματα χρήσης.

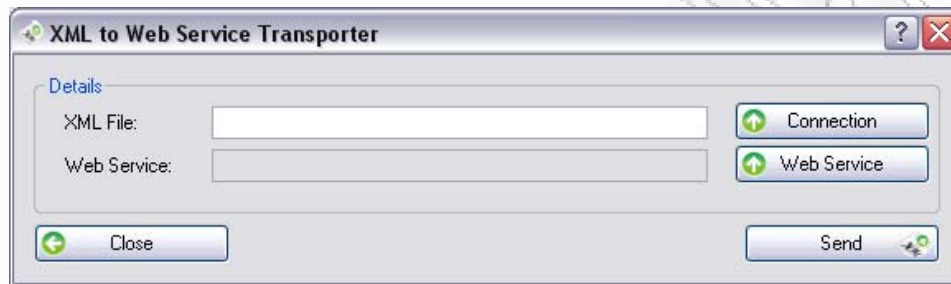


Εικόνα 3.7.50 Αλλαγή μεγέθους σχημάτων

3.8 XML to Web Service Transporter

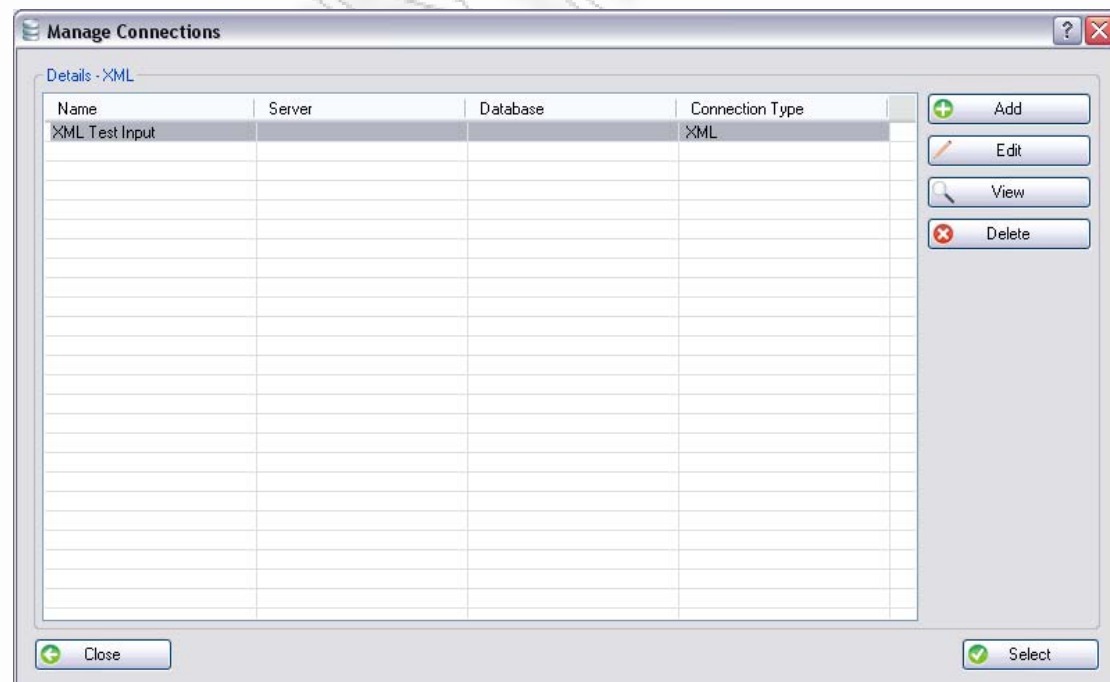
Από τη φόρμα XML to Web Service Transporter, μπορεί να γίνει η αποστολή ενός XML σε κάποιο Web Service. Η διαδικασία είναι πολύ εύκολη και μπορεί να χρησιμοποιηθεί όπως έχουμε ήδη αναφέρει και απευθείας από τον Designer μετά την έξοδο του τελικού σχήματος, τύπου XML.

Η αρχική φόρμα για τη διαδικασία μεταφοράς παρουσιάζεται στην οθόνη της Εικόνα 3.8.1.



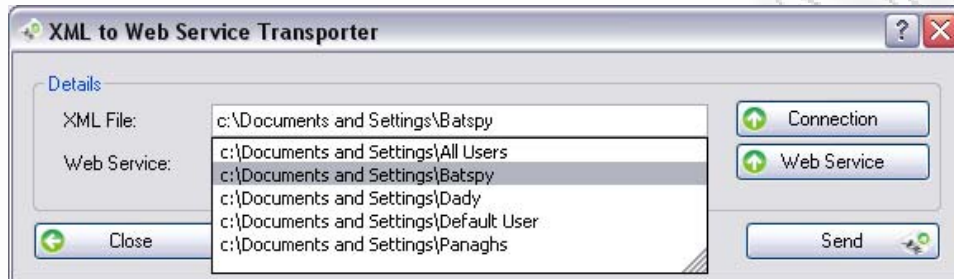
Εικόνα 3.8.1 Αρχική φόρμα αποστολής XML σε Web Service

Μέσα από το κουμπί Connection μπορεί να γίνει η επιλογή κάποιας σύνδεσης τύπου XML (Εικόνα 3.8.2). Η διαχειριστική όμως φόρμα δεν δείχνει όλα τα είδη σύνδεσης. Δείχνει μόνο τις συνδέσεις τύπου XML. Με την επιλογή της σύνδεσης αυτόματα το πεδίο «Input XML», φορτώνεται στο textbox «XML File». για να διαβάσει το αρχείο ανάγνωσης και να είναι αυτό που θα χρησιμοποιηθεί τελικά για την αποστολή.



Εικόνα 3.8.2 Διαχειριστική φόρμα συνδέσεων φιλτραρισμένη μόνο για XML

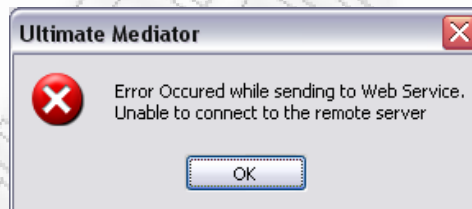
Ένας ακόμα τρόπος για να γίνει η επιλογή του XML, είναι να το πληκτρολογήσει ο χρήστης. Γι' αυτόν το λόγο έχει προστεθεί μία έξτρα δυνατότητα σύμφωνα με την οποία, την ώρα που πληκτρολογεί τη διαδρομή του δίσκου να του δείχνει ένα μενού με τους διαθέσιμους φακέλους (Εικόνα 3.8.3).



Εικόνα 3.8.3 Πληκτρολόγηση διαδρομής δίσκου με βοήθεια

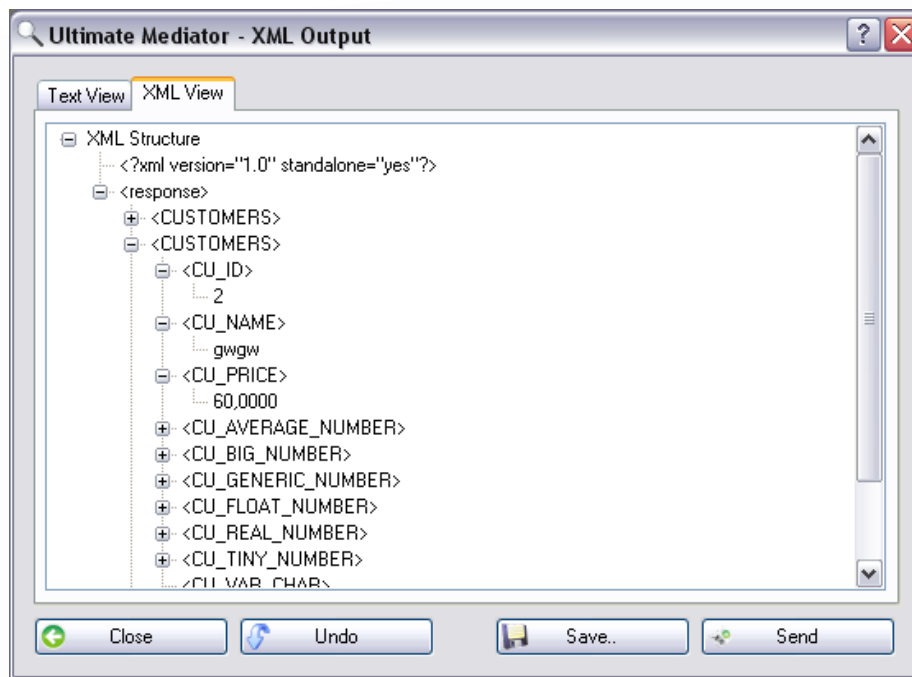
Εφόσον έχει επιλεγεί η σύνδεση, παρόμοια με τη χρήση του κουμπιού «Web Service», μπορεί να επιλέξει από τη λίστα με τα διαθέσιμα web Services.

Πατώντας αποστολή «Send», αν το σύστημα δεν μπορεί να συνδεθεί με το web service θα εμφανίσει μήνυμα λάθους όπως αυτό φαίνεται στην Εικόνα 3.8.4., και θα εμφανίσει το επόμενο παράθυρο διαλόγου (όπως θα δούμε πιο κάτω) με τα επιστρεφόμενα XML δεδομένα κενό.



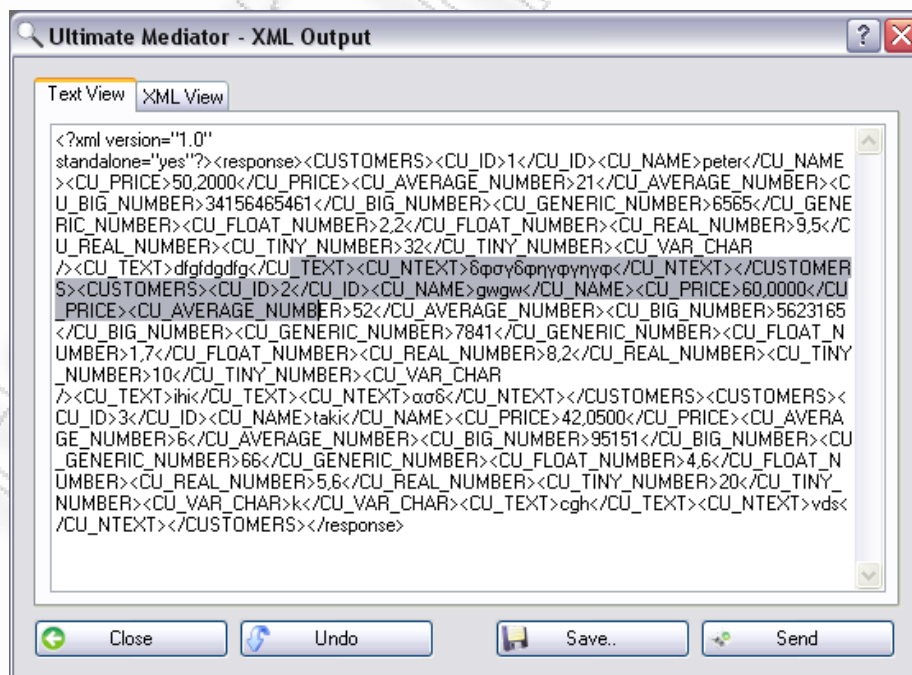
Εικόνα 3.8.4 Μήνυμα σφάλματος όταν το σύστημα δεν μπορεί να επικοινωνήσει με κάποιο Web Service

Αν είναι επιτυχής η σύνδεση και η αποστολή των δεδομένων, θα εμφανιστεί το παράθυρο διαλόγου της Εικόνα 3.8.5. Τα δεδομένα ανοίγουν σε νέα φόρμα και προβάλλονται μέσα από ειδικές οθόνες, προβολής, επεξεργασίας και αποθήκευσης δεδομένων.



Εικόνα 3.8.5 Οθόνη εμφάνισης XML δεδομένων στη δενδρική τους μορφή

Επειδή τα δεδομένα στην προηγούμενη εικόνα ήταν τύπου XML το Tab «XML View», ενεργοποιήθηκε αυτόματα. Πέραν της δενδρικής μορφής τα δεδομένα μπορεί να τα δει και σαν απλό κείμενο όπου εκεί μπορεί να επιφέρει αλλαγές αν το επιθυμεί. Για να δει τα δεδομένα σαν απλό κείμενο αρκεί η χρήση του «Text View» (Εικόνα 3.8.6).



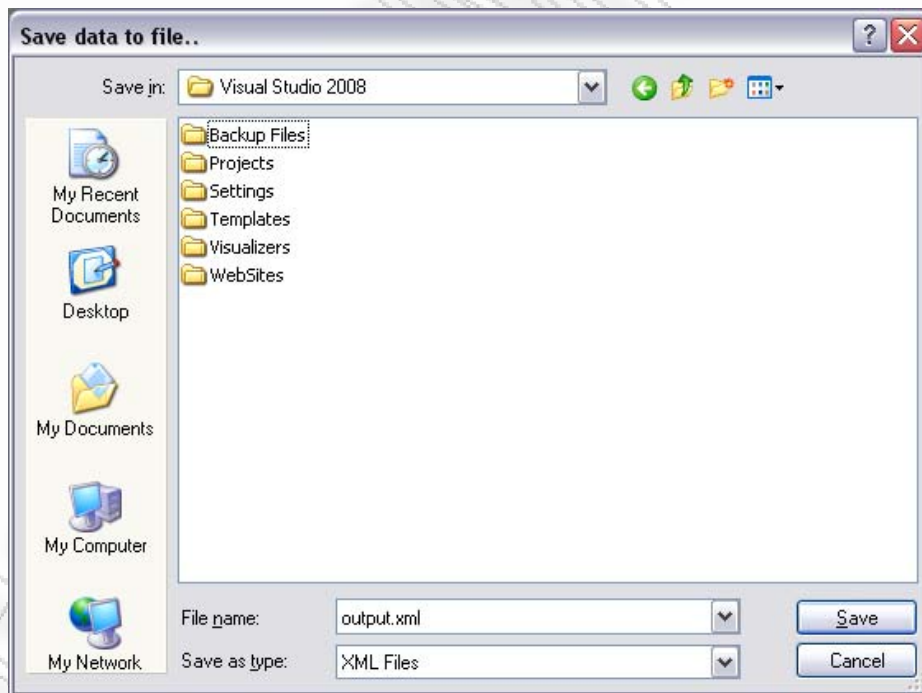
Εικόνα 3.8.6 Οθόνη εμφάνισης XML δεδομένων σε μορφή κειμένου

Αν χρειάζεται να γίνει αλλαγή των δεδομένων ο χρήστης μπορεί, στην απλή προβολή κειμένου των δεδομένων, να κάνει ότι αλλαγές θέλει. Ένας μικρός έλεγχος για διευκόλυνση του χρήστη, έτσι ώστε να δει αν οι αλλαγές είναι σωστές, είναι ο ακόλουθος.

Εφόσον έχει επεξεργαστεί το κείμενο, πατώντας στο Tab «XML View» αν του εμφανίσει το επιθυμητό XML, τότε σημαίνει ότι οι αλλαγές που έκανε ήταν έγκυρες. Αν δεν είναι έγκυρες συνήθως δεν εμφανίσει κανένα Element του XML, παρά μόνο αυτά τα οποία δεν εξαρτιόντουσαν από τις αλλαγές.

Μία ακόμα βοήθεια που προσφέρει το πρόγραμμα είναι η επιστροφή των δεδομένων στην αρχική κατάσταση. Αυτό μπορεί να πραγματοποιηθεί μέσα από το κουμπί «Undo». Έτσι αν ο χρήστης κάνει αρκετές αλλαγές οι οποίες δεν είναι έγκυρες, μπορεί να πάρει πάλι τα δεδομένα όπως ήταν αρχικά πριν να κάνει τις αλλαγές.

Η φόρμα αυτή δίνει επιπλέον δυνατότητες όπως αυτή της αποθήκευσης των δεδομένων σε αρχείο. Στην περίπτωση που το παράθυρο προήλθε μετά από τη μεταφορά δεδομένων σε XML, μέσα από τη φόρμα σχεδιασμού, τότε τα δεδομένα είναι έτσι και αλλιώς αποθηκευμένα στον «output» προορισμό που έχει ορίσει στις Συνδέσεις (*Manage Connections*). Στην περίπτωση της αποστολής XML σε Web Service, τα δεδομένα είναι η απάντηση που δίνει το Web Service. Σε αυτό το σημείο η απάντηση δεν αποθηκεύεται απευθείας πουθενά. Θα πρέπει να χρησιμοποιήσει ο χρήστης το κουμπί «Save as...». Έτσι θα εμφανιστεί ένα άλλο παράθυρο το οποίο θα δίνει τη δυνατότητα να ορίζεται το σημείο και το όνομα αποθήκευσης (Εικόνα 3.8.7).



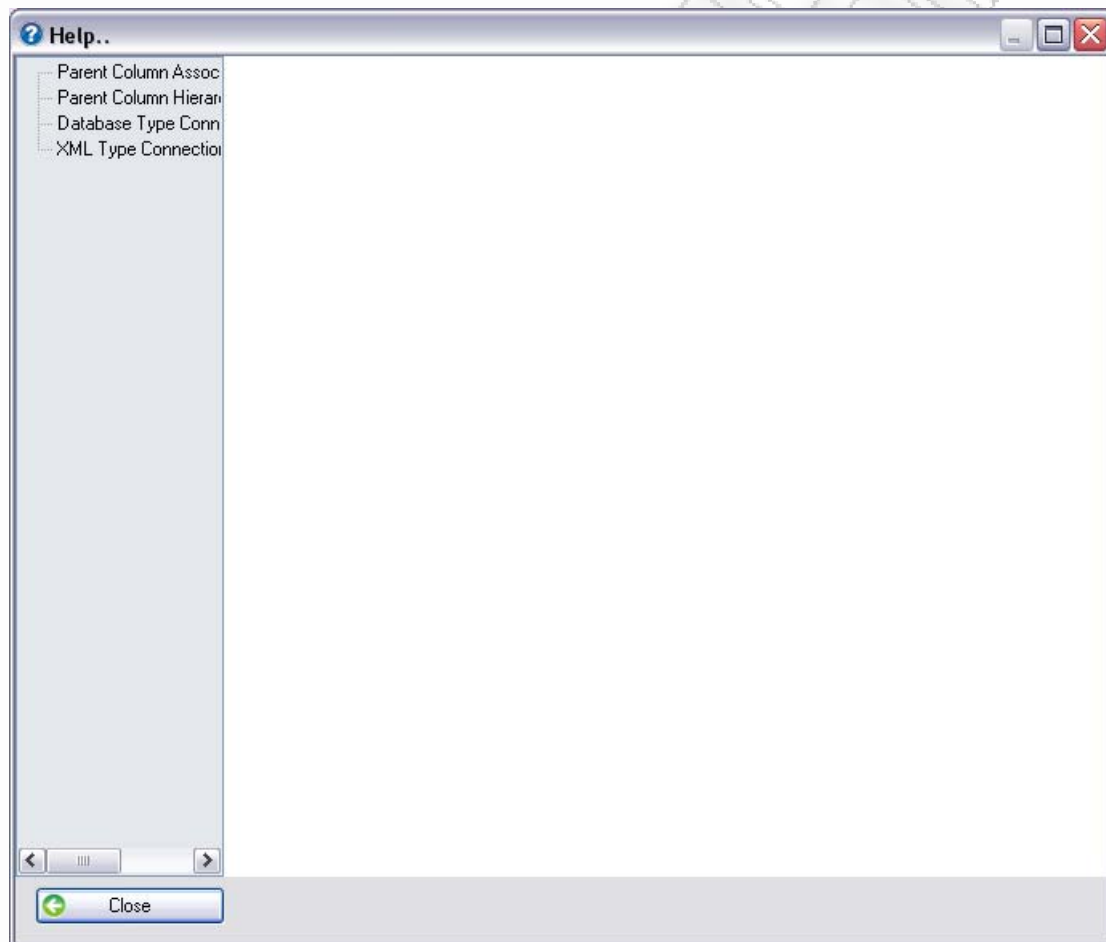
Εικόνα 3.8.7 Παράθυρο επιλογής αρχείου για αποθήκευση δεδομένων

Μία τελευταία δυνατότητα που παρέχεται μέσα από αυτήν τη φόρμα είναι η αποστολή των δεδομένων σε Web Service. Αυτό γίνεται χρησιμοποιώντας το κουμπί «Send». Εκεί ανοίγει πάλι η διαχειριστική φόρμα επιλογής Web Service (Εικόνα 3.6.2). Εφόσον επιλέξει να σταλούν τα δεδομένα, η απάντηση θα αναρτηθεί σε ένα

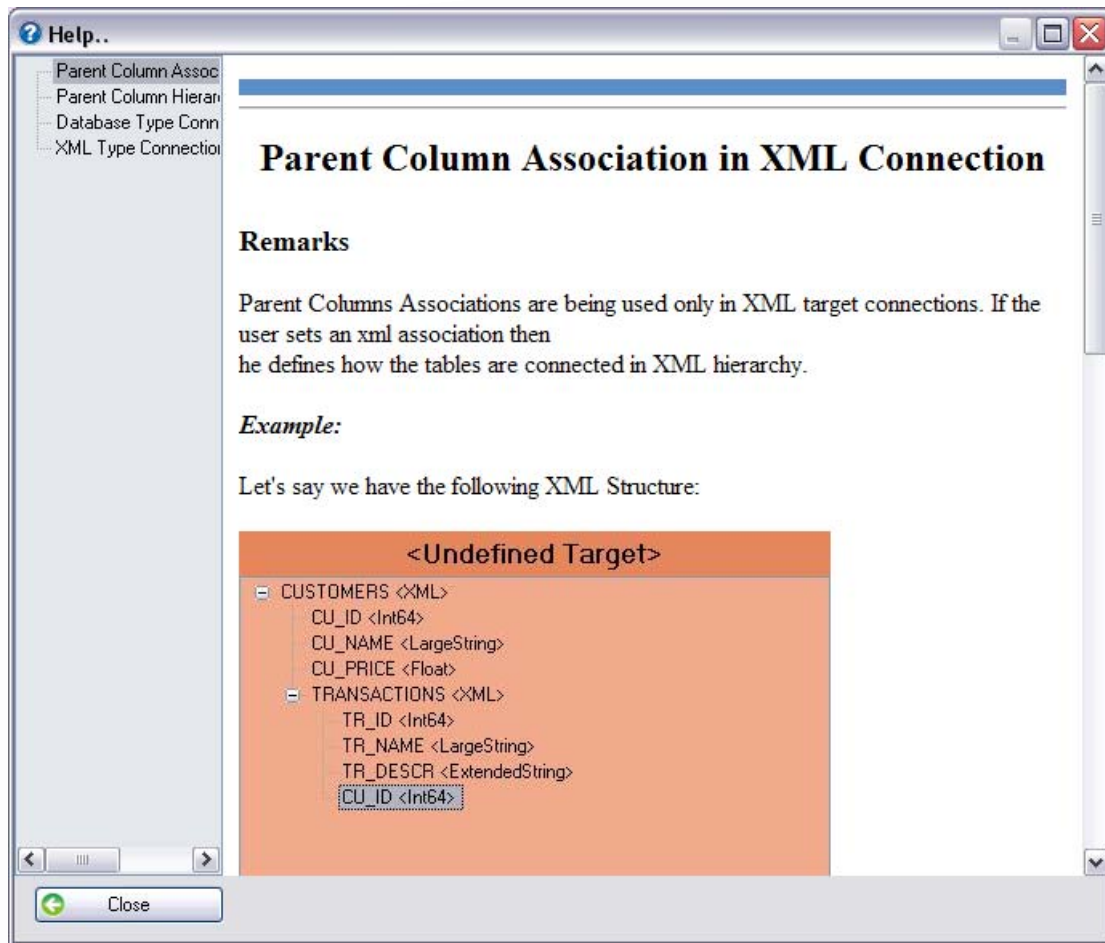
καινούργιο ίδιο με αυτό της Εικόνα 3.8.5, από όπου θα έχει όλες τις δυνατότητες που αναφέρθηκαν παραπάνω, μαζί με την ξανά αποστολή των δεδομένων σε Web Service. Αυτή η διαδικασία συνεχόμενης επιλογής / προβολής / επεξεργασίας / αποστολής δεδομένων δίνει πολλές δυνατότητες σε χρήστες που επιθυμούν να πραγματοποιούν πολλαπλές μεταφορές δεδομένων.

3.9 Help και About

Η χρήση της βοήθειας είναι πολύ σημαντική για μία εφαρμογή. Ειδικά όταν αυτή περιέχει δύσκολες έννοιες που μπορούν να προκαλέσουν σύγχυση στο χρήστη του προγράμματος. Για το λόγο αυτό χρησιμοποιήθηκε και εδώ η λειτουργία της βοήθειας σε κάποια κομμάτια του προγράμματος. Από το μενού που υπάρχει όταν ανοίγουμε την εφαρμογή, μπορούμε να ανοίξουμε όλα τα έγγραφα βοήθειας που είναι διαθέσιμα. Ανοίγοντας, θα δούμε την Εικόνα 3.9.1. όπου αριστερά υπάρχουν τα διαθέσιμα topics. Εκεί ο χρήστης μπορεί να τα επιλέξει για να δει τη βοήθεια στον κενό χώρο στα δεξιά (Εικόνα 3.9.1). Το παράθυρο διαλόγου μπορεί να μεγαλώσει όσο επιθυμείται για να γίνεται καλύτερη η ανάγνωση της βοήθειας.

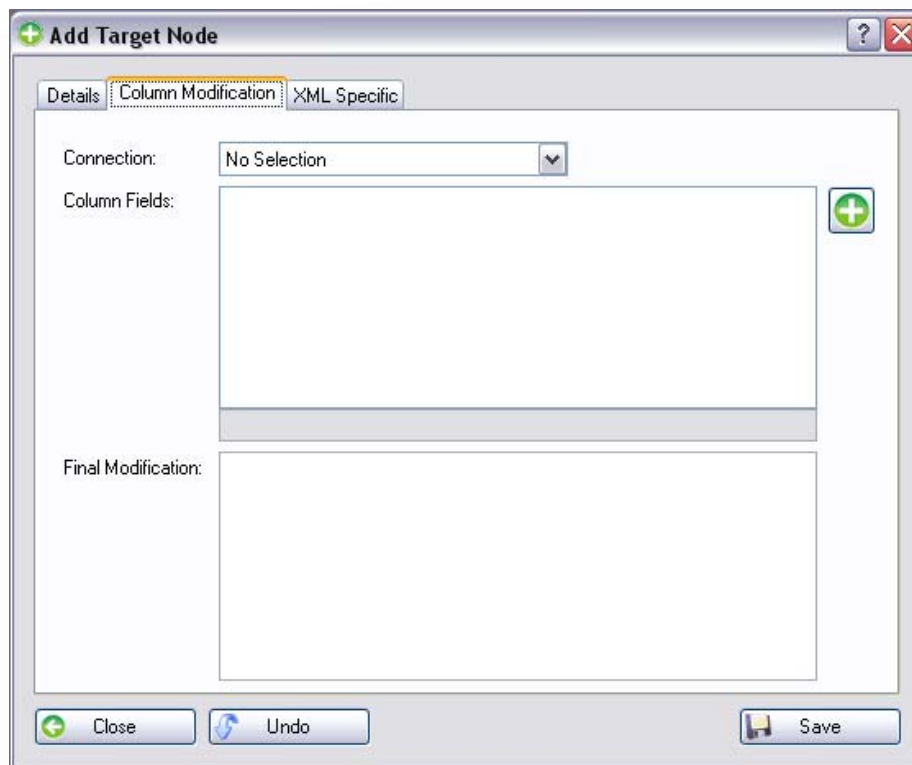


Εικόνα 3.9.1 Οθόνη βοήθειας με όλα τα διαθέσιμα topics



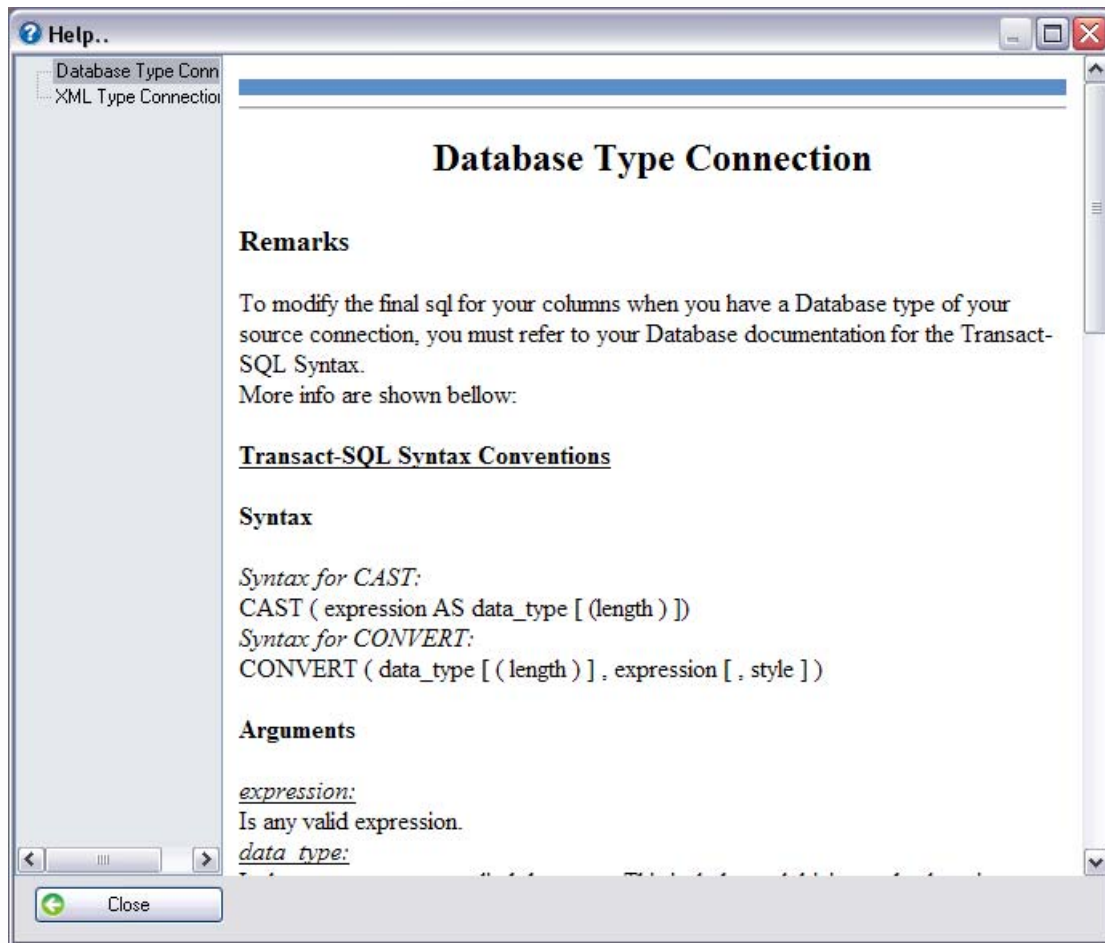
Εικόνα 3.9.2 Προβολής επιλεγμένης βοήθειας στον κενό χώρο

Ο χρήστης μπορεί όμως να προσπελάσει τη βοήθεια, απευθείας, μέσα από κάποιο πεδίο κάποιας φόρμας. Έτσι, αν για παράδειγμα κατά την προσθήκη ενός νέου κόμβου, ο χρήστης επιθυμεί να δει πως μπορεί να γράψει την παράσταση για να πάρει κάποια τιμή ο κόμβος, τότε μπορεί να επιλέξει από το πάνω δεξιά μέρος του παραθύρου το ερωτηματικό (Εικόνα 3.9.3).



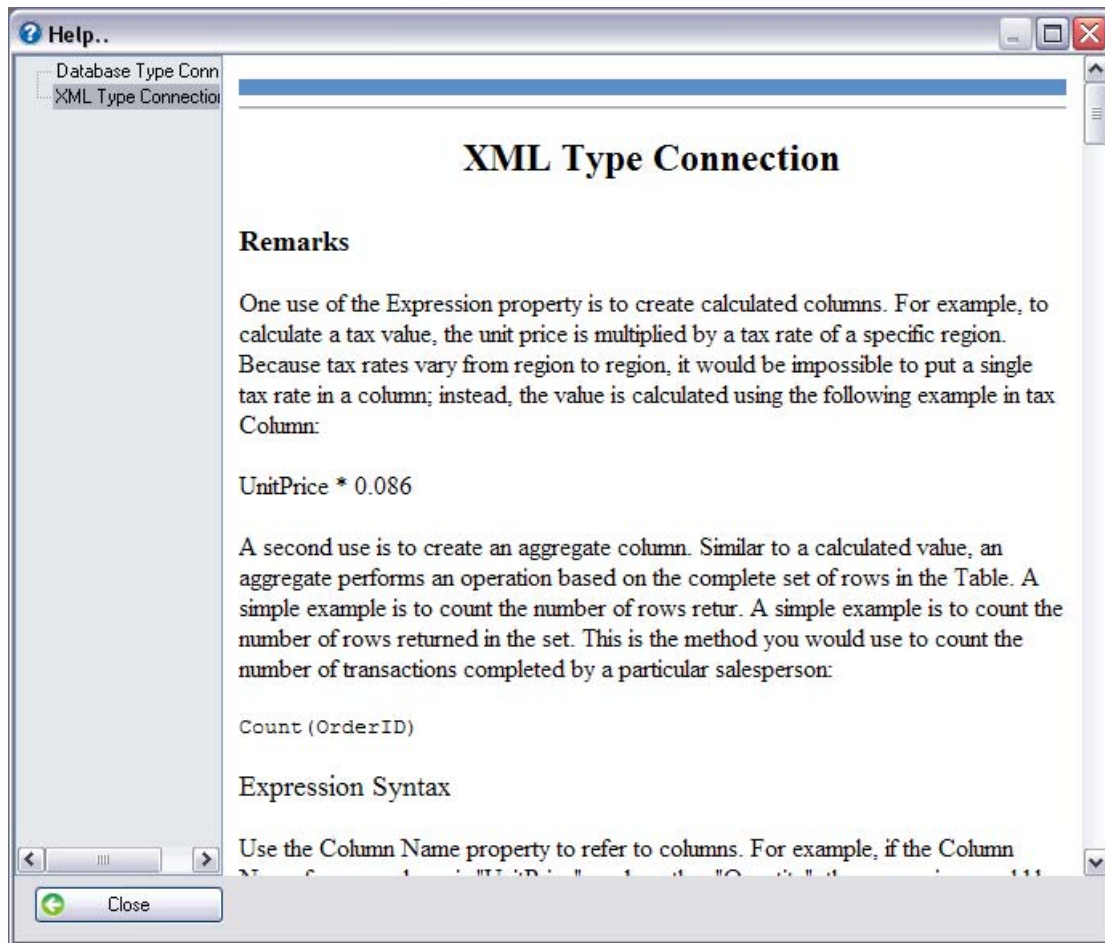
Εικόνα 3.9.3 Οθόνη που περιέχει το ερωτηματικό για την προβολή βοήθειας κάποιου πεδίου

Πατώντας το ερωτηματικό αυτομάτως ο δείκτης του ποντικού αλλάζει σε δείκτη με ερωτηματικό, δίνοντας έτσι την ευκαιρία να επιλέξει κάποιο πεδίο επάνω στη φόρμα στο οποίο θέλει να λάβει βοήθεια. Αν πατήσει για παράδειγμα το πεδίο Final Modification, τότε θα του εμφανιστεί το παράθυρο βοήθειας (Εικόνα 3.9.4).



Εικόνα 3.9.4 Προβολή βοήθειας μετά από την επιλογή του πεδίου Final Modification

Από ό,τι μπορούμε να δούμε, το παράθυρο αυτό στα αριστερά έχει δύο κείμενα βοήθειας και όχι ένα. Αυτό μπορεί να συμβεί σε πολλά πεδία, και ο λόγος είναι γιατί τα πεδία αυτά μπορεί να διαφοροποιούνται ανάλογα με τις συνθήκες με τις οποίες έγιναν οι κλήση τους. Στην περίπτωση αυτή το πεδίο Final Modification, χρησιμοποιεί άλλο είδος παράστασης όταν το τελικό σχήμα είναι XML, και άλλο όταν είναι Βάση. Ανάλογα με το τι είναι, θα προ-επιλεγθεί η κατάλληλη βοήθεια, και θα δοθεί η δυνατότητα μέσω της εμφάνισης και των άλλων, να γίνει η επιλογή και για τις άλλες παραπλήσιες βοήθειες (Εικόνα 3.9.5).



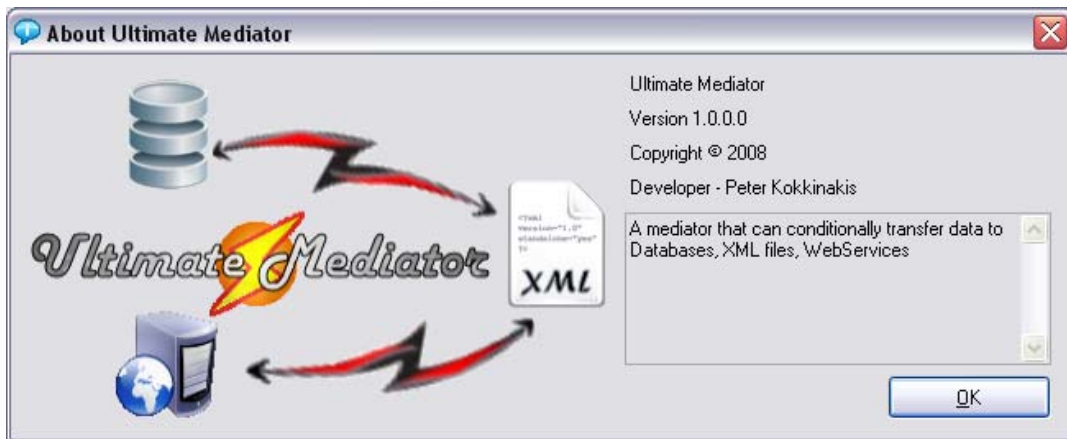
Εικόνα 3.9.5 Επιλογή άλλης παραπλήσιας βοήθειας

Σε αυτήν την έκδοση του λογισμικού υπάρχουν στο σύνολο τέσσερα αρχεία βοήθειας για τρία πεδία. Αν η επιλογή του πεδίου δεν περιέχει κάποιο τέτοιο αρχείο, τότε θα εμφανιστεί ένα μήνυμα στο χρήστη, υποδηλώνοντας ότι δεν υπάρχει διαθέσιμη βοήθεια (Εικόνα 3.9.6).



Εικόνα 3.9.6 Μήνυμα που δείχνει ότι δε βρέθηκε βοήθεια για το πεδίο αυτό

Το About είναι μία φόρμα η οποία δείχνει πληροφορίες για το πρόγραμμα, όπως έκδοση, έτος παραγωγής, Προγραμματιστής, κ.λ.π.. Η Εικόνα 3.9.7, δείχνει την φόρμα About.



Εικόνα 3.9.7 Φόρμα About του Ultimate Mediator

4 Περιπτώσεις χρήσης

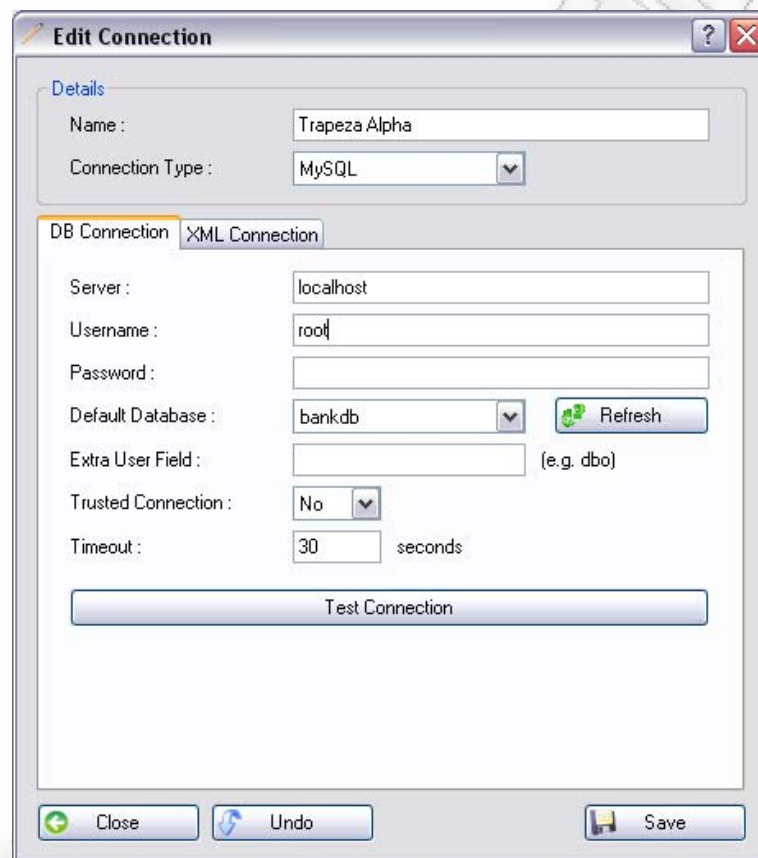
Σε αυτό το κεφάλαιο θα παρουσιαστούν κάποιες περιπτώσεις χρήσης της εφαρμογής, για τη μεταφορά δεδομένων. Η εστίαση θα γίνει κυρίως σε δοκιμαστικά μοντέλα, τα οποία μπορεί να προκύψουν από ανάγκες μιας εταιρίας ή οργανισμού, για τη διασύνδεση προγραμμάτων. Τα προγράμματα αυτά μπορεί να είναι προγράμματα αποθήκης, διαχείρισης πελατολογίου όπως CRM κ.λ.π. Μέσα από τα παραδείγματα θα παρουσιαστούν κάποιες διαδικασίες με τις οποίες μπορεί να γίνει Data Migration.

Στην ενότητα *Μεταφορά από Βάση σε XML*, θα γίνει λεπτομερής αναφορά όλης της διαδικασίας για την εκτέλεση της μεταφοράς. Πολλά κομμάτια της διαδικασίας, επειδή θα είναι κοινά (όπως δημιουργία σύνδεσης στη βάση δεδομένων), δεν θα παρουσιαστούν στις επόμενες ενότητες. Παρόλα αυτά αν ο αναγνώστης επιθυμεί περισσότερες πληροφορίες, τότε θα πρέπει να ανατρέξει στην αντίστοιχη ενότητα του εγχειριδίου σχετικά με το θέμα που τον ενδιαφέρει.

Η πρώτη ενότητα θα ξεκινήσει με μία περιληπτική διαδικασία, της μεταφοράς δεδομένων. Επειδή όμως πολλές ενέργειες θα είναι κοινές και για τις άλλες ενότητες, για λόγους ευκολίας θα παραληφθούν και θα παρουσιάζονται μόνο συνοπτικά.

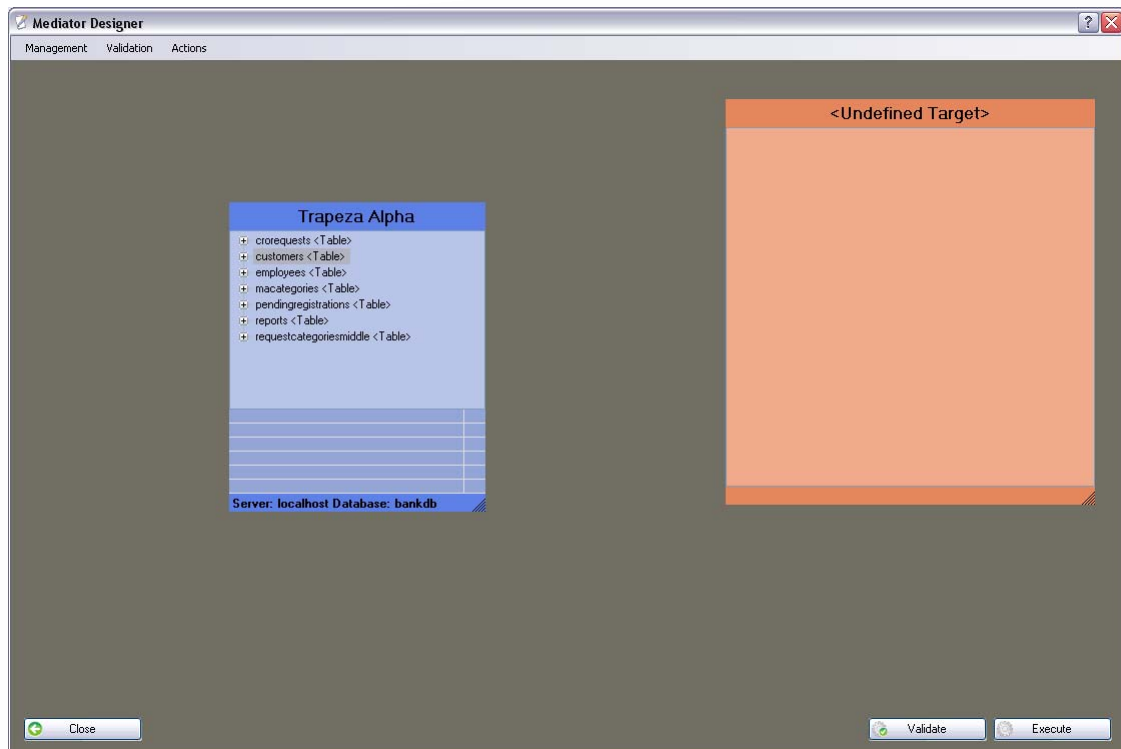
4.1 Μεταφορά από Βάση σε XML

Ας υποθέσουμε ότι υπάρχει η ανάγκη να περαστούν όλα τα δεδομένα ενός πίνακα σε ένα αρχείο XML, ώστε να χρησιμοποιηθούν για την εισαγωγή τους σε ένα τρίτο σύστημα. Για να γίνει αυτό είναι απαραίτητο αρχικά να οριστεί η σύνδεση (έστω MySQL) από την οποία θα γίνει η λήψη των δεδομένων. Θα πρέπει να συμπληρωθούν όλα τα απαιτούμενα πεδία, και εφόσον γίνει το τεστ για να επιβεβαιωθεί ότι τα στοιχεία σύνδεσης είναι σωστά, τότε αποθηκεύεται (Εικόνα 4.1.1).



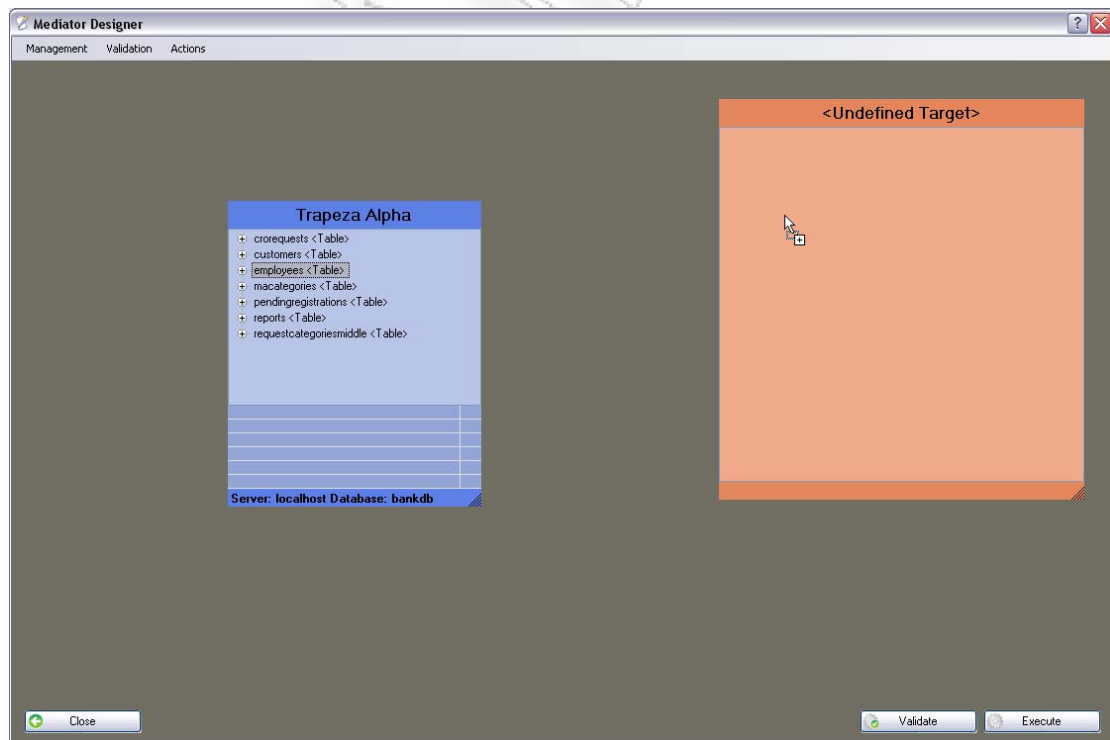
Εικόνα 4.1.1 Εμφάνιση σύνδεσης για τη βάση μίας τράπεζας

Εφόσον έχει οριστεί η σύνδεση, ο χρήστης μπορεί να ανοίξει τη φόρμα σχεδιασμού (Designer), και να επιλέξει τη σύνδεση από το Add Connection μενού. Αυτομάτως θα εμφανιστούν στη φόρμα σχεδιασμού, όλοι οι διαθέσιμοι πίνακες, που περιέχει η σύνδεση (Εικόνα 4.1.2).

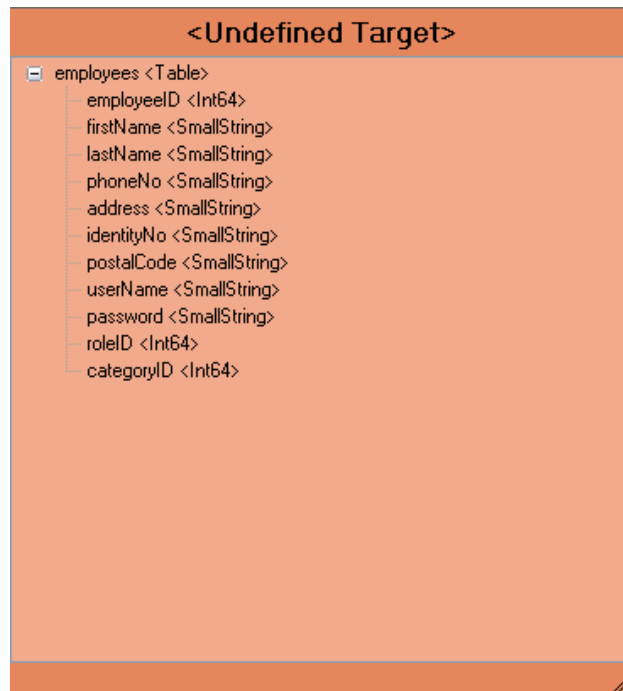


Εικόνα 4.1.2 Εικόνα προβολής πινάκων της σύνδεσης «Trapeza Alpha»

Από τη φόρμα σχεδιασμού ο χρήστης επιλέγει τον πίνακα «customers», και τον σέρνει στο τελικό σχήμα (Εικόνα 4.1.3). Έτσι το τελικό σχήμα περιέχει όλη τη δομή των customers (Εικόνα 4.1.4).

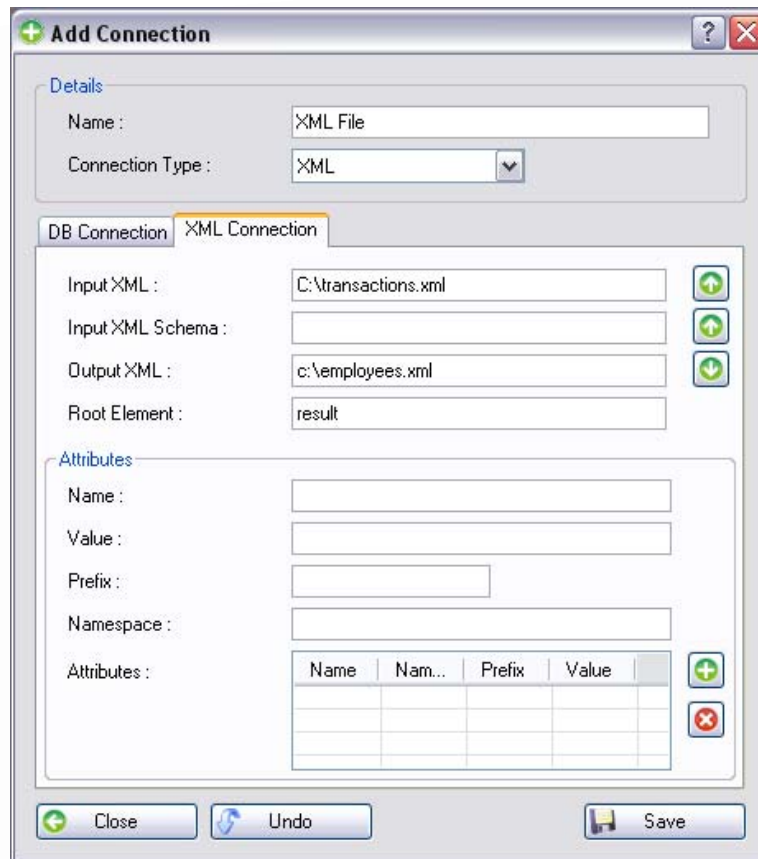


Εικόνα 4.1.3 Μεταφορά του πίνακα customers στο τελικό σχήμα



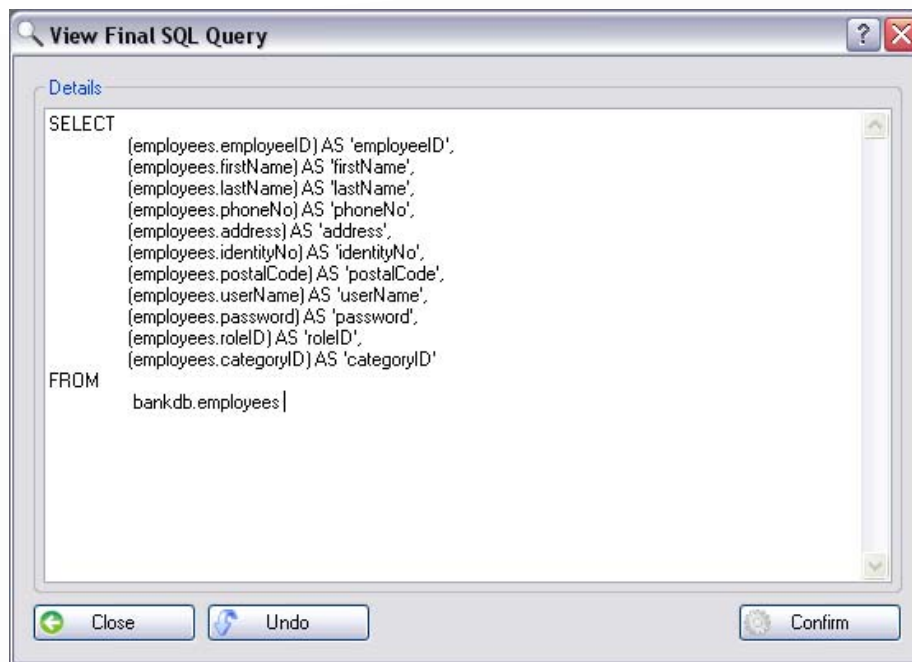
Εικόνα 4.1.4 Τελικό σχήμα με τον πίνακα customers

Έχοντας το τελικό σχήμα, δημιουργούμε την σύνδεση η οποία θα είναι τύπου xml (Εικόνα 4.1.5), και πατάμε το Validate για να βεβαιωθούμε ότι δεν υπάρχουν προβλήματα. Όπως έχει αναφερθεί και στο εγχειρίδιο χρήσης, ακόμα και αν ο χρήστης πατήσει Execute Now αμέσως, θα γίνει πρώτα έλεγχος για σφάλματα και αν δεν υπάρχουν τότε θα συνεχιστεί η διαδικασία (*Execution*).



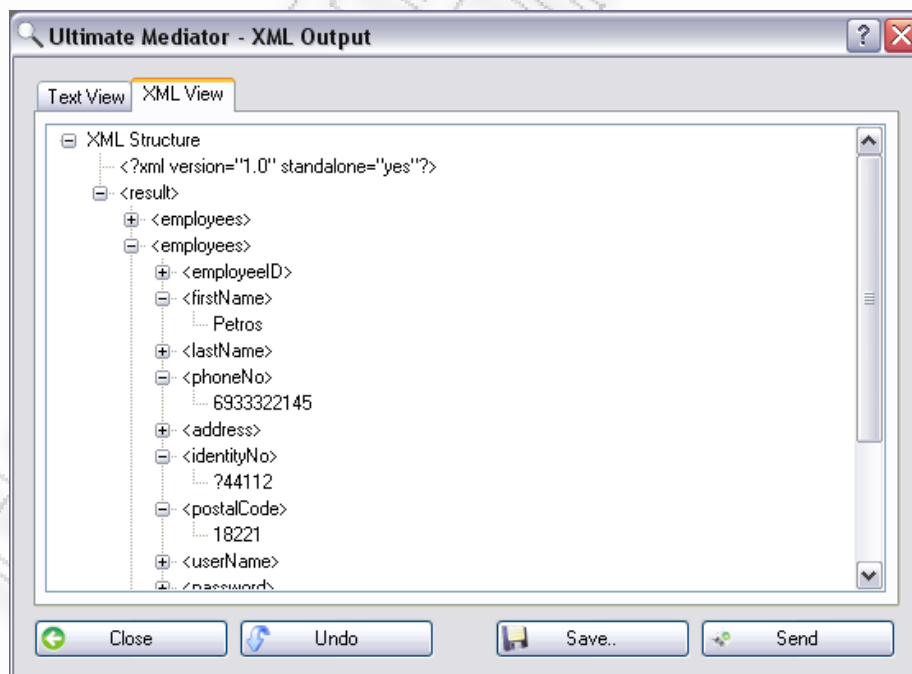
Εικόνα 4.1.5 Δημιουργία σύνδεσης τύπου XML

Πατώντας αμέσως μετά το Execute Now, θα εμφανιστεί η οθόνη που δείχνει το ερώτημα, σε sql, που θα εκτελεστεί στη βάση δεδομένων (Εικόνα 4.1.6). Εφόσον στην περίπτωση που εξετάζεται, δεν επιθυμείται καμία αλλαγή στα δεδομένα, παρά μόνο μεταφορά αυτών, ο χρήστης θα πρέπει να επιλέξει το «Confirm», για να εκτελεστεί η μεταφορά.



Εικόνα 4.1.6 Εμφάνιση τελικού sql ερωτήματος

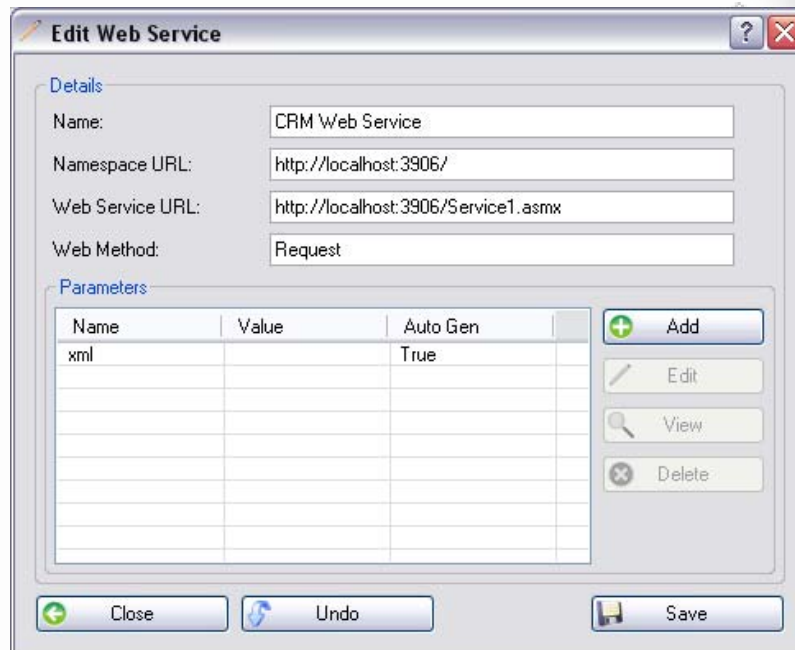
Το τελικό αποτέλεσμα της άνω διαδικασίας είναι να παραχθεί το XML αρχείο employees.xml, το οποίο να περιέχει τα δεδομένα που φαίνονται και στην οθόνη που εμφανίζει η εφαρμογή μετά το τέλος της μεταφοράς (Εικόνα 4.1.7).



Εικόνα 4.1.7 Το xml που δημιουργείται μετά το τέλος της μεταφοράς δεδομένων

Εφόσον τα δεδομένα έχουν ήδη παραχθεί σε xml, ολοκληρώνουμε τη μελέτη περίπτωσης με την αποστολή σε κάποιο web service. Έπειτα επιλέγουμε «Send»,

όπου εμφανίζεται η διαχειριστική φόρμα με τα web services. Δημιουργούμε μία νέα σύνδεση για web services (Εικόνα 4.1.8), και την επιλέγουμε σαν προορισμό. Αν η σύνδεση και η αποστολή των δεδομένων ολοκληρωθούν με επιτυχία, τότε θα ανοίξει ένα άλλο παράθυρο, το οποίο θα περιέχει την απάντηση του web service.



Εικόνα 4.1.8 Στοιχεία του web service που δημιουργήθηκε

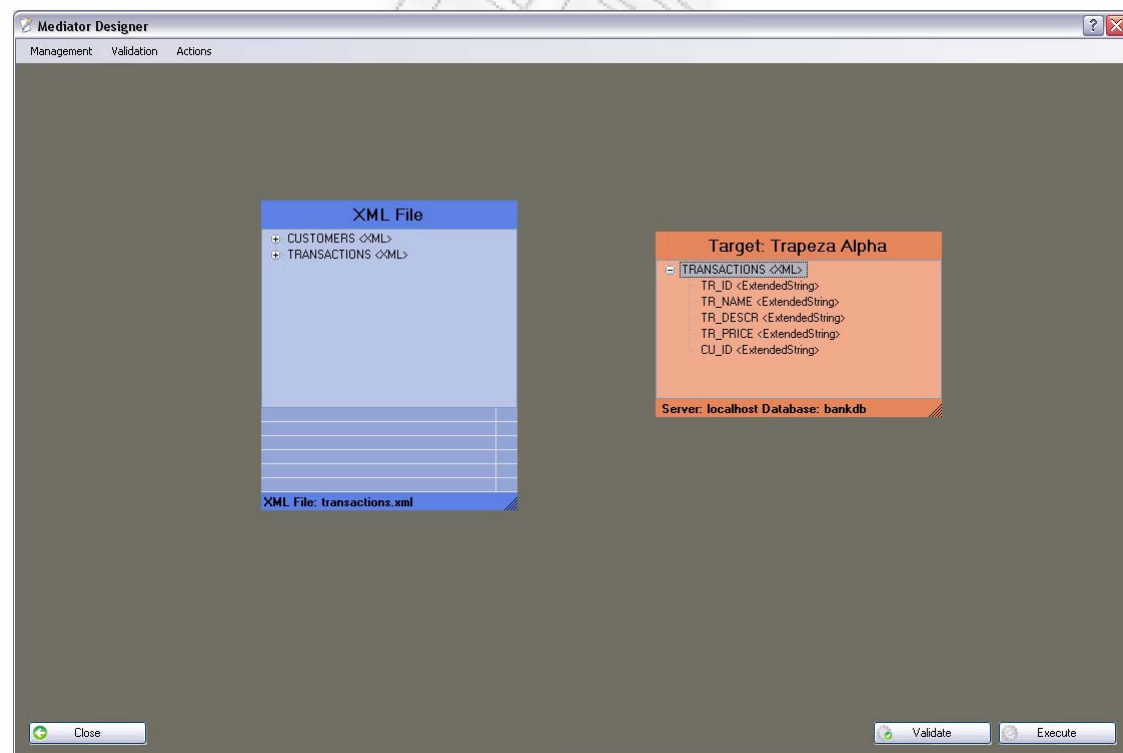
4.2 XML σε Βάση Δεδομένων

Σε αυτήν την ενότητα θα δοκιμάσουμε την περίπτωση όπου ένα σύστημα παράγει xml αρχεία, τα οποία θα πρέπει ένας χρήστης να τα στείλει σε κάποια βάση δεδομένων για κάποιο άλλο σύστημα (ή έστω για να τα χρησιμοποιήσει για κάποιον άλλο λόγο). Θεωρούμε ότι οι συνδέσεις για είσοδο / έξοδο xml αρχείου και για τη βάση MySQL έχουν ήδη οριστεί και αποθηκευτεί από την ενότητα *Μεταφορά από Βάση σε XML*.

Σημείωση:

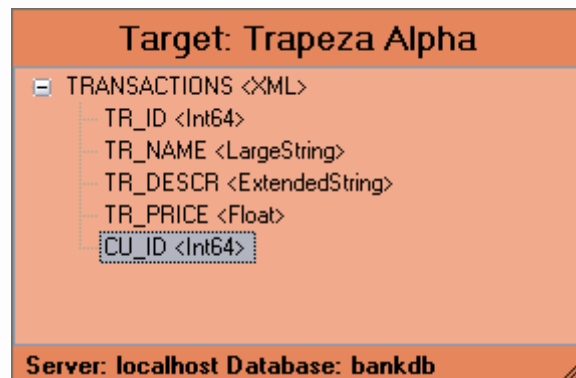
Στο παράδειγμα χρησιμοποιούμε τη βάση MySQL, αλλά θα μπορούσαμε να χρησιμοποιούσαμε και κάποια άλλη από τις διαθέσιμες βάσεις. Ο λόγος που χρησιμοποιείται η ίδια είναι για να μην παρουσιάζεται κάθε φορά η διαδικασία εισαγωγής μίας άλλης σύνδεσης. Σε επόμενες ενότητες θα χρησιμοποιηθεί και άλλη βάση, και ειδικά όταν παρουσιαστεί η περίπτωση μεταφοράς δεδομένων από βάση σε βάση.

Ο χρήστης εισάγει στη φόρμα σχεδίασης τη σύνδεση που είναι τύπου xml. Οι πίνακες και οι στήλες που περιέχονται, θα προβληθούν στη φόρμα σχεδίασης. Από εκεί μπορεί να κάνει την επιλογή του, έστω τον πίνακα «transactions», για να ορίσει το τελικό σχήμα (Εικόνα 4.2.1).



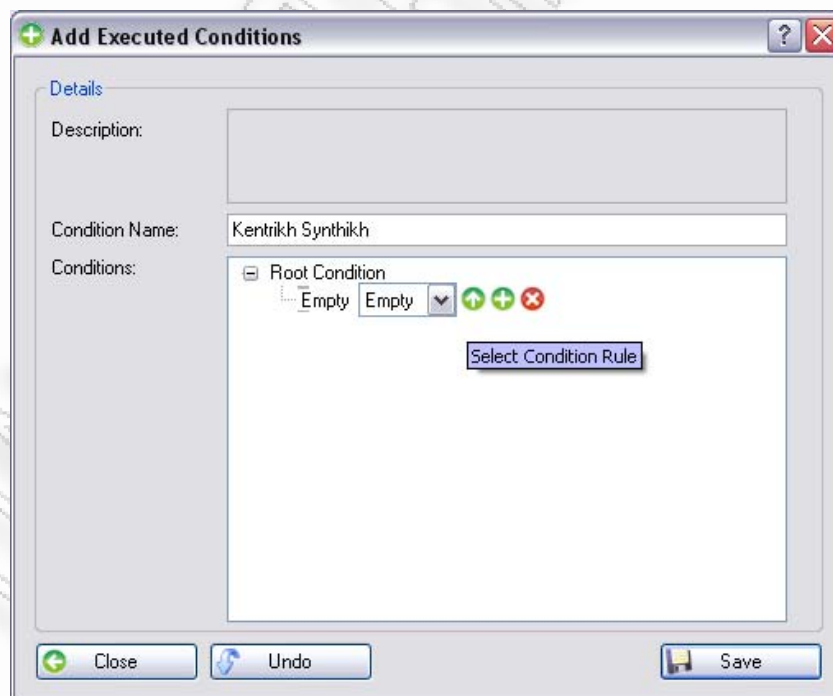
Εικόνα 4.2.1 Μεταφορά του πίνακα «TRANSACTIONS» στο τελικό σχήμα

Επειδή όμως η λήψη των δεδομένων έγινε από αρχείο xml, ο τύπος των δεδομένων δεν έχει οριστεί σωστά. Γι' αυτόν το λόγο θα πρέπει οι κόμβοι να επεξεργαστούν επάνω στο τελικό σχήμα και να οριστεί ο σωστός τύπος δεδομένων από το χρήστη (Εικόνα 4.2.2).



Εικόνα 4.2.2 Αλλαγή του τύπου δεδομένων στο τελικό σχήμα

Έστω όμως ότι είναι επιθυμητό να μη ληφθούν όλα τα δεδομένα του πίνακα «transactions», αλλά μόνο εκείνα τα οποία έχουν CU_ID (δηλ. το id του πελάτη) ίσο με 1. Θα πρέπει λοιπόν ο χρήστης να ορίσει μία συνθήκη η οποία να περιλαμβάνει μόνο αυτό το συγκεκριμένο id. Για να γίνει αυτό θα ανοίξει τη φόρμα διαχείρισης συνθηκών (Manage Conditions), και εκεί θα προσθέσει μία καινούργια (Εικόνα 4.2.3).



Εικόνα 4.2.3 Προσθήκη νέας ομάδας συνθηκών

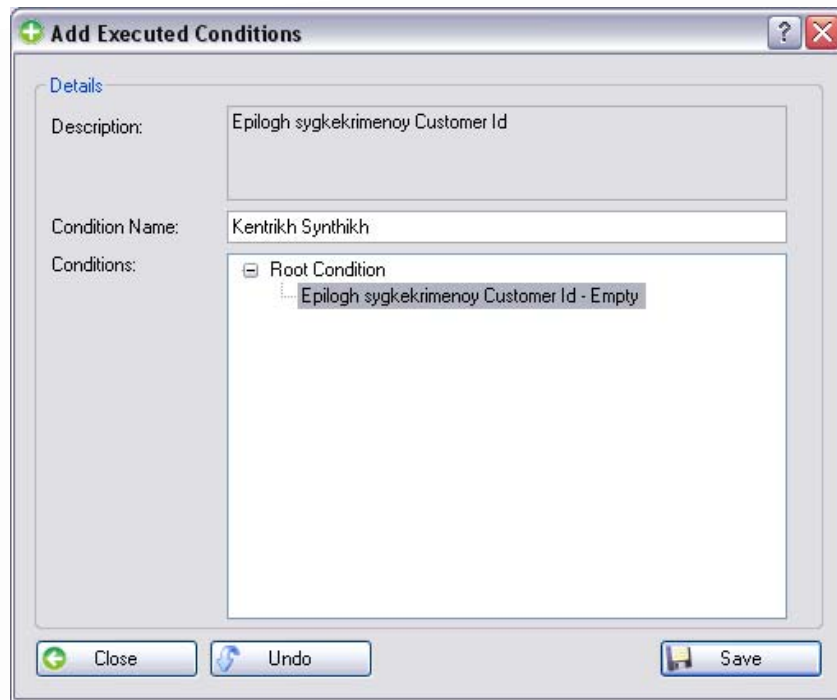
Από εκεί κάνει επιλογή απλής συνθήκης και επειδή δεν έχει οριστεί καμία, θα δημιουργήσει καινούργια μέσα από το κουμπί προσθήκης. Στη συγκεκριμένη φόρμα θα ορίσει τη νέα συνθήκη όπως αναφέρθηκε νωρίτερα (Εικόνα 4.2.4).

The screenshot shows the 'Add Condition' dialog box with the following details:

- Name:** Epiologh sygkekrimenoy Customer Id
- Select Field:** A list containing TR_NAME <LargeString>, TR_DESCR <ExtendedString>, TR_PRICE <Float>, and CU_ID <Int64> (selected).
- Condition:** equal to
- Static Value Provider:**
 - Data Type:** Int64
 - Static Value:** (empty field with a plus icon)
- Values:** A table with one row containing the value '1'.
- Buttons:** Close, Undo, and Save.

Εικόνα 4.2.4 Δημιουργία απλής συνθήκης με βάση το CU_ID

Εφόσον την αποθηκεύσει, την επιλέγει από τη λίστα και έτσι εμφανίζεται στην οθόνη με τις σύνθετες συνθήκες (Εικόνα 4.2.5). Εκεί πατάει αποθήκευση και κλείνει τη διαχειριστική φόρμα συνθηκών.



Εικόνα 4.2.5 Πλήρης συνθήκη έχοντας μόνο μία απλή συνθήκη

Πατώντας απευθείας «Execute Now», το σύστημα ακυρώνει τη μεταφορά γιατί βρήκε λάθη κατά τη διαδικασία ελέγχου εγκυρότητας του τελικού σχήματος.

Validation Messages				
	Description	Node	Parent	Extra
?	Table does not exist and it will be created.			TableInv...
✗	In a DB Connection you must set the root Field: 'TRANSACTIONS' type of <Table>.			

Εικόνα 4.2.6 Προβλήματα κατά τον έλεγχο τελικού σχήματος

Από τα προβλήματα που αναφέρονται μπορούμε να δούμε ότι ο πίνακας που ορίσαμε στο τελικό σχήμα δεν υπάρχει στην τελική βάση δεδομένων. Για να σιγουρέψουμε ότι όντως ισχύει αυτό και ότι δεν έχουμε γράψει απλά λάθος το όνομα, μπορούμε να δούμε τη δομή της βάσης. Η Εικόνα 4.2.7 δείχνει την δομή της MySQL βάσης.

Πίνακας	Ενέργεια	Εγγραφές	Τύπος	Collation	Μέγεθος	Επιβάρυνση
<input type="checkbox"/> corequests		4	MyISAM	latin1_swedish_ci	2,2 KB	-
<input type="checkbox"/> customers		2	MyISAM	latin1_swedish_ci	2,1 KB	-
<input type="checkbox"/> employees		6	MyISAM	latin1_swedish_ci	2,5 KB	-
<input type="checkbox"/> macategories		3	MyISAM	latin1_swedish_ci	2,1 KB	-
<input type="checkbox"/> pendingregistrations		0	MyISAM	latin1_swedish_ci	1,0 KB	-
<input type="checkbox"/> reports		0	MyISAM	latin1_swedish_ci	1,0 KB	-
<input type="checkbox"/> requestcategoriesmiddle		0	MyISAM	latin1_swedish_ci	1,0 KB	-
7 Πίνακας/Πίνακες	Σύνολο	15	MyISAM	latin1_swedish_ci	11,8 KB	0 Bytes

Εικόνα 4.2.7 Διαθέσιμοι πίνακες από την υπάρχουσα δομή της βάσης.

Εφόσον λοιπόν δεν υπάρχει ο πίνακας, τότε μπορούμε να επιβεβαιώσουμε τη δημιουργία του πίνακα. Κάνουμε λοιπόν διπλό κλικ επάνω στο συγκεκριμένο μήνυμα και το επιβεβαιώνουμε. Αν κοιτάξουμε τώρα ξανά τη βάση, θα δούμε ότι ο νέος πίνακας δημιουργήθηκε και τα πεδία του ορίστηκαν ακριβώς όπως τα όρισε ο χρήστης στο τελικό σχήμα (Εικόνα 4.2.8).

Πεδίο	Τύπος	Collation	Χαρακτηριστικά	Κενό	Προκαθορισμένο	Πρόσθετα	Ενέργεια
<input type="checkbox"/> TR_ID	bigint(20)			Ναι	NULL		
<input type="checkbox"/> TR_NAME	varchar(2000)	latin1_swedish_ci		Ναι	NULL		
<input type="checkbox"/> TR_DESCR	varchar(8000)	latin1_swedish_ci		Ναι	NULL		
<input type="checkbox"/> TR_PRICE	float			Ναι	NULL		
<input type="checkbox"/> CU_ID	bigint(20)			Ναι	NULL		

Εικόνα 4.2.8 Εμφάνιση νέου πίνακα που δημιουργήθηκε στην βάση δεδομένων

Το δεύτερο μήνυμα είναι το γεγονός ότι δεν μπορούμε να ορίσουμε ένα πεδίο πίνακα να είναι τύπου xml. Αυτό το διορθώνουμε εύκολα μέσω της επεξεργασίας του συγκεκριμένου κόμβου. Πατώντας λοιπόν «Execute Now», τα δεδομένα θα μεταφερθούν και θα γεμίσει ο πίνακας (Εικόνα 4.2.9).

← T →	TR_ID	TR_NAME	TR_DESCR	TR_PRICE	CU_ID
<input type="checkbox"/>	1	Trans #1	AGORA 10 SFOUGARAKIA	1226	1
<input type="checkbox"/>	3	Trans #2	AGORA MAXAIROIPIROYNA	12041	1

Εικόνα 4.2.9 Οθόνη ένδειξης ότι οι εγγραφές καταχωρήθηκαν

Για να επαληθεύσουμε ότι δεν έγινε η καταχώρηση όλων των εγγραφών, μπορούμε να δούμε ότι το κελί CU_ID περιέχει τιμές μόνο με αριθμό 1. Αν δούμε το xml αρχείο θα δούμε ότι υπήρχαν και άλλες τιμές που δεν μεταφέρθηκαν λόγω της συνθήκης (Εικόνα 4.2.10).

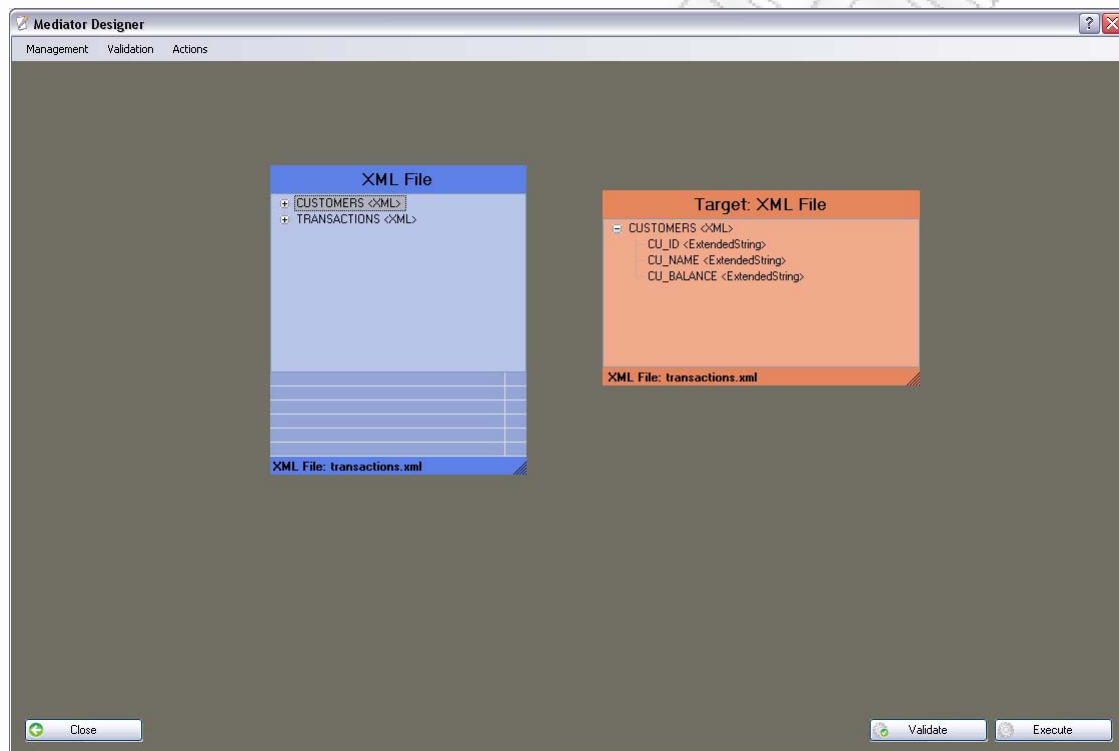
```
<?xml version="1.0" standalone="yes" ?>
- <requests>
- <CUSTOMERS>
  <CU_ID>1</CU_ID>
  <CU_NAME>Petros Kokkinakis</CU_NAME>
  <CU_BALANCE>50,2000</CU_BALANCE>
- <TRANSACTIONS>
  <TR_ID>1</TR_ID>
  <TR_NAME>Trans #1</TR_NAME>
  <TR_DESCR>AGORA 10 SFOUGARAKIA</TR_DESCR>
  <TR_PRICE>12.26</TR_PRICE>
  <CU_ID>1</CU_ID>
</TRANSACTIONS>
- <TRANSACTIONS>
  <TR_ID>3</TR_ID>
  <TR_NAME>Trans #2</TR_NAME>
  <TR_DESCR>AGORA MAXAIROIPIROYNA</TR_DESCR>
  <TR_PRICE>120.41</TR_PRICE>
  <CU_ID>1</CU_ID>
</TRANSACTIONS>
</CUSTOMERS>
- <CUSTOMERS>
  <CU_ID>2</CU_ID>
  <CU_NAME>Antwnis Papantwniou</CU_NAME>
  <CU_BALANCE>60,0000</CU_BALANCE>
- <TRANSACTIONS>
  <TR_ID>2</TR_ID>
  <TR_NAME>Trans #1</TR_NAME>
  <TR_DESCR>AGORA PATATES THIBAS 3 KILA</TR_DESCR>
  <TR_PRICE>1.5</TR_PRICE>
  <CU_ID>2</CU_ID>
</TRANSACTIONS>
</CUSTOMERS>
</requests>
```

Εικόνα 4.2.10 Τα δεδομένα του xml αρχείου που χρησιμοποιήθηκε

4.3 XML σε XML

Το σύστημα παρέχει επιπλέον τη δυνατότητα της τροποποίησης και μεταφοράς δεδομένων από xml αρχεία σε άλλα xml αρχεία. Η διαδικασία αυτή μπορεί να χρησιμοποιηθεί από πολλά συστήματα, μιας και η χρήση τους έχει αυξηθεί όλο και περισσότερο τα τελευταία χρόνια, τόσο που πάρα πολλά προγράμματα εισάγουν τη δυνατότητα εισαγωγής και εξαγωγής από xml.

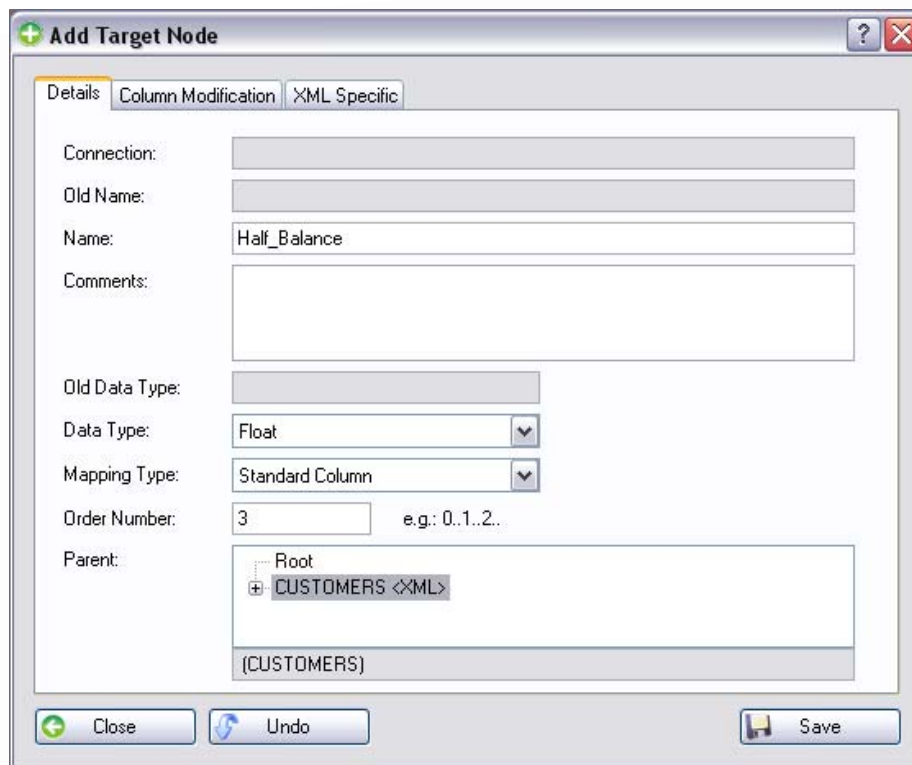
Έστω λοιπόν ότι χρησιμοποιούμε την ίδια σύνδεση για είσοδο και έξοδο xml δεδομένων, που φτιάξαμε στην πρώτη ενότητα. Σε αυτό το σημείο όμως μεταφέρουμε τα δεδομένα του πίνακα «customers», και τα περνάμε στο τελικό σχήμα (Εικόνα 4.3.1).



Εικόνα 4.3.1 Μεταφορά του xml πίνακα «customers»

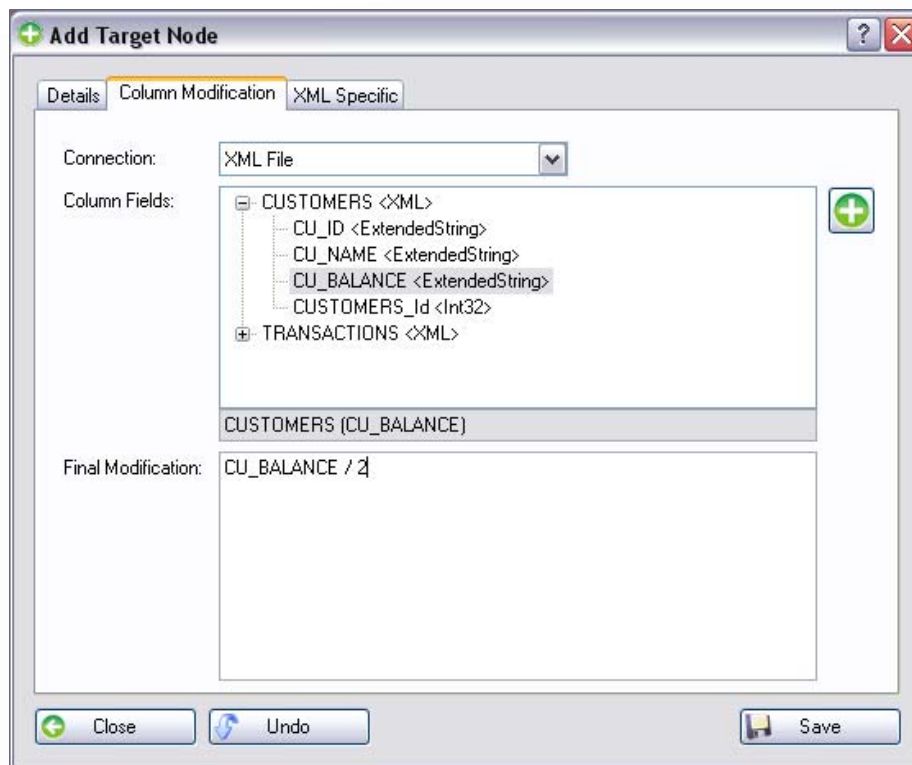
Έστω ότι ο χρήστης θέλει να ορίσει και ένα καινούργιο πεδίο στο τελικό σχήμα το οποίο να λέγεται «Half_Balance». Το νέο αυτό πεδίο θα πρέπει να παίρνει τιμές από το Balance και συγκεκριμένα, θα πρέπει να ορίσει να παίρνει το μισό του Balance, για τιμή. Το πεδίο αυτό θα πρέπει να είναι και δεκαδικός αριθμός, μιας και είναι πολύ πιθανό η διαίρεση να φέρει δεκαδικό μέρος.

Για να γίνει αυτό θα πρέπει αρχικά να πατήσουμε στον κενό χώρο του τελικού σχήματος (με δεξιά πλήκτρο) για να εμφανιστεί το μενού με τις λειτουργίες, και να επιλέξουμε τη δημιουργία νέου κόμβου. Στην οθόνη που θα εμφανιστεί θα ορίσουμε το νέο πεδίο ακριβώς όπως το περιγράψαμε παραπάνω.



Εικόνα 4.3.2 Προσθήκη νέου κόμβου σε xml

Εφόσον έχουμε ορίσει το νέο κόμβο, θα πρέπει να καθορίσουμε πως αυτό θα παίρνει τιμή. Για το λόγο αυτό επιλέγουμε από επάνω το tab «Column Modification», και εκεί στο πεδίο «Final Modification», ορίζουμε την παράσταση με την οποία αυτό θα πάρει τιμή (Εικόνα 4.3.3).



Εικόνα 4.3.3 Καθορισμός τιμής στο νέο κόμβο

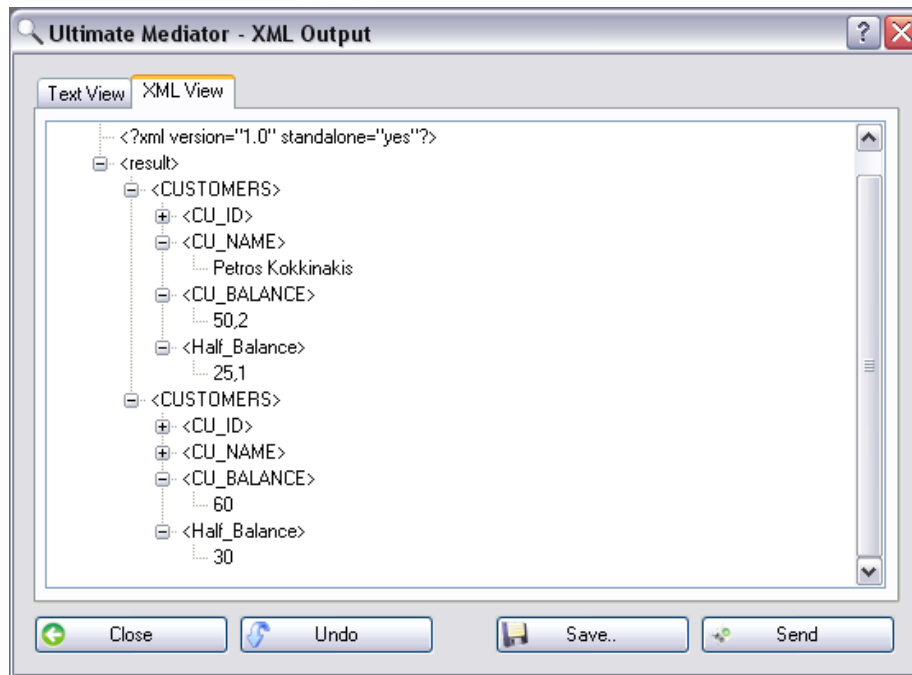
Όπως φαίνεται και από την Εικόνα 4.3.3, για διευκόλυνση δική μας, ορίζουμε τη σύνδεση και έπειτα επιλέγουμε το πεδίο που θέλουμε. Αυτό δεν είναι απαραίτητο. Γίνεται μόνο και μόνο αν η παράσταση χρειάζεται να χρησιμοποιήσει κάποιο άλλο πεδίο. Θα μπορούσε ο χρήστης να ορίσει απλά έναν αριθμό.

Σημείωση:

Τα πεδία που επρόκειτο να επιλεγθούν, θα πρέπει να ορίζονται στο τελικό σχήμα, όταν σε αυτό έχει οριστεί σύνδεση τύπου xml. Ο λόγος είναι γιατί η παράσταση υπολογισμού μπορεί να βρει μόνο αυτά τα πεδία. Αν όμως η σύνδεση είναι τύπου βάσης δεδομένων, τότε μπορεί να χρησιμοποιηθούν πεδία από τους πίνακες που υπάρχουν στη συγκεκριμένη σύνδεση.

Αν ο χρήστης επιθυμεί να ορίσει ένα πεδίο το οποίο δεν θέλει να εμφανιστεί στην έξοδο της σύνδεσης, τότε μπορεί να το βάλει στο τελικό σχήμα ορίζοντάς το ως κρυφό κόμβο.

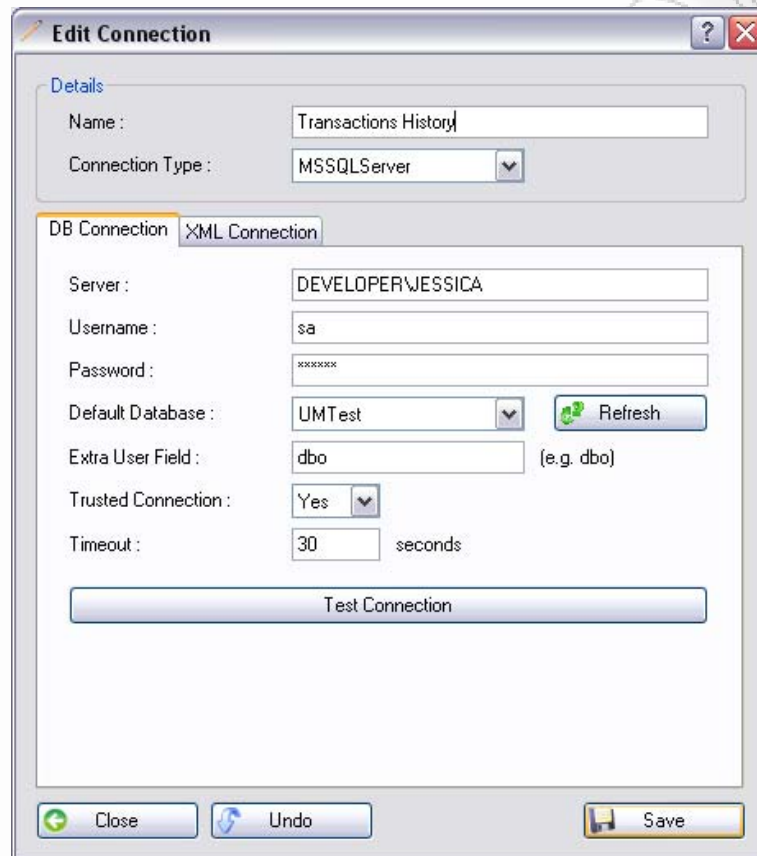
Το αποτέλεσμα της παραπάνω μεταφοράς είναι να δημιουργηθεί το xml στον προορισμό που υποδεικνύει η σύνδεση. Το xml προβάλλεται από το σύστημα και παρουσιάζεται στην Εικόνα 4.3.4. Από την οθόνη αυτή, μπορούμε να παρατηρήσουμε ότι ο νέος κόμβος «Half_Balance» πήρε ακριβώς τη μισή τιμή του πεδίου «Balance», και στους δύο επιστρεφόμενους «CUSTOMERS» πίνακες.



Εικόνα 4.3.4 Επιστρεφόμενο xml αποτέλεσμα που φαίνεται η τιμή του νέου πεδίου

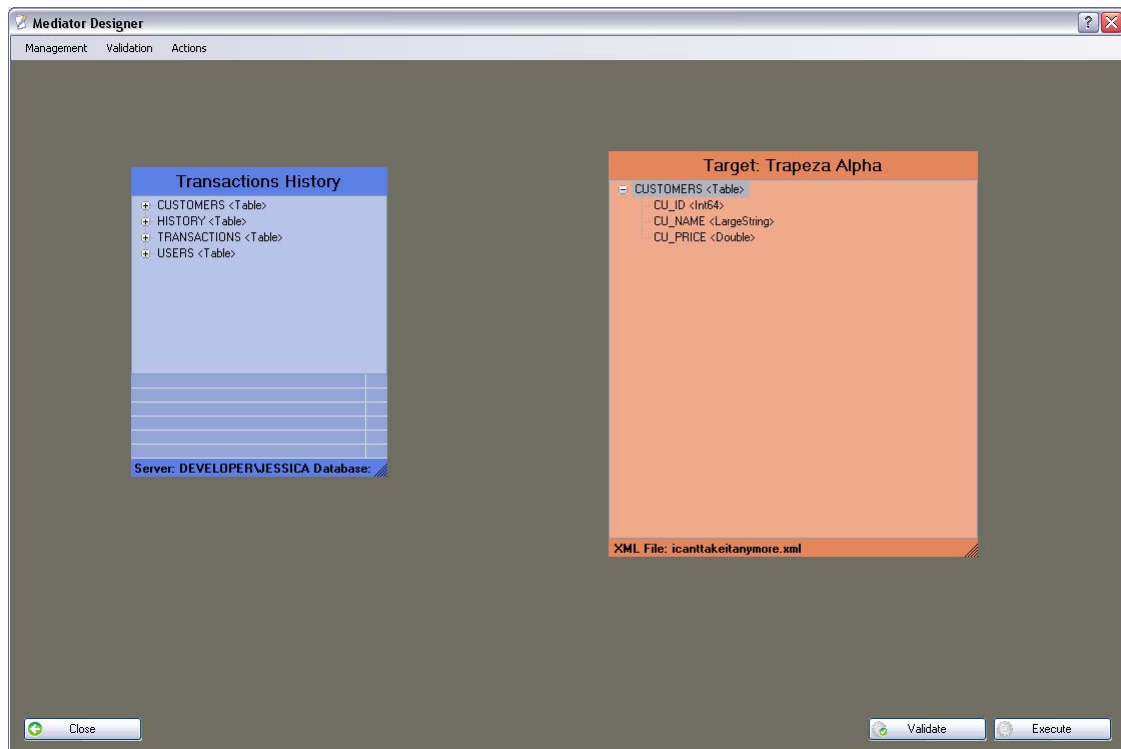
4.4 Μεταφορά δεδομένων από Βάση σε Βάση

Σε αυτήν την ενότητα θα εξετάσουμε το ενδεχόμενο μεταφοράς δεδομένων από μία βάση σε μία άλλη. Γι' αυτόν το λόγο δημιουργούμε και μία επιπλέον σύνδεση πέραν της MySQL, που είχαμε δημιουργήσει σε προηγούμενη ενότητα. Έστω ότι ο χρήστης δημιουργεί μία σύνδεση σε SQL Server (Εικόνα 4.4.1).



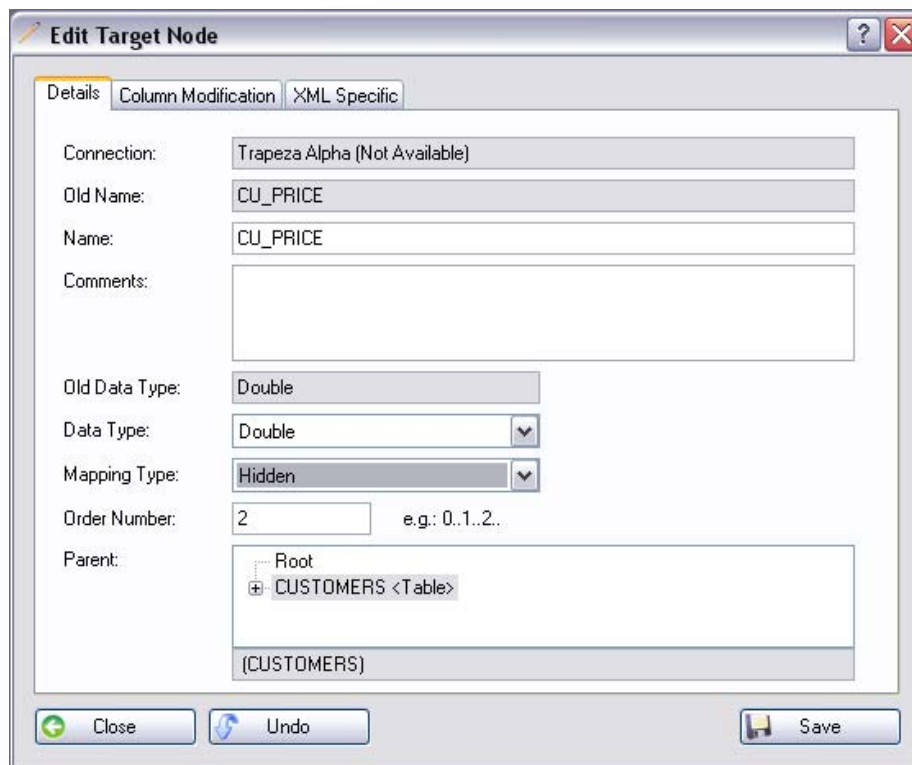
Εικόνα 4.4.1 Δημιουργία σύνδεσης με MSSQLServer

Ορίζει λοιπόν ο χρήστης τη σύνδεση στο τελικό σχήμα που θα είναι η MySQL, και επιλέγει να πάρει δεδομένα από τη σύνδεση στον MSSQLServer. Έπειτα μεταφέρει τον πίνακα «Customers» στο τελικό σχήμα (Εικόνα 4.4.2).



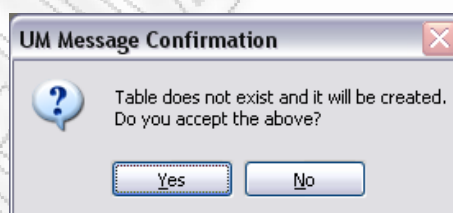
Εικόνα 4.4.2 Μεταφορά πίνακα customers στο τελικό σχήμα

Μπορεί ο χρήστης να επιθυμεί να το κάνει κρυφό και να μην το σβήσει από το τελικό σχήμα. Έτσι κάνει επεξεργασία τον κόμβο CU_PRICE και τον κάνει κρυφό «Hidden» (Εικόνα 4.4.3). Με αυτό τον τρόπο η έξοδος του τελικού σχήματος δεν θα περιλαμβάνει το συγκεκριμένο κόμβο.

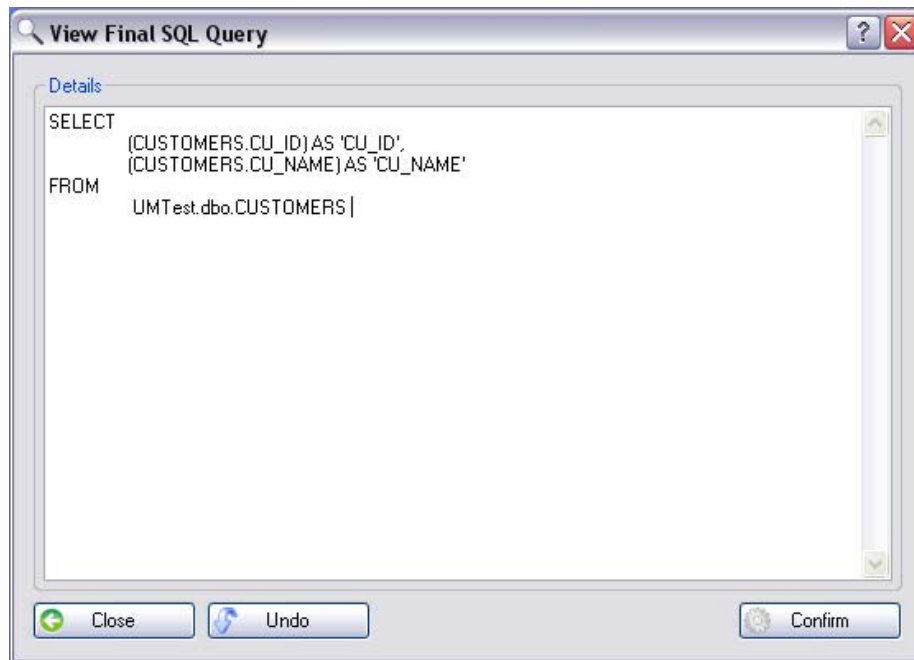


Εικόνα 4.4.3 Ορισμός κρυφού κόμβου

Εφόσον το ορίσει κρυφό, τότε πατάει το «Execute Now», αλλά επειδή ο πίνακας δεν υπάρχει, θα ερωτηθεί ο χρήστης για τη δημιουργία του (Εικόνα 4.4.4). Εφόσον επικυρώσει τη διαδικασία, θα εμφανιστεί ένα παράθυρο που θα δείχνει το τελικό ερώτημα στο χρήστη (Εικόνα 4.4.5). Αυτό μπορεί να φανεί πολύ χρήσιμο κυρίως όταν ο χρήστης επιθυμεί να κάνει κάποια τελευταία τροποποίηση επάνω στο ερώτημα sql, πριν αυτό εκτελεστεί.



Εικόνα 4.4.4 Ερώτημα επικύρωσης για δημιουργία πίνακα



Εικόνα 4.4.5 Εμφάνιση τελικού ερωτήματος για τη λήψη των δεδομένων

Εφόσον επιβεβαιώσει για τελευταία φορά το sql ερώτημα, η μεταφορά θα ξεκινήσει και μετά το πέρας της διαδικασίας θα εμφανιστεί το μήνυμα στην Εικόνα 4.4.6. Έπειτα μπορούμε να πάμε στη βάση δεδομένων της MySQL, και να δούμε ότι δημιουργήθηκε ένας πίνακας με δύο στήλες μόνο, χωρίς να περιλαμβάνεται η CU_PRICE (Εικόνα 4.4.7).



Εικόνα 4.4.6 Μήνυμα επιτυχούς αποστολής δεδομένων

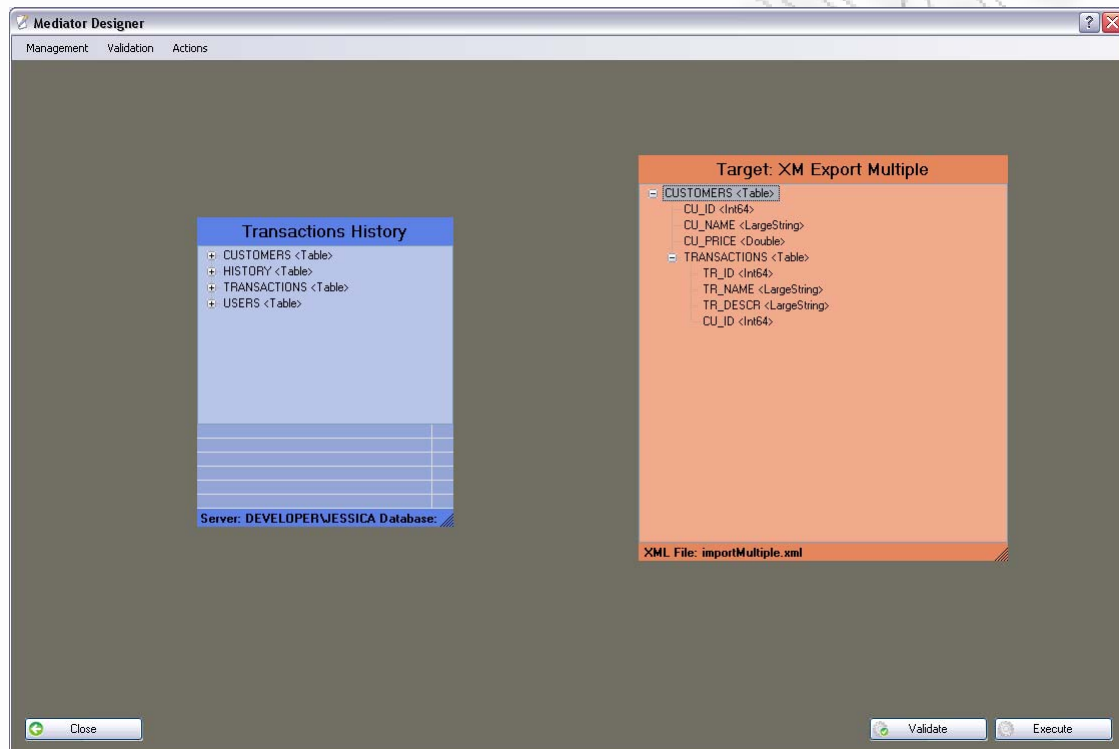
← T →			CU_ID	CU_NAME
<input type="checkbox"/>			1	peter
<input type="checkbox"/>			2	gwgw
<input type="checkbox"/>			3	taki

Εικόνα 4.4.7 Πίνακας στη MySQL βάση

4.5 Σύνθετη περίπτωση από Βάση σε XML

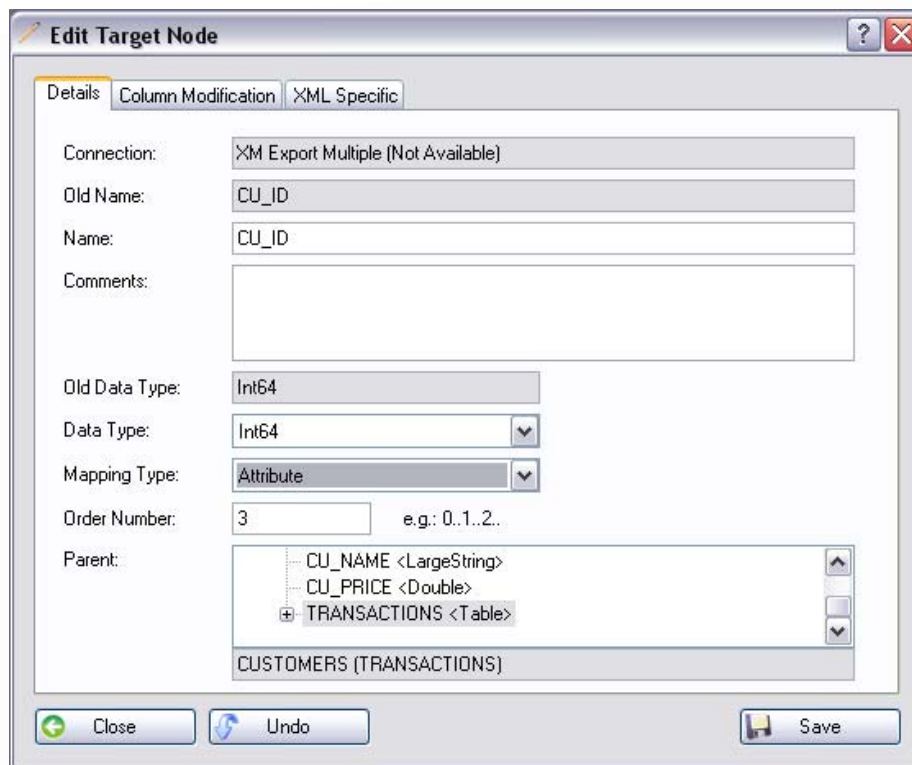
Σε αυτήν την ενότητα θα παρουσιαστούν πιο σύνθετες περιπτώσεις μεταφοράς δεδομένων από βάση σε XML. Το τελικό σχήμα θα είναι πιο σύνθετο, και θα περιλαμβάνει παραπάνω του ενός επίπεδα στη δομή δεδομένων.

Αρχικά επιλέγουμε σαν αρχική βάση δεδομένων μία σύνδεση σε SQL Server. Μεταφέρουμε τον πίνακα Customers και Transactions όπως φαίνεται στην Εικόνα 4.5.1.



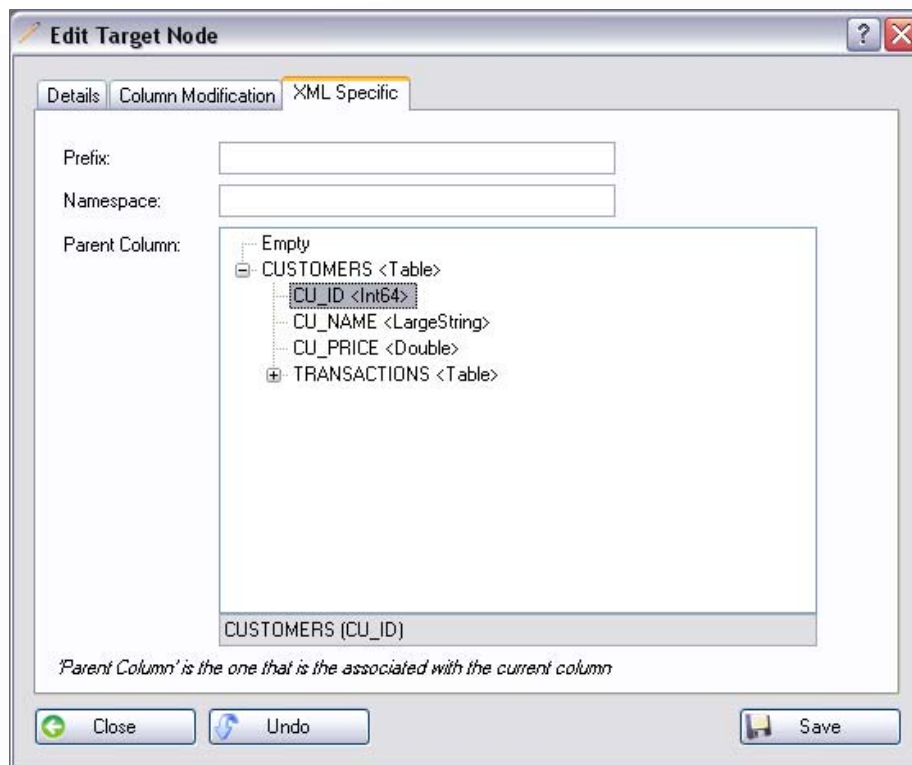
Εικόνα 4.5.1 Μεταφορά δύο πινάκων στο τελικό σχήμα

Έπειτα ορίζουμε τον κόμβο CU_ID του πίνακα «TRANSACTIONS», να μην είναι απλό πεδίο αλλά ιδιότητα / «Attribute» του πατέρα του (δηλ. του προαναφερθέντα πίνακα). Αυτό μπορεί να γίνει με το να επεξεργαστούμε τον κόμβο και να ορίσουμε το Mapping Type στην επιθυμητή τιμή (Εικόνα 4.5.2).



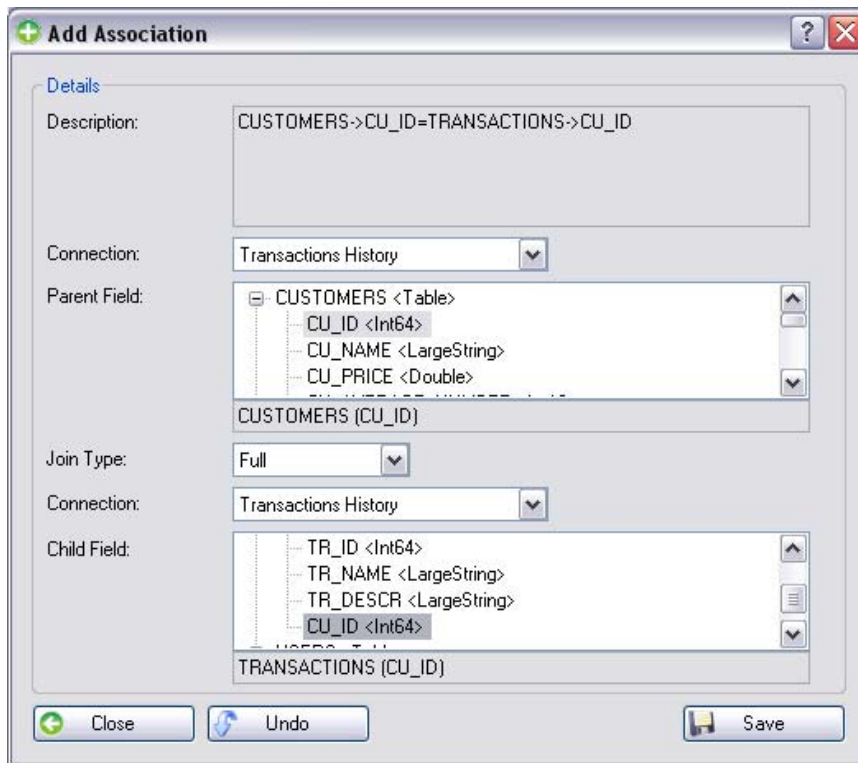
Εικόνα 4.5.2 Ορισμός ενός κόμβου ως ιδιότητα (Attribute)

Επειδή όμως έχουμε ορίσει στο τελικό σχήμα ότι θα έχουμε δύο επίπεδα, θα πρέπει να ορίσουμε και ποια θα είναι η συσχέτιση μεταξύ των πινάκων που ορίζουν τα επίπεδα αυτά. Για να γίνει αυτό θα πρέπει να ανοίξουμε την καρτέλα «XML Specific» που εμφανίζεται κατά την επεξεργασία του κόμβου CU_ID, του πίνακα «Transactions». Εκεί θα ορίσουμε τον πατέρα κόμβο συσχέτισης. Στην περίπτωση μας ο κόμβος «CU_ID» στον πίνακα «CUSTOMERS», είναι ο πατέρας κόμβος, οπότε και αυτόν θα πρέπει να επιλέξουμε (Εικόνα 4.5.3). Η λογική είναι ίδια σαν να ορίζαμε ένα ξένο κλειδί στο πρωτεύον του «CUSTOMERS».



Εικόνα 4.5.3 Ορισμός του πατέρα συσχέτισης στο τελικό σχήμα

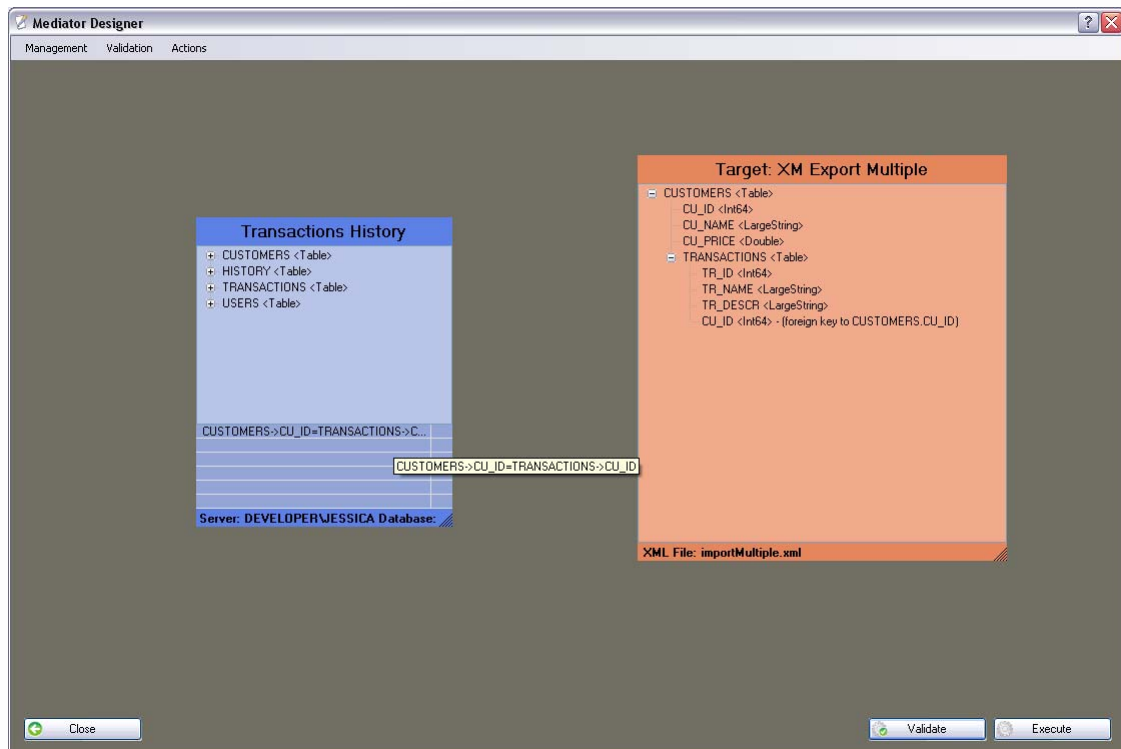
Εφόσον ορίσαμε τη συσχέτιση του τελικού σχήματος, θα πρέπει να ορίσουμε και τη σύνδεση των πινάκων που χρησιμοποιήσαμε. Αυτό για να υλοποιηθεί θα πρέπει να πάμε στο σχήμα πηγής (από όπου μεταφέραμε τους πίνακες), και με το δεξί πλήκτρο του ποντικού να πατήσουμε το «Manage Associations» από το μενού που θα εμφανιστεί. Εκεί θα κάνουμε προσθήκη μίας νέας σύνδεσης και θα συμπληρώσουμε όλα τα απαιτούμενα πεδία της φόρμας, όπως δείχνει και η Εικόνα 4.5.4.



Εικόνα 4.5.4 Προσθήκη συσχέτισης μεταξύ των πινάκων στα σχήματα πηγής

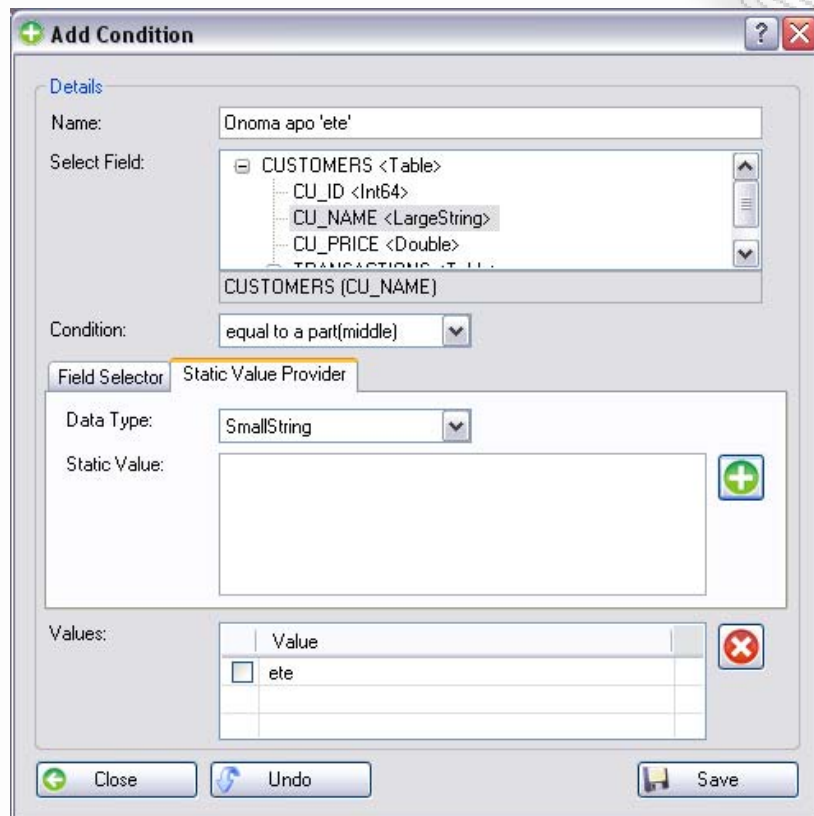
Από τη φόρμα μπορούμε να δούμε ότι επιλέγουμε το πρωτεύον κλειδί του πρώτου πίνακα με το εξωτερικό κλειδί του δεύτερου όπως αντίστοιχα έγινε και για το τελικό σχήμα.

Η προσθήκη της συσχέτισης εγγράφεται και στο σχήμα πηγής που φαίνεται στη σχεδιαστική φόρμα. Έτσι ο χρήστης έχει εποπτεία των συσχετίσεων που έχει ορίσει ακόμα και όταν αυτές είναι μέσα στο ίδιο σχήμα πηγής (Εικόνα 4.5.5).



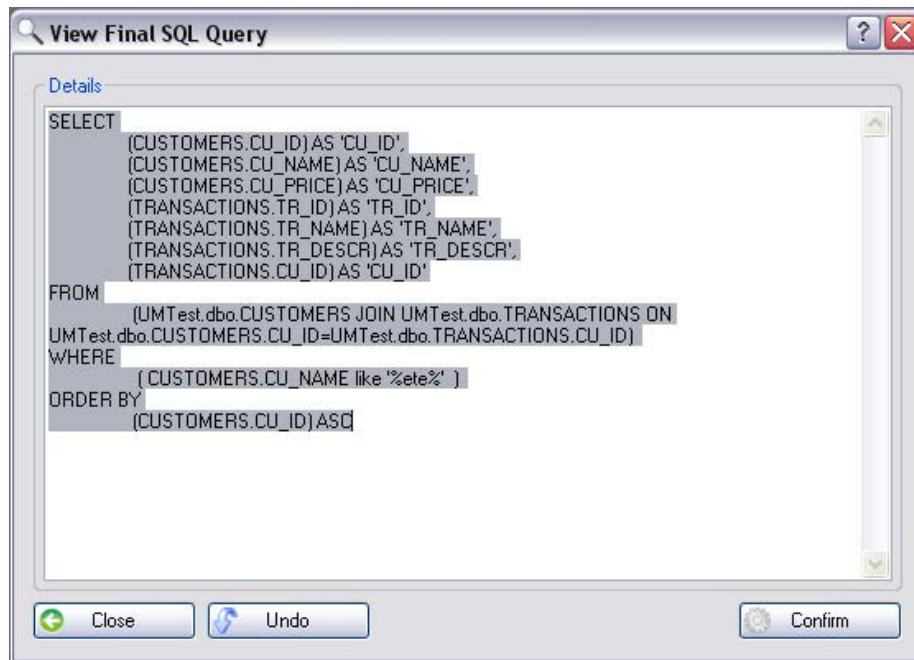
Εικόνα 4.5.5 Εμφάνιση συσχέτισης στη φόρμα σχεδιασμού στο σχήμα πηγής

Ολοκληρώνοντας αυτό το παράδειγμα προσθέτουμε και μία συνθήκη επάνω στο όνομα του πελάτη. Για παράδειγμα, ας αναφέρουμε ότι θέλουμε μόνο εκείνους τους πελάτες που το όνομά τους περιέχει τους χαρακτήρες «ete». Η διαδικασία είναι αντίστοιχη με αυτήν που περιγράψαμε σε προηγούμενη ενότητα, με τη διαφορά ότι τώρα στο πεδίο «Condition», επιλέγουμε το «equal to a part(middle)».



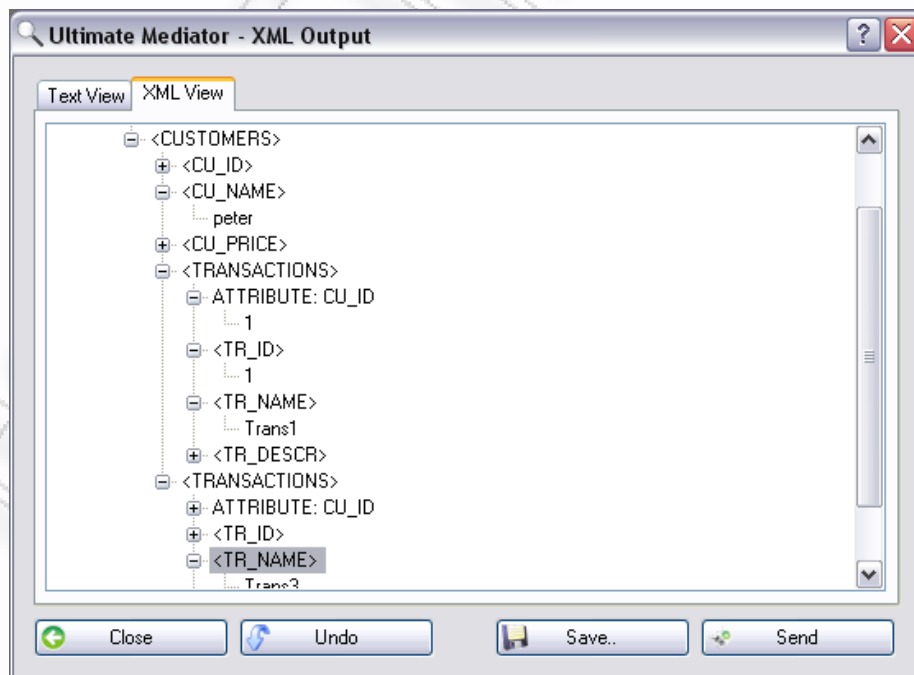
Εικόνα 4.5.6 Προσθήκη νέας συνθήκης με βάση το όνομα του πελάτη

Αφού ορίσαμε όλα τα παραπάνω μπορούμε να προχωρήσουμε με την εκτέλεση της μεταφοράς. Πριν τη μεταφορά θα εμφανιστεί το παράθυρο με τον SQL κώδικα που θα τρέξει για να πάρει τα δεδομένα (Εικόνα 4.5.7). Σε αυτό το σημείο μπορούμε να κάνουμε όποιες αλλαγές θεωρούμε απαραίτητες και μετά να το επιβεβαιώσουμε («Confirm»).



Εικόνα 4.5.7 Εμφάνιση SQL ερωτήματος πριν την εκτέλεση για τη λήψη των δεδομένων

Στο αποτέλεσμα της εξόδου του τελικού σχήματος μπορούμε να δούμε ότι επεστράφη μόνο ένας πελάτης που περιέχει τους χαρακτήρες «ete» (Εικόνα 4.5.8). Επίσης φαίνεται ότι έχει δημιουργηθεί μία ιδιότητα την οποία την ξεχωρίζουμε από την περιγραφή μπροστά της (ATTRIBUTE: CU_ID).



Εικόνα 4.5.8 Τελικό αποτέλεσμα εξόδου του τελικού σχήματος

4.6 Σύνθετη περίπτωση από XML σε XML

Μία ακόμα περίπτωση που θα γίνει γνωστή είναι αυτή της μεταφοράς δεδομένων με περισσότερα επίπεδα στο τελικό σχήμα. Αρχικά σε αυτό το σημείο θα παρουσιάσουμε το αρχικό σχήμα που θα είναι XML σε δύο εικόνες: Εικόνα 4.6.1 και Εικόνα 4.6.2.

```

<?xml version="1.0" standalone="yes" ?>
- <requests>
- <CUSTOMERS>
  <CU_ID>1</CU_ID>
  <CU_NAME>Peter Kokkinakis</CU_NAME>
  <CU_BALANCE>25,32</CU_BALANCE>
- <TRANSACTIONS>
  <TR_ID>1</TR_ID>
  <TR_NAME>Trans #1</TR_NAME>
  <TR_DESCR>200 TEMAXIA MELANIA</TR_DESCR>
  <CU_ID>1</CU_ID>
- <PRODUCT_DETAILS>
  <TR_ID>1</TR_ID>
  <TL_NAME>EPSON STYLOUS COLOR 670</TL_NAME>
</PRODUCT_DETAILS>
</TRANSACTIONS>
- <TRANSACTIONS>
  <TR_ID>3</TR_ID>
  <TR_NAME>Trans #2</TR_NAME>
  <TR_DESCR>120 PALETES XARTI</TR_DESCR>
  <CU_ID>1</CU_ID>
- <PRODUCT_DETAILS>
  <TR_ID>3</TR_ID>
  <TL_NAME>80 MEGETHOUS A4</TL_NAME>
</PRODUCT_DETAILS>
- <PRODUCT_DETAILS>
  <TR_ID>3</TR_ID>
  <TL_NAME>40 MEGETHOUS A3</TL_NAME>
</PRODUCT_DETAILS>
</TRANSACTIONS>
- <HISTORY>
  <CU_ID>1</CU_ID>
  <HI_NAME>TELEUTAIA AGORA 20/03/2008</HI_NAME>
</HISTORY>
</CUSTOMERS>

```

Εικόνα 4.6.1 Πρώτο κομμάτι αρχείου XML με δεδομένα συναλλαγών

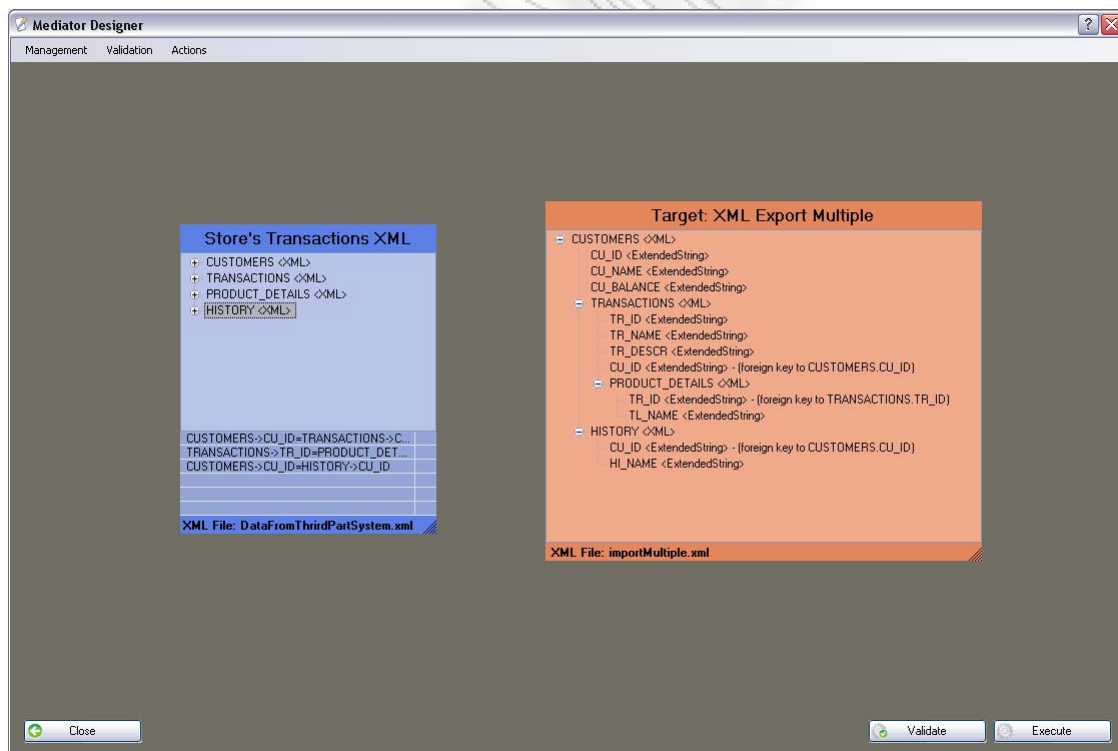

```

- <CUSTOMERS>
  <CU_ID>2</CU_ID>
  <CU_NAME>XRHSTOS DAMARTZHS</CU_NAME>
  <CU_BALANCE>520</CU_BALANCE>
- <TRANSACTIONS>
  <TR_ID>2</TR_ID>
  <TR_NAME>Trans #1</TR_NAME>
  <TR_DESCR>AGORA 7 LAPTOP</TR_DESCR>
  <CU_ID>2</CU_ID>
- <PRODUCT_DETAILS>
  <TR_ID>2</TR_ID>
  <TL_NAME>TURBO-X IRON</TL_NAME>
</PRODUCT_DETAILS>
</TRANSACTIONS>
- <HISTORY>
  <CU_ID>2</CU_ID>
  <HI_NAME>TELEYTAIA AGORA 15/05/2008</HI_NAME>
</HISTORY>
</CUSTOMERS>
</requests>

```

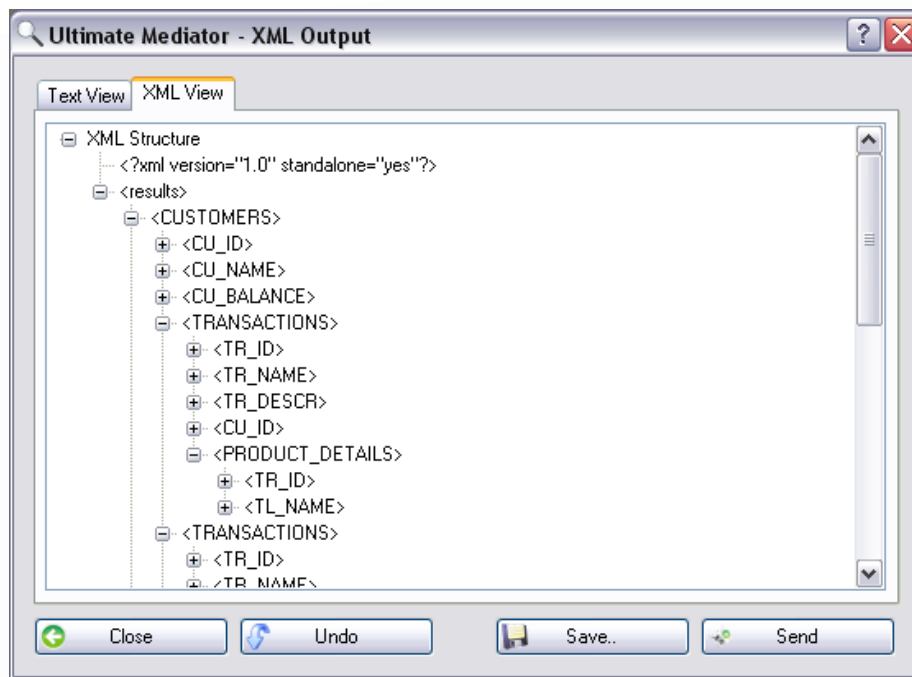
Εικόνα 4.6.2 Δεύτερο κομμάτι XML με δεδομένα συναλλαγών

Ορίζουμε το τελικό σχήμα και βάζουμε όλες τις συσχετίσεις, και στο σχήμα πηγής και στο τελικό έτσι ώστε να το φέρουμε στη μορφή που έχει η Εικόνα 4.6.3.



Εικόνα 4.6.3 Φόρμα σχεδίασης μετά τον ορισμό όλων των συσχετίσεων

Αν εκτελέσουμε απευθείας τη μεταφορά θα πρέπει να πάρουμε σαν έξοδο ακριβώς τα ίδια δεδομένα με την είσοδο (Εικόνα 4.6.1 και Εικόνα 4.6.2).



Εικόνα 4.6.4 Δεδομένα μετά από έξοδο όπως είναι από το αρχικό τους σχήμα

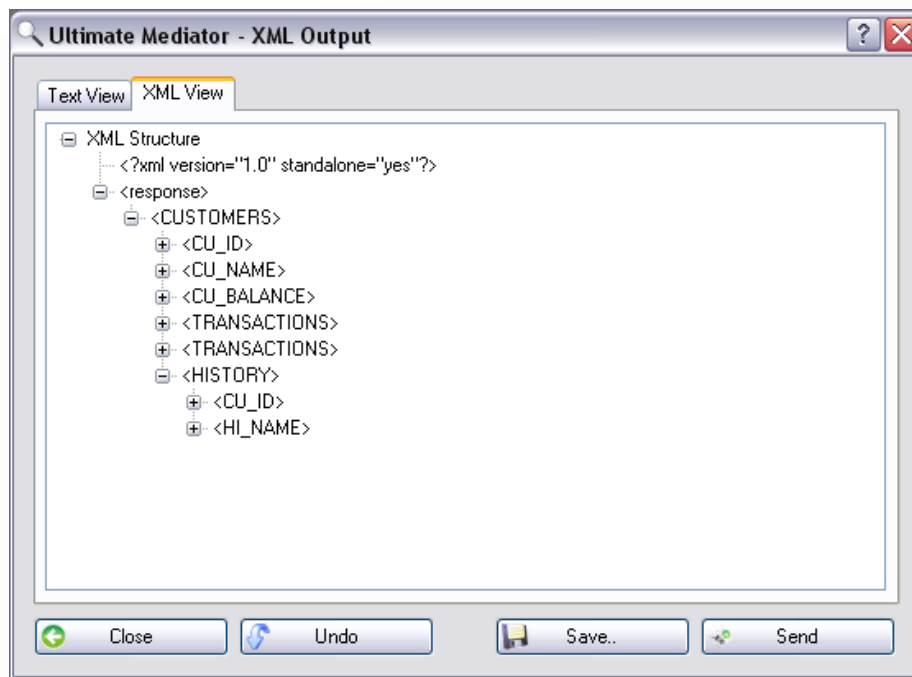
Έστω όμως ότι τα δεδομένα μας δεν ήταν αυτά της Εικόνα 4.6.1 και της Εικόνα 4.6.2, αλλά τα παρακάτω (Εικόνα 4.6.5). Μπορούμε να δούμε ότι τα δεδομένα για το δεύτερο πελάτη δεν είναι πλήρη. Συγκεκριμένα λείπει ο πίνακας που αναφέρεται στο ιστορικό (*HISTORY*), και αυτός που αναφέρεται στις πληροφορίες του προϊόντος (*PRODUCT_DETAILS*). Σε μία τέτοια περίπτωση μία σύνδεση ολική «Full Join» σύνδεση, δεν θα έφερνε δεδομένα για το δεύτερο πελάτη (*XRHSTOS DAMARTZHS*). Αν δηλαδή πατήσουμε το κουμπί εκτέλεσης, θα πάρουμε το αποτέλεσμα της Εικόνα 4.6.6.

```

<?xml version="1.0" standalone="yes" ?>
- <requests>
- <CUSTOMERS>
  <CU_ID>1</CU_ID>
  <CU_NAME>Peter Kokkinakis</CU_NAME>
  <CU_BALANCE>25,32</CU_BALANCE>
- <TRANSACTIONS>
  <TR_ID>1</TR_ID>
  <TR_NAME>Trans #1</TR_NAME>
  <TR_DESCR>200 TEMAXIA MELANIA</TR_DESCR>
  <CU_ID>1</CU_ID>
- <PRODUCT_DETAILS>
  <TR_ID>1</TR_ID>
  <TL_NAME>EPSON STYLOUS COLOR 670</TL_NAME>
</PRODUCT_DETAILS>
</TRANSACTIONS>
- <TRANSACTIONS>
  <TR_ID>3</TR_ID>
  <TR_NAME>Trans #2</TR_NAME>
  <TR_DESCR>120 PALETES XARTI</TR_DESCR>
  <CU_ID>1</CU_ID>
- <PRODUCT_DETAILS>
  <TR_ID>3</TR_ID>
  <TL_NAME>80 MEGETHOUS A4</TL_NAME>
</PRODUCT_DETAILS>
- <PRODUCT_DETAILS>
  <TR_ID>3</TR_ID>
  <TL_NAME>40 MEGETHOUS A3</TL_NAME>
</PRODUCT_DETAILS>
</TRANSACTIONS>
- <HISTORY>
  <CU_ID>1</CU_ID>
  <HI_NAME>TELEUTAIA AGORA 20/03/2008</HI_NAME>
</HISTORY>
</CUSTOMERS>
- <CUSTOMERS>
  <CU_ID>2</CU_ID>
  <CU_NAME>XRHSTOS DAMARTZHS</CU_NAME>
  <CU_BALANCE>520</CU_BALANCE>
- <TRANSACTIONS>
  <TR_ID>2</TR_ID>
  <TR_NAME>Trans #1</TR_NAME>
  <TR_DESCR>AGORA 7 LAPTOP</TR_DESCR>
  <CU_ID>2</CU_ID>
</TRANSACTIONS>
</CUSTOMERS>
</requests>

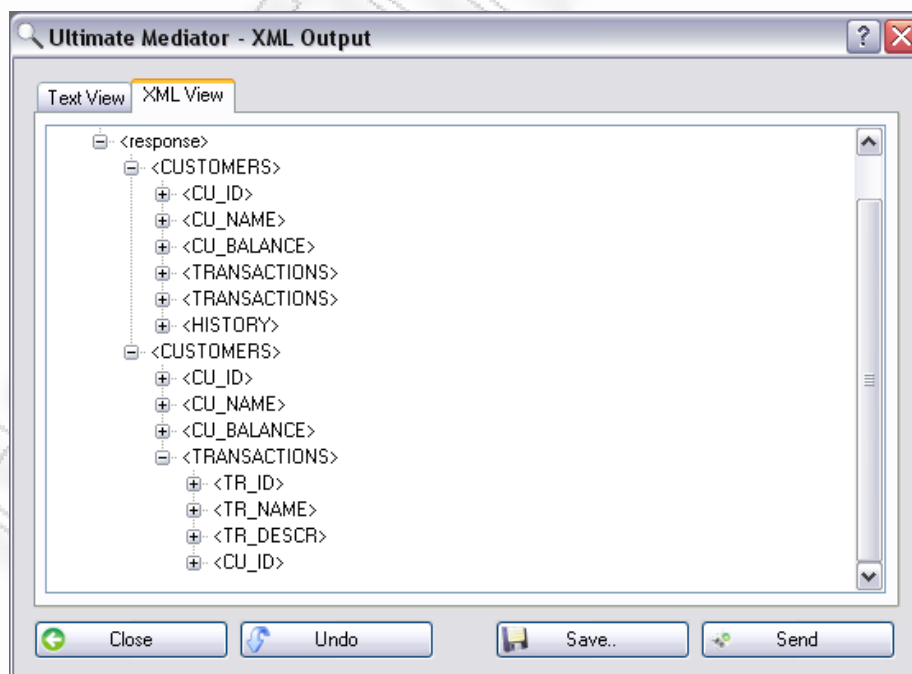
```

Εικόνα 4.6.5 Περίπτωση με δεδομένα μερικώς ορισμένα



Εικόνα 4.6.6 Αποτέλεσμα εκτέλεσης ολικής σύνδεσης με ελλιπές δεδομένα

Αν όμως αλλάξουμε τις συσχετίσεις στο σχήμα πηγής και ορίσουμε ότι θα έχουν «Join Type» ίσο με Left, (δηλαδή αν υπάρχει ο Customer, τότε ακόμα και αν δεν υπάρχει κάποιο ιστορικό θα πρέπει ο πίνακας να εμφανίζεται με κενές τιμές), τότε το αποτέλεσμα θα έχει την παρακάτω μορφή (Εικόνα 4.6.7).

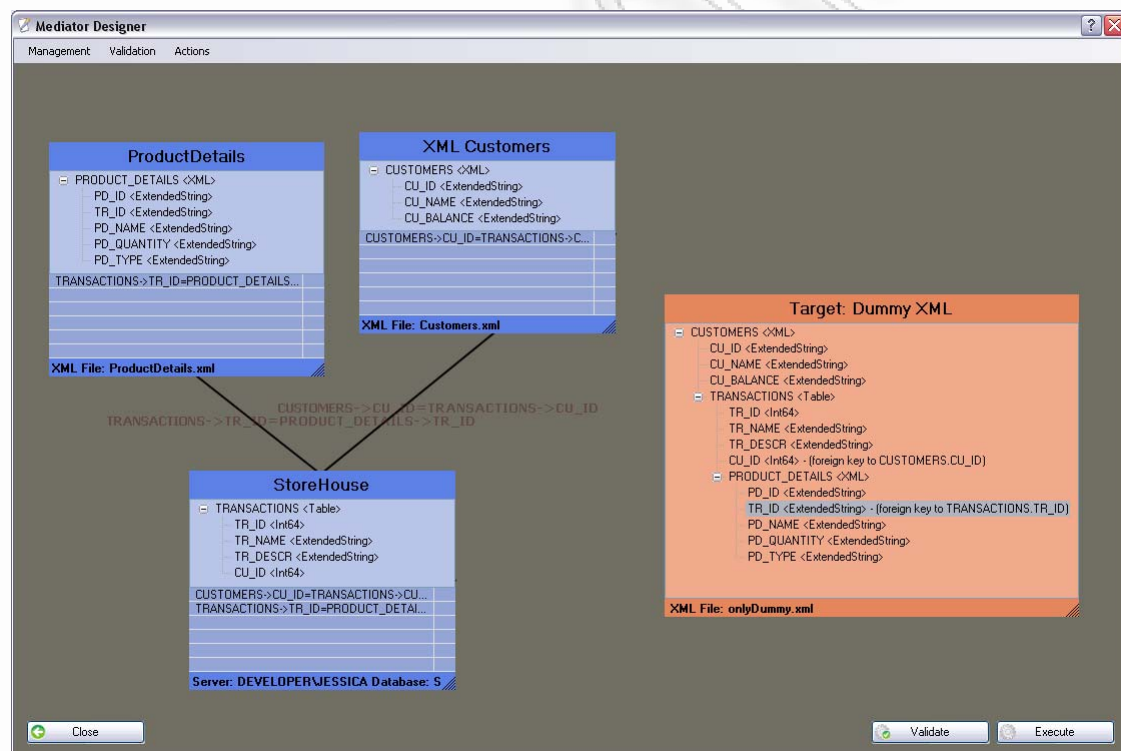


Εικόνα 4.6.7 Έξοδος δεδομένων με αλλαγή του «Join Type» στις συσχετίσεις

4.7 Χρήση πολλαπλών συνδέσεων εισόδου

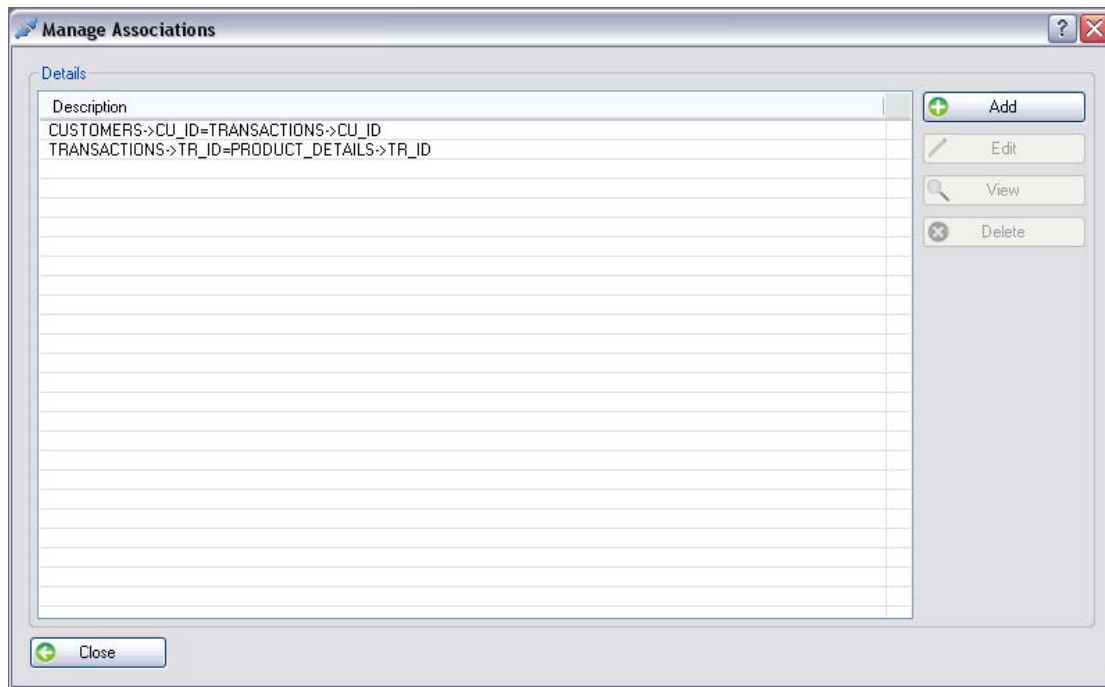
Μία επιπλέον σημαντική δυνατότητα που παρέχει το λογισμικό είναι η ικανότητα να επιλέξει πολλαπλές ροές εισόδου δεδομένων και αυτές να τις συνδυάσει για να βγάλει μία τελική έξοδο σε βάση ή xml αρχείο. Η διαδικασία που χρησιμοποιείται δεν είναι διαφορετική από την υπάρχουσα που περιγράφηκε σε προηγούμενες ενότητες. Ο χρήστης το μόνο που έχει να κάνει είναι να επιλέξει τις επιθυμητές συνδέσεις, να φτιάξει τις συσχετίσεις μεταξύ τους και έπειτα να ορίσει όπως θα έκανε και στην απλή περίπτωση το τελικό σχήμα.

Οι συνδέσεις που μπορεί να ορίσει δεν περιορίζονται μόνο σε βάση ή σε xml αρχεία. Μπορεί να είναι ένα ή και περισσότερα από τα παραπάνω. Η Εικόνα 4.7.1 δείχνει μία τέτοια περίπτωση στην οποία έχουν επιλεγθεί τρεις συνδέσεις, από τις οποίες η μία είναι σε βάση MSSQLServer και οι άλλες δύο σε XML αρχεία. Η επιλογή είναι τυχαία και δεν περιορίζεται σε αυτόν μόνο το συνδυασμό. Μπορεί να υπάρχει οποιοσδήποτε συνδυασμός συνδέσεων. Το τελικό σχήμα έχει οριστεί μέσα από αυτές τις συνδέσεις.



Εικόνα 4.7.1 Οθόνη σχεδίασης με πολλαπλές συνδέσεις εισόδου

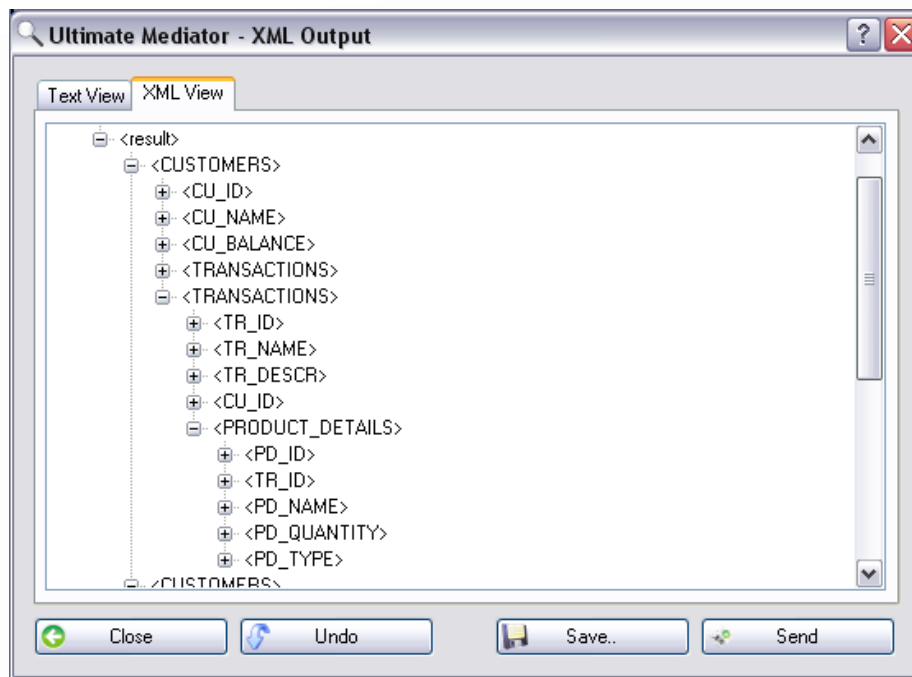
Όπως φαίνεται και από την οθόνη σχεδίασης, έχουν οριστεί συσχετίσεις ανάμεσα στο ProductDetails και στο StoreHouse αλλά και ανάμεσα στο StoreHouse και το XMLCustomers. Στην Εικόνα 4.7.2 φαίνονται οι δύο εγγραφές των συσχετίσεων στη διαχειριστική φόρμα.



Εικόνα 4.7.2 Διαχειριστική φόρμα συσχετίσεων της περίπτωσης πολλαπλής εισόδου δεδομένων

Σαν τελική σύνδεση ορίζεται η έξοδος να παραχθεί σε xml αρχείο, έτσι ώστε να μπορούμε να έχουμε αυτήν ακριβώς τη δομή που ορίστηκε στο σχήμα. Το είδος των συνδέσεων το αφήσαμε σε ολοκληρωτική σύνδεση (Full Join), γιατί ο σκοπός αυτής της περίπτωσης δεν εστιάζεται σε πιο σύνθετη επεξεργασία. Μπορεί όμως ο χρήστης να εκτελέσει όλες τις πιθανές διαδικασίες επεξεργασίας των δεδομένων όπως αυτές ορίστηκαν και παρουσιάστηκαν σε προηγούμενες ενότητες.

Εφόσον έχουμε ορίσει και τις συσχετίσεις στις αρχικές συνδέσεις και τις τελικές συσχετίσεις του τελικού σχήματος (Εικόνα 4.7.1), συνεχίζουμε στην εκτέλεση της μεταφοράς. Το αποτέλεσμα όπως δείχνει η Εικόνα 4.7.3, είναι να δημιουργηθεί ένα δενδροειδές xml της μορφής που ορίστηκε στο τελικό σχήμα.

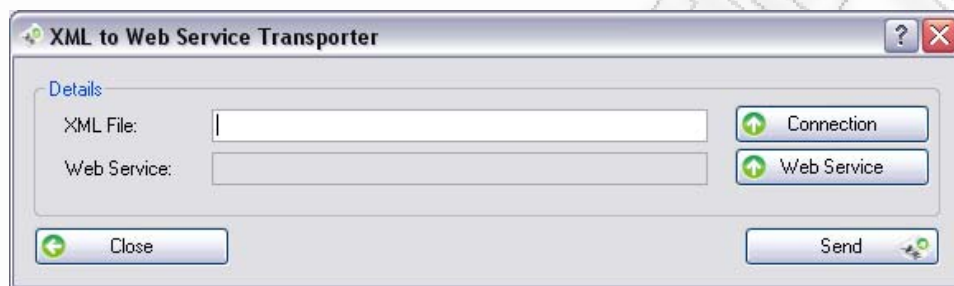


Εικόνα 4.7.3 Αποτέλεσμα σε περίπτωση πολλαπλών συνδέσεων

4.8 Άμεση αποστολή XML σε Web Service

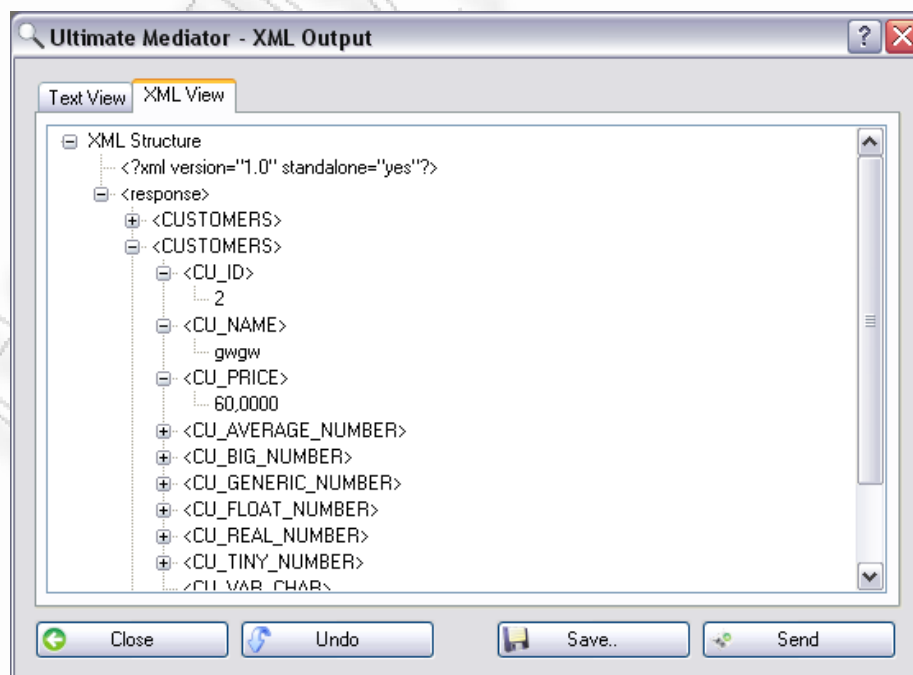
Για μία ακόμα δυνατότητα που παρέχεται από το πρόγραμμα είναι η άμεση αποστολή XML σε ένα Web Service. Αυτό εξυπηρετεί πολλές φορές το χρήστη που έχει είδη τα δεδομένα που θα αποστείλει και απλά θέλει έναν ενδιάμεσο που θα κάνει τη μεταφορά.

Αρχικά θα πρέπει από την κύρια οθόνη εισόδου του προγράμματος ο χρήστης να επιλέξει από το μενού Transporter, «XML to Web Service». Εκεί εμφανίζεται η οθόνη όπως φαίνεται στην Εικόνα 4.8.1.



Εικόνα 4.8.1 Φόρμα άμεσης αποστολής xml σε web service

Εφόσον επιλέξει το αρχείο, είτε μέσω κάποιας xml σύνδεσης, είτε γράφοντας ο ίδιος τη διαδρομή στον δίσκο, πατάει «Send». Από αυτό το σημείο και έπειτα η διαδικασία είναι ίδια. Το παράθυρο που θα εμφανιστεί με τα δεδομένα απάντησης θα είναι το ακόλουθο (Εικόνα 4.8.2). Από εκεί και πέρα ο χρήστης έχει όλες εκείνες τις επιλογές που έχουν ήδη περιγραφτεί για την επεξεργασία / αποθήκευση ή και αποστολή των δεδομένων αυτών.



Εικόνα 4.8.2 Οθόνη προβολής απάντησης από web service

Τα δεδομένα δεν είναι απαραίτητο να είναι XML, μπορούν να είναι και ένα απλό αρχείο text το οποίο να περιλαμβάνει διάφορα δεδομένα (διαφορετικής μορφής από αυτής του xml).

5 Τεχνολογικά - Επιχειρηματικά Οφέλη

Για ένα σύστημα σαν τον Ultimate Mediator υπάρχουν αρκετά σημαντικά οφέλη που μπορούν να εκμεταλλευτούν οι επιχειρήσεις, οι οργανισμοί αλλά και η επιστημονική κοινότητα. Αυτά παρουσιάζονται και αναλύονται παρακάτω.

Ο Ultimate Mediator εισάγει τη Διαλειτουργικότητα στην επιχειρησιακή διαδικασία αφού παρέχει ένα ολοκληρωμένο σύστημα επιλογής, επεξεργασίας, ενοποίησης και μεταφοράς δεδομένων. Δίνει τη δυνατότητα για εκτέλεση σημαντικών λειτουργιών σε μικρό χρονικό διάστημα μέσω κατάλληλων διαμορφωμένων οθόνων. Αυτές περιέχουν όλες εκείνες τις απαραίτητες λειτουργίες και ευχρηστίες που καταστούν την πλήρη διαδικασία θέμα μερικών λεπτών. Επίσης παρέχει μία πολύ καλή βάση για την ανάπτυξή του (σε κάποια επόμενη έκδοση), στην οποία θα μπορεί να παρέχει αυτοματισμούς και συστήματα αποφάσεων workflow.

Μέχρι σήμερα οι εταιρίες που επιθυμούσαν να συνδέσουν διάφορα συστήματα μεταξύ τους, οδηγούνταν στην διαδικασία πληρωμής συγκεκριμένων ατόμων (προγραμματιστές), οι οποίοι έπρεπε να εκπαιδευτούν κατάλληλα, έτσι ώστε να μάθουν τα συστήματα, την αλληλεπίδρασή τους με άλλα συστήματα, καθώς και τον τύπο εισόδου και εξόδου τους. Εύκολα συμπεραίνουμε ότι οι εταιρίες έτσι, θα έπρεπε να προσλάβουν συγκεκριμένο προσωπικό (με γνώσεις προγραμματισμού και γνώσεις διαχείρισης έργων), το οποίο κατέληγε σε αύξηση του κόστους υλοποίησης. Το τελικό σύστημα όμως, θα ήταν προσανατολισμένο ρητά στον τρόπο επικοινωνίας που απαιτούσαν τα συστήματα αλλά και στο είδος ανταλλαγής δεδομένων τους. Έτσι, αν χρειαζόταν να χρησιμοποιηθεί για κάποια άλλα συστήματα, αυτό δε θα ήταν πάντα εφικτό, γιατί θα έπρεπε να διαμορφωθεί ξανά, για να μπορέσει να επικοινωνήσει σύμφωνα με τους νέους κανόνες που ορίζονται από αυτά (τα νέα συστήματα). Επιπλέον αν χρειαζόταν να χρησιμοποιηθεί το λογισμικό αργότερα από άλλους προγραμματιστές, χωρίς να βρίσκονται στην εταιρία οι δημιουργοί τους, αυτό δεν θα ήταν τόσο εύκολο, και θα πρόσθετε επιπλέον χρόνο και κόστος για την παραμετροποίησή του υπάρχοντα λογισμικού. Πολλές φορές αυτό είχε αποδειχτεί αδύνατο και ήταν αναγκαία η διαδικασία δημιουργίας νέου συστήματος από την αρχή.

Πολλές εταιρίες δε θέλουν να προσλάβουν άτομα, και έτσι αναγκάζονται να δώσουν το πρόγραμμα να το υλοποιήσουν άλλες εταιρίες εξειδικευμένες σε προγραμματισμό. Αυτό όμως τελικά μπορεί να επιφέρει επιπρόσθετο κόστος για την επιχείρηση, έχοντας όμως και πάλι το ίδιο πρόβλημα με αυτό που αναφέρθηκε νωρίτερα. Δεν είναι πάντα εφικτή η διαδικασία επαναχρησιμοποίησης σε άλλα συστήματα.

Ο Ultimate Mediator παρέχει ένα ενοποιημένο σύστημα, το οποίο μπορεί να εκτελέσει όλες τις λειτουργίες επιλογής, μεταφοράς, επεξεργασίας και αποστολής δεδομένων σε πολλές οντότητες όπως βάσεις δεδομένων, xml αρχεία και web services. Αυτό μπορεί εύκολα να το κάνει χωρίς να υπάρχει ανάγκη χρήσης κάποιου τρίτου συστήματος. Έτσι μειώνεται το συνολικό κόστος υλοποίησης της διαδικασίας. Επιπροσθέτως, το σύστημα υπάγεται στην κατηγορία προγραμμάτων ελεύθερου λογισμικού, το οποίο σημαίνει ότι με μηδενικό κόστος μπορεί κάποιος χρήστης να το δοκιμάσει και να το ενσωματώσει στη δουλειά του, όπως επίσης και να το διαμορφώσει στις ανάγκες του. Άρα το τελικό κόστος που έχουν άλλες, αντίστοιχες εφαρμογές συγκριτικά με αυτήν είναι κατά κόρον μεγαλύτερο. Τέλος αξίζει να

σημειωθεί, ότι το σύστημα είναι φτιαγμένο μέσω καλά δομημένων φορμών, το οποίο το καταστούν ευανάγνωστο και σε επίπεδο εγγραφής κώδικα, αλλά και σε επίπεδο χρήσης του λογισμικού.

Ένα ακόμα θετικό που απορρέει από τη χρήση του είναι ότι οι διαδραστικές οθόνες και ο τρόπος σχεδίασης του προγράμματος, δίνουν το πλεονέκτημα της γρήγορης και εύκολης χρήσης του, έτσι, που να μπορεί να χρησιμοποιηθεί από οποιοδήποτε άτομο, χωρίς να απαιτεί πολύ ειδικευμένες γνώσεις π.χ. γνώσεις προγραμματισμού ή άλλες όπως γνώσεις κατανόησης βάσεων δεδομένων. Επειδή όλα παρέχονται μέσα από μενού και φόρμες, ο χρήστης το μόνο που έχει να κάνει είναι να εξοικειωθεί με αυτές και να έχει μια μικρή γνώση επάνω σε βάσεις δεδομένων και xml δομές, για να μπορέσει να διαμορφώσει σωστά το τελικό σχήμα.

Τα δεδομένα που διαχειρίζεται το πρόγραμμα δεν είναι αναγκαίο να εισέρχονται σε όλο το σύνολό τους (από τις βάσεις πηγής). Ο Ultimate Mediator δίνει τη δυνατότητα να εισαχθούν μόνο ένα μέρος των δεδομένων προς επεξεργασία, μέσω των κανόνων που θέτει. Έτσι ο χρόνος λήψης των δεδομένων για την τελική τους επεξεργασία είναι αισθητικά μικρότερος. Αυτό μπορεί να αποβεί πολύ κερδοφόρο, κυρίως κατά τη διαχείριση βάσεων δεδομένων που περιέχουν μεγάλο όγκο πληροφορίας, καθιστώντας έτσι το σύστημα πιο γρήγορο σε τέτοιες περιπτώσεις.

Ο Ultimate Mediator όπως ήδη αναφέραμε, υποστηρίζει και xml αρχεία, γεγονός που προσθέτει απεριόριστες δυνατότητες σε επιχειρήσεις και οργανισμούς που χρησιμοποιούν SOA στην αρχιτεκτονική τους. Αυτό συμβαίνει, γιατί μπορεί να γίνει η μεταφορά των δεδομένων σε προγράμματα που υποστηρίζουν διαδικασίες από όλα τα επίπεδα της επιχείρησης, μέσω του συγκεκριμένου πρωτοκόλλου και την αποστολή δεδομένων σε xml.

Ένα επιπλέον πλεονέκτημα που έχει το παρόν λογισμικό, είναι ότι δίνει τη δυνατότητα να συνδυάζονται πολλαπλά σχήματα εισόδου δεδομένων, ακόμα και όταν αυτά δεν έχουν τον ίδιο τύπο σύνδεσης. Μπορεί να χρησιμοποιηθούν ταυτόχρονα διάφορες βάσεις δεδομένων και xml αρχεία για τη χρήση τους, ως σχήματα εισόδου, στη δημιουργία της τελικής εξόδου.

Αυτό είναι και ένα από τα πιο δυνατά του σημεία αφού πολλές επιχειρήσεις, αντιμετωπίζουν συχνά το ακόλουθο πρόβλημα. Χρησιμοποιούν, για παράδειγμα, τρία συστήματα από τα οποία, το ένα έχει βάση δεδομένων σε Oracle, το άλλο σε Microsoft SQL Server και το άλλο παρέχει είσοδο και έξοδο μόνο μέσω xml αρχείων. Έτσι, έστω ότι η εταιρία αυτή μπορεί να έχει ένα ακόμα σύστημα το οποίο να είναι κάποιο web service. Την έξοδο των προηγούμενων τριών συστημάτων είναι πιθανό να τη θέλει σε συνδυασμό, για να ενημερώσει το service το οποίο με τη σειρά του θα παρέχει ενημερωμένα δεδομένα σε χρήστες ή κάποιο άλλο σύστημα. Με τη χρήση του προγράμματος που παρουσιάστηκε, όλα αυτά μπορούν να γίνουν εύκολα και γρήγορα, μέσω των ειδικών φορμών επιλογής και εισαγωγής σχημάτων. Λίγα συστήματα, από τα ήδη υπάρχοντα, παρέχουν τέτοιες δυνατότητες, συνδυασμού δεδομένων για την παραγωγή μίας τελικής εξόδου, η οποία θα έχει τη δυνατότητα ενημέρωσης ενός web service.

Το xml αρχείο που θα παραχθεί από μία διαδικασία όπως η παραπάνω μπορεί είτε να αποθηκευτεί σε πολλά διαφορετικά σημεία του δίσκου, είτε να αποσταλεί απευθείας σε κάποιο web service. Δε δίνεται η άμεση αποστολή, για να μπορεί να δει ο χρήστης του Ultimate Mediator τα δεδομένα πριν τα στείλει. Αυτό μπορεί να

αποβεί πολύ χρήσιμο σε περιπτώσεις λαθών από το χρήστη του συστήματος. Μπορεί δηλαδή πριν πραγματοποιήσει την τελική του αποστολή σε κάποιο web service, να μειώσει με ειδικούς κανόνες τα εξαγόμενα δεδομένα, για να δει αν ήταν αυτά που περίμενε. Έτσι μπαίνει ένας ακόμα αρκετά χρήσιμος έλεγχος για την αποφυγή λαθών.

Κάτι που αξίζει να σημειωθεί επίσης, είναι ότι η αποστολή δεδομένων σε κάποιο web service δεν απαιτεί τα δεδομένα αυτά να είναι τύπου xml. Μπορεί να είναι και απλά δεδομένα κειμένου τα οποία να χρειάζονται για τη λήψη δεδομένων είτε xml είτε απλού κειμένου. Μέσα από τη διαδικασία «Data to Web Service», που περιγράφηκε στην ενότητα *Σχεδίαση Συστήματος*, ο χρήστης δεν είναι απαραίτητο να χρησιμοποιήσει xml αρχείο για είσοδο. Μπορεί να δώσει οποιοδήποτε αρχείο κειμένου για αποστολή.

Τέλος αξίζει να αναφερθεί ότι ένα θετικό που μπορεί να αποκομίσει ο αναγνώστης από τις ικανότητες των τεχνολογιών Linq και κατ' επέκταση .Net, είναι ότι παρέχουν τη δυνατότητα να μας δώσουν γρήγορες, εύκολες, ισχυρές και ανταγωνιστικές λύσεις στη δημιουργία ενός τέτοιου λογισμικού. Ειδικά με την εισαγωγή (τα τελευταία χρόνια) της Linq, μπορούμε να εφαρμόσουμε ακόμα πιο ευέλικτα ερωτήματα επάνω στα δεδομένα, χωρίς ο προγραμματιστής να χρειαστεί να φτιάξει δική του υποδομή, για ερωτήματα σε αντικείμενα της μνήμης. Αυτό κατ' επέκταση τελικά σημαίνει, ότι οι αλλαγές και η ανάπτυξη του συγκεκριμένου λογισμικού μπορεί να πάρει μεγάλες διαστάσεις. Αυτό το κάνει ακόμα πιο ισχυρό και ανταγωνιστικό σε σχέση με τα ήδη υπάρχοντα, δίνοντας έτσι μία ολοκληρωμένη και ασφαλής λύση σε εταιρίες και οργανισμούς με τέτοιες ανάγκες.

6 Προοπτικές Ανάπτυξης

Το λογισμικό Ultimate Mediator παρέχει νέες δυνατότητες και παράλληλα υποστηρίζει αρκετές από αυτές που περιέχουν τα άλλα προγράμματα που υπάρχουν ήδη μέχρι σήμερα. Ένα τέτοιο όμως σύστημα, σαφώς, θα μπορούσε να αναπτυχθεί και να βελτιωθεί, για να παρέχει ακόμα περισσότερες λειτουργίες, αλλά και βελτιώσεις στις ήδη υπάρχουσες.

Αρχικά θα μπορούσε να βελτιωθεί από άποψη απόδοσης εκτέλεσης των ερωτημάτων που πραγματοποιούνται. Επειδή τα δεδομένα όλα αποθηκεύονται στη μνήμη, στις περιπτώσεις που έχουμε πολύ μεγάλο αριθμό δεδομένων αν ο χρήστης δεν ορίσει σωστά τις συνθήκες, τότε θα καθυστερήσει πολύ μέχρι να πραγματοποιήσει την ανάγνωση, επεξεργασία και επιλογή αυτών. Σε αυτό το σημείο θα μπορούσε η διαδικασία να βελτιωθεί και να γίνεται σταδιακά η εξαγωγή των δεδομένων για να μην πραγματοποιείται μαζικά η διαχείριση αυτών.

Επιπλέον ο Ultimate Mediator θα μπορούσε να περιλαμβάνει διαδικασίες προσθήκης επιπλέον οντοτήτων, οι οποίες χρησιμοποιούνται από άλλα συστήματα. Αυτό θα ήταν δυνατό να γίνει μέσω της εισαγωγής κάποιας βιβλιοθήκης (για παράδειγμα κλάσεις από αρχεία κατάληξης “.dll”) που θα έχει συγκεκριμένες συναρτήσεις για την πραγματοποίηση συνδέσεων και ερωτημάτων.

Θα μπορούσαν να φτιαχτούν οι φόρμες διαχείρισης των ερωτημάτων που έχουν προδιαγραφθεί στη βάση δεδομένων του λογισμικού. Σε αυτή την έκδοση, όπως έχει ήδη αναφερθεί, οι αλλαγές μπορούν να γίνουν, μόνο μέσα από το Microsoft Access.

Άλλη λειτουργία που είναι σκόπιμο να δημιουργηθεί στην επόμενη έκδοση του προγράμματος, είναι αυτή της αποθήκευσης της κατάστασης του προγράμματος. Ο χρήστης θα πρέπει να μπορεί εύκολα να αποθηκεύσει τον τρέχοντα σχεδιασμό του τελικού σχήματος, έτσι ώστε να μπορεί οποιαδήποτε στιγμή να επανέλθει σε αυτόν για να συνεχίσει ή να εκτελέσει ξανά τη μεταφορά.

Μία ακόμα λειτουργία θα είναι αυτή του προγραμματισμού εκτέλεσης εργασιών, ανά ορισμένη χρονική περίοδο. Αυτή μπορεί να γίνει μέσα από τη δημιουργία αρχείων εκτέλεσης που παίρνουν σαν είσοδο δεδομένα που ορίζει ο χρήστης κατά τη σχεδίαση. Έπειτα χρησιμοποιώντας τον προγραμματιστή εργασιών (Scheduler) των Windows, θα παρέχεται μία ολοκληρωμένη δυνατότητα προγραμματισμού εργασιών ενοποίησης, μετατροπής και μεταφοράς δεδομένων.

Παράλληλα θα ήταν επίσης εύκολο να χρησιμοποιηθούν οι triggers των βάσεων δεδομένων, για να ενεργοποιήσουν την εκτέλεση του λογισμικού όταν κάποια συνθήκη πραγματοποιηθεί. Έτσι θα είχαμε on-demand εκτέλεση, δίνοντας ακόμα περισσότερες δυνατότητες στο λογισμικό. Αυτό ουσιαστικά υπάγεται στη λογική της χρησιμοποίησης workflows στο πρόγραμμα, το οποίο θα μπορούσε να μπει σαν επιπλέον δυνατότητα για χρήστες που έχουν αντίστοιχες απαιτήσεις.

Τέλος, η ευχρηστία του συστήματος ίσως να αποτελεί μία επιπλέον παράμετρο που πρέπει να ληφθεί υπόψη. Κάποιες βελτιώσεις από σχεδιαστικής απόψεως και ευχρηστίας στις οθόνες του χρήστη, ίσως να μπορέσουν να δώσουν ένα ακόμα πιο ευέλικτο και πιο εύχρηστο περιβάλλον.

Παραπομπές

- [1] Wikipedia - The free encyclopaedia, <http://wikipedia.org> (25/08/2008)
- [2] MSDN .Net Framework Development Center, <http://msdn.microsoft.com> (22/08/2008)
- [3] W3C Extensible Markup Language (XML), <http://www.w3.org/xml/> (22/08/2008)
- [4] W3C XML Schema, www.w3.org/XML/Schema (22/08/2008)
- [5] W3C Recommendation 24 June 2003, “SOAP Version 1.2 Part 1: Messaging Framework”, <http://www.w3.org/TR/soap12-part1/>, (18/08/2008)
- [6] W3Schools <http://www.w3schools.com/soap/default.asp>, (18/08/2008)
- [7] Enterprise application integration http://en.wikipedia.org/wiki/Enterprise_application_integration (25/08/2008)
- [8] Park, Jinsoo and Sudha Ram, (2004), Information Systems Interoperability: What Lies Beneath. ACM Transactions on Information Systems. 22(4), 595-632.
- [9] WebSphere Integration Developer, Features and benefits, <http://www-01.ibm.com/software/integration/wid/about/>, (28/09/2008)
- [10] Eclipse Project, <http://www.eclipse.org>, (04/10/2008)
- [11] Ενδιάμεσο Λογισμικό Διαλειτουργικότητας (ΕΛΔ), http://www.infosoc.gr/infosoc/el-GR/services/elibrary/reports_list/prodiagrafes_dialeitourg_pliourof_system/eld.htm (01/01/2004)
- [12] IEEE INTERNET COMPUTING, Asynchronous Mediation for Integrating Business and Operational Processes (02/09/2008)
- [13] Enterprise application integration, Information Management Journal, Mar/Apr 2002, by Gable, Julie, http://findarticles.com/p/articles/mi_qa3937/is_200203/ai_n9019202, (26/08/2008)
- [14] Enterprise Application Integration, <http://java.sun.com/developer/Books/j2ee/connectorch01.pdf>, (26/08/2008)
- [15] S. Radhakrishnan, (25/04/2005), Integrating Enterprise Applications: Backgrounder
- [16] Extensible Markup Language (XML), <http://www.w3.org/XML/>, (26/08/2008)

- [17] Semantic interoperability in building design: Methods and tools, Q.Z. Yang, Y. Zhang, Computer-Aided Design 38 (2006) 1099–1112
- [18] WebSphere Integration Developer, <http://www-01.ibm.com/software/integration/wid/>, (12/09/2008)
- [19] W3C XML Protocol Activity, <http://www.w3.org/2000/xml/>, (22/08/2008)
- [20] ILS 530 - Information Systems Analysis & Design, Fall 2006 - Dr. Chang Suk Kim, Jennifer Macaulay, September 17, 2006
- [21] Mediators in the architecture of future information systems, IEEE, Wiederhold, G., Dept. of Comput. Sci., Stanford Univ., CA., Mar 1992, Page(s): 38-49 (01/03/1992)
- [22] The conceptual basis for mediation services, IEEE, Wiederhold, G.; Genesereth, M., IEEE Expert, Volume 12, Issue 5, Sep/Oct 1997 Page(s):38-47, (01/10/1997)
- [23] EAI as a Service - Combining the Power of Executable EAI Patterns and SaaS, Scheibler, Thorsten Mietzner, Ralph Leymann, Frank, Page (s):15-19 Sept. 2008, 107-116, (IEEE), (19/10/2008)
- [24] How Do Investments in Enterprise Application Integration Drive Stock Prices?, IEEE, Narcyz Roztock; Heinz Roland Weistroffer, System Sciences, 2007. HICSS 2007. 40th Annual Hawaii International Conference on Volume, Jan. 2007 Page(s):221a - 221a, (01/01/2007)