



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

**Πρότυπο Σύστημα Διαχείρισης Μονοπατιών σε  
Ιεραρχίες Συναφών Καταλόγων**

**ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ**

**ΑΓΓΕΛΙΚΗ Α. ΚΟΥΚΟΥΤΣΑΚΗ**

**Επιβλέπων :** Τιμολέων Σελλής  
Καθηγητής Ε.Μ.Π.

Αθήνα, Ιούλιος 2004





# ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΤΟΜΕΑΣ ΤΕΧΝΟΛΟΓΙΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ

## **Πρότυπο Σύστημα Διαχείρισης Μονοπατιών σε Ιεραρχίες Συναφών Καταλόγων**

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ΑΓΓΕΛΙΚΗ Α. ΚΟΥΚΟΥΤΣΑΚΗ

**Επιβλέπων :** Τιμολέων Σελλής  
Καθηγητής Ε.Μ.Π.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 13<sup>η</sup> Ιουλίου 2004 .

.....  
Τιμολέων Σελλής  
Καθηγητής Ε.Μ.Π.

.....  
Ανδρέας-Γεώργιος Σταφυλοπάτης  
Καθηγητής Ε.Μ.Π.

.....  
Νεκτάριος Κοζύρης  
Επίκουρος Καθηγητής Ε.Μ.Π.

Αθήνα, Ιούλιος 2004

.....  
ΑΓΓΕΛΙΚΗ Α. ΚΟΥΚΟΥΤΣΑΚΗ

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

Copyright © ΑΓΓΕΛΙΚΗ Α. ΚΟΥΚΟΥΤΣΑΚΗ 2004

Με επιφύλαξη παντός δικαιώματος. All rights reserved.

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

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

## Περίληψη

Το διαδίκτυο είναι η μεγαλύτερη πηγή πληροφορίας και προσφέρει έναν ενιαίο και απλό τρόπο ανταλλαγής και μετάδοσης πληροφοριών. Πολλοί κόμβοι του παγκόσμιου ιστού παρέχουν στο χρήστη πληροφορίες σχετικές με ένα πεδίο γνώσης, όπως για παράδειγμα ηλεκτρονικά καταστήματα ή πύλες που παρέχουν υλικό για μουσεία, μουσική κλπ. Χαρακτηριστικό αυτών των πυλών είναι ότι η πληροφορία οργανώνεται σε ιεραρχίες. Σκοπός αυτής της διπλωματικής εργασίας είναι η ανάπτυξη ενός συστήματος για τη διαχείριση μονοπατιών σε ιεραρχίες συναφών καταλόγων. Το σύστημα παρέχει δυνατότητες για επιλογή, προβολή, καρτεσιανό γινόμενο, ένωση, τομή και διαφορά πάνω σε δεδομένα δενδρικής δομής, δηλαδή δεδομένα που οργανώνονται σε δενδρικές ιεραρχίες. Η διαφορά σε σχέση με την σχεσιακή άλγεβρα είναι ότι αυτές οι πράξεις εφαρμόζονται σε δένδρα και όχι σε απλές σχέσεις. Η γλώσσα ερωτήσεων του συστήματος στοχεύει στον χειρισμό πολλών καταλόγων που, αν και αποθηκεύουν ίδιου τύπου αντικείμενα, έχουν διαφορετική οργάνωση της ιεραρχίας τους. Οι πράξεις βοηθούν στον προσδιορισμό μονοπατιών που, αν και διαφορετικά, οδηγούν σε ίδια προϊόντα, και δίνουν στον χρήστη την δυνατότητα να επιλέξει τα μονοπάτια που θέλει ώστε να φέρει την ιεραρχία στην επιθυμητή γι' αυτόν μορφή.

### Λέξεις Κλειδιά:

Δέντρο, ιεραρχία, μονοπάτι, αντικείμενο, TSR σχήμα, επιλογή, προβολή, καρτεσιανό γινόμενο, ένωση, τομή, διαφορά, XML, RDF.



## **Abstract**

Internet is nowadays the richest source of information. Many Web information nodes, like portals, provide the user with huge volumes of information related to various knowledge domains, for example cultural information, music records, etc. The main characteristic of these nodes-portals is the hierarchical organization of information on a category/subcategory basis. The aim of this Diploma Thesis is the development of a prototype system to manipulate paths in hierarchies of similar knowledge domains. The system provides selection, projection, cartesian product, union, intersection, and difference operations on data organized in hierarchies. The operations differ from those of relational algebra, since they are applied on paths of hierarchies and not on relations. The query language implemented aims at the manipulation of catalogs used in portals, which organize same data in different ways. The operations detect paths in such catalogs through which the user can reach same data. Such paths can then be re-organized, giving new hierarchies according to the user's information need.

## **Keywords:**

Tree, hierarchy, path, item, TSR schema, selection, projection, cartesian product, union, intersection, difference, XML, RDF.





## Ευχαριστίες

Η διπλωματική εργασία εκπονήθηκε στο Εργαστήριο Συστημάτων Βάσεων Γνώσεων και Δεδομένων (ΕΣΒΓΔ) του Εθνικού Μετσόβιου Πολυτεχνείου. Αποτελέσει μία πολύ καλή αφορμή για την ενασχόληση μου με το χώρο των συστημάτων διαχείρισης καταλόγων στο WWW καθώς και με το χώρο του προγραμματισμού. Στο σημείο αυτό θα ήθελα να ευχαριστήσω τον καθηγητή κ. Τιμολέων Σελλή για την καθοδήγηση και τη βοήθεια που μου προσέφερε με τις παρατηρήσεις του. Επίσης, θέλω να ευχαριστήσω θερμά τον διδάκτορα Θοδωρή Δαλαμάγκα για τις πολύτιμες συμβουλές, τις ιδέες και τις παρατηρήσεις του αλλά και για τον χρόνο που αφιέρωσε για όλα τα θέματα που ανέκυψαν κατά τη διάρκεια της εκπόνησης της διπλωματικής εργασίας μου. Τέλος, δε θα μπορούσα να μην ευχαριστήσω τους γονείς μου για τις θυσίες, την υλική αλλά κυρίως την ηθική και ψυχολογική στήριξη που μου παρέχουν τόσα χρόνια. Χωρίς αυτούς σίγουρα όλα θα ήταν διαφορετικά.



*Στους γονείς μου, Ανδρέα και Αθηνά,  
στην Παναγιώτα και στον Παναγιώτη*



## Πίνακας περιεχομένων

<b>1</b>	<b>Εισαγωγή.....</b>	<b>17</b>
1.1	Αντικείμενο της διπλωματικής .....	18
1.2	Οργάνωση του τόμου.....	20
<b>2</b>	<b>Περιγραφή Θέματος.....</b>	<b>23</b>
2.1	Θεωρητικό υπόβαθρο.....	23
2.1.1	<i>XML Γλώσσα.....</i>	<i>24</i>
2.1.2	<i>RDF Μοντέλο.....</i>	<i>26</i>
2.2	Σχετικές εργασίες.....	27
2.2.1	<i>Γλώσσες ερωτήσεων δεδομένων ιστού.....</i>	<i>27</i>
2.2.2	<i>Γλώσσες ερωτήσεων ημιδομημένων δεδομένων.....</i>	<i>27</i>
2.2.3	<i>Άλγεβρες ημιδομημένων δεδομένων.....</i>	<i>30</i>
2.2	Στόχος.....	33
<b>3</b>	<b>Θεωρητική μελέτη .....</b>	<b>35</b>
3.1	Δικτυακός κατάλογος - Μονοπάτι - Tree Structure Relation (TSR) .....	35
3.2	$R_p$ - $R_i$ Μοντέλο .....	39
3.2.1	<i>Περιγραφή μοντέλου.....</i>	<i>39</i>
3.2.2	<i>Κατασκευή - Μετάβαση από TSR σχήμα στο μοντέλο <math>R_p</math> - <math>R_i</math>.....</i>	<i>39</i>
3.3	Σύγκριση μονοπατιών .....	41
3.3.1	<i>Ισότητα - Ανισότητα.....</i>	<i>41</i>
3.3.2	<i>Αυστηρό υποσύνολο - Αυστηρό υπερσύνολο .....</i>	<i>42</i>
3.3.3	<i>Χαλαρό υποσύνολο - Χαλαρό υπερσύνολο .....</i>	<i>42</i>
3.4	Ερωτήσεις - Πράξεις.....	43
3.4.1	<i>Επιλογή (selection).....</i>	<i>44</i>
3.4.2	<i>Προβολή (projection).....</i>	<i>45</i>
3.4.3	<i>Καρτεσιανό γινόμενο (cartesian product) .....</i>	<i>46</i>
3.4.4	<i>Ένωση (union).....</i>	<i>49</i>
3.4.5	<i>Τομή (intersection).....</i>	<i>49</i>

3.4.6	Διαφορά ( <i>difference</i> ) .....	50
<b>4</b>	<b>Ανάλυση και σχεδίαση .....</b>	<b>53</b>
4.1	Ανάλυση - Περιγραφή Αρχιτεκτονική.....	53
4.1.1	Διαχωρισμός υποσυστημάτων.....	53
4.1.2	Περιγραφή υποσυστημάτων.....	56
4.1.2.1	Υποσύστημα αποθήκευσης TSR σχημάτων .....	56
4.1.2.2	Υποσύστημα ανάγνωσης TSR σχημάτων.....	56
4.1.2.3	Υποσύστημα απεικόνισης TSR σχημάτων .....	56
4.1.2.4	Υποσύστημα διαπροσωπείας χρήστη .....	57
4.1.2.5	Υποσύστημα διαγραφής TSR σχημάτων .....	58
4.1.2.6	Υποσύστημα ερωτήσεων - πράξεων.....	58
4.1.2.7	Υποσύστημα ανάγνωσης ερωτήσεων από αρχείο.....	58
4.1.2.8	Υποσύστημα ελέγχου εγκυρότητας TSR σχημάτων.....	59
4.1.2.9	Υποσύστημα δημιουργίας TSR σχημάτων από RDFS αρχείο.....	59
4.1.2.10	Υποσύστημα προσθήκης εγγραφών TSR σχημάτων .....	60
4.2	Σχεδίαση συστήματος.....	61
4.2.1	Εφαρμογή διαχείρισης XML αρχείου.....	61
4.2.2	Εφαρμογή διαχείρισης βάσης δεδομένων .....	63
4.2.3	Εφαρμογή απεικόνισης αποθηκευμένου TSR σχήματος.....	64
4.2.4	Εφαρμογή διαγραφής TSR σχήματος .....	65
4.2.5	Εφαρμογή μεταφοράς TSR σχήματος από τη βάση σε XML αρχείο.....	66
4.2.6	Εφαρμογή μεταφοράς TSR σχήματος από XML αρχείο στη βάση.....	66
4.2.7	Εφαρμογή αλλαγής χαρακτηριστικών στοιχείων TSR σχήματος .....	68
4.2.8	Εφαρμογή εισαγωγής ερώτησης.....	70
4.2.9	Εφαρμογή εισαγωγής ερώτησης από αρχείο .....	71
4.2.10	Εφαρμογή εισαγωγής TSR σχημάτων από XML αρχείο .....	72
4.2.11	Εφαρμογή εισαγωγής TSR σχημάτων από RDFS αρχείο .....	73
4.2.12	Εφαρμογή προσθήκης εγγραφών TSR σχημάτων.....	76
<b>5</b>	<b>Υλοποίηση.....</b>	<b>79</b>
5.1	Πλατφόρμες και προγραμματιστικά εργαλεία .....	79
5.2	Λεπτομέρειες υλοποίησης.....	80

5.2.1	<i>Βάση δεδομένων</i> .....	80
5.2.2	<i>Μεταγλωττιστής</i> .....	81
5.2.2.1	Γραμματική της γλώσσας.....	82
5.2.2.2	Συντακτική ανάλυση - Χειρισμός εντολών.....	83
5.2.3	<i>Υποσύστημα απεικόνισης TSR σχημάτων</i> .....	83
5.2.4	<i>XML, RDFS, TXT QRY αρχεία</i> .....	85
5.2.5	<i>Περιγραφή κλάσεων</i> .....	86
<b>6</b>	<b>Έλεγχος</b> .....	<b>119</b>
6.1	Μεθοδολογία Ελέγχου .....	119
6.2	Αναλυτική παρουσίαση έλεγχου.....	119
<b>7</b>	<b>Επίλογος</b> .....	<b>147</b>
7.1	Σύνοψη και συμπεράσματα.....	147
7.2	Μελλοντικές επεκτάσεις .....	149
<b>8</b>	<b>Βιβλιογραφία</b> .....	<b>151</b>





# 1

## *Εισαγωγή*

Το διαδίκτυο (internet) είναι η μεγαλύτερη πηγή πληροφορίας και προσφέρει έναν ενιαίο και απλό τρόπο ανταλλαγής και μετάδοσης πληροφοριών. Παρά τον τεράστιο όγκο δεδομένων και πληροφοριών που προωθείται και αναζητείται στον ιστό καθημερινά, δεν υπάρχει ένας ενιαίος και αυστηρός τρόπος οργάνωσης των δεδομένων, με αποτέλεσμα να υπάρχουν πολλές δυσκολίες στην ανταλλαγή και επεξεργασία τους [1]. Ο Ιστός για να εκμεταλλευτεί τη δυναμική του και να γίνει μία παγκόσμια προσβάσιμη πλατφόρμα επεξεργασίας και ανταλλαγής δεδομένων από πληροφοριακά συστήματα, θα πρέπει να εξελιχθεί στο Σημασιολογικό Ιστό [2,3].

Ο Σημασιολογικός Ιστός είναι μία επέκταση του ιστού στην οποία η πληροφορία αποκτά σημασία ώστε να υποστηριχθούν διαδικασίες αναζήτησης, αυτοματισμού και ενοποίησης δεδομένων. Ο Σημασιολογικός Ιστός δίνει δομή, οργάνωση και σημασιολογία στο περιεχόμενο του ιστού αλλά μπορεί να λειτουργήσει μόνο αν η διαθέσιμη πληροφορία γίνει περισσότερο κατανοητή σε επίπεδο μηχανής. Για το λόγο αυτό νέες γλώσσες και μοντέλα έχουν προταθεί ώστε να εμπλουτιστεί η διαθέσιμη πληροφορία στον ιστό με σημασιολογία και δομή. Δύο γνωστές τεχνολογίες που έχουν αναπτυχθεί και υποστηρίζουν τη λειτουργία του σημασιολογικού ιστού είναι η γλώσσα XML (eXtensible Markup Language) [4] και το μοντέλο RDF (Resource Description Framework) [5], οι οποίες θα αναλυθούν στη συνέχεια. Τα ιεραρχικά σχήματα παίζουν σημαντικό ρόλο στο Σημασιολογικό Ιστό καθώς

χρησιμοποιούνται για να εμπλουτίσουν σημασιολογικά την παρεχόμενη πληροφορία. Μέχρι τώρα οι ιεραρχίες έχουν χρησιμοποιηθεί σαν απλοί σημασιολογικοί οδηγοί για διάσχιση και αναζήτηση πληροφορίας, όπως για παράδειγμα στη γλώσσα XQuery [22]. Όμως, στο περιβάλλον του Ιστού, όπου η αναζήτηση πληροφορίας σε ένα πεδίο γνώσης απαιτεί επεξεργασία σε πολλές πηγές σχετικές με το πεδίο αυτό, προσδιορίζεται η ανάγκη της ανάδειξης των ιεραρχιών σε οντότητες πρώτης τάξης, έτσι ώστε να διαχειρίζονται ως ολοκληρωμένες οντότητες και όχι ως σύνολα από μεμονωμένους κόμβους.

Ένα σημαντικό θέμα που προκύπτει σε σχέση με τις ιεραρχίες των πυλών καταλόγων είναι ο προσδιορισμός των σημασιολογικά ίδιων κόμβων. Το πρόβλημα αυτό είναι γνωστό ως *ταίριασμα σχημάτων* [30]. Τα ταίριασμα σχημάτων προσδιορίζει τις κατηγορίες σε μία ιεραρχία που είναι ίδιες σημασιολογικά με κατηγορίες από μία άλλη ιεραρχία, έστω και αν χρησιμοποιούνται διαφορετικές λέξεις που τις περιγράφουν. Για παράδειγμα, σε μία ιεραρχία η τιμή ενός προϊόντος περιγράφεται με τη λέξη *price*, ενώ σε μία άλλη με τη λέξη *cost*. Για τις ανάγκες αυτής της διπλωματικής εργασίας θεωρήσαμε το πρόβλημα του ταυριάσματος σχημάτων λυμένο.

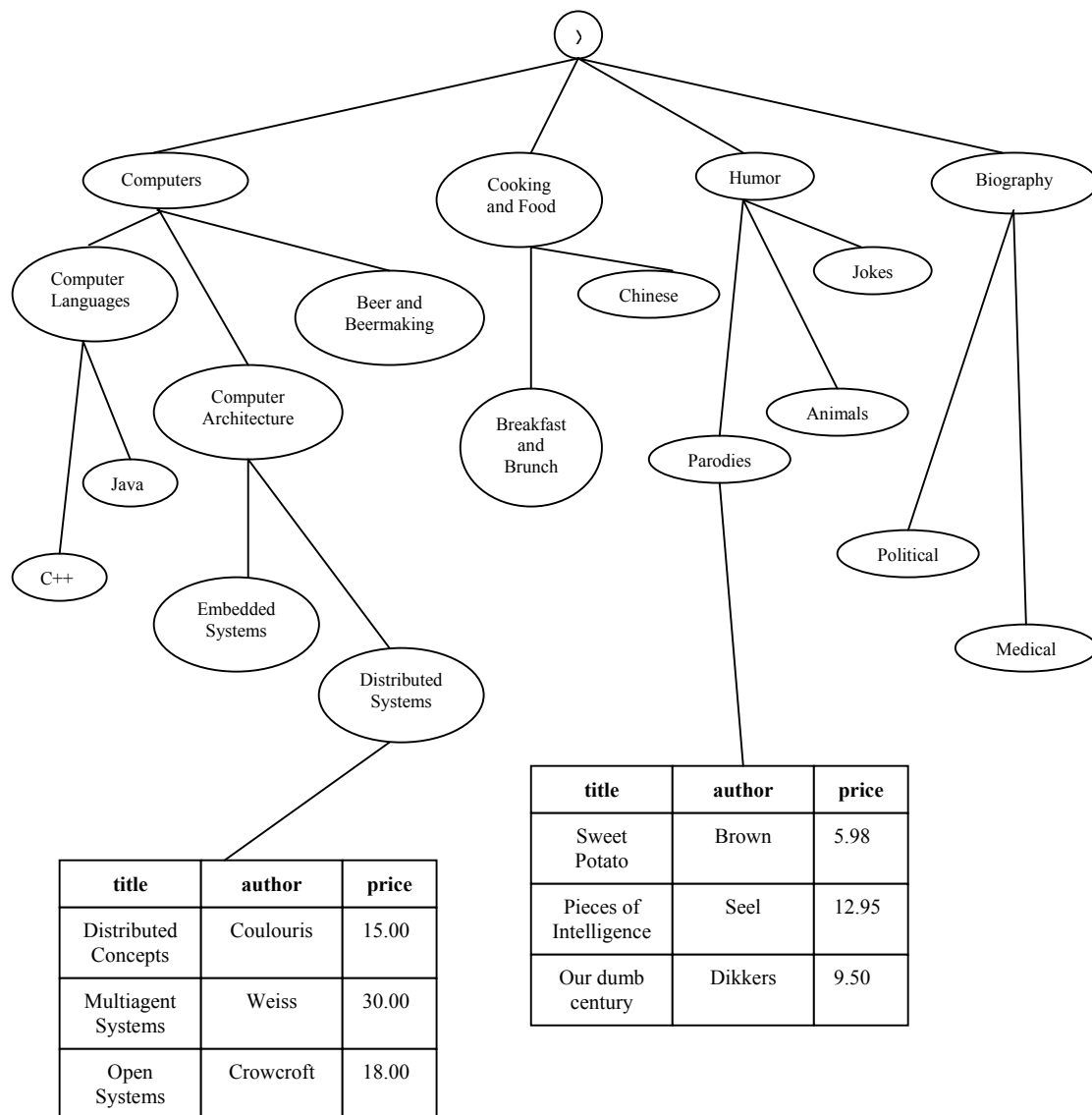
Επίσης, ένα ακόμα πρόβλημα σχετικά με τις ιεραρχίες είναι η διαφορετική οργάνωση των μονοπατιών σε ιεραρχίες συναφών καταλόγων, δηλαδή σε καταλόγους που παρέχουν στο χρήστη πληροφορίες σχετικές με ένα πεδίο γνώσης. Για παράδειγμα, θεωρούμε δύο καταλόγους που παρέχουν πληροφορίες για φωτογραφικό εξοπλισμό οργανωμένες σε θεματικές ιεραρχίες με κατηγορίες και υποκατηγορίες που υποστηρίζουν αναζήτηση και διάσχιση. Για την αναζήτηση πληροφοριών για φακούς στον πρώτο κατάλογο, πρέπει να ακολουθήσουμε το μονοπάτι `"/cameras and lenses/lenses'`, ενώ στον δεύτερο κατάλογο χρειαζόμαστε το μονοπάτι `"/photo/35mmSystems/lenses'`.

## ***1.1 Αντικείμενο της διπλωματικής***

Η διπλωματική εργασία ασχολείται με το δεύτερο πρόβλημα που προκύπτει σχετικά με τις ιεραρχίες συναφών καταλόγων, δηλαδή τη διαφορετική οργάνωση των μονοπατιών τους. Συγκεκριμένα, στοχεύει στο σχεδιασμό και στην υλοποίηση ενός νέου συστήματος για τη διαχείριση αυτών των μονοπατιών. Η γλώσσα ερωτήσεων του συστήματος χειρίζεται πολλούς καταλόγους που, αν και αποθηκεύουν ίδιου τύπου αντικείμενα, έχουν διαφορετική οργάνωση της ιεραρχίας τους. Οι πράξεις της γλώσσας ερωτήσεων βοηθούν στον προσδιορισμό μονοπατιών που, αν και διαφορετικά, οδηγούν σε ίδια προϊόντα, και δίνουν στο χρήστη τη δυνατότητα να επιλέξει τα μονοπάτια που θέλει ώστε να φέρει την ιεραρχία

στην επιθυμητή γι' αυτόν μορφή. Η γλώσσα ερωτήσεων που υλοποιείται αποτελείται από ένα σύνολο ερωτήσεων ανάλογων των ερωτήσεων της σχεσιακής άλγεβρας, οι οποίες όμως αποτιμώνται σε δενδρικές δομές δεδομένων, δίνοντας ως αποτέλεσμα ένα σύνολο μονοπατιών που οδηγούν σε εγγραφές δεδομένων.

Ένας δικτυακός τύπος με μεγάλο όγκο δεδομένων συγκεκριμένου αντικειμένου ή τομέα ονομάζεται κάθετος δικτυακός κατάλογος (vertical portal catalog). Αυτοί οι δικτυακοί τύποι παρέχουν καλύτερες δυνατότητες ερωτήσεων από ότι οι παραδοσιακές μηχανές ερωτήσεων. Χρησιμοποιώντας το μοντέλο της XML, ο τεράστιος όγκος πληροφοριών του καταλόγου οργανώνεται με συγκεκριμένη ιεραρχία, προσθέτοντας ταυτόχρονα το στοιχείο της δομικής πληροφορίας το οποίο δεν υπάρχει στις κλασσικές βάσεις δεδομένων. Το Σχήμα 1.1 παρουσιάζει την ιεραρχία για τον κάθετο κατάλογο παρουσίασης βιβλίων powells (www.powells.com).



Σχήμα 1.1

Στα φύλλα της ιεραρχίας βρίσκονται τοποθετημένες ομάδες εγγραφών με τα χαρακτηριστικά τους για κάθε αντίστοιχη κατηγορία βιβλίων με τη μορφή των κλασικών σχέσεων. Με τη βοήθεια της γλώσσας ερωτήσεων που θα υλοποιηθεί, θα μπορούν να εκφραστούν ερωτήματα πάνω στα μονοπάτια της ιεραρχίας αλλά και ερωτήσεις σχεσιακής επιλογής και προβολής πάνω στις εγγραφές των φύλλων της ιεραρχίας. Επίσης δίνεται η δυνατότητα συνδυασμού δύο ή περισσότερων ιεραρχιών με τις πράξεις της ένωσης, της τομής, της διαφοράς και του καρτεσιανού γινομένου. Με αυτόν τον τρόπο παράγεται μία καινούρια ιεραρχία με τις επιθυμητές εγγραφές προϊόντων και τα επιθυμητά μονοπάτια που καταλήγουν σε αυτές.

Δύο παραδείγματα δυνατών ερωτημάτων είναι τα εξής:

- 1) Από ένα σχήμα φέρε όλες τις εγγραφές των βιβλίων για κατανεμημένα συστήματα με τιμή μικρότερη από \$20.00 με την προϋπόθεση ότι τα μονοπάτια που οδηγούν σε αυτές περιέχουν το μονοπάτι /Computer Architecture/Distributed Systems.
- 2) Από πολλά σχήματα φέρε όλες τις εγγραφές των βιβλίων για τις παρωδίες με την προϋπόθεση ότι τα μονοπάτια που οδηγούν σε αυτές είναι ίδια με το μονοπάτι /Humor/Parodies (η ερώτηση αυτή έχει χαρακτηριστικά ένωσης).

Η διπλωματική εργασία εκτός από τη γλώσσα ερωτήσεων που αναπτύσσει, παρέχει στο χρήστη ένα εύχρηστο σύστημα για την εισαγωγή της επιθυμητής ερώτησης, την απεικόνιση των ιεραρχιών σε γραφική μορφή και τη διαχείριση των μονοπατιών αλλά και των καταλόγων που τα περιέχουν.

## ***1.2 Οργάνωση του τόμου***

Η διπλωματική εργασία οργανώνεται στα παρακάτω κεφάλαια:

- 1) Στο 2<sup>ο</sup> Κεφάλαιο γίνεται μία σύντομη παρουσίαση του θεωρητικού υπόβαθρου και των εργασιών με συναφές θέμα.
- 2) Στο 3<sup>ο</sup> Κεφάλαιο παρουσιάζεται η θεωρητική μελέτη του θέματος όπου δίνονται οι απαραίτητοι ορισμοί και ορίζονται οι βασικοί τελεστές των πράξεων επιλογής (selection), προβολής (projection), καρτεσιανού γινομένου (cartecian product), ένωσης (union), τομής (intersection) και διαφοράς (difference).

- 3) Στο 4<sup>ο</sup> Κεφάλαιο παρουσιάζεται η αρχιτεκτονική του συστήματος. Αρχικά παρουσιάζονται τα ανεξάρτητα υποσυστήματα που το αποτελούν, ενώ στη συνέχεια παρουσιάζονται όλες οι εφαρμογές που υλοποιεί το σύστημα.
- 4) Στο 5<sup>ο</sup> Κεφάλαιο παρουσιάζεται η υλοποίηση του συστήματος και αναφέρονται εν συντομία οι πλατφόρμες και τα προγραμματιστικά εργαλεία που χρησιμοποιήθηκαν.
- 5) Στο 6<sup>ο</sup> Κεφάλαιο δίνεται ένα σενάριο ελέγχου του συστήματος με το οποίο γίνεται σύγκριση των αποτελεσμάτων με τα αναμενόμενα.
- 6) Στο 7<sup>ο</sup> Κεφάλαιο επιχειρείται μία σύνοψη της εργασίας και παρουσιάζονται πιθανές μελλοντικές επεκτάσεις και θέματα γύρω από το αντικείμενο αυτής της διπλωματικής εργασίας.
- 7) Στο 8<sup>ο</sup> Κεφάλαιο παρουσιάζεται η σχετική βιβλιογραφία.



# 2

## *Περιγραφή Θέματος*

Το κεφάλαιο αυτό παρουσιάζει το θεωρητικό υπόβαθρο της διπλωματικής εργασίας. Μέρος του κεφαλαίου παρουσιάζεται στην αναφορά [27].

### *2.1 Θεωρητικό υπόβαθρο*

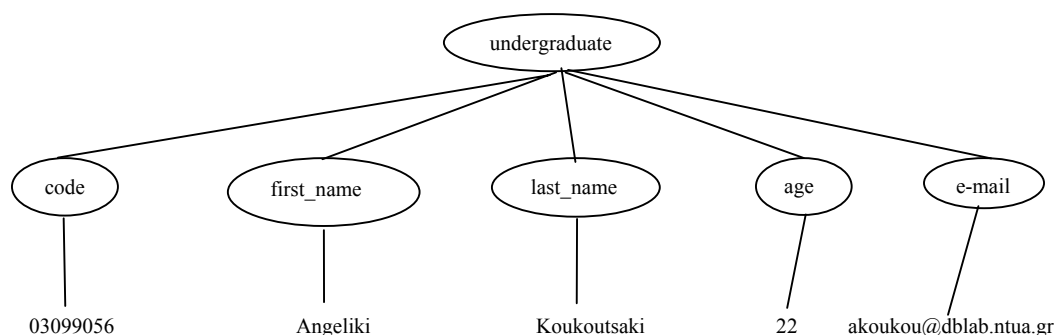
Τα δεδομένα του ιστού χαρακτηρίζονται ως *ημιδομημένα δεδομένα* [1], με την έννοια ότι δεν υπάρχει αυστηρά καθορισμένο σχήμα που να τα συνοδεύει, όπως γίνεται με τα δομημένα δεδομένα που διαχειρίζονται οι παραδοσιακές βάσεις δεδομένων. Το σχήμα στα ημιδομημένα δεδομένα εμπεριέχεται στα ίδια τα δεδομένα. Τα μοντέλα αναπαράστασης ημιδομημένων δεδομένων έχουν συνήθως μορφή γράφου ή δέντρου. Στην συνέχεια της ενότητας αυτής γίνεται μία σύντομη παρουσίαση της γλώσσας XML (eXtensible Markup Language), η οποία είναι ένα μοντέλο ημιδομημένων δεδομένων, και του RDF μοντέλου (Resource Description Framework), που είναι ένα μοντέλο για την περιγραφή μεταδεδομένων, δηλαδή δεδομένων για τα δεδομένα.

### 2.1.1 XML Γλώσσα

Με τη βοήθεια της XML γλώσσας, οι σελίδες και τα δεδομένα του ιστού σημαδεύονται με ετικέτες (tags). Οι ετικέτες βοηθούν τα προγράμματα του ιστού να ανακαλύψουν ευκολότερα τα δεδομένα, δίνοντας τους σημασιολογία και δομή. Η γλώσσα XML δίνει μία κωδικοποίηση στα δεδομένα που βοηθάει την ανταλλαγή και επαναχρησιμοποίηση τους από πολλές εφαρμογές. Βασικές μονάδες της XML είναι τα *στοιχεία*, τα οποία είναι κομμάτια κειμένου που περικλείονται από ετικέτες και μπορεί να περιέχουν άλλα φωλιασμένα στοιχεία (*υποστοιχεία*) ή απλό κείμενο. Τα στοιχεία μπορούν να έχουν *ιδιότητες* που αντιστοιχούν στα χαρακτηριστικά από το χώρο των μοντέλων δεδομένων. Μία ιδιότητα εμφανίζεται μόνο μία φορά σε κάθε στοιχείο σε αντίθεση με τα υποστοιχεία που μπορούν να εμφανιστούν πολλές φορές. Για παράδειγμα, η πληροφορία για τον συγγραφέα αυτής της διπλωματικής εργασίας κωδικοποιείται σε ένα αρχείο XML με το στοιχείο `undergraduate` και τα υποστοιχεία `first_name`, `last_name`, `age`, `e-mail` ως εξής:

```
<undergraduate code = "03099056">  
  <first_name>Angeliki</first_name>  
  <last_name>Koukoutsaki</last_name>  
  <age>22</age>  
  <e-mail>akoukou@dblab.ntua.gr</e-mail>  
</undergraduate>
```

Στο Σχήμα 2.1 παρουσιάζεται μία δενδρική αναπαράσταση του προηγούμενου XML αρχείου.



Σχήμα 2.1

Τα XML αρχεία δεν έχουν πάντα την ίδια δομή και το ίδιο λεξιλόγιο ακόμα και αν προέρχονται από τα ίδια πεδία εφαρμογής. Απαραίτητη είναι λοιπόν η ύπαρξη ενός σχήματος



που θα περιορίζει την κωδικοποίηση των δεδομένων σε συγκεκριμένη δομή και λεξιλόγιο. Τα DTD (Document Type Definition) [4] και XML Schemas [6] είναι τέτοια σχήματα. Ένα DTD είναι μία γραμματική χωρίς συμφραζόμενα για XML αρχεία και αποτελεί μέρος της XML γλώσσας. Ένα έγκυρο XML αρχείο είναι αυτό που έχει DTD και είναι συμβατό μαζί του. Για παράδειγμα στη συνέχεια δίνονται παραδείγματα κανόνων DTD που βάζουν περιορισμούς στη μορφή των XML αρχείων στα οποία αναφέρονται.

```
<!ELEMENT undergraduate (first_name,last_name,age,e-mail)>
<!ELEMENT first_name (#PCDATA)>
<!ELEMENT last_name (#PCDATA)>
<!ELEMENT age (#PCDATA)>
<!ELEMENT e-mail (#PCDATA)>
<!ATTLIST undergraduate code CDATA #REQUIRED>
```

Σύμφωνα με τους παραπάνω κανόνες το στοιχείο undergraduate πρέπει να αποτελείται από τα υποστοιχεία first\_name, last\_name, age και e-mail με την ίδια σειρά που εμφανίζονται. Όλα τα παραπάνω υποστοιχεία πρέπει να περιέχουν απλό κείμενο ενώ το στοιχείο undergraduate πρέπει να έχει μία ιδιότητα με όνομα code.

Για πιο πλούσιους περιορισμούς στη δομή και το λεξιλόγιο XML αρχείων, συμπεριλαμβανομένου και πληροφορίας τύπων δεδομένων, μπορεί να χρησιμοποιηθεί το σχήμα XML Schema. Οι παρακάτω κανόνες XML Schema απαιτούν ότι α) το στοιχείο undergraduate πρέπει να αποτελείται από τα υποστοιχεία first\_name, last\_name, age και e-mail με την ίδια σειρά με την οποία εμφανίζονται, β) όλα τα υποστοιχεία του α) πρέπει να είναι συμβολοσειρές εκτός από το age που πρέπει να είναι ακέραιος και γ) το στοιχείο undergraduate πρέπει να έχει ως ιδιότητα τη συμβολοσειρά code.

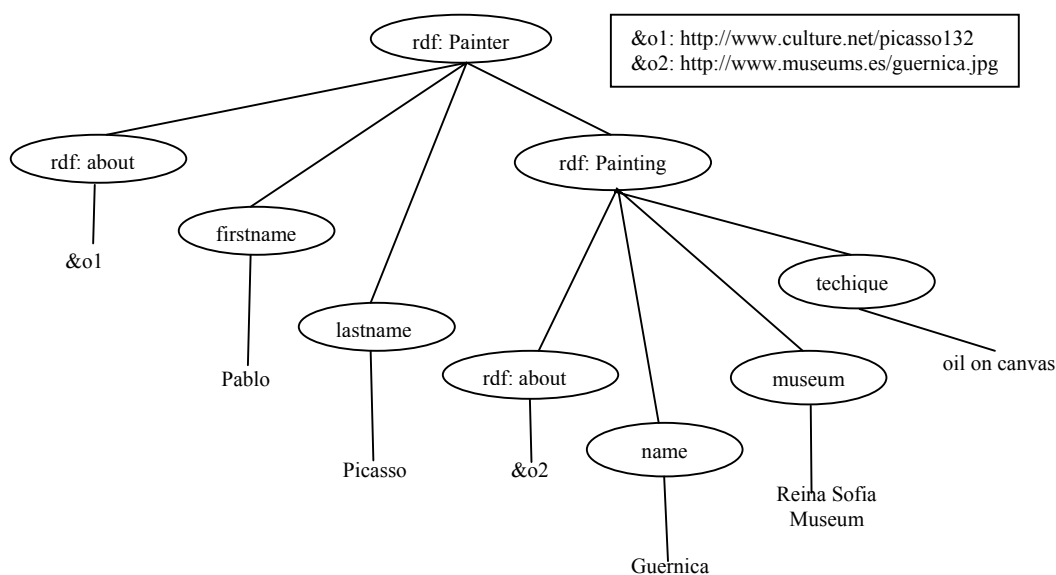
```
<xs element name = "undergraduate">
  <xs:complexType>
    <xs:sequence>
      <xs element name = "first_name" type = "xs:string"/>
      <xs element name = "last_name" type = "xs:string"/>
      <xs element name = "age" type = "xs:integer"/>
      <xs element name = "e-mail" type = "xs:string"/>
    </xs:sequence>
    <xs:attribute name = "code" type = "xs:string"/>
  </xs:complexType>
</xs:element>
```

## 2.1.2 RDF Μοντέλο

Η γλώσσα XML είναι ένα μόνο κομμάτι μιας μεγαλύτερης εικόνας τεχνολογιών του ιστού. Τα XML αρχεία μπορούν να παραστήσουν και *μεταδεδομένα*, δηλαδή δεδομένα για τα δεδομένα. Το πλαίσιο RDF είναι ένα *εργαλείο αναπαράστασης μεταδεδομένων*. Χρησιμοποιεί τριάδες της μορφής *υποκείμενο, ρήμα, αντικείμενο*, οι οποίες και αποτελούν δηλώσεις για τα αντικείμενα του ιστού όπως σελίδες, συγγραφείς, προγράμματα κτλ. Όπως η γραμματικές DTD ενεργούν ως σχήμα για τα XML αρχεία, έτσι και τα σχήματα RDF Schema [5] ενεργούν ως σχήματα για τα RDF αρχεία, δηλώνοντας τάξεις και χαρακτηριστικά για να συσχετίσουν ή να ομαδοποιήσουν τα αντικείμενα. Έστω το παρακάτω XML αρχείο το οποίο κωδικοποιεί RDF δηλώσεις:

```
<?xml version = "1.0"?>
<rdf: Painter rdf:about = "http://www.culture.net/picasso132">
  <firstname>Pablo</firstname>
  <lastname>Picasso</lastname>
  <rdf: Painting rdf:about = "http://www.museums.es/guernica.jpg">
    <name>Guernica</name>
    <technique>oil on canvas</technique>
    <museum>Reina Sofia Museum</museum>
  </rdf: Painting>
</rdf: Painter>
```

Το Σχήμα 2.2 παρουσιάζει μία γραφική αναπαράσταση του προηγούμενου XML αρχείου:



Σχήμα 2.2

## **2.2 Σχετικές εργασίες**

Αν θεωρήσουμε τον ιστό σαν μία τεράστια βάση δεδομένων, απαιτείται η παροχή δυνατοτήτων αναζήτησης που ξεπερνούν το κλασικό παράδειγμα των ερωτήσεων με χρήση λέξεων κλειδιών που χρησιμοποιείται στις παραδοσιακές μηχανές αναζήτησης. Η δομή στον ιστό και συγκεκριμένα η εσωτερική δομή των σελίδων αλλά και η εξωτερική δομή που ορίζουν οι σύνδεσμοι, θα πρέπει να συμπεριληφθούν στους μηχανισμούς αναζήτησης. Στην ενότητα αυτή παρουσιάζονται περιληπτικά γλώσσες ερωτήσεων δεδομένων ιστού, γλώσσες ερωτήσεων ημιδομημένων δεδομένων και άλγεβρες ημιδομημένων δεδομένων.

### **2.2.1 Γλώσσες ερωτήσεων δεδομένων ιστού**

Οι γλώσσες ερωτήσεων δεδομένων ιστού διαχωρίζονται σε γλώσσες πρώτης και δεύτερης γενιάς [7]. Συγκεκριμένα η πρώτη γενιά των γλωσσών αυτών συνδυάζει ερωτήσεις περιεχομένου των κλασικών μηχανών αναζήτησης με ερωτήσεις δομής. Οι ερωτήσεις περιεχομένου ανιχνεύουν ταιριάσματα προτύπων κειμένου (λέξεις κλειδιά) πάνω στο περιεχόμενο των ιστοσελίδων. Οι ερωτήσεις δομής ανιχνεύουν ταιριάσματα δενδρικών προτύπων ή προτύπων γράφων πάνω στην εσωτερική δομή των ιστοσελίδων ή στην εξωτερική δομή που ορίζουν οι σύνδεσμοι. Παραδείγματα τέτοιων γλωσσών είναι οι W3QL [8], Web-SQL [9], WebLog [10]. Όμως οι γλώσσες αυτές αντιμετωπίζουν τα κείμενα του ιστού ως ατομικά αντικείμενα και ελέγχουν απλά αν περιέχουν ή όχι πρότυπα κειμένου ή αν έχουν συνδέσμους προς άλλα κείμενα. Δεν παρέχουν τη δυνατότητα επεξεργασίας επιμέρους κομματιών κειμένου. Η δεύτερη γενιά γλωσσών ερωτήσεων δεδομένων ιστού έχει αυξημένες εκφραστικές δυνατότητες. Οι γλώσσες αυτές αντιμετωπίζουν τα κείμενα του ιστού ως σύνθετες οντότητες και παρέχουν δυνατότητες αναδόμησης, δημιουργώντας σύνθετα αντικείμενα, ενώ συγχρόνως δίνουν έμφαση στο χειρισμό ημιδομημένων δεδομένων. Τέτοιες γλώσσες είναι οι WebOQL [11], STRUQL [12], FLORID [13].

### **2.2.2 Γλώσσες ερωτήσεων ημιδομημένων δεδομένων**

Μία από τις πιο γνωστές γλώσσες ερωτήσεων ημιδομημένων δεδομένων είναι η Lorel [14] η οποία χρησιμοποιεί εκφράσεις μονοπατιών και δέσιμο μεταβλητών για να ανακαλύψει ταιριάσματα σε κατευθυνόμενους γράφους με ετικέτες που αναπαριστούν ημιδομημένα

δεδομένα. Για παράδειγμα η παρακάτω ερώτηση σε γλώσσα Lorel βρίσκει όλους τους προπτυχιακούς φοιτητές που έχουν ηλικία μεγαλύτερη από 26:

```
select X.name
from ntua.students.undergraduate X
where X.age>26
```

Η γλώσσα UnQL [15] χρησιμοποιεί πρότυπα για να εκφράσει συνθήκες ερώτησης πάνω σε δομές. Τα πρότυπα δίνουν ταιριάσματα μεταβλητών πάνω σε κόμβους. Οι UnQL ερωτήσεις μεταφράζονται σε δομικές αναδρομές, οι οποίες μοιάζουν με τα πρότυπα που χρησιμοποιούνται σε συναρτησιακές γλώσσες προγραμματισμού. Για παράδειγμα η παρακάτω ερώτηση βρίσκει όλους τους προπτυχιακούς φοιτητές που είναι πάνω από 26 ετών:

```
select {ntua:{student:{undergraduate:{name: X}}}}
where {ntua:{student:{undergraduate:{name: X, age: Y}}}} in db, Y>26
```

Η γλώσσα XML-QL [16,17] συνδυάζει τη σύνταξη της XML με δυνατότητες ερωτήσεων βασισμένες σε εκφράσεις μονοπατιού και δέσιμο μεταβλητών πάνω σε κατευθυνόμενους γράφους με ετικέτες. Για παράδειγμα η παρακάτω ερώτηση βρίσκει όλους τους προπτυχιακούς φοιτητές που είναι μεγαλύτεροι από 26 ετών, εκτελώντας ταυτόχρονα αναδόμηση στο αποτέλεσμα με τον τελεστή CONSTRUCT:

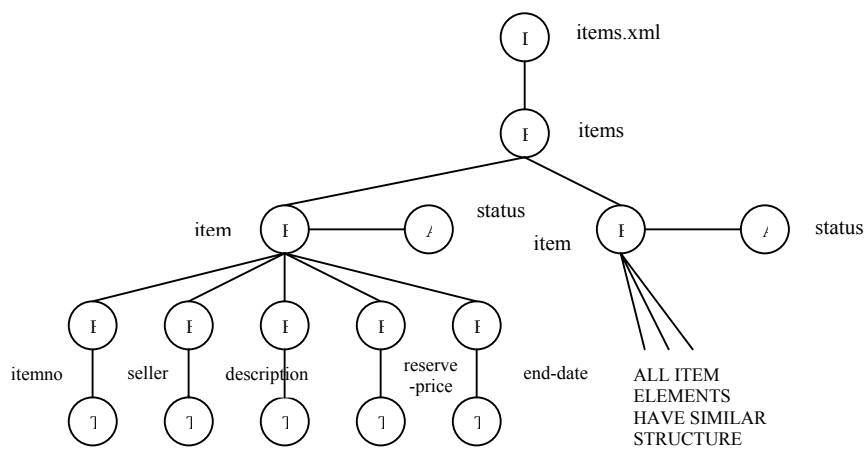
```
WHERE <ntua><student><undergraduate>
  <name>$X</name>
  <age>$Y</age>
</undergraduate></student></ntua>
in www.ntua.gr/db.xml, $Y>26
CONSTRUCT <ntua><old-students>
  <name>$X</name>
</old-students></ntua>
```

Άλλες γλώσσες ερωτήσεων για XML δεδομένα είναι οι: XML-GL [18], XQL [19], QUILT [20] και XQuery [21,22].

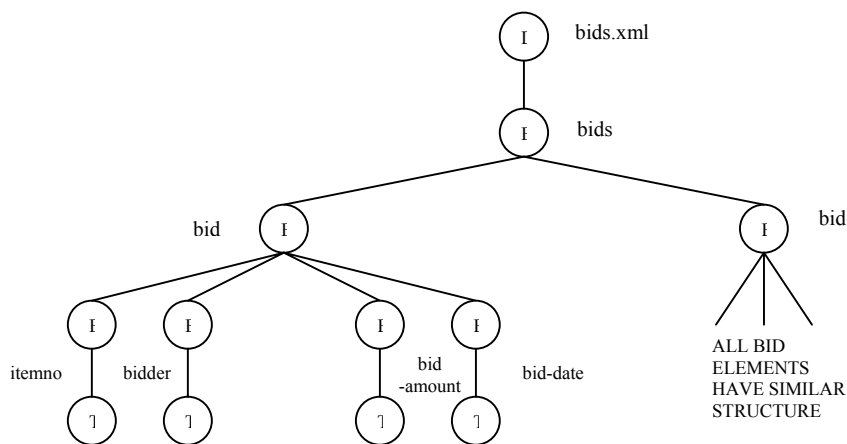
Η τελευταία είναι σήμερα η βασική γλώσσα ερωτήσεων για XML δεδομένα και ο σχεδιασμός της στηρίζεται σε παλαιότερες γλώσσες (XQL, XML-QL, Lorel κλπ). Είναι μία συναρτησιακή γλώσσα με εκφράσεις μονοπατιού, κατασκευαστές στοιχείων, κλήσεις συναρτήσεων, αριθμητικές και λογικές εκφράσεις, εκφράσεις συνθηκών κλπ. Βασική λειτουργία που προσφέρει η γλώσσα XQuery είναι οι εκφράσεις FLWR, που περιέχουν επαναληπτικό βρόχο

(For), έκφραση ανάθεσης (Let), έκφραση συνθήκης (Where) και έκφραση επιστροφής – κατασκευής του αποτελέσματος (Return). Στα Σχήματα 2.3 και 2.4 παρουσιάζονται δύο μοντέλα δεδομένων από τα items.xml και bids.xml αρχεία αντίστοιχα. Στα σχήματα, οι κόμβοι με τις επιγραφές D, E, A και T αντιπροσωπεύουν έγγραφο (Document), στοιχεία (Element), χαρακτηριστικά (Attributes) και κόμβους κειμένων (Text nodes) αντίστοιχα. Ένα απλό ερώτημα σε XQuery θα μπορούσε να επιλέγει από το έγγραφο items.xml όλα τα αντικείμενα που πωλούνται από κάποιο άτομο με όνομα Koukoutsaki:

```
document ("items.xml")/* / item[seller = "Koukoutsaki"] / description
```



Σχήμα 2.3



Σχήμα 2.4

Η XQuery εκτός από τη δυνατότητα επιλογής κόμβων και αντικειμένων από ένα μοντέλο δεδομένων, προσφέρει και τη δυνατότητα κατασκευής νέων κόμβων. Για παράδειγμα με το παρακάτω ερώτημα κατασκευάζεται ένα νέο αντικείμενο με όνομα highbid που έχει τη μεγαλύτερη προσφορά. Οι τιμές τις ιδιότητας status και των κόμβων – κειμένου itemno

και `bid-amount` δεν είναι σταθερές, αλλά προκύπτουν από το αρχικό μοντέλο του Σχήματος 2.3 μέσω των μεταβλητών `$s`, `$i` που αντιστοιχούν στην ιδιότητα `status` και στο αντικείμενο `item` αντίστοιχα:

```
<highbid status = "{$s}">
  <itemno> {$i} </itemno>
  <bid-amount>{ max($bids[itemno=$i] / bid=amount)} </bid-amount>
</highbid>
```

Αν θέλουμε για κάθε αντικείμενο του Σχήματος 2.3 που περιέχει πάνω από δέκα προσφορές που παρουσιάζονται στο Σχήμα 2.4 να κατασκευάσουμε ένα νέο αντικείμενο με όνομα `popular-item` που περιέχει τον αριθμό του (`itemno`), την περιγραφή (`description`) και το πλήθος των προσφορών, πρέπει να συντάξουμε την παρακάτω ερώτηση:

```
for $i in document ("items.xml") /*/ item
let $b := document ("bids.xml") /*/ bid[itemno = #i / itemno]
where count ($b) > 10
return
  <popular-item>
    (
      $i / itemno
      $i / description
      <bid-count> (count ($b) ) </bid-count>
    )
  </popular-item>
```

Στο παραπάνω ερώτημα ορίζουμε ως μεταβλητή για κάθε αντικείμενο του `items.xml` τη `$i` και μεταβλητή κάθε προσφοράς από το `bids.xml` το `$b`. Έτσι οι εκφράσεις `$i/itemno` και `$i/description` δίνουν τον αριθμό και την περιγραφή κάθε αντικειμένου. Η συνάρτηση `count` μετράει το πλήθος των προσφορών `$b` κάθε αντικειμένου. Τέλος η έκφραση `where` εξασφαλίζει ότι το αντικείμενο `popular-item` κατασκευάζεται μόνο όταν οι προσφορές για το `$i` είναι τουλάχιστον 10.

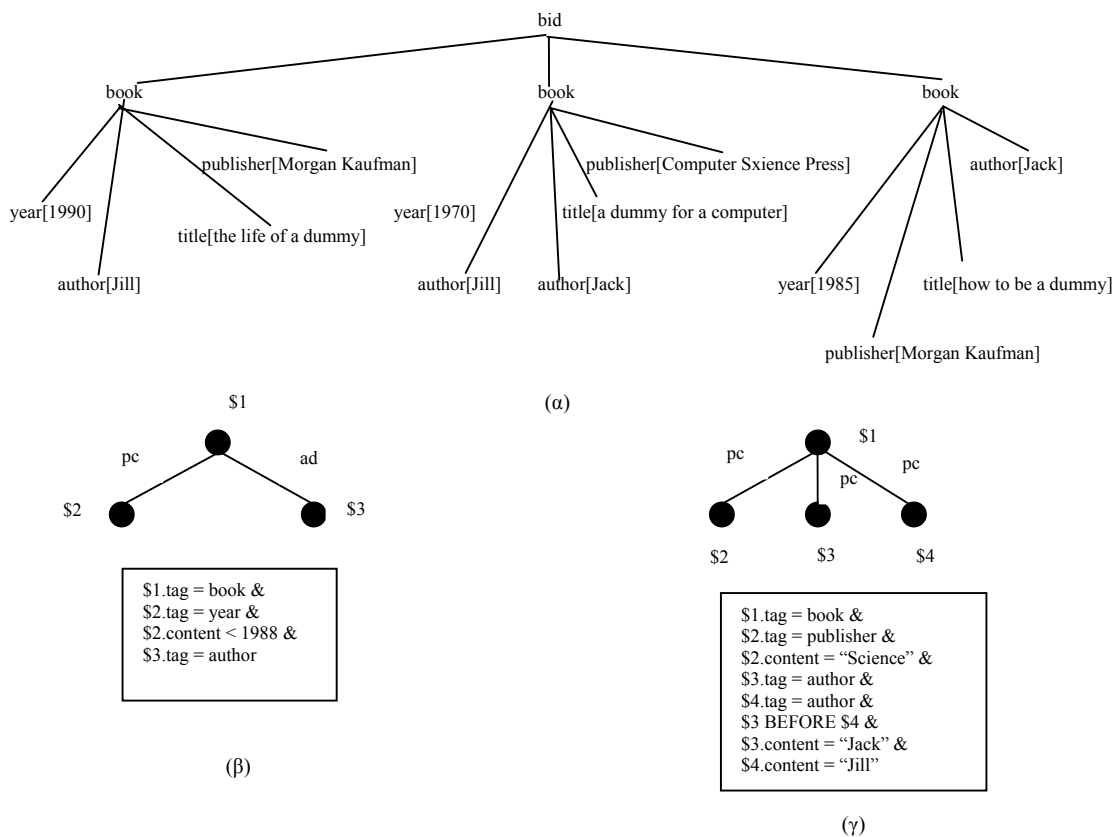
### 2.2.3 Άλγεβρες ημιδομημένων δεδομένων

Για τη διαχείριση ημιδομημένων δεδομένων εκτός από τις γλώσσες ερωτήσεων έχουν προταθεί και σχετικές άλγεβρες. Ο στόχος τους είναι να αποτελέσουν μία κοινή βάση για τη μετατροπή όλων των γλωσσών ερωτήσεων XML δεδομένων και τη βελτιστοποίηση εκτέλεσής τους. Μία λογική άλγεβρα που έχει προταθεί ονομάζεται SAL [23] και χρησιμοποιεί τελεστές που εφαρμόζονται σε λίστες.

Έχει προταθεί επίσης η άλγεβρα TAX [24] η οποία διαφέρει από άλλες άλγεβρες καθώς εφαρμόζεται σε σύνολα από διατεταγμένα δέντρα με ετικέτες, αντιμετωπίζοντάς τα ως

οντότητες πρώτης τάξης και όχι ως σύνολα από μεμονωμένους κόμβους. Σε αναλογία με τη σχεσιακή άλγεβρα, προτείνει τελεστές επιλογής, προβολής, ένωσης, τομής, διαφοράς, καρτεσιανού γινομένου, συνάθροισης κλπ, καθώς και δυνατότητες αναδόμησης δέντρων. Η TAX δίνει έμφαση σε μηχανισμούς επιλογής και αναδόμησης κομματιών δεντρικών δομών που αναπαριστούν XML κείμενα.

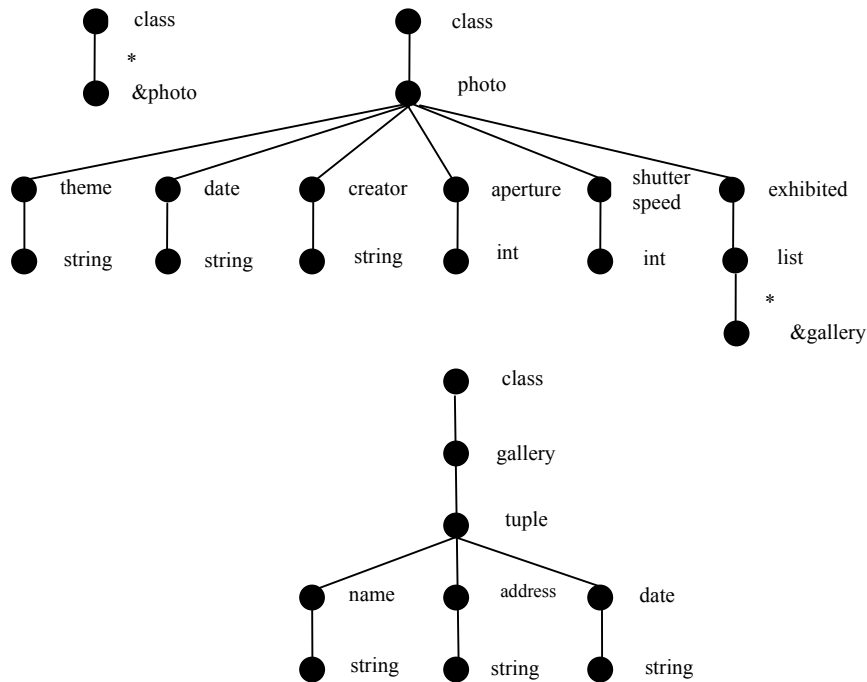
Στο Σχήμα 2.5(α) βλέπουμε ένα παράδειγμα XML βάσης δεδομένων και στο (β) μία ερώτηση που βρίσκει τα βιβλία που εκδόθηκαν πριν το 1988 και έχουν έναν τουλάχιστον συγγραφέα. Στο (γ) παρουσιάζεται η ερώτηση που δίνει τα βιβλία που εκδόθηκαν από εκδοτικό οίκο με όνομα που περιέχει το “Science”:



Σχήμα 2.5

Μία διαφορετική αντιμετώπιση ακολουθείται στην άλγεβρα YAT [25,26]. Αρχικά η YAT προτάθηκε ως μηχανισμός αποτίμησης ερωτήσεων σε συστήματα ενοποίησης δεδομένων. Η YAT XML άλγεβρα βασίζεται σε κλασικούς αντικειμενοστρεφείς τελεστές που διαχειρίζονται πλειάδες και επιπλέον παρέχει δύο τελεστές που εφαρμόζονται σε δεντρικές δομές και είτε τις αναδομούν είτε τις εκφυλλίζουν. Η ιδέα πίσω από την άλγεβρα YAT είναι ο διαχωρισμός της επεξεργασίας των δεντρικών δομών από τις κλασικές σχεσιακές δομές. Το μοντέλο της YAT είναι ένα γενικό σύστημα τύπων που αναπαριστά σχήματα αλλά και

δεδομένα, και άρα είναι κατάλληλο για την αναπαράσταση ημιδομημένων δεδομένων. Το Σχήμα 2.6 παρουσιάζει ένα παράδειγμα αναπαράστασης YAT για ένα σύνολο φωτογραφιών σε διάφορους μουσειακούς χώρους. Οι φωτογραφίες μοντελοποιούνται ως οντότητες κάτω από το μονοπάτι class/photo, έχοντας τα item, date, creator, aperture, shutter speed ως απλά χαρακτηριστικά και το exhibited ως αναφορά σε ένα σύνολο από μουσειακούς χώρους galleries.



Σχήμα 2.6

Η άλγεβρα YAT χρησιμοποιεί κλασσικούς αντικειμενοστρεφείς τελεστές σε εκφυλισμένες δομές που προκύπτουν από τον τελεστή Δέσιμο. Έπειτα η εκφυλισμένες δομές αναδομούνται με τον τελεστή Δέντρο.

1) Ο τελεστής Δέσιμο: βρίσκει δεδομένα σύμφωνα με ένα πρότυπο δέντρου με μεταβλητές και επιστρέφει μία δομή που ονομάζεται TAB, και έχει τη μορφή σχέσης που έχει ως ιδιότητες τις μεταβλητές που έχουν οριστεί στο πρότυπο δέντρου.

2) Ο τελεστής Δέντρο: αναδομεί τη δομή TAB που έχει προκύψει από τον τελεστή Δέσιμο και κατασκευάζει ξανά μία δεντρική δομή, σύμφωνα με κάποιο πρότυπο δέντρου ως είσοδο.



## 2.3 Στόχος

Στόχος αυτής της διπλωματικής εργασίας είναι η ανάπτυξη ενός συστήματος για τη διαχείριση μονοπατιών σε ιεραρχίες συναφών καταλόγων. Η γλώσσα του συστήματος στοχεύει στο χειρισμό πολλών καταλόγων που αν και αποθηκεύουν ίδιου τύπου αντικείμενα, έχουν διαφορετική οργάνωση της ιεραρχίας τους. Οι δυνατές πράξεις σε δενδρικές ιεραρχίες είναι: επιλογή (selection), προβολή (projection), καρτεσιανό γινόμενο (cartesian product), ένωση (union), τομή (intersection) και διαφορά (difference). Οι πράξεις βοηθούν στον προσδιορισμό μονοπατιών που αν και διαφορετικά, οδηγούν σε ίδια προϊόντα, και δίνουν στο χρήστη τη δυνατότητα να επιλέξει τα μονοπάτια που επιθυμεί ώστε να φέρει την ιεραρχία στην επιθυμητή για αυτόν μορφή. Η διπλωματική εργασία παρέχει στο χρήστη ένα εύχρηστο σύστημα με το οποίο μπορεί να εισάγει την επιθυμητή πράξη, να δει σε γραφική μορφή τα αποτελέσματα, ενώ συγχρόνως έχει τη δυνατότητα να επιτελέσει αρκετές ακόμη λειτουργίες πάνω στις ιεραρχίες.



# 3

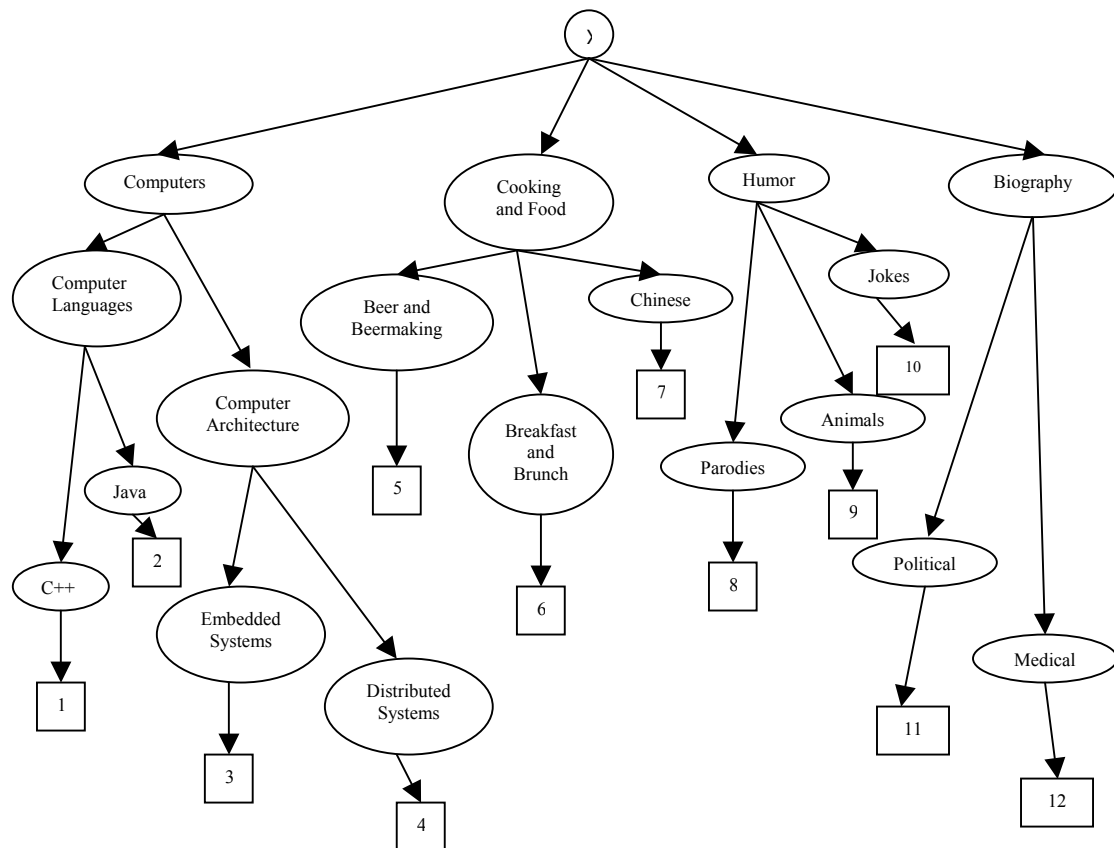
## ***Θεωρητική Μελέτη***

Στο κεφάλαιο αυτό δίνονται εν συντομία ορισμένοι χρήσιμοι ορισμοί των βασικών δομών της γλώσσας ερωτήσεων του συστήματος καθώς και του μοντέλου  $R_p - R_i$  που χρησιμοποιείται για την αναπαράσταση των δομών αυτών. Για περισσότερες λεπτομέρειες ο αναγνώστης παραπέμπεται στην αναφορά [28]. Επίσης ορίζονται οι βασικοί τελεστές των πράξεων της γλώσσας: επιλογή (selection), προβολή (projection), καρτεσιανό γινόμενο (cartesian product), ένωση (union), τομή (intersection) και διαφορά (difference).

### ***3.1 Δικτυακός κατάλογος – Μονοπάτι - Tree Structure Relation (TSR)***

Τα περιεχόμενα ενός δικτυακού καταλόγου οργανώνονται σε μία ιεραρχία με μορφή δέντρου που ονομάζεται *σχήμα καταλόγου* (catalog schema). Το σχήμα ενός καταλόγου αποτελείται από μία μοναδική *ρίζα* (root), από *κόμβους* με ετικέτες που αντιστοιχούν σε θεματικές κατηγορίες και από *φύλλα* (ή κόμβους – αντικείμενα) που περιέχουν εγγραφές των περιεχόμενων αντικειμένων. Κάθε κόμβος – αντικείμενο έχει μία σειρά από ιδιότητες. Στο παρακάτω Σχήμα (3.1) φαίνεται ένα μέρος του σχήματος του δικτυακού καταλόγου `powells`

(www.powells.com) ο οποίος περιέχει βιβλία προς πώληση. Οι κόμβοι – φύλλα αυτού του σχήματος παριστάνουν τους κόμβους – αντικείμενα του σχήματος καταλόγου. Οι κόμβοι – αντικείμενα περιέχουν κάθε φορά κατάλληλες εγγραφές. Για παράδειγμα, ο κόμβος – αντικείμενο 1 περιέχει εγγραφές για βιβλία της γλώσσας προγραμματισμού C++ με ιδιότητες Title, Author και Price για τον τίτλο, τον συγγραφέα και την τιμή αντίστοιχα. Ο κόμβος αυτός έχει τη μορφή που φαίνεται στο Σχήμα 3.2. Σημειώνεται ότι για απλούστευση, στο σχήμα δεν εισάγονται οι εγγραφές του κόμβου – αντικειμένου.



**Σχήμα 3.1**

C++ Books		
Title	Author	Price

**Σχήμα 3.2**

Ακολουθούν οι ορισμοί του μονοπατιού και του *TSR* σχήματος.

**Ορισμός 3.1.** *Μονοπάτι του σχήματος καταλόγου είναι μια ακολουθία κόμβων  $v_0$  (ρίζα),  $v_1$ ,  $v_2$ , ...,  $v_{k-1}$  με  $k \in \mathbb{N}$  που οδηγούν μέσω ακμών στον κόμβο – αντικείμενο  $v_k$ .*

Για παράδειγμα, στο σχήμα καταλόγου του Σχήματος 3.1 για να προσπελάσουμε το αντικείμενο Distributed Systems που έχει τις εγγραφές των βιβλίων για τα καταναμεμημένα συστήματα υπολογιστών πρέπει να ακολουθήσουμε το μονοπάτι /Computers/Computer Architecture/Distributed Systems.

Μονοπάτια από πολλά σχήματα καταλόγων που οδηγούν στον ίδιο κόμβο – αντικείμενο σχηματίζουν ένα δέντρο το οποίο ονομάζεται TSR σχήμα. Ένα TSR σχήμα κρατά εναλλακτικά μονοπάτια που οδηγούν στον ίδιο κόμβο – αντικείμενο.

**Ορισμός 3.2.** *Έστω  $S = \{CS_1, CS_2, \dots, CS_n\}$  ένα σύνολο σχημάτων καταλόγων και ένας κόμβος – αντικείμενο που υπάρχει σε όλα τα σχήματα. Ένα σχήμα TSR είναι ένας AND - OR γράφος που αποτελείται από μία ρίζα, τον κοινό κόμβο – αντικείμενο και συνδυασμούς μονοπατιών από τα σχήματα του  $S$  που καταλήγουν σε αυτόν.*

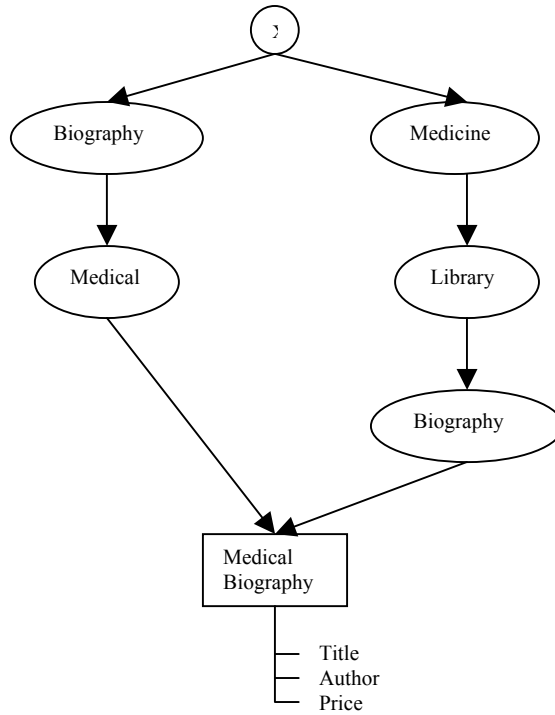
Σε ένα σχήμα TSR:

1. *OR ονομάζονται τα μονοπάτια που αντιστοιχούν στην ύπαρξη περισσότερων του ενός δρόμων προσπέλασης του κόμβου αντικειμένου.*
2. *Τα AND μονοπάτια ορίζουν μια σύνθετη έννοια κατά αντιστοιχία με αυτή που προκύπτει από την εφαρμογή καρτεσιανού γινομένου σε σχέσεις στη σχεσιακή άλγεβρα.*
3. *OR συνιστώσα ενός TSR σχήματος ονομάζεται ο συνδυασμός των γειτονικών AND και των ανεξάρτητων OR μονοπατιών.*

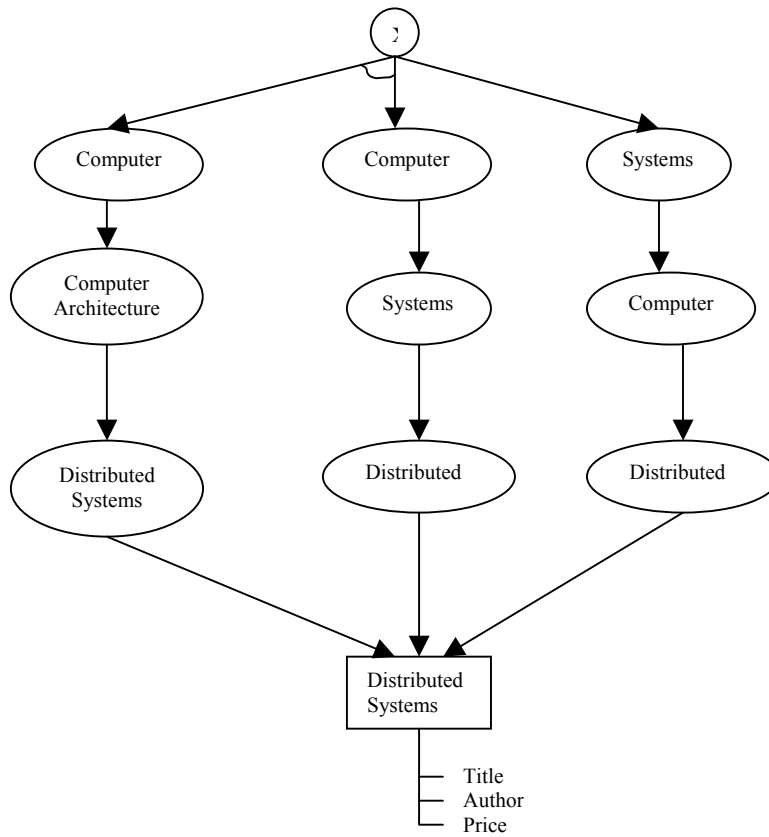
Στο Σχήμα 3.3 απεικονίζεται μία εκφυλισμένη μορφή ενός TSR σχήματος, το οποίο αποτελείται από δύο OR μονοπάτια. Σε αυτή την περίπτωση τα δύο ανεξάρτητα OR μονοπάτια (/Biography/Medical και /Medicine/Library/Biography) σχηματίζουν τις δύο OR συνιστώσες του σχήματος. Αυτές οι δύο συνιστώσες δείχνουν τους διαφορετικούς τρόπους προσπέλασης του κόμβου – αντικειμένου.

Στο Σχήμα 3.4 φαίνεται ένα TSR σχήμα τα μονοπάτια του οποίου οδηγούν σε εγγραφές για βιβλία καταναμεμημένων συστημάτων. Ένα TSR σχήμα με AND μονοπάτια προκύπτει από συνδυασμό μονοπατιών πολλών σχημάτων καταλόγων το οποία οδηγούν στον ίδιο κόμβο – αντικείμενο. Όπως φαίνεται στο σχήμα τα γειτονικά AND μονοπάτια /Computer/Computer Architecture/Distributed Systems και /Computer/Systems/Distributed σχηματίζουν την πρώτη OR συνιστώσα της TSR, ενώ το μονοπάτι

/Systems/Computer/Distributed σχηματίζει τη δεύτερη. Σημειώνεται ότι τα AND μονοπάτια παριστάνονται ενώνοντας τις ακμές τους που ξεκινούν από τη ρίζα με ένα τόξο.



**Σχήμα 3.3**



**Σχήμα 3.4**

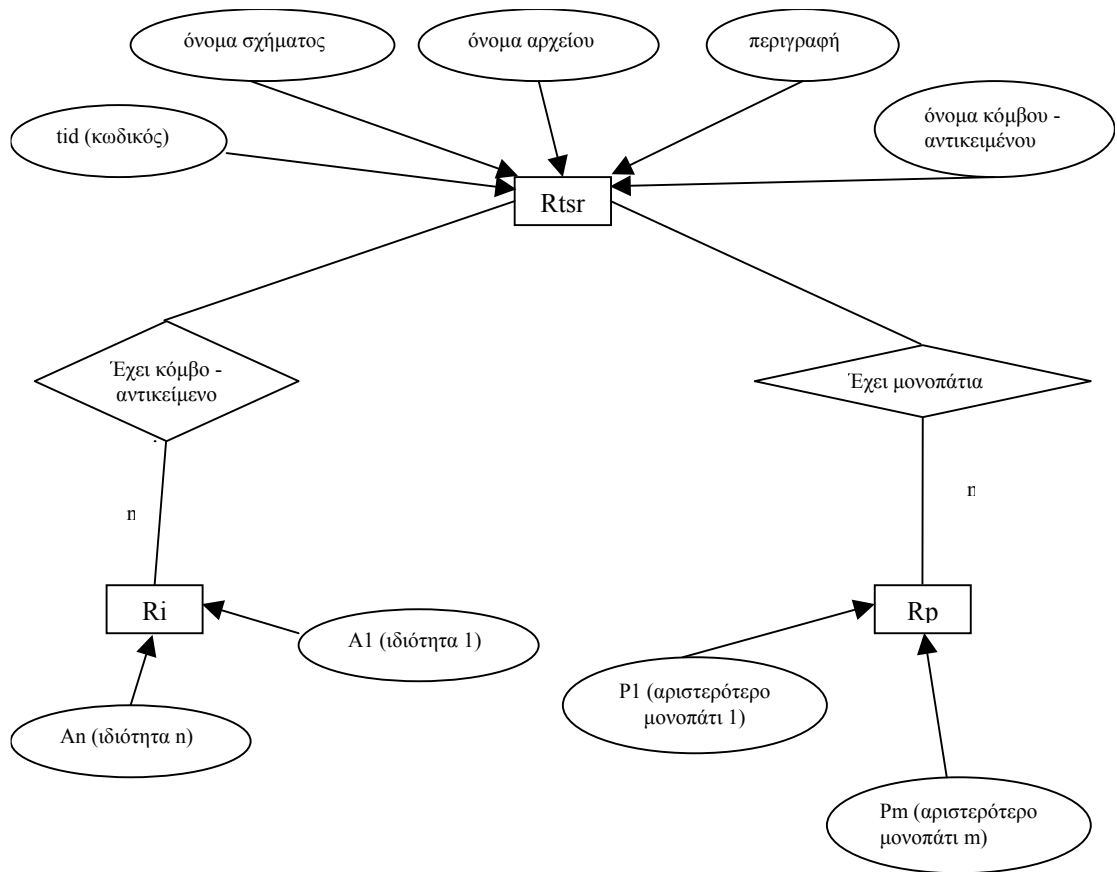
## 3.2 $R_p - R_i$ μοντέλο

### 3.2.1 Περιγραφή του μοντέλου

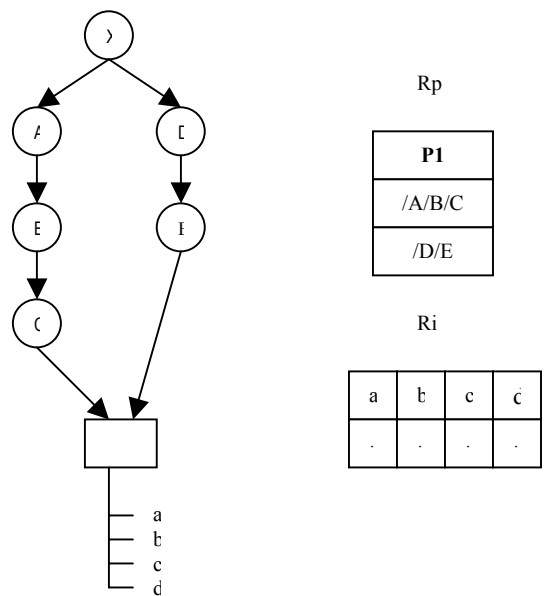
Για την αποθήκευση και την προσπέλαση ενός TSR σχήματος είναι απαραίτητο ένα μοντέλο το οποίο θα ακολουθείται κάθε φορά που πρόκειται να αποθηκευθεί ή να προσπελαστεί ένα σχήμα. Το μοντέλο  $R_p - R_i$  αναπαριστά ένα TSR σχήμα με σχέσεις της σχεσιακής άλγεβρας. Έτσι, λοιπόν, για κάθε κόμβο – αντικείμενο ορίζουμε τη σχέση  $R_i$ , η οποία περιέχει τις ιδιότητες και τις εγγραφές του, και τη σχέση  $R_p$ , η οποία περιέχει τα μονοπάτια που οδηγούν σε αυτόν. Επιπλέον ορίζεται και η σχέση  $R_{tsr}$  η οποία περιέχει πληροφορίες για το TSR σχήμα. Συγκεκριμένα για κάθε TSR σχήμα φυλάσσονται το όνομά του, το όνομα του σχήματος καταλόγου από το οποίο προήλθε, μία σύντομη περιγραφή του καθώς επίσης και το όνομα του κόμβου – αντικειμένου. Από το διάγραμμα του μοντέλου οντοτήτων συσχετίσεων (Σχήμα 3.5) παρατηρούμε ότι η οντότητα  $R_p$  έχει ως γνωρίσματα τις μεταβλητές μονοπατιών (*path variables*) που αντιστοιχούν στα AND μονοπάτια κάθε OR συνιστώσας του TSR σχήματος, ενώ στην οντότητα  $R_i$  υπάρχουν γνωρίσματα τα οποία αντιστοιχούν στις ιδιότητες (*attributes*) του κόμβου αντικειμένου. Όπως θα εξηγηθεί αναλυτικά στη συνέχεια, μία μεταβλητή μονοπατιών (*path variable*) αναφέρεται σε κάποιο AND μονοπάτι (πρώτο, δεύτερο,...) κάθε OR συνιστώσας ενός TSR σχήματος ξεκινώντας τη μέτρηση από αριστερά.

### 3.2.2 Κατασκευή – Μετάβαση από TSR σχήμα στο μοντέλο $R_p - R_i$

Στη συνέχεια δίνονται δύο παραδείγματα μετάβασης από ένα TSR σχήμα στο μοντέλο  $R_p - R_i$ . Στο Σχήμα 3.6 παρουσιάζονται οι σχέσεις  $R_p$  και  $R_i$  για ένα TSR με δύο OR συνιστώσες. Για λόγους απλότητας, τα ονόματα των κόμβων και των ιδιοτήτων του κόμβου – αντικειμένου έχουν αντικατασταθεί με κεφαλαία και μικρά αναγνωριστικά, ενώ οι εγγραφές του κόμβου – αντικειμένου δεν εισάγονται στη σχέση  $R_i$ . Από το σχήμα παρατηρούμε ότι κάθε OR συνιστώσα αποτελείται από ένα μόνο μονοπάτι. Άρα υπάρχει μόνο μία μεταβλητή μονοπατιών, η P1.



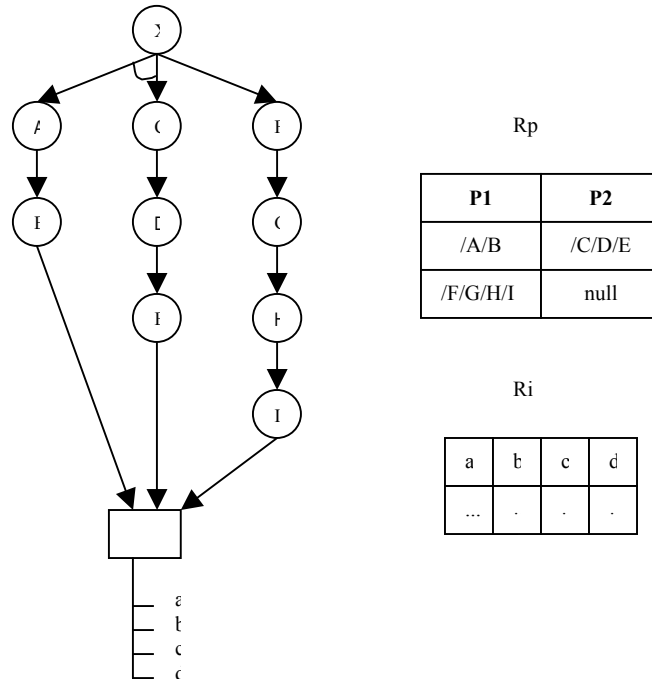
Σχήμα 3.5



Σχήμα 3.6



Στο Σχήμα 3.7 υπάρχουν δύο OR συνιστώσες στο σχήμα της TSR με αποτέλεσμα τη δημιουργία δύο πλειάδων στη σχέση  $R_p$ . Η πρώτη συνιστώσα αποτελείται από τα AND μονοπάτια  $/A/B$  και  $/C/D/E$ , ενώ η δεύτερη από το  $/F/G/H/I$ . Επομένως υπάρχουν δύο μεταβλητές μονοπατιών, η P1 και η P2. Επειδή η δεύτερη συνιστώσα έχει μόνο ένα AND μονοπάτι, η πλειάδα της στην  $R_p$  σχέση έχει στην δεύτερη μεταβλητή μονοπατιού τιμή null.



Σχήμα 3.7

### 3.3 Σύγκριση μονοπατιών

Στο κεφάλαιο αυτό ορίζουμε το σύνολο των τελεστών που καθορίζουν τη σχέση μεταξύ δύο μονοπατιών.

#### 3.3.1 Ισότητα – Ανισότητα

**Ορισμός 3.3.** Για δύο μονοπάτια  $P_1$  και  $P_2$  ισχύει  $P_1 = P_2$  αν και μόνο αν αυτά αποτελούνται από τους ίδιους κόμβους με την ίδια σειρά προηγούμενου - επόμενου.

Παράδειγμα:  $/A/B = /A/B$

**Ορισμός 3.4.** Για δύο μονοπάτια  $P_1$  και  $P_2$  ισχύει  $P_1 \neq P_2$  αν και μόνο αν οι κόμβοι τους δεν είναι ίδιοι ή δεν είναι με την ίδια σειρά προηγούμενου – επόμενου.

Παράδειγμα:  $/A/B/C \neq /A/C/B$   
 $/A/B \neq /C/D$

### 3.3.2 Αυστηρό υποσύνολο – Αυστηρό υπερσύνολο

**Ορισμός 3.5.** Για δύο μονοπάτια  $P_1$  και  $P_2$  ισχύει  $P_1 \mathbf{N} P_2$  ( $P_1$  αυστηρό υποσύνολο του  $P_2$ ) αν και μόνο αν όλοι οι κόμβοι του πρώτου μονοπατιού υπάρχουν στο δεύτερο με την ίδια ακριβώς σειρά προηγούμενου – επόμενου, χωρίς να παρεμβάλλονται ενδιάμεσα άλλοι κόμβοι.

Παράδειγμα:  $/A/B \mathbf{N} /C/A/B$

**Ορισμός 3.6.** Για δύο μονοπάτια  $P_1$  και  $P_2$  ισχύει  $P_1 \mathbf{K} P_2$  ( $P_1$  αυστηρό υπερσύνολο του  $P_2$ ) αν και μόνο αν όλοι οι κόμβοι του δεύτερου μονοπατιού υπάρχουν στο πρώτο με την ίδια ακριβώς σειρά προηγούμενου – επόμενου, χωρίς να παρεμβάλλονται ενδιάμεσα άλλοι κόμβοι.

Παράδειγμα:  $/A/B/C/D \mathbf{K} /B/C$

### 3.3.3 Χαλαρό υποσύνολο – Χαλαρό υπερσύνολο

**Ορισμός 3.7.** Για δύο μονοπάτια  $P_1$  και  $P_2$  ισχύει  $P_1 \mathbf{M} P_2$  ( $P_1$  χαλαρό υποσύνολο του  $P_2$ ) αν και μόνο αν όλοι οι κόμβοι του πρώτου μονοπατιού περιέχονται στο δεύτερο με την ίδια σειρά προηγούμενου – επόμενου, ανεξάρτητα από το αν παρεμβάλλονται ενδιάμεσα και άλλοι κόμβοι.

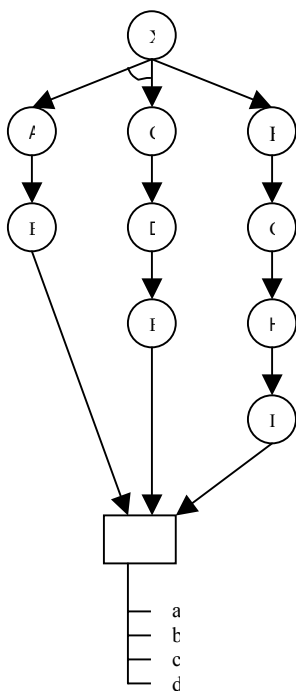
Παράδειγμα:  $/A/B \mathbf{M} /A/C/B$

**Ορισμός 3.8.** Για δύο μονοπάτια  $P_1$  και  $P_2$  ισχύει  $P_1 \mathbf{I} P_2$  ( $P_1$  χαλαρό υπερσύνολο του  $P_2$ ) αν και μόνο αν όλοι οι κόμβοι του δεύτερου μονοπατιού περιέχονται στο πρώτο με την ίδια σειρά προηγούμενου – επόμενου, ανεξάρτητα από το αν παρεμβάλλονται ενδιάμεσα και άλλοι κόμβοι.

Παράδειγμα:  $/A/B/C/D \mathbf{I} /A/D$

### 3.4 Ερωτήσεις – Πράξεις

Σε αυτό το κεφάλαιο ορίζεται το σύνολο των πράξεων που μπορούν να εφαρμοστούν σε TSR σχήματα. Μέρος του κεφαλαίου έχει παρουσιαστεί και στην αναφορά [29]. Σημειώνεται ότι το σύνολο των τελεστών αυτών των πράξεων πρέπει να ικανοποιεί την απαίτηση της κλειστότητας, δηλαδή σε κάθε πράξη είσοδος να είναι ένα ή περισσότερα TSR σχήματα και το αποτέλεσμα να είναι επίσης ένα TSR σχήμα. Οι επιτρεπτές πράξεις χωρίζονται σε δύο υποσύνολα. Το πρώτο περιλαμβάνει τις μοναδιαίες πράξεις, δηλαδή τις πράξεις οι οποίες δέχονται για είσοδο ένα μόνο TSR σχήμα. Αυτές είναι οι πράξεις της επιλογής (selection) και της προβολής (projection). Το δεύτερο υποσύνολο περιλαμβάνει τις δυαδικές πράξεις, οι οποίες δέχονται για είσοδο δύο TSR σχήματα, δηλαδή τις πράξεις του καρτεσιανού γινομένου (cartesian product), της ένωσης (union), της τομής (intersection) και της διαφοράς (difference).



Σχήμα 3.8

Πριν προχωρήσουμε στην αναλυτική περιγραφή των παραπάνω πράξεων είναι απαραίτητο να εξηγήσουμε τους όρους μεταβλητή μονοπατιών (path index variable) και OR μεταβλητή (OR index variable), τους οποίους θα χρησιμοποιήσουμε στη συνέχεια. Όπως έχει ήδη αναφερθεί, η μεγαλύτερη OR συνιστώσα ενός TSR σχήματος, δηλαδή αυτή με τα περισσότερα AND μονοπάτια, καθορίζει τον αριθμό των μεταβλητών μονοπατιών. Μία μεταβλητή μονοπατιών, η οποία συμβολίζεται με  $S_i$ , αναφέρεται στο  $i$ -οστό μονοπάτι κάθε

OR συνιστώσας ενός TSR σχήματος ξεκινώντας τη μέτρηση από αριστερά. Για παράδειγμα στο Σχήμα 3.8, η μεταβλητή μονοπατιού \$1 αντιστοιχεί στα /A/B και /F/G/H/I μονοπάτια ενώ η \$2 στο /C/D/E μονοπάτι αφού η δεύτερη OR συνιστώσα αποτελείται μόνο από ένα AND μονοπάτι. Επίσης με το σύμβολο \$\_ αναφερόμαστε σε οποιοδήποτε μονοπάτι κάθε OR συνιστώσας του TSR σχήματος. Μία OR μεταβλητή, η οποία συμβολίζεται με #j, αναφέρεται στην j-οστή OR συνιστώσα του TSR σχήματος ξεκινώντας τη μέτρηση από τα αριστερά. Στο Σχήμα 3.8 για παράδειγμα, το #2 αναφέρεται στη δεύτερη OR συνιστώσα του TSR σχήματος η οποία αποτελείται από το μονοπάτι /F/G/H/I.

### 3.4.1 Επιλογή (selection)

Σε αυτήν την πράξη απαιτούνται δύο λίστες συνθηκών – μία για τις συνθήκες των μονοπατιών και μία για τις συνθήκες των ιδιοτήτων. Επιλέγονται οι OR συνιστώσες που οδηγούν στον κόμβο – αντικείμενο του TSR σχήματος, τα μονοπάτια των οποίων ικανοποιούν τις συνθήκες μονοπατιών, αλλά και τα δεδομένα των εγγραφών που ικανοποιούν τις συνθήκες ιδιοτήτων. Πιο συγκεκριμένα έχουμε:

*Λίστα συνθηκών ιδιοτήτων:* ένωση ή τομή boolean συνθηκών που αναφέρονται στις ιδιότητες του κόμβου – αντικειμένου και έχουν τη μορφή ιδιότητα τελεστής τιμή ή ιδιότητα τελεστής ιδιότητα, όπου η ιδιότητα είναι μία από τις ιδιότητες του κόμβου – αντικειμένου και ο τελεστής είναι ένας από τους {=, ≠, <, >, ≤, ≥}.

*Λίστα συνθηκών μονοπατιών:* Ένωση ή τομή boolean συνθηκών που αναφέρονται στα μονοπάτια που οδηγούν στον κόμβο αντικείμενο και έχουν τη μορφή μεταβλητή μονοπατιών τελεστής τιμή ή μεταβλητή μονοπατιών τελεστής μεταβλητή μονοπατιών, όπου ο τελεστής είναι ένας από τους {=, ≠, M, I, N, K}.

Η σύνταξη της πράξης έχει ως εξής:

SELECT <Λίστα συνθηκών ιδιοτήτων> <Λίστα συνθηκών μονοπατιών> (TSR)

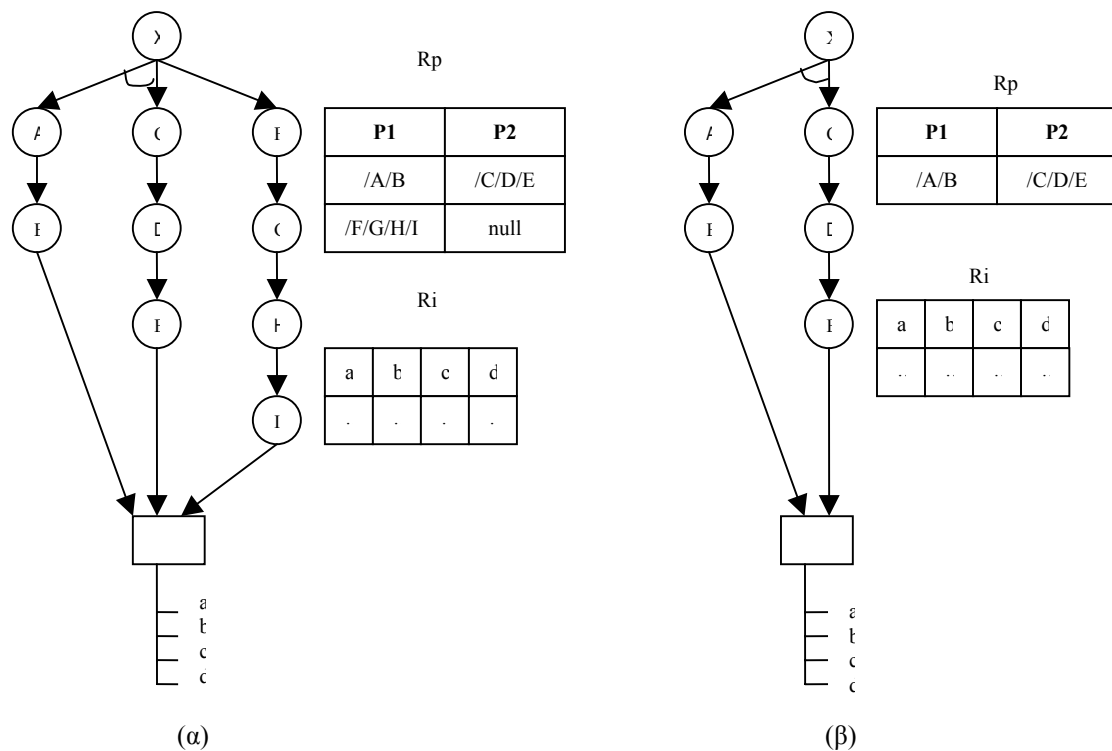
#### Παράδειγμα:

Θεωρούμε την TSR  $R_1$  του Σχήματος 3.9(α). Θέλουμε με χρήση της πράξης της επιλογής να κατασκευάσουμε μία νέα TSR της οποίας τα αριστερότερα μονοπάτια κάθε OR συνιστώσας θα είναι αυστηρά υπερσύνολα του /A/B, ενώ ο κόμβος – αντικείμενο θα έχει εγγραφές που

ικανοποιούν τη συνθήκη ιδιοτήτων  $a = c$ . Η σύνταξη της ερώτησης που θέλουμε να εκτελέσουμε είναι η εξής:

SELECT  $\langle a = c \rangle \langle \$1 K /A/B \rangle (R_1)$

Το αποτέλεσμα αυτής της ερώτησης φαίνεται στο Σχήμα 3.9(β). Σημειώνουμε ότι για ευκολία δεν εισάγονται οι εγγραφές στους πίνακες  $R_i$ .



Σχήμα 3.9

### 3.4.2 Προβολή (projection)

Με τη βοήθεια της πράξης της προβολής επιλέγονται μερικά από τα AND μονοπάτια κάθε OR συνιστώσας ή ολόκληρες OR συνιστώσες, καθώς και μερικές από τις ιδιότητες του κόμβου – αντικειμένου. Πιο συγκεκριμένα έχουμε:

*Λίστα ιδιοτήτων:* Μία λίστα από ιδιότητες του κόμβου - αντικειμένου του TSR σχήματος οι οποίες θα αποτελούν το σύνολο των ιδιοτήτων του κόμβου – αντικειμένου του TSR σχήματος που θα προκύψει.

*Λίστα μεταβλητών μονοπατιών ή OR μεταβλητών:* Μία λίστα από μεταβλητές μονοπατιών ή OR μεταβλητές οι οποίες καθορίζουν τα AND μονοπάτια και τις OR συνιστώσες του TSR σχήματος που θα προκύψει. Πρέπει να σημειωθεί ότι σε μία προβολή πρέπει να υπάρχει είτε λίστα μεταβλητών μονοπατιών, είτε λίστα OR μεταβλητών για να αποφεύγονται οι συγκρούσεις. Για παράδειγμα, έστω ένα TSR σχήμα με μία OR συνιστώσα η οποία αποτελείται από δύο AND μονοπάτια. Αν στην πράξη της προβολής υπάρχουν τα: \$1 (πρώτη μεταβλητή μονοπατιού) και #1 (πρώτη OR μεταβλητή) τότε υπάρχει σύγκρουση, αφού ζητάμε να προβάλουμε μόνο το αριστερότερο μονοπάτι της OR συνιστώσας αλλά και ολόκληρη την OR συνιστώσα.

Η σύνταξη της πράξης έχει ως εξής:

PROJECT <Λίστα ιδιοτήτων> <Λίστα μεταβλητών μονοπατιών ή OR μεταβλητών> (TSR)

#### **Παράδειγμα:**

Θεωρούμε την TSR  $R_1$  του Σχήματος 3.10(α). Θέλουμε με χρήση της πράξης της προβολής να κατασκευάσουμε μία νέα TSR η οποία θα αποτελείται από τα αριστερότερα μονοπάτια κάθε OR συνιστώσας της TSR εισόδου και ο κόμβος – αντικείμενο της θα έχει εγγραφές με ιδιότητες τις a και c. Η σύνταξη της ερώτησης που θέλουμε να εκτελέσουμε είναι η εξής:

PROJECT <a, c> <\$1> ( $R_1$ )

Το αποτέλεσμα αυτής της ερώτησης φαίνεται στο Σχήμα 3.10(β). Σημειώνουμε ότι για ευκολία δεν εισάγονται οι εγγραφές στους πίνακες  $R_i$ .

#### **3.4.3 Καρτεσιανό γινόμενο (cartesian product)**

Το καρτεσιανό γινόμενο είναι μία από τις δυαδικές πράξεις. Δέχεται δηλαδή σαν είσοδο δύο TSR σχήματα και παράγει μία νέα TSR. Τα μονοπάτια του αποτελέσματος είναι ο συνδυασμός κάθε OR συνιστώσας της πρώτης TSR με κάθε OR συνιστώσα της δεύτερης TSR, ενώ οι ιδιότητες του είναι το σύνολο των ιδιοτήτων των δύο TSRs εισόδου. Τέλος οι εγγραφές του αποτελέσματος είναι ο συνδυασμός κάθε εγγραφής του κόμβου – αντικειμένου της πρώτης TSR με κάθε εγγραφή του κόμβου – αντικειμένου της δεύτερης.

Η σύνταξη της πράξης έχει ως εξής:

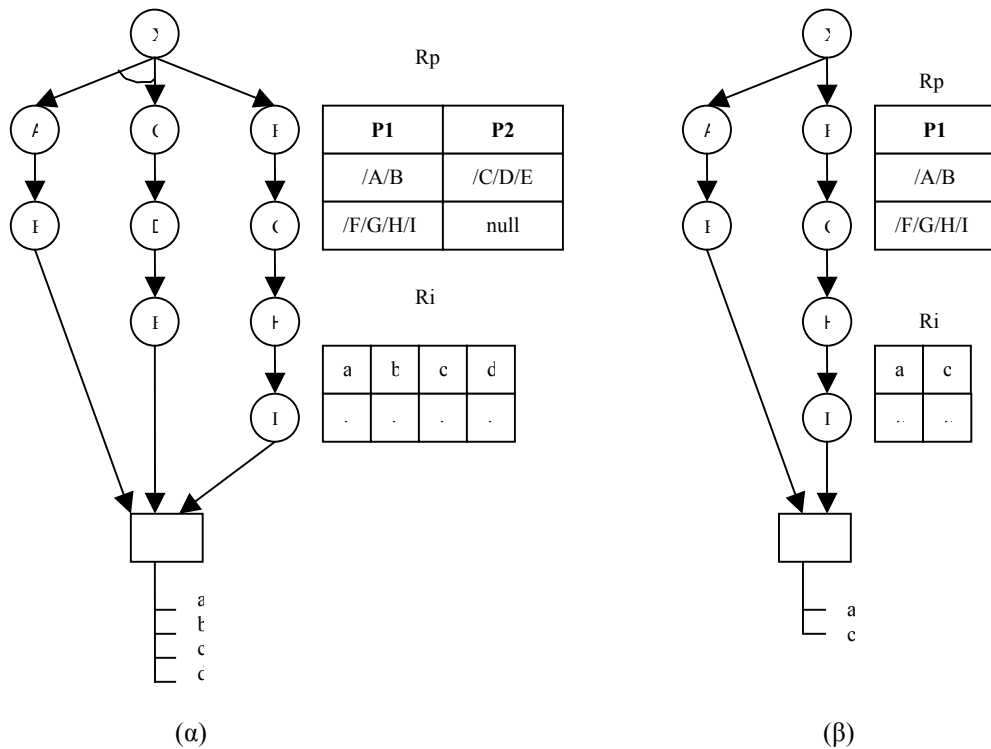
(TSR1) X (TSR2)

**Παράδειγμα:**

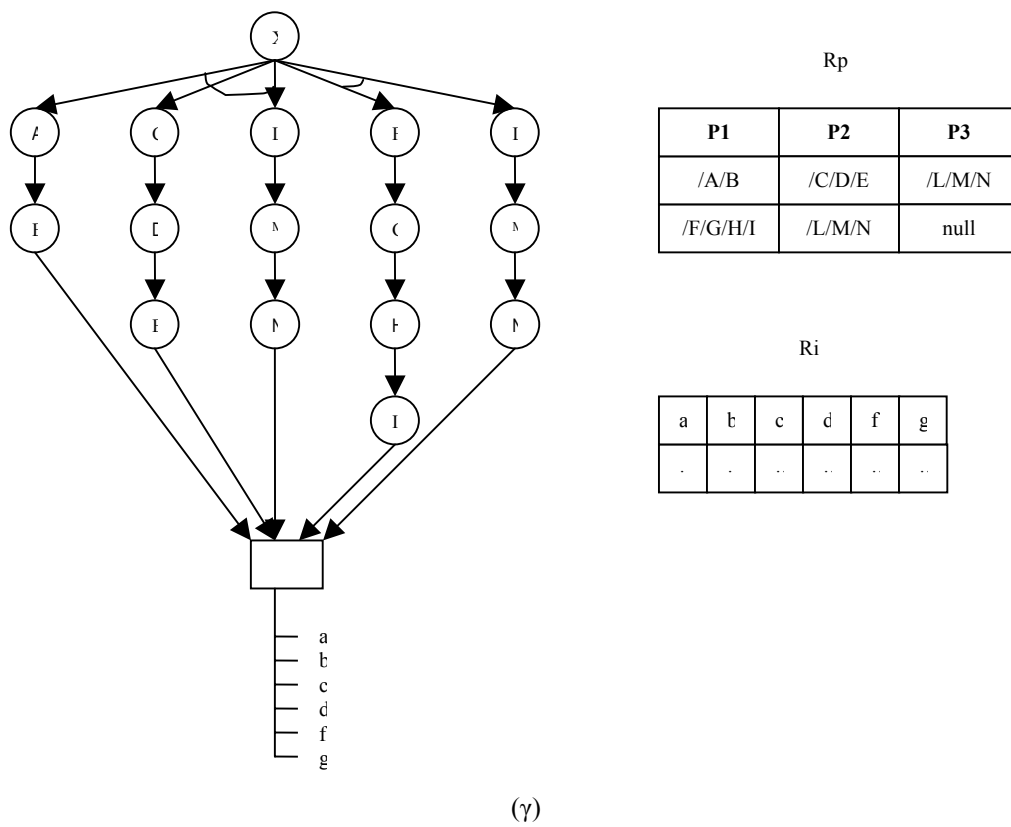
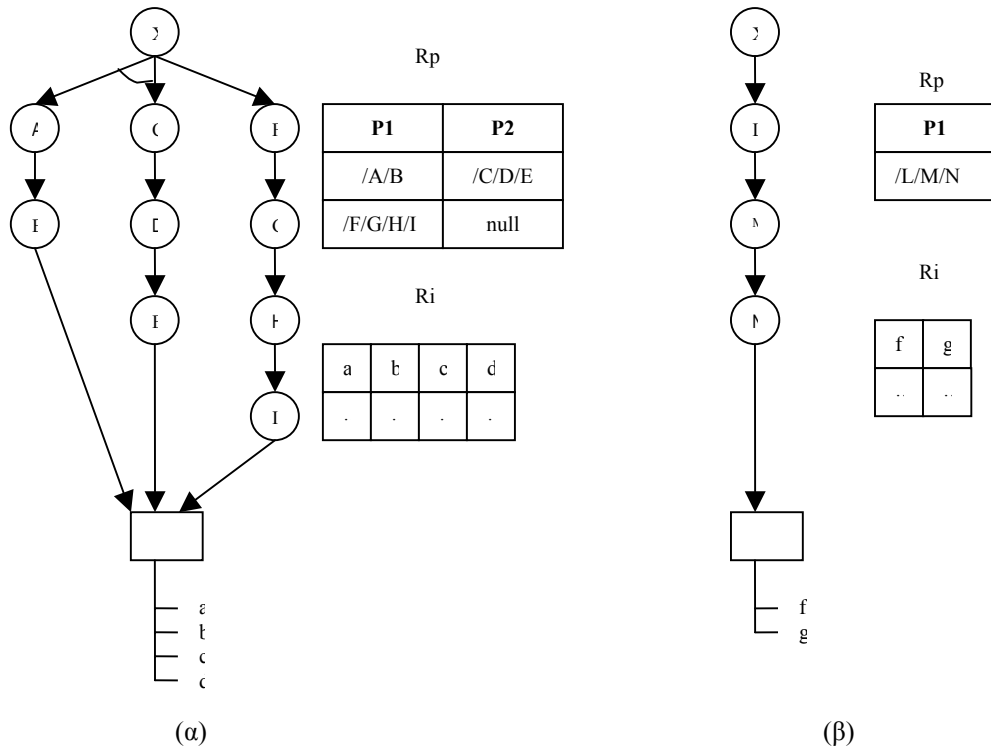
Θεωρούμε τα TSR σχήματα  $R_1$  και  $R_2$  των Σχημάτων 3.11(α) και 3.11(β). Κατασκευάζουμε μία νέα TSR με συνδυασμό των αρχικών χρησιμοποιώντας το καρτεσιανό γινόμενο. Ο συνδυασμός δημιουργεί νέες OR συνιστώσες συνδυάζοντας αυτές της πρώτης TSR με αυτές της δεύτερης, ενώ στον κόμβο – αντικείμενο συνδυάζονται οι εγγραφές των αρχικών σχημάτων με τρόπο ανάλογο με αυτόν που συνδυάζονται οι πλειάδες στο καρτεσιανό γινόμενο της σχεσιακής άλγεβρας. Η σύνταξη της ερώτησης που θέλουμε να εκτελέσουμε είναι η εξής:

$(R_1) X (R_2)$

Το αποτέλεσμα αυτής της ερώτησης φαίνεται στο Σχήμα 3.11(γ). Σημειώνουμε ότι για ευκολία δεν εισάγονται οι εγγραφές στους πίνακες  $R_i$ .



Σχήμα 3.10



Σχήμα 3.11



### 3.4.4 Ένωση (*union*)

Η πράξη της ένωσης γίνεται πάνω σε δύο TSR σχήματα των οποίων οι κόμβοι έχουν τις ίδιες ως προς το όνομα και τον τύπο ιδιότητες. Το TSR σχήμα που παράγεται αποτελείται από τις OR συνιστώσες και των δύο αρχικών TSR σχημάτων, ενώ οι εγγραφές του κόμβου – αντικειμένου του είναι η ένωση των εγγραφών των αρχικών κόμβων – αντικειμένων. Δηλαδή οι εγγραφές του κόμβου – αντικειμένου του αποτελέσματος υπάρχουν είτε στη μία είσοδο είτε στην άλλη. Σημειώνεται ότι αν μία εγγραφή υπάρχει και στους δύο κόμβους – αντικείμενα, τότε αυτή εμφανίζεται στο αποτέλεσμα μόνο μία φορά.

Η σύνταξη της πράξης έχει ως εξής:

$$(TSR1) \cup (TSR2)$$

Παράδειγμα:

Θεωρούμε τα TSR σχήματα  $R_1$  και  $R_2$  των Σχημάτων 3.12(α) και 3.12(β). Κατασκευάζουμε μία νέα TSR η οποία αποτελεί την ένωση των αρχικών. Η νέα TSR περιλαμβάνει τις OR συνιστώσες και των δύο αρχικών σχημάτων. Η σύνταξη της ερώτησης που θέλουμε να εκτελέσουμε είναι η εξής:

$$(R_1) \cup (R_2)$$

Το αποτέλεσμα αυτής της ερώτησης φαίνεται στο Σχήμα 3.12(γ). Σημειώνουμε ότι για ευκολία δεν εισάγονται οι εγγραφές στους πίνακες  $R_i$ .

### 3.4.5 Τομή (*intersection*)

Όμοια με την πράξη της ένωσης, η πράξη της τομής έχει ως είσοδο δύο TSR σχήματα με τις ίδιες ως προς το όνομα και τον τύπο ιδιότητες. Το TSR σχήμα που παράγεται αποτελείται από τις OR συνιστώσες και των δύο αρχικών TSR σχημάτων, ενώ οι εγγραφές του κόμβου – αντικειμένου του είναι η τομή των εγγραφών των κόμβων αντικειμένων των δύο αρχικών TSR σχημάτων. Δηλαδή, οι εγγραφές του κόμβου – αντικειμένου του αποτελέσματος είναι κοινές και στις δύο TSR εισόδου.

Η σύνταξη της πράξης έχει ως εξής:

$$(TSR1) \text{ H } (TSR2)$$

**Παράδειγμα:**

Εφαρμόζοντας την πράξη της τομής στις δύο TSR του προηγούμενου παραδείγματος (Σχήμα 3.12(α) και Σχήμα 3.12(β)) κατασκευάζουμε μία νέα TSR η οποία έχει ακριβώς τις ίδιες OR συνιστώσες με αυτές που είχε το αποτέλεσμα της ένωσης (Σχήμα 3.12(γ)). Όσον αφορά τις εγγραφές του κόμβου – αντικειμένου, αυτός περιέχει μόνο τις κοινές εγγραφές των κόμβων – αντικειμένων των TSR εισόδου.

**3.4.6 Διαφορά (difference)**

Όμοια με την ένωση και την τομή ορίζεται και η πράξη της διαφοράς, η οποία επενεργεί σε δύο TSR σχήματα με τις ίδιες ιδιότητες κόμβου – αντικειμένου ως προς το όνομα και τον τύπο. Το αποτέλεσμα είναι ένα TSR σχήμα το οποίο αποτελείται από τις OR συνιστώσες μόνο της πρώτης TSR εισόδου, ενώ περιέχει τις εγγραφές που υπάρχουν στον πρώτο κόμβο – αντικείμενο και δεν υπάρχουν στον δεύτερο.

Η σύνταξη της πράξης έχει ως εξής:

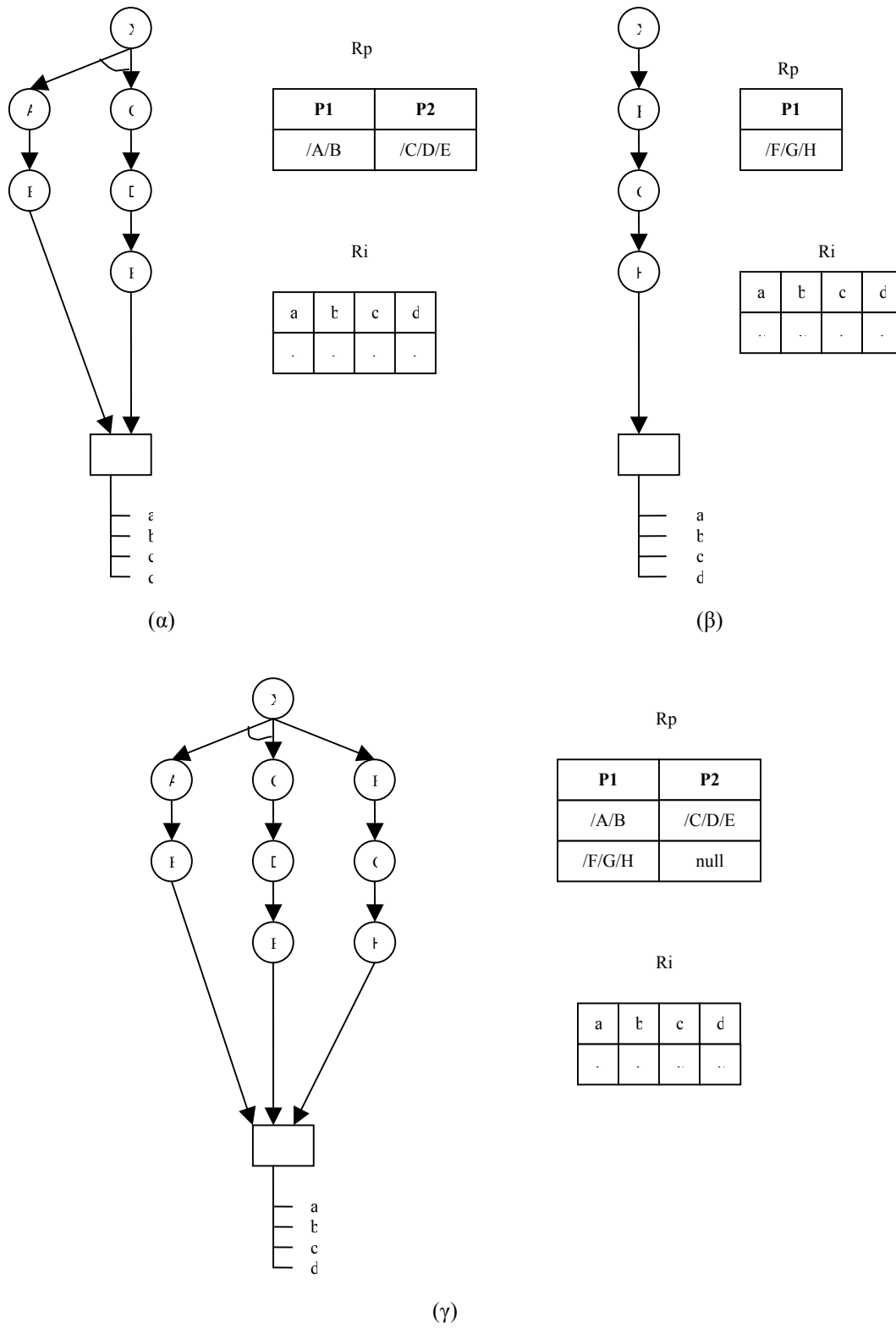
$$(TSR1) - (TSR2)$$

**Παράδειγμα:**

Θεωρούμε τα TSR σχήματα  $R_1$  και  $R_2$  των Σχημάτων 3.13(α) και 3.13(β). Κατασκευάζουμε μία νέα TSR η οποία αποτελεί την διαφορά των αρχικών. Η νέα TSR περιλαμβάνει τις OR συνιστώσες μόνο του πρώτου TSR σχήματος και τις εγγραφές του πρώτου κόμβου – αντικειμένου που δεν υπάρχουν στο δεύτερο TSR σχήμα. Η σύνταξη της ερώτησης που θέλουμε να εκτελέσουμε είναι η εξής:

$$(R_1) - (R_2)$$

Το αποτέλεσμα αυτής της ερώτησης φαίνεται στο Σχήμα 3.13(γ). Σημειώνουμε ότι για ευκολία δεν εισάγονται οι εγγραφές στους πίνακες  $R_i$ .



Rp

P1	P2
/A/B	/C/D/E

Ri

a	b	c	d
.	.	.	.

Rp

P1
/F/G/H

Ri

a	b	c	d
..	..	.	.

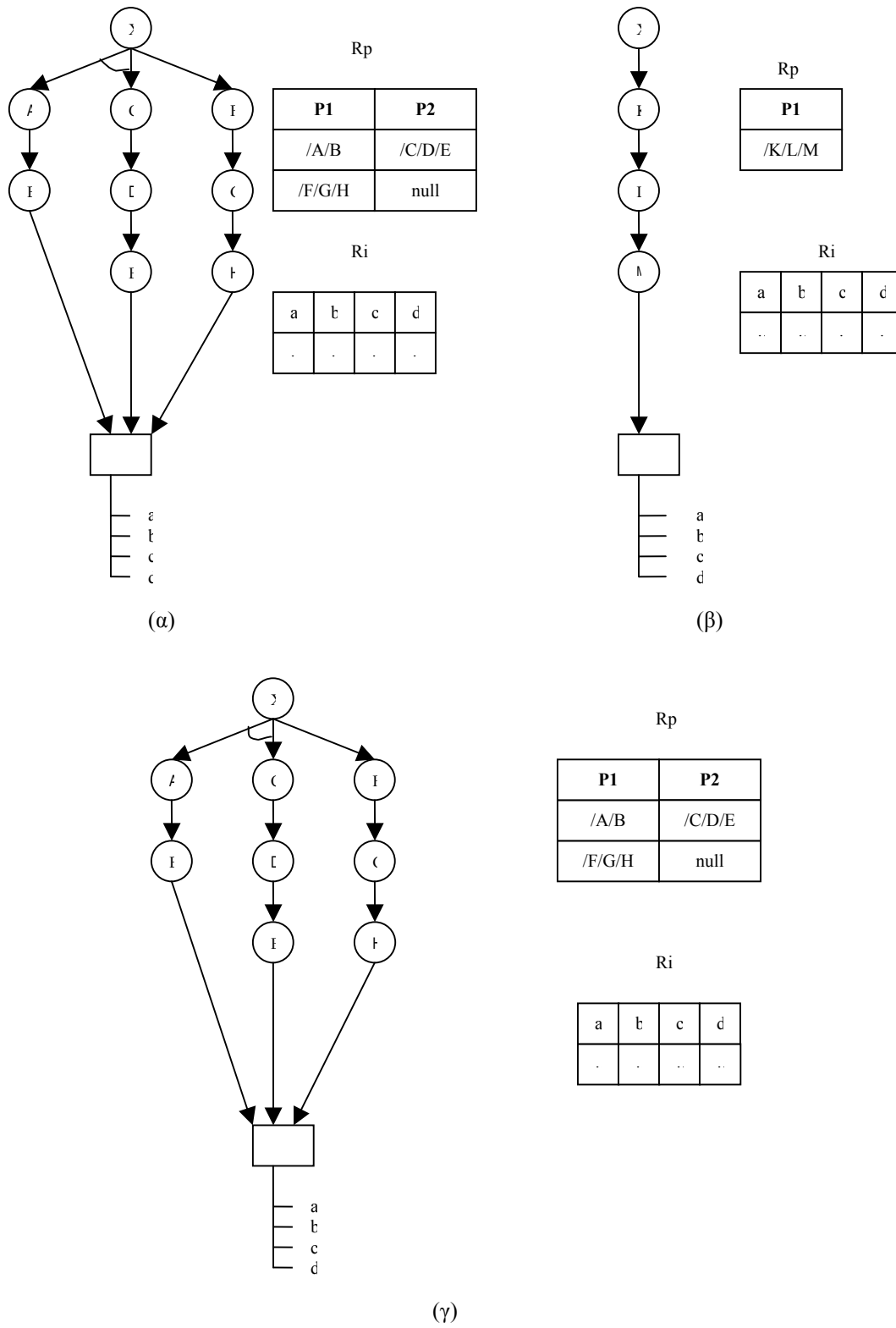
Rp

P1	P2
/A/B	/C/D/E
/F/G/H	null

Ri

a	b	c	d
.	.	..	..

Σχήμα 3.12



Σχήμα 3.13

# 4

## *Ανάλυση και Σχεδίαση*

Στο κεφάλαιο αυτό παρουσιάζεται η αναλυτική μελέτη του συστήματος. Αρχικά γίνεται ο διαχωρισμός του συστήματος σε επιμέρους υποσυστήματα και παρουσιάζονται οι λειτουργικές απαιτήσεις του καθενός. Στη συνέχεια παρουσιάζονται αναλυτικά οι εφαρμογές που υλοποιούν το σύστημα.

### *4.1 Ανάλυση – Περιγραφή Αρχιτεκτονικής*

Η ενότητα αυτή περιγράφει τα επιμέρους υποσυστήματα και παρουσιάζει την αρχιτεκτονική τους.

#### *4.1.1 Διαχωρισμός υποσυστημάτων*

Το σύστημα απαρτίζεται από τα παρακάτω υποσυστήματα:

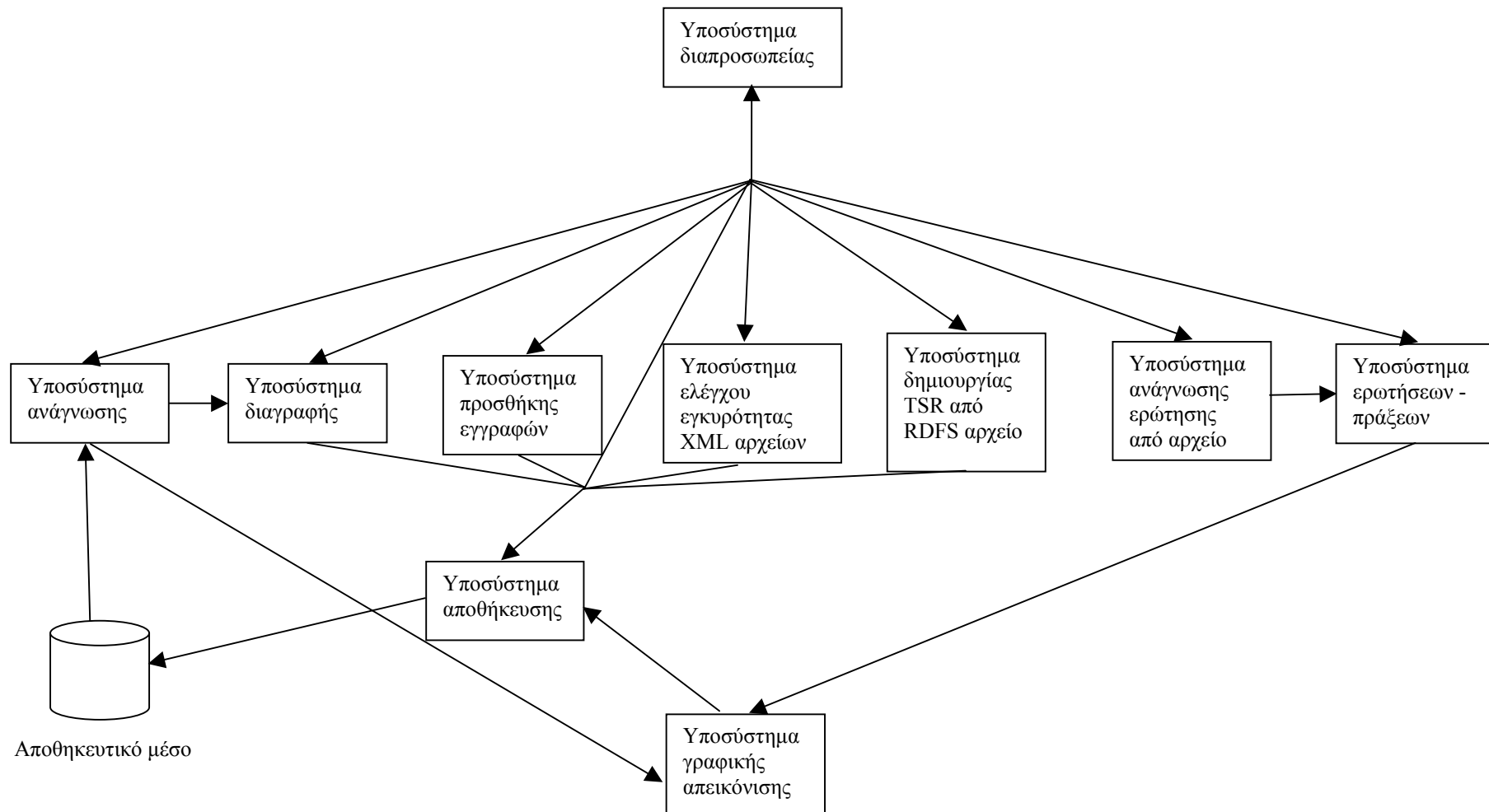
1. Υποσύστημα αποθήκευσης TSR σχημάτων
2. Υποσύστημα ανάγνωσης TSR σχημάτων
3. Υποσύστημα γραφικής απεικόνισης TSR σχημάτων
4. Υποσύστημα διαπροσωπείας χρήστη

5. Υποσύστημα διαγραφής TSR σχημάτων
6. Υποσύστημα ερωτήσεων – πράξεων
7. Υποσύστημα ανάγνωσης ερωτήσεων από αρχείο
8. Υποσύστημα ελέγχου εγκυρότητας XML αρχείων
9. Υποσύστημα δημιουργίας TSR σχημάτων από RDFS αρχείο
10. Υποσύστημα προσθήκης εγγραφών TSR σχημάτων

Όπως θα εξηγηθεί αναλυτικά στη συνέχεια, μέσα από τη διαπροσωπεία χρήστη (interface), ο χρήστης μπορεί να επιλέξει τη λειτουργία που θέλει να επιτελέσει και να εισάγει τα κατάλληλα δεδομένα, π.χ ποια TSR σχήματα θα συμμετάσχουν σε μία ερώτηση. Οι δυνατές λειτουργίες (εφαρμογές) που μπορεί να επιλέξει ο χρήστης του συστήματος είναι:

- απεικόνιση αποθηκευμένου TSR σχήματος
- διαγραφή TSR σχήματος
- μεταφορά του από τη βάση δεδομένων σε XML αρχείο και αντίστροφα
- αλλαγή των χαρακτηριστικών του (όνομα αρχείου, όνομα TSR σχήματος, περιγραφή και όνομα κόμβου – αντικειμένου)
- είσοδος και έλεγχος νέου TSR σχήματος από XML αρχείο
- δημιουργία νέων TSR σχημάτων από RDFS αρχείο
- δημιουργία νέων εγγραφών
- εισαγωγή νέου ερωτήματος
- εισαγωγή ερωτήματος αποθηκευμένου σε αρχείο
- βοήθεια.

Τα TSR σχήματα πάνω στα οποία επενεργούν οι λειτουργίες του συστήματος βρίσκονται στο αποθηκευτικό μέσο του συστήματος, το οποίο μπορεί να είναι είτε η βάση δεδομένων ή κάποια XML αρχεία. Για να εκτελεστούν οι λειτουργίες πρέπει πρώτα τα κατάλληλα TSR σχήματα να ανακτηθούν από το αποθηκευτικό μέσο. Αυτό γίνεται με τη βοήθεια του υποσυστήματος ανάγνωσης TSR σχημάτων. Στη συνέχεια εκτελείται η επιλεγμένη λειτουργία και, εάν αυτό είναι απαραίτητο, το αποτέλεσμα αποθηκεύεται και πάλι στο αποθηκευτικό μέσο με τη βοήθεια του υποσυστήματος αποθήκευσης TSR σχημάτων. Το υποσύστημα απεικόνισης TSR σχημάτων είναι υπεύθυνο για την γραφική παρουσίαση των αποθηκευμένων TSR σχημάτων αλλά και των αποτελεσμάτων των πράξεων. Τα υποσυστήματα ελέγχου εγκυρότητας XML αρχείων και δημιουργίας TSR σχημάτων από RDFS αρχείο συμβάλλουν στην ορθή εισαγωγή TSR σχημάτων τα οποία θα χρησιμοποιηθούν από το σύστημα στη συνέχεια. Το υποσύστημα προσθήκης εγγραφών TSR σχημάτων βοηθάει στη δημιουργία ιδιοτήτων και εγγραφών για έναν κενό κόμβο – αντικείμενο κάποιου TSR σχήματος και στην προσθήκη νέων εγγραφών για έναν κόμβο – αντικείμενο ενός TSR σχήματος ο οποίος περιέχει ήδη κάποιες ιδιότητες και κάποιες εγγραφές. Στο Σχήμα 4.1 παρουσιάζεται η αρχιτεκτονική του συστήματος.



Σχήμα 4.1

### ***4.1.2 Περιγραφή υποσυστημάτων***

Σε αυτήν την παράγραφο παρουσιάζονται αναλυτικά τα διάφορα υποσυστήματα και οι λειτουργίες που πρέπει να επιτελούν.

#### ***4.1.2.1 Υποσύστημα αποθήκευσης TSR σχημάτων***

Το υποσύστημα αυτό είναι υπεύθυνο για την αποθήκευση των TSR σχημάτων. Όταν μία λειτουργία πρέπει να αποθηκεύσει ένα TSR σχήμα ή όταν αυτό ζητηθεί από το χρήστη μέσω του υποσυστήματος απεικόνισης TSR σχημάτων, τότε με κατάλληλο τρόπο καταγράφονται στο αποθηκευτικό μέσο οι ιδιότητες και οι εγγραφές του κόμβου – αντικειμένου που περιέχονται στο TSR σχήμα, καθώς και το σύνολο των μονοπατιών που οδηγούν στον κόμβο αυτό. Στο σύστημα μας ως αποθηκευτικό μέσο χρησιμοποιείται τόσο η βάση δεδομένων όσο και τα XML αρχεία. Το υποσύστημα αυτό είναι υπεύθυνο και για τον έλεγχο των στοιχείων ενός TSR σχήματος (δηλαδή του ονόματος του και του ονόματος του αρχείου του), έτσι ώστε στο ίδιο αποθηκευτικό μέσο να μην υπάρχουν δύο TSR σχήματα με το ίδιο όνομα αρχείου και το ίδιο όνομα σχήματος.

#### ***4.1.2.2 Υποσύστημα ανάγνωσης TSR σχημάτων***

Τα TSR σχήματα που διαχειρίζεται το σύστημα βρίσκονται αποθηκευμένα στο αποθηκευτικό μέσο που αυτό προσφέρει (είναι δηλαδή αποθηκευμένα είτε στη βάση δεδομένων είτε σε κάποιο XML αρχείο). Η επιλογή από το χρήστη μιας λειτουργίας που χρησιμοποιεί ένα ή περισσότερα TSR σχήματα απαιτεί την εύρεση και προσπέλαση τους. Αυτή είναι η βασική λειτουργία του υποσυστήματος ανάγνωσης TSR σχημάτων.

#### ***4.1.2.3 Υποσύστημα απεικόνισης TSR σχημάτων***

Βασική λειτουργία αυτού του υποσυστήματος είναι η γραφική απεικόνιση των TSR σχημάτων που παράγονται από την εκτέλεση πράξεων σε TSR σχήματα, καθώς επίσης και η απλή απεικόνιση των αποθηκευμένων TSR σχημάτων. Πρέπει να πληρούνται οι παρακάτω προδιαγραφές:



- ✓ Απεικόνιση των TSR σχημάτων με τέτοιο τρόπο ώστε να είναι εμφανή τόσο ο κόμβος – αντικείμενο με τις ιδιότητες και τις εγγραφές του, όσο και το σύνολο των μονοπατιών που οδηγούν σε αυτόν.
- ✓ Αν το TSR σχήμα που απεικονίζεται δεν είναι ήδη αποθηκευμένο, πρέπει να παρέχεται η δυνατότητα αποθήκευσης του είτε στη βάση δεδομένων είτε σε κάποιο XML αρχείο.

Τα υποσύστημα αυτό δέχεται σαν είσοδο ένα TSR σχήμα το οποίο ο χρήστης επιθυμεί να απεικονίσει με γραφικό τρόπο. Για κάθε OR συνιστώσα και για κάθε AND μονοπάτι που την αποτελεί το υποσύστημα εντοπίζει τους κόμβους από τους οποίους αποτελείται και οι οποίοι σχηματίζουν το μονοπάτι που οδηγεί στον κόμβο – αντικείμενο του TSR σχήματος. Αυτοί οι κόμβοι και οι ακμές που τους συνδέουν πρέπει να απεικονιστούν στην οθόνη του υπολογιστή του χρήστη. Εκτός από τα μονοπάτια πρέπει να απεικονιστεί τόσο η ρίζα (*root*) του TSR σχήματος όσο και ο κόμβος – αντικείμενο (*resource\_item*) με τις ιδιότητες και τις εγγραφές του.

#### *4.1.2.4 Υποσύστημα διαπροσωπείας χρήστη*

Το υποσύστημα αυτό μαζί με το υποσύστημα απεικόνισης TSR σχημάτων αποτελούν τη διεπαφή ολόκληρου του συστήματος και αναλαμβάνουν την αλληλεπίδρασή του με το χρήστη. Είναι προφανές ότι το υποσύστημα αυτό πρέπει να είναι εύχρηστο και σε περίπτωση οποιουδήποτε λάθους πρέπει ο χρήστης να λαμβάνει κατάλληλα επεξηγηματικά μηνύματα. Με τη βοήθεια του υποσυστήματος αυτού αρχικά επιλέγεται μία από τις λειτουργίες του συστήματος. Οι λειτουργίες αυτές, οι οποίες θα επεξηγηθούν αναλυτικά παρακάτω είναι:

- απεικόνιση αποθηκευμένου TSR σχήματος
- διαγραφή TSR σχήματος
- μεταφορά TSR σχήματος από τη βάση δεδομένων σε XML αρχείο και αντίστροφα
- αλλαγή των χαρακτηριστικών στοιχείων ενός TSR σχήματος
- εισαγωγή και έλεγχος νέου XML αρχείου με TSR σχήματα
- δημιουργία νέων TSR σχημάτων από RDFS αρχείο
- δημιουργία νέων εγγραφών
- εισαγωγή νέας ερώτησης
- εισαγωγή νέας ερώτησης από αρχείο

Αφού επιλεγθεί η επιθυμητή λειτουργία εισάγονται μέσω του υποσυστήματος διαπροσωπείας χρήστη τα απαραίτητα για κάθε λειτουργία δεδομένα και ξεκινά η εκτέλεση της συγκεκριμένης λειτουργίας.

#### *4.1.2.5 Υποσύστημα διαγραφής TSR σχημάτων*

Το υποσύστημα αυτό είναι υπεύθυνο για τη διαγραφή TSR σχημάτων από τα αποθηκευτικά μέσα. Αν ο χρήστης επιλέξει να διαγράψει κάποιο TSR σχήμα, τότε μέσω του υποσυστήματος διαπροσωπείας χρήστη δίνονται οι απαραίτητες πληροφορίες ώστε το υποσύστημα ανάγνωσης TSR σχημάτων να μπορεί να εντοπίσει το επιθυμητό TSR σχήμα στο αποθηκευτικό μέσο. Στη συνέχεια με κατάλληλες ενέργειες το TSR σχήμα διαγράφεται από το αποθηκευτικό μέσο. Σημειώνεται ότι το υποσύστημα διαγραφής TSR σχημάτων δεν ενεργοποιείται μόνο μετά από αίτηση του χρήστη για διαγραφή κάποιου TSR σχήματος, αλλά ενεργοποιείται και από άλλες λειτουργίες όπως θα αναλυθεί στη συνέχεια.

#### *4.1.2.6 Υποσύστημα ερωτήσεων – πράξεων*

Το υποσύστημα αυτό υλοποιεί τους βασικούς τελεστές της γλώσσας οι οποίοι είναι: προβολή (projection), επιλογή (selection), καρτεσιανό γινόμενο (cartesian union), ένωση (union), τομή (intersection) και διαφορά (difference). Από το υποσύστημα διαπροσωπείας χρήστη (ή το υποσύστημα ανάγνωσης ερώτησης από αρχείο που θα αναλυθεί παρακάτω) συλλέγονται όλες οι παράμετροι που απαιτούνται για την εκτέλεση των πράξεων στα TSR σχήματα που έχουν φορτωθεί μέσω του υποσυστήματος ανάγνωσης TSR σχημάτων. Το υποσύστημα ερωτήσεων – πράξεων αναλαμβάνει την εκτέλεση των πράξεων, ενώ το αποτέλεσμα (ένα νέο TSR σχήμα) περνάει στο υποσύστημα απεικόνισης όπου και σχηματίζεται σε δεντρική μορφή.

#### *4.1.2.7 Υποσύστημα ανάγνωσης ερωτήσεων από αρχείο*

Εκτός από τη δυνατότητα εισαγωγής ερωτήσεων μέσω του υποσυστήματος διαπροσωπείας χρήστη που περιγράφηκε στην ενότητα 4.1.2.4, το σύστημα παρέχει έναν εναλλακτικό τρόπο για την εκτέλεση των ερωτήσεων. Οι ερωτήσεις μπορούν να αποθηκεύονται σε κατάλληλη μορφή σε αρχεία και να εκτελούνται μέσα από αυτά. Το αρχείο στο οποίο είναι

αποθηκευμένη η ερώτηση έχει προέκταση .qry. Στην αρχή κάθε αρχείου υπάρχει μία γραμμή με τον τίτλο του αρχείου (`--Query file`) ενώ στην αμέσως επόμενη γραμμή εισάγεται η ερώτηση. Το υποσύστημα ανάγνωσης ερωτήσεων από αρχείο προσφέρει στο χρήστη μια γραμματική σύνταξης ερωτήσεων επιλογής, προβολής, καρτεσιανού γινομένου, ένωσης, τομής και διαφοράς σε TSR σχήματα μέσω των αντίστοιχων εντολών (η γραμματική αυτή περιγράφεται αναλυτικά στο επόμενο κεφάλαιο). Το υποσύστημα αυτό αναλαμβάνει την ανάγνωση της ερώτησης του αρχείου και τη συλλογή των απαραίτητων πληροφοριών για την αποτίμηση της. Συγκεκριμένα χρησιμοποιώντας ένα συντακτικό αναλυτή (`parser`) αναγνωρίζει και ελέγχει την ορθότητα της σύνταξης της εντολής της αντίστοιχης πράξης και αφού συλλέξει τις απαραίτητες παραμέτρους, φορτώνει τα TSR σχήματα που χρειάζονται για την εκτέλεση της ερώτησης με τη βοήθεια του υποσυστήματος ανάγνωσης TSR σχημάτων. Αν αναγνωρισθεί κάποιο λάθος στη σύνταξη της ερώτησης τότε είναι απαραίτητη η εμφάνιση κατάλληλων επεξηγηματικών μηνυμάτων ώστε να είναι εύκολος ο εντοπισμός και η διόρθωση του σφάλματος από το χρήστη.

#### *4.1.2.8 Υποσύστημα ελέγχου εγκυρότητας TSR σχημάτων*

Το υποσύστημα αυτό είναι υπεύθυνο για τον έλεγχο της ορθότητας των TSR σχημάτων ενός XML αρχείου. Για την αποθήκευση ενός καινούριου TSR σχήματος (είτε αυτό είναι το αποτέλεσμα κάποιων πράξεων πάνω σε ένα ή περισσότερα TSR σχήματα είτε εισάγεται για πρώτη φορά στο σύστημα) είναι απαραίτητος ο έλεγχος για το εάν αυτό είναι σύμφωνο με το DTD της εφαρμογής. Σε περίπτωση που στο XML αρχείο εντοπιστεί κάποιο σφάλμα, τότε είναι απαραίτητη η αναλυτική ενημέρωση του χρήστη για το λάθος, ώστε να είναι δυνατή η διόρθωση του και η επανάληψη της διαδικασίας.

#### *4.1.2.9 Υποσύστημα δημιουργίας TSR σχημάτων από RDFS αρχείο*

Όπως έχει εξηγηθεί και νωρίτερα, το RDF (Resource Description Framework) είναι ένα μοντέλο για την αναπαράσταση πληροφοριών για πηγές στο διαδίκτυο. Το RDF σχεδιάστηκε για περιπτώσεις που η πληροφορία που αναπαριστά πρέπει να χρησιμοποιηθεί από εφαρμογές και μάλιστα παρέχει τη δυνατότητα ανταλλαγής πληροφοριών μεταξύ εφαρμογών χωρίς απώλειες. Ένα RDFS αρχείο περιέχει ένα σχήμα για RDF δεδομένα, κατά αναλογία με το DTD που είναι ένα σχήμα για XML δεδομένα.

Το υποσύστημα αυτό είναι υπεύθυνο για τη δημιουργία νέων TSR σχημάτων με βάση τις ιεραρχίες που βρίσκονται σε ένα RDFS αρχείο. Από ένα RDFS αρχείο θα προκύψουν ένα ή περισσότερα TSR σχήματα τα οποία όμως θα αποτελούνται μόνο από μία OR συνιστώσα με ένα μονοπάτι. Επίσης οι κόμβοι – αντικείμενα των TSR σχημάτων που δημιουργούνται είναι κενοί (δεν περιέχουν δηλαδή ούτε ιδιότητες ούτε εγγραφές). Το υποσύστημα αυτό παρέχει έναν εύκολο και γρήγορο τρόπο δημιουργίας TSR σχημάτων.

#### *4.1.2.10 Υποσύστημα προσθήκης εγγραφών TSR σχημάτων*

Το υποσύστημα αυτό είναι υπεύθυνο για την προσθήκη νέων εγγραφών σε κόμβους - αντικείμενα TSR σχημάτων. Οι κόμβοι – αντικείμενα των TSR σχημάτων μπορούν να είναι κενοί από εγγραφές ή όχι. Στην πρώτη περίπτωση πρέπει εκτός από τις εγγραφές να εισαχθούν και οι ιδιότητες. Τόσο οι ιδιότητες όσο και οι εγγραφές επιλέγονται από το χρήστη μέσω του υποσυστήματος διαπροσωπείας χρήστη. Συγκεκριμένα επιλέγονται γνωρίσματα και πλειάδες από έναν πίνακα ο οποίος βρίσκεται αποθηκευμένος στη βάση δεδομένων και ο οποίος περιέχει διάφορα δεδομένα. Με αυτόν τον τρόπο ο χρήστης μπορεί να επιλέξει ποια γνωρίσματα αλλά και ποιες πλειάδες του πίνακα αυτού ταιριάζουν με κάθε TSR σχήμα και να δημιουργήσει για τον κόμβο – αντικείμενο του TSR σχήματος ιδιότητες και εγγραφές. Αφού εισαχθούν οι ιδιότητες, αναγνωρίζεται αυτόματα ο τύπος τους και μαζί με τις πλειάδες αποθηκεύονται στο αποθηκευτικό μέσο όπου βρίσκεται και το TSR σχήμα. Από την άλλη μεριά, όταν ο κόμβος – αντικείμενο του TSR σχήματος αποτελείται από κάποιες ιδιότητες πρέπει οι ιδιότητες που θα επιλεγθούν από το χρήστη να είναι ίδιες ως προς το όνομα και τον τύπο με αυτές που ήδη έχει. Στην περίπτωση που αυτό δε συμβαίνει η εισαγωγή των εγγραφών που έχουν επιλεγθεί είναι αδύνατη. Αντίθετα αν οι ιδιότητες που επιλέχθηκαν είναι ίδιες σε αριθμό, όνομα και τύπο με αυτές που έχει ο κόμβος – αντικείμενο του TSR σχήματος τότε οι εισαγωγές των νέων εγγραφών μπορεί να συνεχιστεί. Οι νέες εγγραφές προστίθεται στον κόμβο – αντικείμενο του TSR σχήματος. Αυτό σημαίνει ότι μετά το τέλος της διαδικασίας ο κόμβος – αντικείμενο του TSR σχήματος θα περιέχει τόσο τις αρχικές όσο και τις νέες εγγραφές. Σημειώνεται ότι η επιλογή των ιδιοτήτων και των εγγραφών του κόμβου – αντικειμένου από το χρήστη γίνεται με την εισαγωγή μίας απλής SQL ερώτησης. Επομένως το υποσύστημα προσθήκης εγγραφών TSR σχημάτων δέχεται σαν είσοδο το TSR σχήμα στο οποίο θέλουμε να προσθέσουμε εγγραφές και την SQL ερώτηση με βάση την οποία θα επιλεγθούν οι εγγραφές και ενδεχομένως οι ιδιότητες του κόμβου – αντικειμένου του TSR σχήματος. Αν δεν προκύψουν σφάλματα το υποσύστημα αυτό δίνει ένα νέο TSR σχήμα του

οποίου ο κόμβος αντικείμενο περιέχει τόσο τις αρχικές (αν υπάρχουν) όσο και τις νέες εγγραφές.

## **4.2 Σχεδίαση του συστήματος**

Στην ενότητα αυτή παρουσιάζονται αναλυτικά οι εφαρμογές του συστήματος.

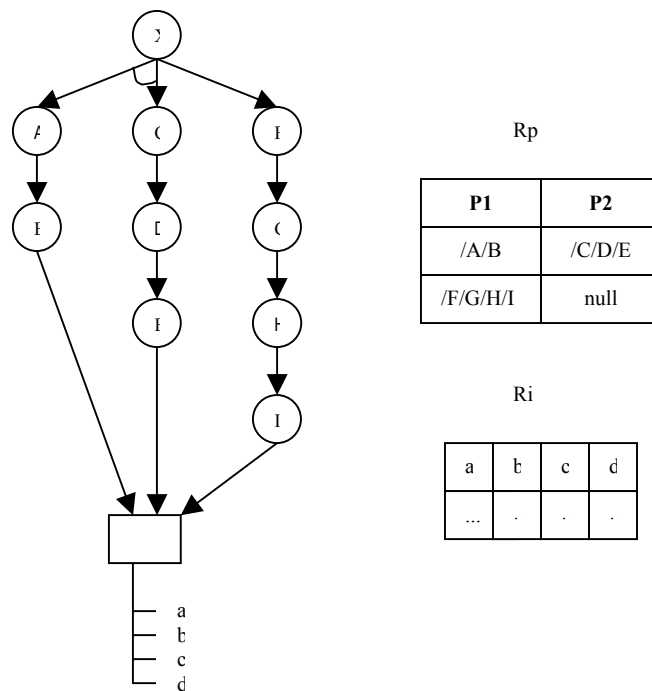
1. Εφαρμογή διαχείρισης XML αρχείου
2. Εφαρμογή διαχείρισης βάσης δεδομένων
3. Εφαρμογή απεικόνισης αποθηκευμένου TSR σχήματος
4. Εφαρμογή διαγραφής TSR σχήματος
5. Εφαρμογή μεταφοράς TSR σχήματος από τη βάση δεδομένων σε XML αρχείο
6. Εφαρμογή μεταφοράς TSR σχήματος από κάποιο XML αρχείο στη βάση δεδομένων
7. Εφαρμογή αλλαγής χαρακτηριστικών στοιχείων TSR σχημάτων
8. Εφαρμογή εισαγωγής ερώτησης
9. Εφαρμογή εισαγωγής ερώτησης από αρχείο
10. Εφαρμογή εισαγωγής TSR σχημάτων από XML αρχείο
11. Εφαρμογή εισαγωγής TSR σχημάτων από RDFS αρχείο
12. Εφαρμογή προσθήκης εγγραφών TSR σχημάτων

### **4.2.1 Εφαρμογή διαχείρισης XML αρχείου**

Η εφαρμογή αυτή είναι υπεύθυνη για την αποθήκευση και τη φόρτωση ενός TSR σχήματος σε και από κάποιο XML αρχείο αντίστοιχα.

Κάθε XML αρχείο κρατάει πληροφορίες για το όνομα του TSR σχήματος, για το σύνολο των μονοπατιών και για τον κόμβο – αντικείμενο του. Συγκεκριμένα, περιγράφεται το  $R_p - R_i$  μοντέλο του σχήματος παραθέτοντας τις OR συνιστώσες και προσθέτοντας σε κάθε μία τα AND μονοπάτια που περιέχει. Κάθε OR συνιστώσα περικλείεται σε `<or>` tag ενώ κάθε AND μονοπάτι σε `<and>` tag. Στη συνέχεια παρατίθεται μέσα σε `<description>` tag η περιγραφή του TSR σχήματος. Για τον κόμβο – αντικείμενο παρατίθεται το όνομα και ο τύπος των ιδιοτήτων του καθώς και οι εγγραφές των περιεχομένων του. Ολόκληρος ο κόμβος – αντικείμενο περικλείεται σε ένα `<item>` tag ενώ οι ιδιότητες και οι εγγραφές του σε `<attribute>` και `<tuple>` tags αντίστοιχα. Σημειώνουμε ότι για τον κόμβο – αντικείμενο κρατείται πληροφορία και για το όνομα του στο `<item>` tag.

Για την αποθήκευση επομένως ενός TSR σχήματος σε κάποιο XML αρχείο, αρκεί η αποθήκευση των μονοπατιών και του κόμβου – αντικειμένου του (δηλαδή των ιδιοτήτων και των εγγραφών του κόμβου – αντικειμένου) στα κατάλληλα tags. Για παράδειγμα ας θεωρήσουμε την TSR R<sub>i</sub> του Σχήματος 4.2, η οποία αποτελείται από δύο OR συνιστώσες. Την πρώτη OR συνιστώσα σχηματίζουν τα γειτονικά AND μονοπάτια /A/B και /C/D/E, ενώ η δεύτερη OR συνιστώσα αποτελείται από το μονοπάτι /F/G/H/I. Στο σχήμα εκτός από την TSR φαίνονται και οι σχέσεις R<sub>p</sub> και R<sub>i</sub> του R<sub>p</sub> – R<sub>i</sub> μοντέλου.



**Σχήμα 4.2**

Αυτό το TSR σχήμα αποθηκεύεται σε XML αρχείο έχοντας την εξής μορφή:

```
<root>
  <tsr name = "R1">
    <or>
      <and> /A/B </and>
      <and> /C/D/E </and>
    </or>
    <or>
      <and>/F/G/H/I</and>
    </or>
    <description> '...' </description>
    <item name = "...">
      <attribute name = "a" type = "..."/>
```

```

        <attribute name = "b" type = "..."/>
        <attribute name = "c" type = "..."/>
        <attribute name = "d" type = "..."/>
        <tuple> '...', '...', '...', '...' </tuple>
        <tuple> '...', '...', '...', '...' </tuple>
    </item>
</tsr>
</root>

```

Σημειώνουμε ότι για λόγους ευκολίας στο παραπάνω XML αρχείο δεν έχουν εισαχθεί εγγραφές, αλλά ούτε και η περιγραφή του TSR σχήματος και το όνομα του κόμβου – αντικειμένου που δεν είναι γνωστά.

Όσον αφορά τη φόρτωση ενός TSR σχήματος που βρίσκεται αποθηκευμένο σε κάποιο XML αρχείο, αρκεί μία απλή προσπέλαση του αρχείου αυτού ώστε να ανακτηθούν όλες οι απαραίτητες πληροφορίες για τα AND μονοπάτια και τις OR συνιστώσες αλλά και για τον κόμβο - αντικείμενο.

#### **4.2.2 Εφαρμογή διαχείρισης βάσης δεδομένων**

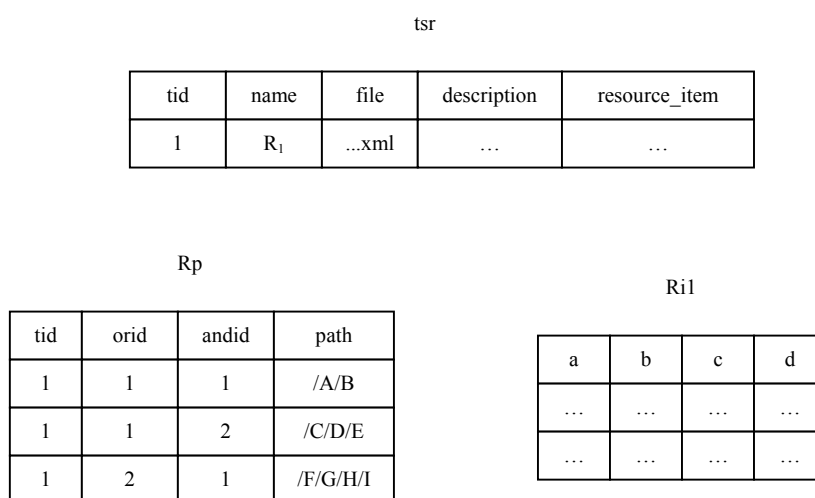
Η εφαρμογή αυτή είναι υπεύθυνη για την αποθήκευση και την προσπέλαση των TSR σχημάτων στην και από τη βάση δεδομένων του συστήματος αντίστοιχα.

Κάθε TSR σχήμα που αποθηκεύεται στη βάση δεδομένων οργανώνεται ακολουθώντας το  $R_p - R_i$  μοντέλο. Σύμφωνα με το μοντέλο αυτό, για κάθε TSR σχήμα που εισάγετε στη βάση δεδομένων δημιουργείται μία πλειάδα στον πίνακα tsr, ο οποίος είναι αντίστοιχος της οντότητας  $R_{tsr}$ , όπου φυλάσσονται διάφορα χαρακτηριστικά στοιχεία του TSR σχήματος. Αυτά είναι ένας κωδικός (tid) για το συγκεκριμένο TSR σχήμα, το όνομά του, το όνομα του αρχείου από το οποίο προέρχεται, η περιγραφή του και το όνομα του κόμβου – αντικειμένου του. Για την οντότητα  $R_i$  ορίζεται ένας νέος πίνακας με όνομα  $R_i$  συν τον κωδικό tid του TSR σχήματος από την αντίστοιχη εγγραφή στον tsr πίνακα. Δηλαδή για κάθε TSR σχήμα που αποθηκεύεται στη βάση δεδομένων δημιουργείται ένας νέος πίνακας για την αποθήκευση των ιδιοτήτων και των εγγραφών του. Όσον αφορά τα μονοπάτια του TSR σχήματος, αυτά αποθηκεύονται στον πίνακα  $R_p$ , εισάγοντας σε αυτόν τόσες πλειάδες όσα είναι και τα AND μονοπάτια του TSR σχήματος. Κάθε πλειάδα περιέχει πληροφορίες για ένα AND μονοπάτι του TSR σχήματος. Αυτές οι πληροφορίες είναι το ίδιο το μονοπάτι (δηλαδή οι κόμβοι που το αποτελούν), η OR συνιστώσα στην οποία ανήκει αλλά και ο αύξοντα αριθμός αυτού του AND μονοπατιού μέσα στην OR συνιστώσα. Με κατάλληλες SQL ερωτήσεις αποθηκεύονται στους κατάλληλους πίνακες τα κατάλληλα δεδομένα από το TSR σχήμα. Για παράδειγμα στο

Σχήμα 4.3 φαίνονται τα γνωρίσματα και οι πλειάδες των πινάκων της βάσης δεδομένων που αντιστοιχούν στις εγγραφές για την TSR του Σχήματος 4.2.

Σημειώνεται ότι όπως και στο παράδειγμα με το XML αρχείο, δεν έχουν εισαχθεί στον πίνακα  $R_i1$  εγγραφές ενώ στον πίνακα  $tsr$  δεν έχουν εισαχθεί το όνομα του αρχείου από το οποίο προέρχεται το TSR σχήμα (*file*), η περιγραφή του (*description*) και το όνομα του κόμβου – αντικειμένου (*resource\_item*) που δεν είναι γνωστά.

Όταν ένα TSR σχήμα πρέπει να ανακτηθεί από τη βάση δεδομένων, τότε, αφού εντοπιστεί η εγγραφή του συγκεκριμένου σχήματος στη βάση, με κατάλληλες SQL ερωτήσεις μπορούμε να ανακτήσουμε τόσο το σύνολο των μονοπατιών όσο και τον κόμβο – αντικείμενο του, και επομένως να φορτώσουμε το επιθυμητό TSR.



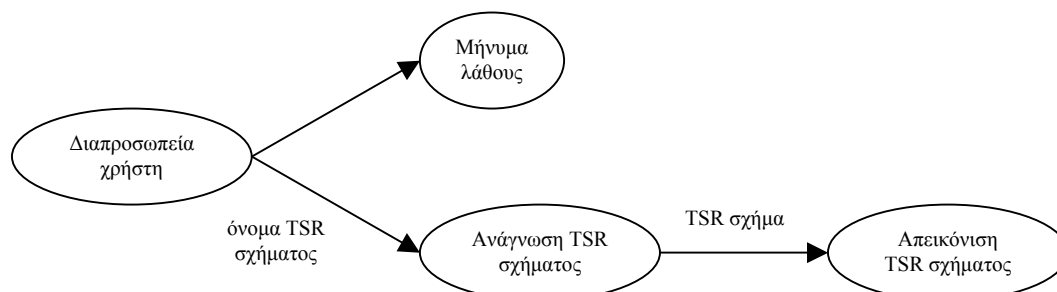
**Σχήμα 4.3**

### **4.2.3 Εφαρμογή απεικόνισης αποθηκευμένου TSR σχήματος**

Με τη βοήθεια αυτής της εφαρμογής παρέχεται στο χρήστη η δυνατότητα να δει κάποιο από τα αποθηκευμένα TSR σχήματα στην οθόνη του υπολογιστή σε γραφική μορφή. Η διαδικασία που ακολουθείται είναι η εξής: μέσω του υποσυστήματος διαπροσωπείας χρήστη επιλέγεται το επιθυμητό TSR σχήμα προς απεικόνιση. Εάν δεν υπάρχουν αποθηκευμένα TSR σχήματα τότε θα εμφανιστεί κατάλληλο μήνυμα. Το υποσύστημα ανάγνωσης TSR σχημάτων αναλαμβάνει την προσπέλαση του TSR σχήματος που επιλέχθηκε, ενώ το υποσύστημα γραφικής απεικόνισης TSR σχημάτων αναλαμβάνει την απεικόνιση του σε γραφική μορφή δέντρου. Σημειώνεται ότι το γραφικό περιβάλλον παρουσίασης TSR σχημάτων δεν παρέχει σε αυτήν την περίπτωση τη δυνατότητα αποθήκευσης του σχήματος στη βάση δεδομένων ή σε κάποιο XML αρχείο. Αυτό συμβαίνει γιατί το TSR που απεικονίζεται είναι ήδη



αποθηκευμένο σε κάποιο από το αποθηκευτικά μέσα. Επομένως αυτή η εφαρμογή έχει σαν στόχο απλά την απεικόνιση κάποιου TSR σχήματος. Στο Σχήμα 4.4 φαίνεται το διάγραμμα ροής της εφαρμογής απεικόνισης αποθηκευμένου TSR σχήματος:

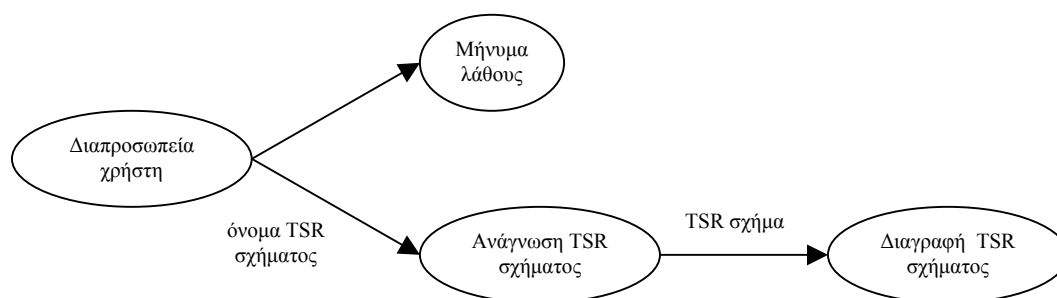


Σχήμα 4.4

#### 4.2.4 Εφαρμογή διαγραφής TSR σχήματος

Η εφαρμογή αυτή παρέχει στο χρήστη τη δυνατότητα διαγραφής κάποιου TSR σχήματος το οποίο βρίσκεται αποθηκευμένο είτε στη βάση δεδομένων είτε σε κάποιο XML αρχείο. Συγκεκριμένα, αν το TSR σχήμα βρίσκεται αποθηκευμένο στη βάση δεδομένων τότε με κατάλληλες SQL ερωτήσεις διαγράφονται από τους πίνακες της βάσης δεδομένων όλες οι εγγραφές που αφορούν το συγκεκριμένο TSR σχήμα. Δηλαδή από τον πίνακα  $t_{sr}$  διαγράφεται η εγγραφή που κρατούσε πληροφορίες για τον κωδικό του TSR σχήματος, το όνομά του, το όνομα του αρχείου από το οποίο προερχόταν, την περιγραφή του και το όνομα του κόμβου – αντικειμένου του. Από τον πίνακα  $R_p$  διαγράφονται οι πλειάδες που κρατούσαν πληροφορίες για τα μονοπάτια του TSR σχήματος, ενώ διαγράφεται ολόκληρος ο πίνακας  $R_{tid}$  (όπου  $tid$  είναι ο κωδικός του TSR σχήματος προς διαγραφή) ο οποίος κρατούσε πληροφορίες για τις ιδιότητες και τις εγγραφές του κόμβου αντικειμένου του TSR σχήματος. Αν το TSR σχήμα είναι αποθηκευμένο σε κάποιο XML αρχείο, τότε απλά σβήνονται από το αρχείο αυτό όλες οι γραμμές που περιείχαν πληροφορίες για το συγκεκριμένο TSR σχήμα. Η διαδικασία που ακολουθείται είναι η εξής: από το υποσύστημα διαπροσωπείας χρήστη επιλέγεται το επιθυμητό TSR σχήμα για διαγραφή. Στη συνέχεια το υποσύστημα ανάγνωσης TSR σχημάτων φορτώνει το επιλεγμένο TSR σχήμα, ενώ το υποσύστημα διαγραφής TSR σχημάτων αναλαμβάνει τη διαγραφή του από το αποθηκευτικό μέσο στο οποίο βρίσκεται. Εάν δεν υπάρχουν αποθηκευμένα TSR σχήματα τότε θα εμφανιστεί κατάλληλο μήνυμα. Σημειώνεται τέλος ότι εάν το TSR σχήμα είναι αποθηκευμένο σε ένα XML αρχείο το οποίο περιέχει μόνο αυτό το TSR σχήμα, η διαγραφή του σχήματος σημαίνει αυτόματα και την

διαγραφή ολόκληρου του XML αρχείου. Στο Σχήμα 4.5 φαίνεται το διάγραμμα ροής της εφαρμογής διαγραφής TSR σχήματος:



Σχήμα 4.5

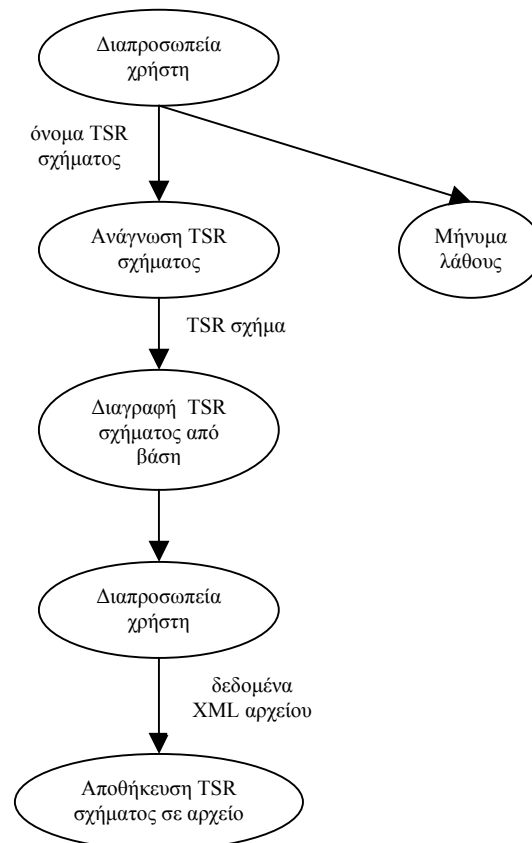
#### ***4.2.5 Εφαρμογή μεταφοράς TSR σχήματος από τη βάση δεδομένων σε XML αρχείο***

Η εφαρμογή αυτή παρέχει τη δυνατότητα αλλαγής αποθηκευτικού μέσου. Συγκεκριμένα, ένα TSR σχήμα που βρίσκεται αποθηκευμένο στη βάση δεδομένων, μπορεί με κατάλληλες ενέργειες να σβηστεί από αυτήν και να αποθηκευτεί σε κάποιο XML αρχείο. Η διαδικασία που ακολουθείται είναι η εξής: αρχικά από το υποσύστημα διαπροσωπείας χρήστη επιλέγεται κάποιο από τα TSR σχήματα που είναι αποθηκευμένα στη βάση δεδομένων. Σημειώνεται ότι αν η βάση δεδομένων δεν περιέχει κανένα TSR σχήμα θα εμφανιστεί κατάλληλο μήνυμα λάθους. Στη συνέχεια με τη βοήθεια του υποσυστήματος ανάγνωσης TSR σχημάτων, το TSR σχήμα που επιλέχθηκε φορτώνεται, ενώ με τη βοήθεια του υποσυστήματος διαγραφής διαγράφεται από τη βάση δεδομένων. Από το υποσύστημα διαπροσωπείας χρήστη εισάγονται τα απαραίτητα στοιχεία για την αποθήκευση του TSR σχήματος σε κάποιο XML αρχείο, όπως είναι το όνομα του αρχείου, το όνομα του TSR σχήματος και η περιγραφή του. Τέλος το υποσύστημα αποθήκευσης TSR σχημάτων αναλαμβάνει την αποθήκευση του TSR σχήματος στο XML αρχείο που επιλέχθηκε. Στο Σχήμα 4.6 φαίνεται το διάγραμμα ροής της εφαρμογής μεταφοράς ενός TSR σχήματος από τη βάση δεδομένων σε κάποιο XML αρχείο.

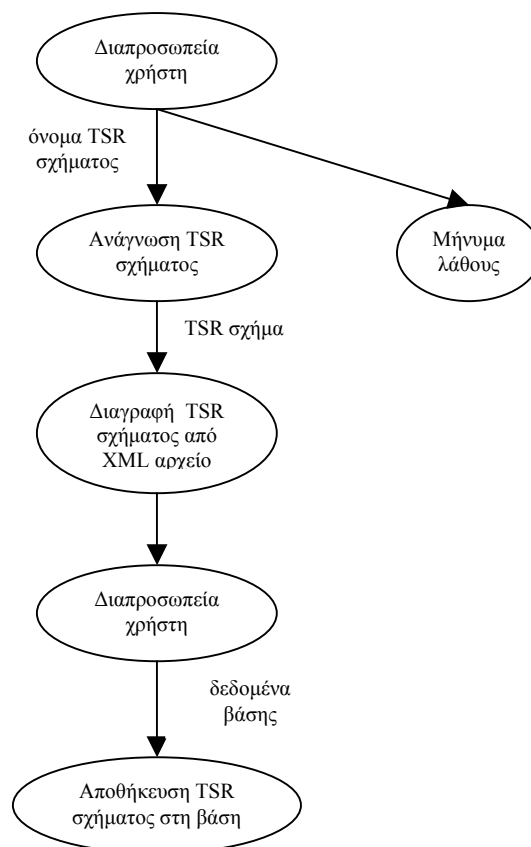
#### ***4.2.6 Εφαρμογή μεταφοράς TSR σχήματος από XML αρχείο στη βάση δεδομένων***

Με τη βοήθεια αυτής της εφαρμογής παρέχεται ακόμη ένας τρόπος αλλαγής αποθηκευτικού μέσου. Συγκεκριμένα ένα TSR σχήμα που βρίσκεται αποθηκευμένο σε κάποιο XML αρχείο,

μπορεί με κατάλληλες ενέργειες να σβηστεί από αυτό το XML αρχείο και να αποθηκευτεί στη βάση δεδομένων. Αρχικά, λοιπόν, επιλέγεται κάποιο από τα TSR σχήματα που βρίσκονται αποθηκευμένα σε XML αρχεία. Αν δεν υπάρχουν καθόλου XML αρχεία τότε εμφανίζεται ένα μήνυμα λάθους. Στη συνέχεια με τη βοήθεια του υποσυστήματος ανάγνωσης TSR σχημάτων, το TSR σχήμα που έχει επιλεγεί προσπελαύνεται και έπειτα με τη βοήθεια του υποσυστήματος διαγραφής TSR σχημάτων, διαγράφεται από το XML αρχείο στο οποίο βρισκόταν. Το υποσύστημα διαπροσωπείας χρήστη αναλαμβάνει τη συλλογή των απαραίτητων στοιχείων για την αποθήκευση στη βάση δεδομένων. Εισάγονται, λοιπόν, από το χρήστη το όνομα του TSR σχήματος, μία περιγραφή του και το όνομα του αρχείου από το οποίο προέρχεται. Τέλος το υποσύστημα αποθήκευσης TSR σχημάτων αναλαμβάνει την αποθήκευση του TSR σχήματος στη βάση δεδομένων. Στο Σχήμα 4.7 φαίνεται το διάγραμμα ροής της εφαρμογής μεταφοράς ενός TSR σχήματος από XML αρχείο στη βάση δεδομένων.



**Σχήμα 4.6**

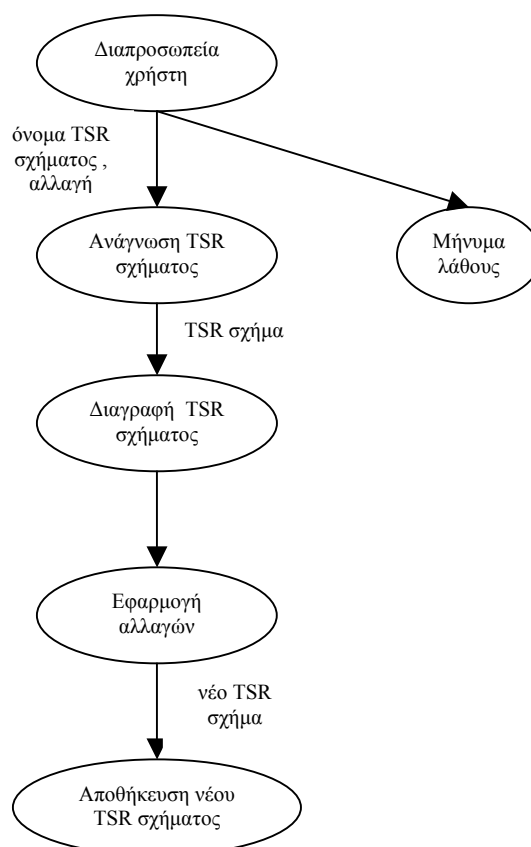


**Σχήμα 4.7**

#### **4.2.7 Εφαρμογή αλλαγής χαρακτηριστικών στοιχείων TSR σχήματος**

Η εφαρμογή αυτή είναι υπεύθυνη για την αλλαγή των χαρακτηριστικών στοιχείων ενός TSR σχήματος. Αυτά τα χαρακτηριστικά στοιχεία είναι το όνομα του TSR σχήματος, το όνομα του αρχείου στο οποίο είναι αποθηκευμένο ή από το οποίο προέρχεται, η περιγραφή του σχήματος και το όνομα του κόμβου – αντικειμένου του. Η διαδικασία που ακολουθείται είναι η εξής: αρχικά μέσω του υποσυστήματος διαπροσωπείας χρήστη επιλέγεται το TSR σχήμα στο οποίο ο χρήστης επιθυμεί να εφαρμόσει κάποια αλλαγή, καθώς επίσης και η επιθυμητή αλλαγή. Στη συνέχεια εισάγονται τα νέα χαρακτηριστικά στοιχεία του TSR σχήματος. Συγκεκριμένα αν η επιθυμητή αλλαγή που έχει ζητηθεί είναι η αλλαγή του ονόματος του TSR σχήματος τότε πρέπει να εισαχθεί το νέο όνομα της TSR. Αν η επιθυμητή αλλαγή είναι η αλλαγή του ονόματος του αρχείου του TSR σχήματος, τότε ζητείται η εισαγωγή του νέου ονόματος του αρχείου. Τέλος αν η επιθυμητή αλλαγή είναι αυτή της περιγραφής του TSR σχήματος τότε ο χρήστης πρέπει να εισάγει τη νέα περιγραφή, ενώ αν επιθυμείται η αλλαγή του ονόματος του κόμβου – αντικειμένου ο χρήστης πρέπει να εισάγει το νέο όνομα του. Με τη βοήθεια, λοιπόν του υποσυστήματος διαπροσωπείας χρήστη συλλέγονται όλα τα

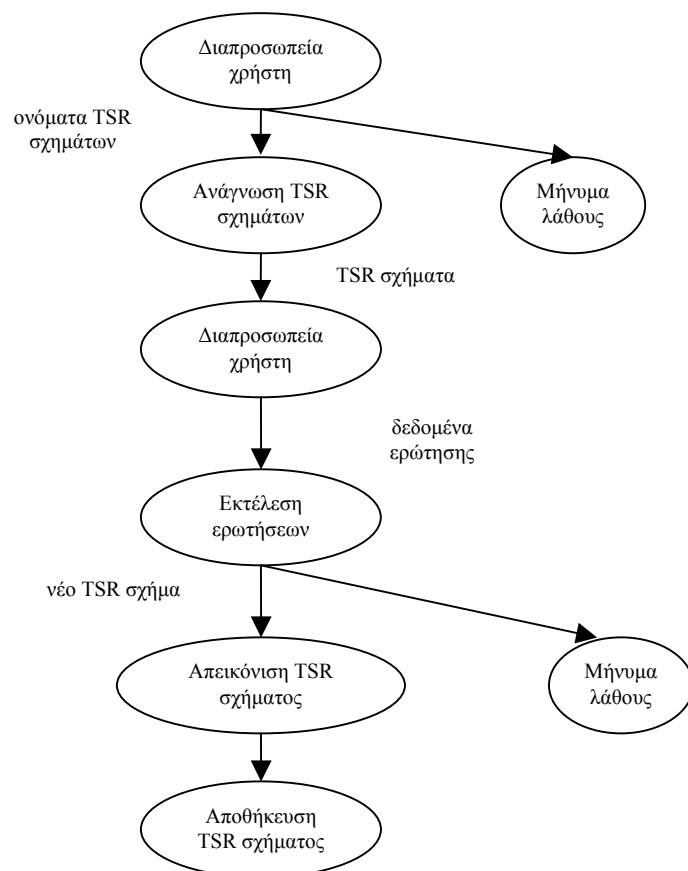
απαραίτητα δεδομένα για την εκτέλεση της αλλαγής. Στη συνέχεια το υποσύστημα ανάγνωσης TSR σχημάτων αναλαμβάνει την προσέλαση του επιλεγμένου TSR σχήματος και το υποσύστημα διαγραφής TSR σχημάτων αναλαμβάνει τη διαγραφή του από το αποθηκευτικό μέσο στο οποίο βρισκόταν. Στη συνέχεια εκτελείται η αλλαγή που έχει επιλεγεί πάνω στο TSR σχήμα. Με αυτόν τον τρόπο δημιουργείται ένα νέο TSR σχήμα το οποίο όσον αφορά τα μονοπάτια και τις ιδιότητες και τις εγγραφές του κόμβου – αντικείμενου δεν έχει καμία διαφορά με το αρχικό. Έχει όμως αλλαχθεί κάποιο από τα βασικά χαρακτηριστικά στοιχεία του. Τέλος το υποσύστημα αποθήκευσης TSR σχημάτων αναλαμβάνει την αποθήκευση του νέου TSR σχήματος στο ίδιο αποθηκευτικό μέσο στο οποίο βρισκόταν και το αρχικό TSR σχήμα. Σημειώνεται ότι η αλλαγή του ονόματος του XML αρχείου ενός TSR σχήματος πρόκειται ουσιαστικά για μεταφορά TSR σχημάτων μέσα στο ίδιο αποθηκευτικό μέσο. Στο Σχήμα 4.8 φαίνεται το διάγραμμα ροής της εφαρμογής αλλαγής χαρακτηριστικών ενός TSR σχήματος:



**Σχήμα 4.8**

#### 4.2.8 Εφαρμογή εισαγωγής ερώτησης

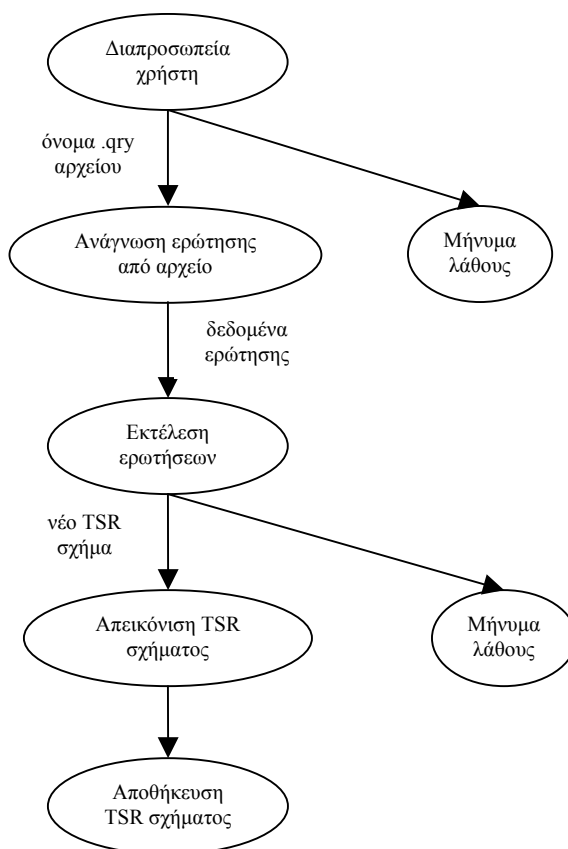
Η εφαρμογή αυτή είναι η βασική εφαρμογή του συστήματος αφού δίνει τη δυνατότητα στο χρήστη να εισάγει ερωτήσεις και πράξεις πάνω σε ένα ή περισσότερα TSR σχήματα. Μέσω αυτής ο χρήστης μπορεί να εκφράσει ερωτήματα πάνω στα μονοπάτια ενός TSR σχήματος αλλά και ερωτήσεις επιλογής και προβολής στις εγγραφές του κόμβου – αντικειμένου. Επιπλέον δίνεται η δυνατότητα συνδυασμού δύο ή περισσότερων TSR σχημάτων με τις πράξεις της ένωσης, της τομής, της διαφοράς και του καρτεσιανού γινομένου. Αρχικά, λοιπόν, μέσω του υποσυστήματος διαπροσωπείας χρήστη επιλέγονται όλα τα TSR σχήματα τα οποία θα συμμετάσχουν στην ερώτηση και στη συνέχεια φορτώνονται μέσω του υποσυστήματος ανάγνωσης TSR σχημάτων. Έπειτα με τη βοήθεια του υποσυστήματος διαπροσωπείας χρήστη και πάλι, συλλέγονται όλες οι απαραίτητες πληροφορίες για την εκτέλεση της ερώτησης. Το υποσύστημα ερωτήσεων – πράξεων αναλαμβάνει την εκτέλεση της ερώτησης, ενώ το αποτέλεσμα περνάει στο υποσύστημα απεικόνισης TSR σχημάτων με τη βοήθεια του οποίου το TSR σχήμα που έχει προκύψει απεικονίζεται σε μορφή δέντρου. Τέλος ο χρήστης μπορεί προαιρετικά να επιλέξει την αποθήκευση του TSR σχήματος στη βάση δεδομένων ή σε κάποιο XML αρχείο. Στο Σχήμα 4.9 φαίνεται το διάγραμμα ροής της εφαρμογής εισαγωγής ερώτησης.



Σχήμα 4.9

#### 4.2.9 Εφαρμογή εισαγωγής ερώτησης από αρχείο

Η εφαρμογή αυτή παρέχει στο χρήστη έναν εναλλακτικό τρόπο για την εισαγωγή μιας ερώτησης πάνω σε ένα ή περισσότερα TSR σχήματα. Η ερώτηση που πρόκειται να εκτελεστεί είναι αποθηκευμένη σε κατάλληλο αρχείο με τρόπο ο οποίος αναλύθηκε προηγουμένως. Με τη βοήθεια του υποσυστήματος διαπροσωπείας χρήστη επιλέγεται το αρχείο το οποίο περιέχει την ερώτηση που πρέπει να εκτελεστεί. Το υποσύστημα ανάγνωσης ερώτησης από αρχείο αναλαμβάνει τον έλεγχο της σωστής ή όχι σύνταξης της ερώτησης καθώς και τη συλλογή των απαραίτητων παραμέτρων για την εκτέλεση της. Το υποσύστημα ανάγνωσης TSR σχημάτων αναλαμβάνει την προσπέλαση των TSR σχημάτων που συμμετέχουν στην ερώτηση. Στη συνέχεια με τη βοήθεια του υποσυστήματος ερωτήσεων – πράξεων εκτελείται η ερώτηση ενώ το υποσύστημα απεικόνισης TSR σχημάτων αναλαμβάνει την απεικόνιση του αποτελέσματος σε γραφική μορφή δέντρου. Όπως και στην εφαρμογή εισαγωγής ερώτησης παρέχεται η δυνατότητα για αποθήκευση του TSR σχήματος που προέκυψε από την ερώτηση είτε στη βάση δεδομένων είτε σε κάποιο XML αρχείο. Στο Σχήμα 4.10 φαίνεται το διάγραμμα ροής της εφαρμογής εισαγωγής ερώτησης από αρχείο.



Σχήμα 4.10

#### 4.2.10 Εφαρμογή εισαγωγής TSR σχημάτων από XML αρχείο

Η εφαρμογή αυτή είναι υπεύθυνη για τον έλεγχο ενός νέου XML αρχείου το οποίο περιέχει ένα ή περισσότερα TSR σχήματα, καθώς επίσης και για την εισαγωγή αυτών των TSR σχημάτων ώστε να είναι δυνατός ο χειρισμός τους.

Κάθε XML αρχείο που περιέχει TSRs πρέπει να οργανώνεται σύμφωνα με το DTD της εφαρμογής το οποίο είναι το εξής:

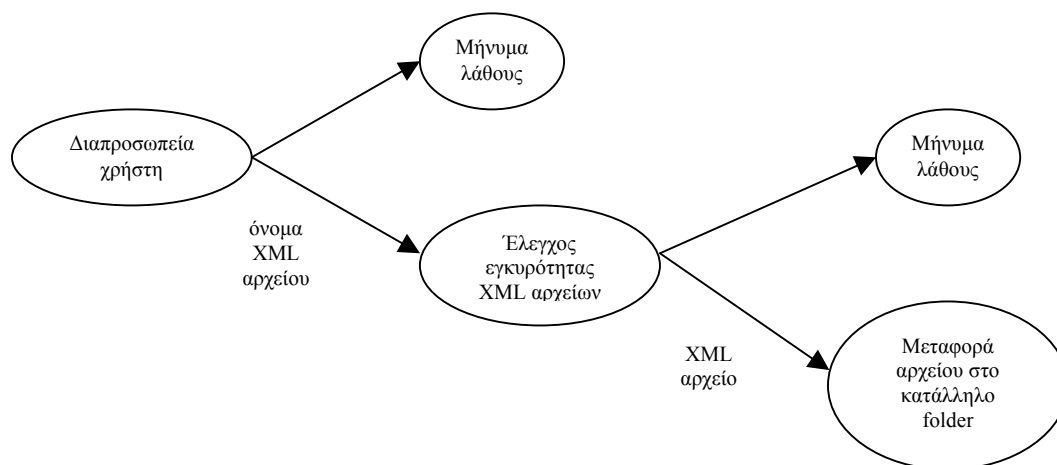
```
<!ELEMENT root (tsr)+>
<!ELEMENT tsr ((or)*,description,item)>
<!ATTLIST tsr name CDATA "">
<!ELEMENT or ((and)+)>
<!ELEMENT and (#PCDATA)>
<!ELEMENT description (#PCDATA)>
<!ELEMENT item ((attribute)*,(tuple)*)>
<!ATTLIST item name CDATA "">
<!ELEMENT attribute EMPTY>
<!ATTLIST attribute name CDATA "" type CDATA "">
<!ELEMENT tuple (#PCDATA)>
```

Σύμφωνα, λοιπόν, με το προηγούμενο DTD, κάθε XML αρχείο μπορεί να περιέχει ένα ή περισσότερα TSR σχήματα (*tsr*) τα οποία θα έχουν κάποιο όνομα. Κάθε TSR σχήμα αποτελείται από καμία ή περισσότερες OR συνιστώσες (*or*), από ακριβώς μία περιγραφή (*description*) και από ακριβώς έναν κόμβο – αντικείμενο (*item*). Κάθε OR συνιστώσα αποτελείται από ένα ή περισσότερα AND μονοπάτια (*and*), ενώ ο κόμβος – αντικείμενο έχει ένα όνομα (*name*) και αποτελείται από καμία ή περισσότερες ιδιότητες (*attribute*) και από καμία ή περισσότερες εγγραφές (*tuple*). Κάθε ιδιότητα έχει ένα όνομα (*name*) και έναν τύπο (*type*). Αν στο XML αρχείο υπάρχει κάποιο λάθος τότε το TSR σχήμα δεν εισάγεται. Σε αυτήν την περίπτωση είναι απαραίτητη η εμφάνιση κατάλληλων επεξηγηματικών μηνυμάτων ώστε ο χρήστης να μπορεί εύκολα να εντοπίσει το λάθος, να το διορθώσει και να επαναλάβει τη διαδικασία.

Συγκεκριμένα, η διαδικασία που ακολουθείται είναι η εξής: αρχικά μέσω του υποσυστήματος διαπροσωπείας χρήστη εισάγεται το όνομα του XML αρχείου που περιέχει ένα ή περισσότερα TSR σχήματα προς εισαγωγή. Στη συνέχεια το υποσύστημα ελέγχου εγκυρότητας XML αρχείων κάνει τον έλεγχο όπως εξηγήθηκε αναλυτικά σε προηγούμενη παράγραφο. Αν διαπιστωθεί από το υποσύστημα κάποιο λάθος σε κάποιο TSR σχήμα του XML αρχείου τότε εμφανίζονται στην οθόνη του χρήστη κατάλληλα επεξηγηματικά μηνύματα. Αν δεν προκύψει κάποιο λάθος, το XML αρχείο περιέχει TSR σχήματα σύμφωνα με το DTD της εφαρμογής



οπότε και μεταφέρεται στο φάκελο στον οποίο βρίσκονται όλα τα έγκυρα XML αρχεία. Στο Σχήμα 4.11 φαίνεται το διάγραμμα ροής της εφαρμογής εισαγωγής νέου XML αρχείου.



Σχήμα 4.11

#### 4.2.11 Εφαρμογή εισαγωγής TSR σχημάτων από RDFS αρχείο

Η εφαρμογή αυτή παρέχει στο χρήστη τη δυνατότητα να δημιουργήσει νέα TSR σχήματα από τις ιεραρχίες που υπάρχουν σε ένα RDFS αρχείο. Υπενθυμίζεται ότι τα TSR σχήματα που δημιουργούνται αποτελούνται από μία OR συνιστώσα με ένα μονοπάτι, ενώ ο κόμβος – αντικείμενο τους είναι κενός (δεν περιέχει ούτε ιδιότητες ούτε εγγραφές).

Παρακάτω δίνεται ένα κομμάτι του περιεχομένου ενός RDFS αρχείου. Το κομμάτι αυτό δεν είναι ολοκληρωμένο αφού ένα RDFS αρχείο περιέχει και άλλα στοιχεία. Όμως για το συγκεκριμένο σύστημα και το σκοπό που θέλουμε να επιτελέσουμε μέσω του RDFS σχήματος ενδιαφερόμαστε μόνο για τα tags 'rdfs:Class' και 'rdfs:subClassOf' προκειμένου να προκύψουν οι ιεραρχίες που χρειαζόμαστε:

```

<?xml version="1.0"?>

<rdf:RDF xml:lang="en"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema#" >

<rdfs:Class rdf:ID="Root"/>
<rdfs:Class rdf:ID="Artist"/>

<rdfs:Class rdf:ID="Artifact"/>
<rdfs:Class rdf:ID="Style"/>

```

```

<rdfs:Class rdf:ID="Museum"/>

<rdfs:Class rdf:ID="Sculptor">
  <rdfs:subClassOf rdf:resource="#Artist"/>
</rdfs:Class>

<rdfs:Class rdf:ID="Painter">
  <rdfs:subClassOf rdf:resource="#Artist"/>
</rdfs:Class>

<rdfs:Class rdf:ID="Cubist">
  <rdfs:subClassOf rdf:resource="#Painter"/>
</rdfs:Class>

<rdfs:Class rdf:ID="Flemish">
  <rdfs:subClassOf rdf:resource="#Painter"/>
</rdfs:Class>

<rdfs:Class rdf:ID="Sculpture">
  <rdfs:subClassOf rdf:resource="#Artifact"/>
</rdfs:Class>

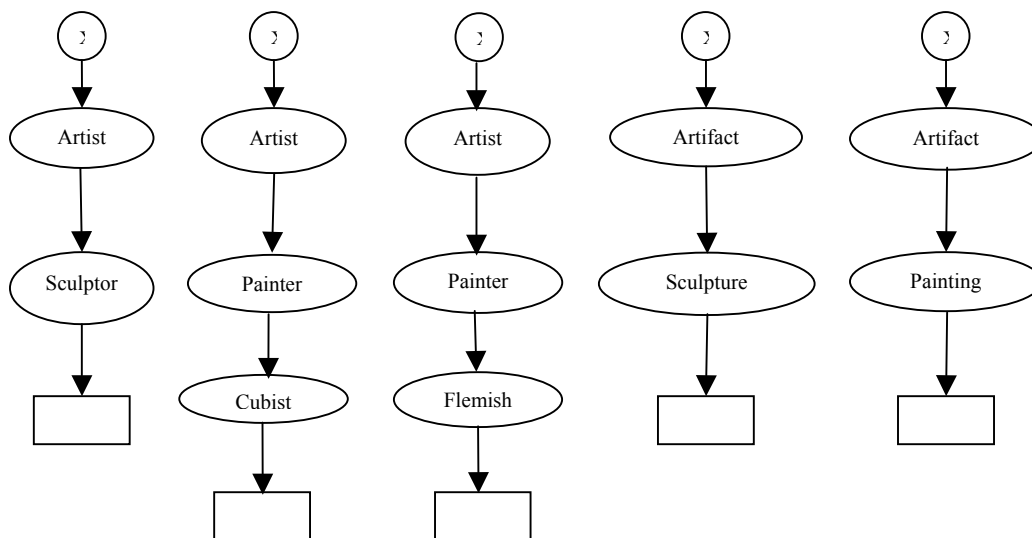
<rdfs:Class rdf:ID="Painting">
  <rdfs:subClassOf rdf:resource="#Artifact"/>
</rdfs:Class>

<rdf:Property rdf:ID="creates">
  <rdfs:domain rdf:resource="#Artist"/>
  <rdfs:range rdf:resource="#Artifact"/>
</rdf:Property>

<rdf:Property rdf:ID="paints">
  <rdfs:domain rdf:resource="#Painter"/>
  <rdfs:range rdf:resource="#Painting"/>
  <rdfs:subPropertyOf rdf:resource="#creates"/>
</rdf:Property>
.
.
.
.
</rdf:RDF>

```

Τα TSR σχήματα που προκύπτουν από το προηγούμενο RDFS αρχείο φαίνονται στο Σχήμα 4.12:

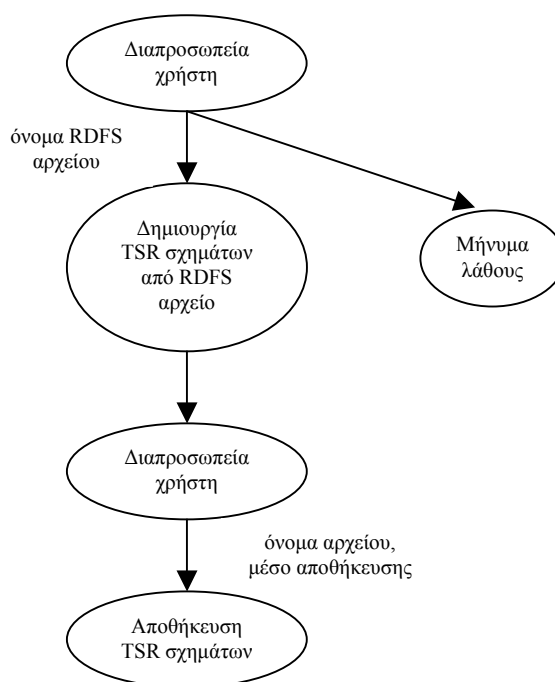


**Σχήμα 4.12**

Οι κόμβοι – αντικείμενα των TSR σχημάτων που δημιουργούνται από RDFS αρχεία δεν περιέχουν ούτε ιδιότητες ούτε εγγραφές. Αυτό συμβαίνει γιατί στο RDFS αρχείο εντοπίζουμε και χρησιμοποιούμε μόνο τα μονοπάτια που οδηγούν στους κόμβους – αντικείμενα (εντοπίζουμε δηλαδή μόνο τις ιεραρχίες). Οι ιδιότητες και οι εγγραφές του κόμβου – αντικειμένου ενός TSR σχήματος μπορούν να προστεθούν αργότερα με τη βοήθεια της προσθήκης εγγραφών σε TSR σχήματα, η οποία θα αναλυθεί στη συνέχεια. Τα TSR σχήματα που δημιουργούνται παίρνουν τα ονόματά τους από τον τελευταίο κόμβο της ιεραρχίας τους. Δηλαδή στα παράδειγμα του Σχήματος 4.12 τα ονόματα των TSR σχημάτων που δημιουργούνται είναι: `Sculptor`, `Cubist`, `Flemish`, `Sculpture` και `Painting` αντίστοιχα. Τα ονόματα των κόμβων – αντικειμένων των νέων TSR σχημάτων είναι ίδια με τα ονόματα των TSR σχημάτων με το πρόθεμα ‘\_’. Δηλαδή τα ονόματα των κόμβων αντικειμένων των TSR σχημάτων του Σχήματος 4.12 είναι: `_Sculptor`, `_Cubist`, `_Flemish`, `_Sculpture` και `_Painting` αντίστοιχα. Στα νέα TSR σχήματα που δημιουργούνται δε δίνεται καμία περιγραφή. Σημειώνεται ότι τόσο τα ονόματα των TSR σχημάτων όσο και η περιγραφή τους και τα ονόματα των κόμβων – αντικειμένων τους που δίνονται αυτόματα από το σύστημα, μπορούν να αλλάξουν με τη βοήθεια της εφαρμογής αλλαγής χαρακτηριστικών στοιχείων TSR σχημάτων η οποία θα αναλυθεί σε επόμενη ενότητα.

Η διαδικασία που ακολουθείται είναι η εξής: αρχικά μέσω του υποσυστήματος διαπροσωπείας χρήστη επιλέγεται το RDFS αρχείο από το οποίο θα εξαχθούν τα νέα TSR σχήματα. Αν δεν υπάρχουν RDFS αρχεία τότε εμφανίζεται ένα κατάλληλο επεξηγηματικό μήνυμα. Στη συνέχεια με τη βοήθεια του υποσυστήματος δημιουργίας TSR σχημάτων από RDFS αρχείο δημιουργούνται τα νέα TSR σχήματα. Μέσω του υποσυστήματος

διαπροσωπείας χρήστη επιλέγεται το αποθηκευτικό μέσο στο οποίο θα αποθηκευτούν τα νέα TSR σχήματα ενώ συγχρόνως εισάγεται και το όνομα του αρχείου τους. Τέλος το υποσύστημα αποθήκευσης TSR σχημάτων αναλαμβάνει την αποθήκευση των νέων TSRs στο αποθηκευτικό μέσο που επιλέχθηκε. Στο Σχήμα 4.13 φαίνεται το διάγραμμα ροής της εφαρμογής εισαγωγής TSR σχημάτων από κάποιο RDF αρχείο:

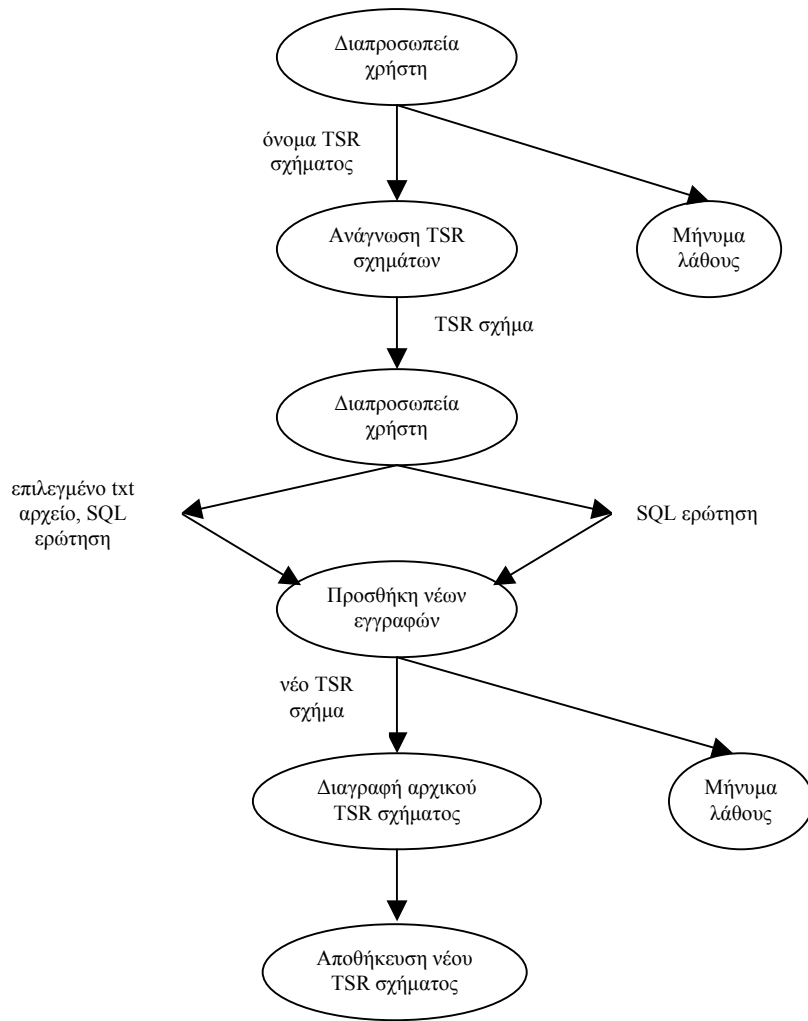


Σχήμα 4.13

#### 4.2.12 Εφαρμογή προσθήκης εγγραφών TSR σχημάτων

Η εφαρμογή αυτή δίνει στο χρήστη τη δυνατότητα να εισάγει νέες εγγραφές σε TSR σχήματα οι κόμβοι – αντικείμενα των οποίων περιέχουν ήδη κάποιες ιδιότητες και ενδεχομένως κάποιες εγγραφές, αλλά και σε TSR σχήματα οι κόμβοι – αντικείμενα των οποίων δεν περιέχουν καμία ιδιότητα και καμία εγγραφή (όπως είναι τα TSR σχήματα που προκύπτουν από κάποιο RDFS αρχείο). Συγκεκριμένα, αρχικά μέσω του υποσυστήματος διαπροσωπείας χρήστη επιλέγεται το TSR σχήμα στο οποίο ο χρήστης επιθυμεί να προσθέσει κάποιες εγγραφές. Το υποσύστημα ανάγνωσης TSR σχημάτων αναλαμβάνει την ανάγνωση του TSR σχήματος που επιλέχθηκε από το αποθηκευτικό μέσο στο οποίο αυτό βρίσκεται. Στη συνέχεια μέσω του υποσυστήματος διαπροσωπείας χρήστη, δίνεται η δυνατότητα στο χρήστη να επιλέξει αν τις εγγραφές που επιθυμεί να προσθέσει θα τις πάρει μέσα από ένα txt αρχείο το

οποίο πρέπει να έχει συγκεκριμένη μορφή ή μέσα από κάποιον πίνακα της βάσης δεδομένων ο οποίος έχει δημιουργηθεί για αυτόν τον σκοπό. Στην περίπτωση που ο χρήστης επιλέξει να διαβάσει τα δεδομένα από κάποιο txt αρχείο, τότε πρέπει να δώσει το όνομα του txt αρχείου καθώς επίσης και την SQL ερώτηση με βάση την οποία θα επιλεγθούν οι εγγραφές και ενδεχομένως οι ιδιότητες του κόμβου – αντικειμένου του TSR σχήματος. Σε αυτήν την περίπτωση όλο το txt αρχείο αποθηκεύεται στον πίνακα DATA της βάσης δεδομένων για να είναι πιο γρήγορη και αποτελεσματική η εύρεση των δεδομένων που ζητά ο χρήστης. Αν αντίθετα ο χρήστης δεν επιλέξει να εισάγει τα δεδομένα από κάποιο txt αρχείο, αλλά προτιμήσει να τα διαβάσει από έναν συγκεκριμένο πίνακα της βάσης δεδομένων στον οποίο έχει εισάγει διάφορες πληροφορίες, τότε θα του ζητηθεί απλά η SQL ερώτηση με τη βοήθεια της οποίας θα γίνει η επιλογή των εγγραφών και ενδεχομένως των ιδιοτήτων του κόμβου – αντικειμένου του TSR σχήματος. Το υποσύστημα προσθήκης εγγραφών TSR σχημάτων αναλαμβάνει στη συνέχεια να εισάγει τις νέες εγγραφές στον κόμβο – αντικείμενο του επιλεγμένου TSR σχήματος. Σημειώνεται ότι αν προκύψει οποιοδήποτε σφάλμα το σύστημα θα ενημερώσει το χρήστη με κατάλληλα επεξηγηματικά μηνύματα. Μερικά από τα σφάλματα που ενδέχεται να προκύψουν είναι: προσπάθεια εισαγωγής νέων εγγραφών σε TSR σχήμα ο κόμβος – αντικειμένου του οποίου αποτελείται από ιδιότητες που δεν είναι ακριβώς όμοιες στον αριθμό, στο όνομα και στον τύπο με αυτές που έχει επιλέξει ο χρήστης, εσφαλμένη εισαγωγή SQL ερώτησης ή σφάλμα στη δημιουργία του txt αρχείου που επιλέχθηκε για την εύρεση των ιδιοτήτων και των εγγραφών. Αν δεν προκύψει κάποιο σφάλμα το αρχικό σύστημα διαγράφεται από το αποθηκευτικό μέσο στο οποίο βρισκόταν μέσω του υποσυστήματος διαγραφής TSR σχημάτων. Τέλος το νέο TSR σχήμα που προκύπτει αποθηκεύεται μέσω του υποσυστήματος αποθήκευσης TSR σχημάτων στο αποθηκευτικό μέσο στο οποίο βρισκόταν και το αρχικό TSR σχήμα. Στο Σχήμα 4.14 που ακολουθεί φαίνεται το διάγραμμα ροής της εφαρμογής προσθήκης εγγραφών TSR σχημάτων:



**Σχήμα 4.14**

# 5

## *Υλοποίηση*

Στο κεφάλαιο αυτό αναφέρονται τα προγραμματιστικά εργαλεία που χρησιμοποιήθηκαν για την ανάπτυξη του συστήματος. Επίσης δίνονται ορισμένες λεπτομέρειες της υλοποίησης της βάσης δεδομένων, του μεταγλωττιστή, του υποσυστήματος απεικόνισης και των αρχείων που χρησιμοποιούνται από το σύστημα, καθώς επίσης και η περιγραφή των κλάσεων που υλοποιούν το σύστημα.

### *5.1 Πλατφόρμες και προγραμματιστικά εργαλεία*

Για την υλοποίηση του προγράμματος του συστήματος χρησιμοποιήθηκε η γλώσσα προγραμματισμού Java. Η έκδοση που χρησιμοποιήθηκε ήταν η Java 2 SDK Standard Edition 1.4.1. Επιπλέον πακέτα – βιβλιοθήκες που χρησιμοποιήθηκαν περιέχουν κλάσεις για τη δημιουργία και την ανάγνωση XML εγγράφων καθώς και οδηγούς για τη δημιουργία συνδέσεων με τη βάση δεδομένων. Συγκεκριμένα χρησιμοποιήθηκαν JDBC οδηγό πρόσβασης σε βάση δεδομένων και τα πακέτα dom4j και sax για τη διαχείριση των XML αρχείων. Ειδικά για την υλοποίηση του μεταγλωττιστή χρησιμοποιήθηκε η γεννήτρια συντακτικών αναλυτών JavaCC 3.2. Για τη βάση δεδομένων του συστήματος χρησιμοποιήθηκε η MySQL 4.0.17.

## 5.2 Λεπτομέρειες Υλοποίησης

Στην ενότητα αυτή παρουσιάζονται οι λεπτομέρειες υλοποίησης του συστήματος ξεκινώντας από τη βάση δεδομένων και το σχήμα της. Έπειτα περιγράφεται ο τρόπος δημιουργίας του μεταγλωττιστή, η σύνταξη και ο έλεγχος των ερωτήσεων που μπορούν να διατυπωθούν. Επίσης γίνεται σύντομη περιγραφή του τρόπου απεικόνισης των TSR σχημάτων καθώς επίσης και των αρχείων που χρησιμοποιούνται από το σύστημα. Τέλος γίνεται μία σύντομη περιγραφή των κλάσεων που έχουν συνταχθεί για την υλοποίηση των εφαρμογών του συστήματος.

### 5.2.1 Βάση δεδομένων

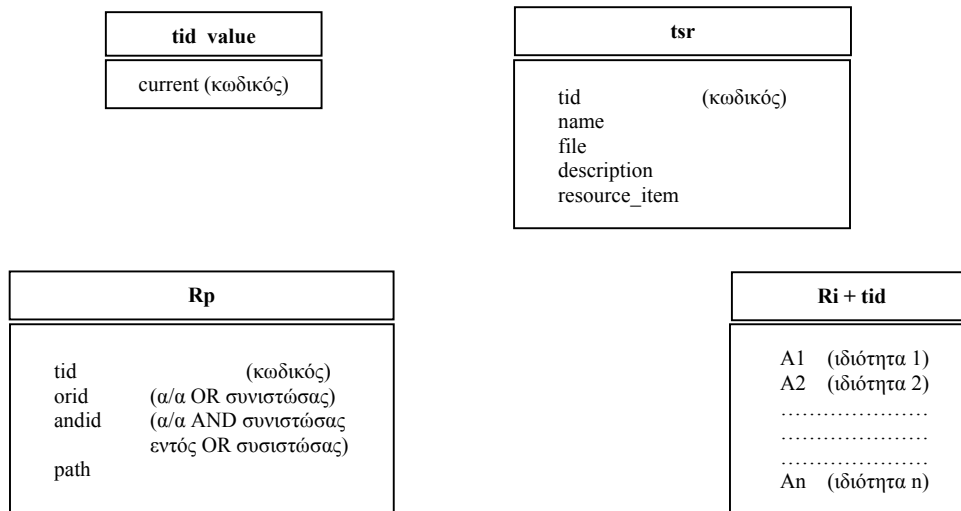
Ένα από τα δύο αποθηκευτικά μέσα του συστήματος είναι η βάση δεδομένων dbTSR στο σύστημα της MySQL. Όπως έχει αναφερθεί αναλυτικά στο προηγούμενο κεφάλαιο, σύμφωνα με το  $R_p - R_i$  μοντέλο για κάθε TSR σχήμα που αποθηκεύεται στη βάση δεδομένων φυλάσσονται το όνομα του, το όνομα του αρχείου από το οποίο προέρχεται, μία σύντομη περιγραφή του και το όνομα του κόμβου – αντικειμένου του. Επίσης κρατούνται πληροφορίες για το σύνολο των μονοπατιών του TSR σχήματος, καθώς επίσης και πληροφορίες για τις ιδιότητες και τις εγγραφές του κόμβου – αντικειμένου. Για τη φύλαξη των παραπάνω πληροφοριών απαιτούνται οι εξής τρεις πίνακες: ο  $tsr$  ο οποίος κρατά πληροφορίες για το TSR σχήμα, ο  $R_p$  ο οποίος κρατά πληροφορίες για τα μονοπάτια του TSR σχήματος και ο  $R_i + tid$  ο οποίος κρατάει πληροφορίες για τις ιδιότητες και τις εγγραφές του κόμβου – αντικειμένου.  $tid$  είναι ένας μοναδικός ακέραιος αριθμός ο οποίος αντιπροσωπεύει τον κωδικό του TSR σχήματος στη βάση δεδομένων. Για την γρήγορη εύρεση του κωδικού ενός TSR σχήματος που πρόκειται να αποθηκευτεί στη βάση δεδομένων απαιτείται ένας βοηθητικός πίνακας, ο πίνακας  $tid\_value$  στον οποίο φυλάσσεται ο κωδικός του τελευταίου TSR σχήματος που έχει αποθηκευτεί στη βάση δεδομένων. Έτσι κάθε φορά που ένα νέο TSR σχήμα εισάγεται στη βάση δεδομένων παραχωρείται σε αυτό ένας κωδικός ο οποίος προκύπτει από την αύξηση κατά ένα του περιεχομένου της εγγραφής του πίνακα  $tid\_value$ . Στο Σχήμα 5.1 φαίνονται όλοι οι πίνακες της βάσης δεδομένων dbtsr με τα γνωρίσματα τους. Στη συνέχεια παρατίθεται ο SQL κώδικας ο οποίος δημιουργεί όλους τους πίνακες της βάσης δεδομένων εκτός από τον  $R_i + tid$  ο οποίος δημιουργείται κάθε φορά που ένα TSR σχήμα εισάγεται στη βάση δεδομένων:



```

DROP DATABASE IF EXISTS dbTSR;
CREATE DATABASE dbTSR;
USE dbTSR;
CREATE TABLE tid_value (
    current          integer          NOT NULL );
CREATE TABLE tsr (
    tid              integer          NOT NULL,
    name             text             NOT NULL,
    file             text             NOT NULL,
    description      text,
    resource_item    text,
    PRIMARY KEY(tid));
CREATE TABLE rp (
    tid              integer          NOT NULL,
    orid             integer          NOT NULL,
    andid            integer          NOT NULL,
    path             text             NOT NULL );
INSERT INTO tid_value VALUES (0);

```



Σχήμα 5.1

### 5.2.2 Μεταγλωττιστής

Ο μεταγλωττιστής υλοποιήθηκε με χρήση της γεννήτριας συντακτικών αναλυτών (parsers) JavaCC (Java Compiler Compiler). Με χρήση του JavaCC ορίζεται αρχικά η γραμματική της γλώσσας και στη συνέχεια παράγονται αυτόματα οι Java κλάσεις που υλοποιούν το συντακτικό αναλυτή που αναγνωρίζει τη γλώσσα αυτή. Ο ορισμός της γραμματικής και η διαχείριση της γλώσσας γίνεται στο πηγαίο αρχείο με επέκταση `.jj`, στο

οποίο γράφονται εκφράσεις της γλώσσας προγραμματισμού Java, αλλά και κανονικές (regular expressions) και ειδικές εκφράσεις που ορίζει το εργαλείο JavaCC.

### 5.2.2.1 Γραμματική της γλώσσας

Η δημιουργία του συντακτικού αναλυτή ξεκινά από τον ορισμό της γλώσσα που αναλύει. Έτσι, λοιπόν, αρχικά ορίζονται οι εκφράσεις και οι λέξεις που διαχειρίζεται η γλώσσα.

Για το σύστημα που υλοποιήσαμε χρησιμοποιήσαμε την γραμματική της γλώσσας TreeSQuerL [28] έχοντας κάνει τις παρακάτω αλλαγές στις λέξεις και στις εκφράσεις της γλώσσας:

- Έχουν καταργηθεί οι εντολές 'LOAD', 'SAVE2DB', 'EXIT' και 'HELP'.
- Έχει καταργηθεί η έκφραση 'AS'.
- Η έκφραση <PATH\_VAR> έχει αλλαχθεί ως εξής:  
    < PATH\_VAR:    "\$\_" | (" \$" | "#") ( ["1"-"9"] )+ >
- Οι εκφράσεις των ερωτήσεων – πράξεων είναι:

#### Επιλογή

```
<SELECT>    "(" <ATTRIBUTE_CONDITION> ")" (" <PATH_CONDITION> ") (" <TSR>") "
```

#### Προβολή

```
<PROJECT>    "(" <ATTRIBUTE_LIST> ")" (" <PATH_LIST> ") (" <TSR>") "
```

#### Καρτεσιανό γινόμενο

```
<CARTESIAN_PRODUCT>    "(" <TSR> ")" (" <TSR> ") "
```

#### Ένωση

```
<UNION>    "(" <TSR> ")" (" <TSR> ") "
```

#### Τομή

```
<INTERSECTION>    "(" <TSR> ")" (" <TSR> ") "
```

## Διαφορά

<DIFFERENCE> “ (“ <TSR> ”) (“ <TSR> ”) ”

Αυτή η σύνταξη των ερωτήσεων – πράξεων πρέπει να ακολουθείται κάθε φορά που ο χρήστης γράφει μία νέα ερώτηση σε ένα αρχείο. Αν δεν ακολουθείται με ακρίβεια το συντακτικό που ορίζεται με αυτή τη γραμματική, η διαδικασία της ανάγνωσης σταματά και εμφανίζονται στην οθόνη κατάλληλα επεξηγηματικά μηνύματα.

### 5.2.2.2 Συντακτική ανάλυση – Χειρισμός εντολών

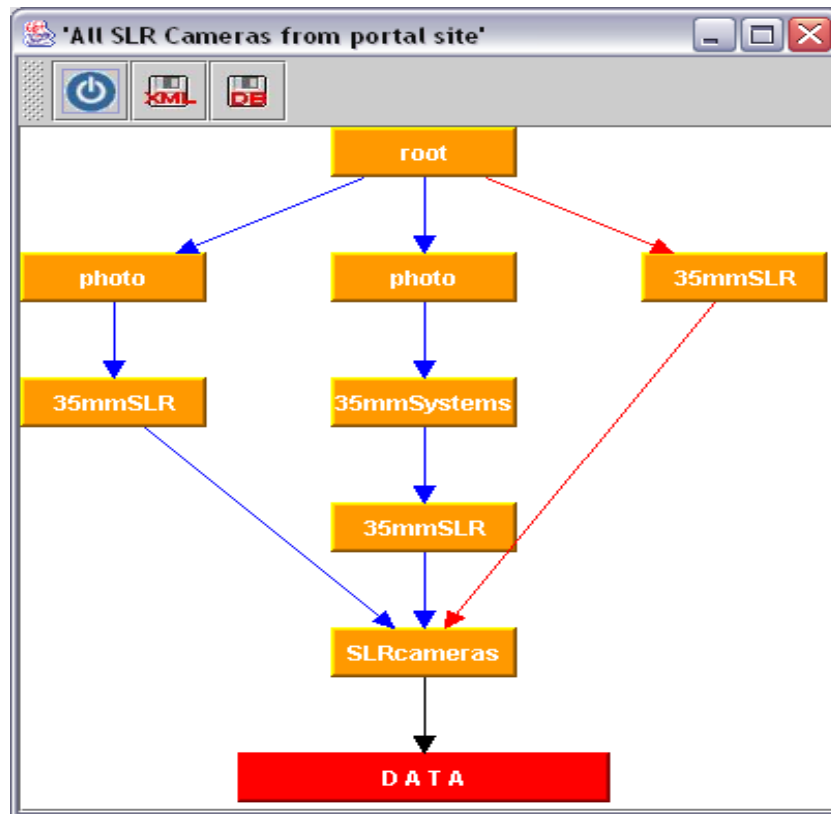
Ο πυρήνας του συντακτικού αναλυτή είναι η συνάρτηση `parseExpression`, η οποία αναλαμβάνει να αναγνωρίσει την εντολή που υπάρχει μέσα στο `.qry` αρχείο και να καλέσει την αντίστοιχη συνάρτηση για κάθε ερώτηση – πράξη. Κάθε εντολή του μεταγλωττιστή υλοποιείται από μία συνάρτηση, στο σώμα της οποίας καλούνται συναρτήσεις που με περαιτέρω ανάλυση συλλέγουν τις παραμέτρους της εντολής. Στη συνέχεια καλούνται συναρτήσεις από το υποσύστημα ερωτήσεων – πράξεων οι οποίες και εκτελούν την ερώτηση.

### 5.2.3 Υποσύστημα απεικόνισης TSR σχημάτων

Κάθε TSR σχήμα προς απεικόνιση αποθηκεύεται προσωρινά σε ένα συγκεκριμένο XML αρχείο από το οποίο το υποσύστημα αυτό διαβάζει τα απαραίτητα στοιχεία για την απεικόνιση. Εάν το TSR σχήμα που απεικονίζεται δε βρίσκεται στο αποθηκευτικό μέσο (δηλαδή είναι το αποτέλεσμα μιας ερώτησης ή μιας πράξης), το γραφικό περιβάλλον παρέχει στο χρήστη τη δυνατότητα αποθήκευσης αυτού του TSR σχήματος είτε στη βάση δεδομένων είτε σε κάποιο XML αρχείο.

Στο Σχήμα 5.2 παρουσιάζεται το γραφικό περιβάλλον που απεικονίζει τα TSR σχήματα. Στο πάνω μέρος του σχήματος διακρίνονται οι επιλογές για κλείσιμο του παραθύρου, για αποθήκευση σε XML αρχείο και για αποθήκευση στη βάση δεδομένων. Για την καλύτερη γραφική αναπαράσταση, τα AND μονοπάτια παριστάνονται με το ίδιο χρώμα ακμών που συνδέουν τα μονοπάτια αυτά με τη ρίζα και τον κόμβο – αντικείμενο. Ο κόμβος με τον οποίο συνδέονται οι ακμές όλων των μονοπατιών παριστάνει τον κόμβο – αντικείμενο του TSR σχήματος, ενώ ο τελευταίος κόμβος (κόκκινος κόμβος με όνομα `DATA`) αντιστοιχεί στο περιεχόμενο του κόμβου – αντικειμένου, δηλαδή τις ιδιότητες και τις εγγραφές. Πατώντας, λοιπόν, πάνω σε αυτόν τον κόμβο με τη βοήθεια του ποντικιού, εμφανίζεται ένας πίνακας ο

οποίος περιέχει όλες τις ιδιότητες και τις εγγραφές του κόμβου – αντικειμένου και για την TSR του Σχήματος 5.2 έχει τη μορφή που φαίνεται στο σχήμα 5.3.



Σχήμα 5.2

brand	model	price
Canon	EOS 3	990
Canon	EOS Rebel Ti	264.95
Canon	EOS Elan 7	424.95
Minolta	Maxxum 3 QD	139.95
Minolta	Maxxum 4 QD	194.95
Minolta	Maxxum 7 QD	599.99
Nikon	N65	205
Pentax	ZX M	149.95

Σχήμα 5.3

Στην πρώτη γραμμή αυτού του πίνακα εμφανίζονται οι ιδιότητες του κόμβου – αντικειμένου, ενώ στις επόμενες γραμμές φαίνονται οι εγγραφές του.

#### 5.2.4 XML, RDFS, TXT, QRY αρχεία

Όλα τα XML αρχεία που περιέχουν TSR σχήματα, πριν ελεγχθούν από την εφαρμογή εισαγωγής νέου XML αρχείου, βρίσκονται αποθηκευμένα στον φάκελο `'org/_3pq/jgraph/my_diplo/sources/insert_files'`. Όταν ένα XML αρχείο ελεγχθεί με τη βοήθεια της εφαρμογής εισαγωγής νέου TSR σχήματος και δεν προκύψει κανένα σφάλμα, τότε ολόκληρο το XML αρχείο μεταφέρεται στον φάκελο `'org/_3pq/jgraph/my_diplo/sources'`. Σε αυτόν τον φάκελο βρίσκονται όλα τα XML αρχεία στα οποία είναι αποθηκευμένα TSR σχήματα και τα οποία χρησιμοποιεί το σύστημα. Κάθε XML αρχείο πρέπει να έχει ως πρώτη δήλωση την εξής: `<!DOCTYPE root SYSTEM "dtdtest.dtd">`, η οποία δηλώνει ότι το XML αρχείο ακολουθεί ένα συγκεκριμένο DTD το οποίο υπάρχει στο αρχείο `dtdtest.dtd`.

Τόσο τα RDFS όσο και τα TXT αρχεία που χρησιμοποιούνται από τις εφαρμογές εισαγωγής TSR σχημάτων από RDFS αρχεία και προσθήκης εγγραφών TSR σχημάτων αντίστοιχα βρίσκονται αποθηκευμένα στον φάκελο `'org/_3pq/jgraph/my_diplo/sources/insert_files'`. Τα TSR σχήματα που προκύπτουν με τη βοήθεια του υποσυστήματος δημιουργίας TSR σχημάτων από RDFS αρχεία αποθηκεύονται, εκτός και αν ο χρήστης επιλέξει να τα αποθηκεύσει στη βάση δεδομένων, στον φάκελο στον οποίο υπάρχουν όλα τα XML αρχεία τα οποία χρησιμοποιεί η εφαρμογή, δηλαδή στον `'org/_3pq/jgraph/my_diplo/sources'`. Τα TXT αρχεία περιέχουν δεδομένα για τη δημιουργία νέων εγγραφών και χρησιμοποιούνται από την εφαρμογή προσθήκης εγγραφών TSR σχημάτων. Εκτός από δεδομένα για τις εγγραφές περιέχουν και τα ονόματα των νέων ιδιοτήτων. Ένα TXT αρχείο που χρησιμοποιείται για αυτό το σκοπό πρέπει να έχει συγκεκριμένη μορφή η οποία αναγνωρίζεται από το σύστημα. Ένα παράδειγμα ενός TXT αρχείου φαίνεται παρακάτω:

```
first_name#last_name#Artist#Artifact#name
Pablo#Picasso#Cubist#Painting#guernica
Pablo#Picasso#Cubist#Painting#woman
August      Rodin#Sculptor#Sculpture#Crucifixion
August      Rodin#Sculptor#Sculpture#Thinker
Michelangelo#Buonarroti#Sculptor#Sculpture#Theslave
Michelangelo#Buonarroti#Sculptor#Sculpture#Descent
```

Η πρώτη γραμμή του TXT αρχείου περιέχει τα ονόματα των ιδιοτήτων ενώ οι υπόλοιπες γραμμές αποτελούν δεδομένα για κάθε ιδιότητα. Για παράδειγμα, τα Pablo, August, Michelangelo αποτελούν δεδομένα της ιδιότητας `first_name` ενώ τα `guernica`, `woman`,

Crucifixion, Thinker, Theslave, Descent είναι δεδομένα της ιδιότητας name. Σε κάθε γραμμή τα στοιχεία χωρίζονται μεταξύ τους με '#'.

Τα QRY αρχεία τα οποία χρησιμοποιούνται από την εφαρμογή εισαγωγής ερώτησης από αρχείο βρίσκονται αποθηκευμένα στον φάκελο 'org/\_3pq/jgrapht/my\_diplo/sources/qry\_files'.

### 5.2.5 Περιγραφή κλάσεων

Στην ενότητα αυτή παρατίθεται οι κλάσεις που υλοποιούν τις εφαρμογές του συστήματος. Για κάθε κλάση παρουσιάζονται τα πεδία και οι μέθοδοι της, καθώς επίσης και μία σύντομη περιγραφή της λειτουργίας της.

#### 5.2.3.1 *public class Attribute*

Η κλάση Attribute υλοποιεί μία ιδιότητα του κόμβου – αντικειμένου.

##### Πεδία

- public String name  
Το όνομα της ιδιότητας.
- public String type  
Ο τύπος της ιδιότητας. Μπορεί να είναι text για κείμενο ή integer,float για αλφαριθμητικό.

##### Μέθοδοι

- public Attribute(String name, String type)  
Ο κατασκευαστής αντικειμένων της κλάσης. Οι παράμετροι name και type αρχικοποιούν τα πεδία.

#### 5.2.3.2 *public class Path*

Η κλάση Path ορίζει το AND μονοπάτι που οδηγεί σε κάποιο κόμβο – αντικείμενο.

##### Πεδία

- private String image  
Απεικόνιση του μονοπατιού image σε μορφή String.

##### Μέθοδοι

- public boolean equal(Path path)  
Τελεστής σύγκρισης μονοπατιών ισότητα.
- public boolean strictSubset(Path path)  
Τελεστής σύγκρισης μονοπατιών αυστηρό υποσύνολο.
- public boolean strictHyperset(Path path)  
Τελεστής σύγκρισης μονοπατιού αυστηρό υπερσύνολο.
- public boolean looseSubset(Path path)

- Τελεστής σύγκρισης μονοπατιών χαλαρό υποσύνολο.  
public boolean looseHyperset(Path path)
- Τελεστής σύγκρισης μονοπατιών χαλαρό υπερσύνολο.  
public String toString()
- Προσπέλαση String απεικόνισης του μονοπατιού.  
public Path(String image)
- Κατασκευαστής αντικειμένων της κλάσης. Η παράμετρος image αρχικοποιεί το αντίστοιχο πεδίο της κλάσης.

### 5.2.3.3 public class OR

Η κλάση OR υλοποιεί την OR συνιστώσα κάθε TSR σχήματος.

#### Πεδία

- private Vector vAND  
Διάνυσμα που περιέχει τα AND μονοπάτια που αποτελούν τη συνιστώσα. Αποτελείται από αντικείμενα της κλάσης Path.

#### Μέθοδοι

- public void addAND(Path path)  
Προσθήκη του AND μονοπατιού path στην OR συνιστώσα.
- public Path getAND(int index)  
Προσπέλαση του index αριστερότερου AND μονοπατιού της OR συνιστώσας.
- public int countANDs()  
Καταμέτρηση περιεχόμενων AND μονοπατιών της OR συνιστώσας.
- public OR()  
Κατασκευαστής αντικειμένων της κλάσης.

### 5.2.3.4 public class Item

Η κλάση Item ορίζει τον κόμβο – αντικείμενο μιας TSR.

#### Πεδία

- public Vector vAttributes  
Διάνυσμα που περιέχει τις ιδιότητες του κόμβου - αντικειμένου. Αποτελείται από αντικείμενα της κλάσης Attribute.
- public Vector vTuples  
Διάνυσμα που περιέχει τις εγγραφές του κόμβου – αντικειμένου.

#### Μέθοδοι

- public void addAttribute(Attribute attr)  
Προσθήκη της ιδιότητας attr στον κόμβο - αντικείμενο.
- public void addTuple(String tuple)  
Προσθήκη της εγγραφής tuple στον κόμβο - αντικείμενο.
- public Attribute getAttribute(int index)  
Προσπέλαση της index ιδιότητας.
- public String getTuple(int index)  
Προσπέλαση της index εγγραφής.
- public int countAttributes()  
Καταμέτρηση των ιδιοτήτων του κόμβου – αντικειμένου.
- public int countTuples()  
Καταμέτρηση των εγγραφών του κόμβου – αντικειμένου.
- public int getAttributeIndexByName(String attrName)

- Εύρεση του αύξοντα αριθμού της ιδιότητας με όνομα attrName στον κόμβο – αντικείμενο.  
public boolean existAttributeByName(String attrName)  
Αναζήτηση της ιδιότητας με όνομα attrName στον κόμβο – αντικείμενο.
- public boolean existAttribute(Attribute attr)  
Αναζήτηση της ιδιότητας attr στον κόμβο – αντικείμενο.
- public boolean existTuple(String tuple)  
Αναζήτηση της εγγραφής tuple στον κόμβο – αντικείμενο.
- public boolean sameAttributes(TSR other)  
Σύγκριση των ιδιοτήτων του αντικειμένου με αυτές της other TSR. Στην περίπτωση που οι ιδιότητες είναι ακριβώς ίδιες (τόσο στο όνομα όσο και στον τύπο) επιστρέφεται true.
- public boolean differentAttributes(TSR other)  
Σύγκριση των ιδιοτήτων του αντικειμένου με αυτές της other TSR. Στην περίπτωση που έστω και μία ιδιότητα του αντικειμένου είναι διαφορετική από τις ιδιότητες της other επιστρέφεται true.
- public boolean differentAttributesByName(TSR other)  
Σύγκριση των ονομάτων των ιδιοτήτων του αντικειμένου με αυτές του αντικειμένου της other TSR. Στην περίπτωση που οι ιδιότητες είναι όλες διαφορετικές μεταξύ τους επιστρέφεται true.
- public Item()  
Κατασκευαστής αντικειμένων της κλάσης.

### 5.2.3.5 public class TSR

Η κλάση TSR ορίζει ένα σχήμα TSR.

#### Πεδία

- public String file  
Το όνομα του XML αρχείου από το οποίο προέρχεται το TSR σχήμα.
- public String name  
Το όνομα του TSR σχήματος.
- public String description  
Η περιγραφή του TSR σχήματος.
- public String resource\_item  
Το όνομα του κόμβου – αντικειμένου του TSR σχήματος.
- private Vector vOR  
Διάνυσμα που περιέχει τις OR συνιστώσες του TSR σχήματος. Αποτελείται από αντικείμενα της κλάσης OR.
- public Item item  
Ο κόμβος – αντικείμενο του TSR σχήματος.

#### Μέθοδοι

- public String toString()  
Απεικόνιση του TSR σχήματος σύμφωνα με το σχήμα του XML αρχείου που χρησιμοποιείται ως αποθηκευτικό μέσο.
- public void addOR(OR or)  
Προσθήκη της OR or συνιστώσας στο TSR σχήμα.
- public OR getOR(int index)  
Προσπέλαση της index OR συνιστώσας του TSR σχήματος.
- public int countORs()  
Καταμέτρηση των OR συνιστωσών του TSR σχήματος.
- public OR getMaxOR()  
Προσπέλαση της μεγαλύτερης OR συνιστώσας του TSR σχήματος, δηλαδή αυτής με τα περισσότερα AND μονοπάτια.
- public boolean existANDindex(int index)  
Αναζήτηση του index αριστερότερου AND μονοπατιού. Με τη μέθοδο αυτή διαπιστώνεται αν το index αριστερότερο μονοπάτι υπάρχει στην OR συνιστώσα με τα περισσότερα μονοπάτια.
- public TSR()



Κατασκευαστής αντικειμένων της κλάσης.

### 5.2.3.6 *public class condition*

Στη σύνταξη των ερωτήσεων επιλογής δίνεται μια σειρά από συνθήκες ιδιοτήτων και μονοπατιών που αποτιμώνται πάνω στις εγγραφές και τα μονοπάτια του κόμβου – αντικειμένου αντίστοιχα. Η κλάση που ορίζει αυτές τις συνθήκες είναι η `Condition`. Η μορφή των συνθηκών μπορεί να είναι μεταβλητή τελεστής μεταβλητή ή μεταβλητή τελεστής τιμή, όπου μεταβλητή είναι είτε μεταβλητή ιδιότητας είτε μεταβλητή μονοπατιών.

#### **Πεδία**

- `public String term1`  
Πρώτος όρος της συνθήκης. Είναι πάντα μεταβλητή.
- `public String term2`  
Δεύτερος όρος της συνθήκης. Ανάλογα με τον τύπο της συνθήκης είναι είτε μεταβλητή είτε τιμή.
- `public boolean type`  
Ο τύπος της συνθήκης. Είναι `true` αν πρόκειται για σύγκριση μεταβλητών και `false` αν συγκρίνεται μεταβλητή με τιμή.
- `public String operator`  
Ο τελεστής της συνθήκης. Είναι είτε τελεστής σύγκρισης ιδιοτήτων είτε τελεστής σύγκρισης μονοπατιών.
- `private String previousBitwise`  
Το πεδίο αυτό παίρνει τις τιμές `AND` ή `OR` ανάλογα με το πώς συνδέεται η συνθήκη με τη αμέσως προηγούμενη της.

#### **Μέθοδοι**

- `public Condition(String term1, String op, String term2, boolean type, String previousBitwise)`  
Κατασκευαστής αντικειμένων της κλάσης. Οι παράμετροι `term1`, `op`, `term2`, `type` και `previousBitwise` αρχικοποιούν τα αντίστοιχα πεδία της κλάσης.

### 5.2.3.7 *public class DBschema*

Με τη βοήθεια της κλάσης `DBschema` γίνεται η διαχείριση της βάσης δεδομένων.

#### **Πεδία**

- `private Connection pConn`  
Η σύνδεση με τη βάση δεδομένων.

#### **Μέθοδοι**

- `private ResultSet execQuery(String strQuery)`  
Εκτέλεση του ερωτήματος `strQuery` στη βάση δεδομένων.
- `private int getRowSize(ResultSet rs)`  
Εύρεση του μεγέθους μιας γραμμής του `rs`.
- `public int getTSRTid(String strTSRFilename, String strTSRName)`  
Εύρεση του κωδικού του `TSR` σχήματος με χαρακτηριστικά το όνομα του αρχείου `strTSRFilename` και το όνομα της `TSR` `strTSRName`.
- `private int getNextTid()`  
Εύρεση του επόμενου κωδικού που μπορεί να δοθεί σε μία νέα καταχώρηση `TSR` σχήματος στη βάση δεδομένων.
- `public TSR getTSR(String strTSRFilename, String strTSRName)`

Προσπέλαση του TSR σχήματος με χαρακτηριστικά strTSRFilename και strTSRName από τη βάση δεδομένων.

- public void addTSR(TSR tsr)  
Προσθήκη του σχήματος tsr στη βάση δεδομένων.
- public boolean existTSR(String strTSRFilename, String strTSRName)  
Αναζήτηση του TSR σχήματος με χαρακτηριστικά strTSRFilename και strTSRName στη βάση δεδομένων.
- public int countdbTSR()  
Καταμέτρηση των TSR σχημάτων που υπάρχουν στη βάση δεδομένων.
- public String[][] findAllTSR(int TSRnumber)  
Εύρεση των ονομάτων των TSR σχημάτων και των ονομάτων των αρχείων από τα οποία προέρχονται.
- public String getTSRdescription(String filename,String tsrname)  
Εύρεση της περιγραφής του TSR σχήματος με χαρακτηριστικά filename και tsrname.
- public DBschema(Connection conn)  
Κατασκευαστής αντικειμένων της κλάσης.

#### 5.2.3.8 *public class XMLschema*

Με τη βοήθεια της κλάσης XMLschema γίνεται η διαχείριση ενός XML αρχείου.

##### **Πεδία**

- private Document source  
Το αντικείμενο της κλάσης Document που δημιουργείται από το XML αρχείο που θα διαχειριστεί.
- private SAXReader parser  
Ο parser που θα προσπελάσει τα στοιχεία του XML αρχείου.

##### **Μέθοδοι**

- public TSR getTSR(String XMLfile, String TSRname)  
Προσπέλαση του TSR σχήματος με όνομα TSRname στο XML αρχείο με όνομα XMLfile.
- public void addTSR(TSR tsr)  
Προσθήκη του σχήματος tsr στο XML αρχείο.
- public boolean existTSR(String strTSRFilename, String strTSRName)  
Έλεγχος για την ύπαρξη TSR σχήματος με όνομα strTSRName στο XML αρχείο με όνομα strTSRFilename.
- public int countxmlTSR(String filename)  
Εύρεση του πλήθους των TSR σχημάτων που υπάρχουν στο XML αρχείο με όνομα αρχείου filename.
- public String[] findALLTSR(String filename)  
Εύρεση των ονομάτων των TSR σχημάτων που υπάρχουν στο XML αρχείο με όνομα filename.
- public String getTSRdescription(String XMLfile, String TSRname)  
Εύρεση της περιγραφής του TSR σχήματος με όνομα TSRname στο XML αρχείο με όνομα XMLfile.
- public String getresource\_item(String XMLfile, String TSRname)  
Εύρεση του ονόματος του κόμβου – αντικειμένου του TSR σχήματος με όνομα TSRname στο XML αρχείο με όνομα XMLfile.
- public XMLschema()  
Κατασκευαστής αντικειμένων της κλάσης.

#### 5.2.3.9 *public class TSRdbDialog*

Η κλάση TSRdbDialog ορίζει ένα βοηθητικό παράθυρο το οποίο αναλαμβάνει τη συλλογή στοιχείων για την αποθήκευση του TSR σχήματος στη βάση δεδομένων.

## Πεδία

- private JButton btOk  
Το κουμπί Ok.
- private JButton btCancel  
Το κουμπί Cancel.
- private JTextField tfTSRfile  
Textfield για την εισαγωγή του ονόματος του αρχείου από το οποίο προέρχεται το TSR σχήμα.
- private JTextField tfTSRname  
TextField για την εισαγωγή του ονόματος του TSR σχήματος που θα αποθηκευτεί.
- private JTextField tfTSRdescription  
TextField για την εισαγωγή της περιγραφής του TSR σχήματος που θα αποθηκευτεί.
- public boolean onOk  
Μεταβλητή που καθορίζει αν ο χρήστης ζήτησε την εισαγωγή του TSR σχήματος στη βάση δεδομένων.
- public String TSRfile  
Το όνομα του TSR αρχείου από το οποίο προέρχεται το TSR σχήμα, όπως δίνεται από τον χρήστη.
- public String TSRname  
Το όνομα του TSR σχήματος που θα αποθηκευτεί, όπως δίνεται από τον χρήστη
- public String TSRdescription  
Η περιγραφή του TSR σχήματος που θα αποθηκευτεί, όπως δίνεται από τον χρήστη.

## Μέθοδοι

- public TSRdbDialog(JFrame frame, String TSRfile, String TSRname, String TSRdescription )  
Κατασκευαστής αντικειμένων της κλάσης.
- private void jblnit()  
Μέθοδος για την αρχικοποίηση του παραθύρου.
- public void actionPerformed(ActionEvent e)  
Μέθοδος για το χειρισμό των ενεργειών του χρήστη.

### 5.2.3.10 public class TSRxmlDialog

Η κλάση TSRxmlDialog ορίζει ένα βοηθητικό παράθυρο το οποίο αναλαμβάνει τη συλλογή στοιχείων για την αποθήκευση του TSR σχήματος σε XML αρχείο.

## Πεδία

- private JButton btOk  
Το κουμπί Ok.
- private JButton btCancel  
Το κουμπί Cancel.
- private JTextField tfTSRfile  
Textfield για την εισαγωγή του ονόματος του αρχείου στο οποίο θα αποθηκευτεί το TSR σχήμα.
- private JTextField tfTSRname  
TextField για την εισαγωγή του ονόματος του TSR σχήματος που θα αποθηκευτεί.
- private JTextField tfTSRdescription  
TextField για την εισαγωγή της περιγραφής του TSR σχήματος που θα αποθηκευτεί.
- public boolean onOk  
Μεταβλητή που καθορίζει αν ο χρήστης ζήτησε την εισαγωγή του TSR σχήματος στο XML αρχείο.
- public String TSRfile  
Το όνομα του TSR αρχείου στο οποίο θα αποθηκευτεί το TSR σχήμα, όπως δίνεται από τον χρήστη.

- `public String TSRname`  
Το όνομα του TSR σχήματος που θα αποθηκευτεί, όπως δίνεται από τον χρήστη
- `public String TSRdescription`  
Η περιγραφή του TSR σχήματος που θα αποθηκευτεί, όπως δίνεται από τον χρήστη.

#### Μέθοδοι

- `public TSRxmlDialog(JFrame frame, String TSRfile, String TSRname, String TSRdescription )`  
Κατασκευαστής αντικειμένων της κλάσης.
- `private void jblInit()`  
Μέθοδος για την αρχικοποίηση του παραθύρου.
- `public void actionPerformed(ActionEvent e)`  
Μέθοδος για το χειρισμό των ενεργειών του χρήστη.

#### 5.2.3.11 *public class TSRfunctions*

Η κλάση `TSRfunctions` υλοποιεί τις ερωτήσεις – πράξεις της εφαρμογής πάνω σε σχήματα – αντικείμενα της κλάσης `TSR`.

#### Πεδία

#### Μέθοδοι

- `public TSR select(Vector vAttributeCondition, Vector vPathCondition, TSR tsr)`  
Η πράξη της επιλογής με λίστα συνθηκών ιδιοτήτων `vAttributeCondition` και λίστα συνθηκών μονοπατιών `vPathCondition` στο σχήμα `tsr`.
- `public TSR project(Vector vAttributeVariable, Vector vPathVariable, TSR tsr)`  
Η πράξη της προβολής με λίστα ιδιοτήτων `vAttributeVariable` και λίστα μεταβλητών μονοπατιών `vPathVariable` στο σχήμα `tsr`.
- `public TSR cartecianProduct(TSR tsr1, TSR tsr2)`  
Η πράξη του καρτεσιανού γινομένου των σχημάτων `tsr1` και `tsr2`.
- `public TSR union(TSR tsr1, TSR tsr2)`  
Η πράξη της ένωσης των σχημάτων `tsr1` και `tsr2`.
- `public TSR intersection(TSR tsr1, TSR tsr2)`  
Η πράξη της τομής των σχημάτων `tsr1` και `tsr2`.
- `public TSR difference(TSR tsr1, TSR tsr2)`  
Η πράξη της διαφοράς των σχημάτων `tsr1` και `tsr2`.
- `private boolean existAttributes(Vector vAttributeVariable, TSR tsr)`  
Μέθοδος για τον έλεγχο της ορθότητας των παραμέτρων μιας πράξης. Ελέγχει αν το σχήμα `tsr` περιέχει τις ιδιότητες `vAttributeVariable`.
- `private boolean existPaths(Vector vPathVariable, TSR tsr)`  
Μέθοδος για τον έλεγχο της ορθότητας των παραμέτρων μιας πράξης. Ελέγχει αν το σχήμα `tsr` περιέχει τις μεταβλητές μονοπατιών `vPathVariable`.
- `private boolean evaluate(String term1, String operator, String term2)`  
Μέθοδος για την αποτίμηση της έκφρασης `term1 operator term2`.
- `public boolean isNumber(String term)`  
Μέθοδος για τον καθορισμό του τύπου του `term` (δηλαδή αν είναι αλφαριθμητικό ή αριθμητικό).
- `private boolean updateAccept(boolean accept, String previousBitwise, boolean term)`  
Μέθοδος για τον ολικό έλεγχο της ισχύος της λίστας συνθηκών.
- `public TSRfunctions()`  
Κατασκευαστής αντικειμένων της κλάσης.

### 5.2.3.12 *public class viewTSR*

Με τη βοήθεια της κλάσης *viewTSR* δημιουργείται ένα βοηθητικό παράθυρο για την επιλογή ενός από τα αποθηκευμένα TSR σχήματα για απεικόνιση σε γραφική μορφή.

#### **Πεδία**

- `private static Connection pConn`  
Η σύνδεση με τη βάση δεδομένων.
- `private JFrame viewFrame`  
Το πλαίσιο (JFrame) το οποίο εμφανίζεται στην οθόνη.
- `private JPanel viewPanel`  
Ο τομέας (JPanel) για την εισαγωγή των συστατικών του πλαισίου.
- `private JButton OkBtn`  
Το κουμπί Ok.
- `private JButton CancelBtn`  
Το κουμπί Cancel.
- `private JLabel label1`  
Βοηθητική ετικέτα (JLabel) για την απεικόνιση πληροφοριακού κειμένου.
- `private JLabel label2`  
Βοηθητική ετικέτα (JLabel) για την απεικόνιση πληροφοριακού κειμένου.
- `private JLabel descriptionLabel`  
Βοηθητική ετικέτα (JLabel) για την απεικόνιση της περιγραφής του TSR σχήματος που έχει επιλεγεί.
- `private DBSchema newDBSchema`  
Αντικείμενο της κλάσης DBSchema.
- `private XMLSchema newXMLSchema`  
Αντικείμενο της κλάσης XMLSchema.
- `private JComboBox choosetsr`  
JComboBox για την επιλογή του TSR προς απεικόνιση.
- `private Font labelFont, lightlabelFont`  
Fonts για τις ετικέτες.
- `private String selectedfile`  
Το όνομα του αρχείου του TSR σχήματος που επιλέχθηκε.
- `private String selectedtsr`  
Το όνομα του TSR σχήματος που επιλέχθηκε.
- `private String selection`  
Η επιλογή από το JComboBox. Αποτελείται από το όνομα του αρχείου και το όνομα του TSR σχήματος.
- `private String alldescriptions[]`  
Πίνακας που περιέχει τις περιγραφές όλων των TSR σχημάτων που εμφανίζονται στο JComboBox.
- `private String description`  
Η περιγραφή του TSR σχήματος που επιλέχθηκε.
- `private GridBagLayout blayout`  
Αντικείμενο της κλάσης GridBagLayout (διάταξη δομημένων πλεγμάτων).

#### **Μέθοδοι**

- `private String[] findAlldescription(String[] allTSR)`  
Μέθοδος για την εύρεση και την αποθήκευση σε πίνακα των περιγραφών όλων των TSR σχημάτων, τα ονόματα των οποίων βρίσκονται στον πίνακα *allTSR*.
- `private TSR findTSR(String filename,String TSRname)`  
Αναζήτηση του TSR σχήματος με όνομα *TSRname* και όνομα αρχείου *filename* και προσπέλαση του. Το ζητούμενο TSR σχήμα είναι αποθηκευμένο είτε στη βάση δεδομένων είτε στο XML αρχείο με το δοθέν όνομα.
- `private void addComponent(Component c, Container p, int row, int column, int width, int height)`  
Η μέθοδος αυτή είναι υπεύθυνη για την κατάλληλη τοποθέτηση των συστατικών στο JFrame της εφαρμογής.

- `public void actionPerformed(ActionEvent ae)`  
Μέθοδος για το χειρισμό των ενεργειών του χρήστη.
- `public viewTSR(Connection con)`  
Κατασκευαστής αντικειμένων της κλάσης.

### 5.2.3.13 *public class ExportTSR2DB*

Με τη βοήθεια της κλάσης `ExportTSR2DB` δημιουργείται ένα βοηθητικό παράθυρο για την επιλογή ενός από τα αποθηκευμένα σε XML αρχεία TSR σχήματα, για αποθήκευση του στη βάση δεδομένων.

#### **Πεδία**

- `private static Connection pConn`  
Η σύνδεση με τη βάση δεδομένων.
- `private JFrame Convert2DBFrame`  
Το πλαίσιο (`JFrame`) το οποίο εμφανίζεται στην οθόνη.
- `private JPanel Convert2DBPanel`  
Ο τομέας (`JPanel`) για την εισαγωγή των συστατικών του πλαισίου.
- `private JButton OkBtn`  
Το κουμπί `Ok`.
- `private JButton CancelBtn`  
Το κουμπί `Cancel`.
- `private JLabel label1`  
Βοηθητική ετικέτα (`JLabel`) για την απεικόνιση πληροφοριακού κειμένου.
- `private JLabel label2`  
Βοηθητική ετικέτα (`JLabel`) για την απεικόνιση πληροφοριακού κειμένου.
- `private JLabel descriptionLabel`  
Βοηθητική ετικέτα (`JLabel`) για την απεικόνιση της περιγραφής του TSR σχήματος που έχει επιλεγεί.
- `private DBSchema newDBSchema`  
Αντικείμενο της κλάσης `DBSchema`.
- `private XMLSchema newXMLSchema`  
Αντικείμενο της κλάσης `XMLSchema`.
- `TSR newTSR`  
Αντικείμενο της κλάσης `TSR`.
- `private JComboBox choosetsr`  
`JComboBox` για την επιλογή του TSR προς μεταφορά.
- `private Font labelFont, lightlabelFont`  
Fonts για τις ετικέτες.
- `private String selectedfile`  
Το όνομα του αρχείου του TSR σχήματος που επιλέχθηκε.
- `private String selectedtsr`  
Το όνομα του TSR σχήματος που επιλέχθηκε.
- `private String selection`  
Η επιλογή από το `JComboBox`. Αποτελείται από το όνομα του αρχείου και το όνομα του TSR σχήματος.
- `private String alldescriptions[]`  
Πίνακας που περιέχει τις περιγραφές όλων των TSR σχημάτων που εμφανίζονται στο `JComboBox`.
- `private String description`  
Η περιγραφή του TSR σχήματος που επιλέχθηκε.
- `private GridBagLayout blayout`  
Αντικείμενο της κλάσης `GridBagLayout` (διάταξη δομημένων πλεγμάτων).

#### **Μέθοδοι**

- `private String[] findAlldescription(String[] allTSR)`

Μέθοδος για την εύρεση και την αποθήκευση σε πίνακα των περιγραφών όλων των TSR σχημάτων, τα ονόματα των οποίων βρίσκονται στον πίνακα allTSR.

- `private void addComponent(Component c, Container p, int row, int column, int width, int height)`  
Η μέθοδος αυτή είναι υπεύθυνη για την κατάλληλη τοποθέτηση των συστατικών στο JFrame της εφαρμογής.
- `public void actionPerformed(ActionEvent ae)`  
Μέθοδος για το χειρισμό των ενεργειών του χρήστη.
- `public ExportTSR2DB(Connection con)`  
Κατασκευαστής αντικειμένων της κλάσης.

#### 5.2.3.14 *public class ExportTSR2File*

Με τη βοήθεια της κλάσης `ExportTSR2File` δημιουργείται ένα βοηθητικό παράθυρο για την επιλογή ενός από τα αποθηκευμένα στη βάση δεδομένων TSR σχήματα, για αποθήκευση του σε κάποιο XML αρχείο.

##### **Πεδία**

- `private static Connection pConn`  
Η σύνδεση με τη βάση δεδομένων.
- `private JFrame Convert2FileFrame`  
Το πλαίσιο (JFrame) το οποίο εμφανίζεται στην οθόνη.
- `private JPanel Convert2FilePanel`  
Ο τομέας (JPanel) για την εισαγωγή των συστατικών του πλαισίου.
- `private JButton OkBtn`  
Το κουμπί Ok.
- `private JButton CancelBtn`  
Το κουμπί Cancel.
- `private JLabel label1`  
Βοηθητική ετικέτα (JLabel) για την απεικόνιση πληροφοριακού κειμένου.
- `private JLabel label2`  
Βοηθητική ετικέτα (JLabel) για την απεικόνιση πληροφοριακού κειμένου.
- `private JLabel descriptionLabel`  
Βοηθητική ετικέτα (JLabel) για την απεικόνιση της περιγραφής του TSR σχήματος που έχει επιλεγεί.
- `private DBschema newDBschema`  
Αντικείμενο της κλάσης `DBschema`.
- `private XMLschema newXMLschema`  
Αντικείμενο της κλάσης `XMLschema`.
- `TSR newTSR`  
Αντικείμενο της κλάσης `TSR`.
- `private JComboBox choosetsr`  
JComboBox για την επιλογή του TSR προς μεταφορά.
- `private Font labelFont, lightlabelFont`  
Fonts για τις ετικέτες.
- `private String selectedfile`  
Το όνομα του αρχείου του TSR σχήματος που επιλέχθηκε.
- `private String selectedtsr`  
Το όνομα του TSR σχήματος που επιλέχθηκε.
- `private String selection`  
Η επιλογή από το JComboBox. Αποτελείται από το όνομα του αρχείου και το όνομα του TSR σχήματος.
- `private String alldescriptions[]`  
Πίνακας που περιέχει τις περιγραφές όλων των TSR σχημάτων που εμφανίζονται στο JComboBox.
- `private String description`  
Η περιγραφή του TSR σχήματος που επιλέχθηκε.

- private GridBagLayout blayout  
Αντικείμενο της κλάσης GridBagLayout (διάταξη δομημένων πλεγμάτων).

### Μέθοδοι

- private String[] findAlldescription(String[] allTSR)  
Μέθοδος για την εύρεση και την αποθήκευση σε πίνακα των περιγραφών όλων των TSR σχημάτων, τα ονόματα των οποίων βρίσκονται στον πίνακα allTSR.
- private static ResultSet execQuery(String strQuery)  
Μέθοδος για την εκτέλεση του ερωτήματος strQuery στη βάση δεδομένων.
- private void addComponent(Component c, Container p, int row, int column, int width, int height)  
Η μέθοδος αυτή είναι υπεύθυνη για την κατάλληλη τοποθέτηση των συστατικών στο JFrame της εφαρμογής.
- public void actionPerformed(ActionEvent ae)  
Μέθοδος για το χειρισμό των ενεργειών του χρήστη.
- public ExportTSR2File(Connection con)  
Κατασκευαστής αντικειμένων της κλάσης.

### 5.2.3.15 public class deleteTSR

Με τη βοήθεια της κλάσης deleteTSR δημιουργείται ένα βοηθητικό παράθυρο για την επιλογή ενός από τα αποθηκευμένα TSR σχήματα, για διαγραφή.

### Πεδία

- private static Connection pConn  
Η σύνδεση με τη βάση δεδομένων.
- private JFrame deleteFrame  
Το πλαίσιο (JFrame) το οποίο εμφανίζεται στην οθόνη.
- private JPanel deletePanel  
Ο τομέας (JPanel) για την εισαγωγή των συστατικών του πλαισίου.
- private JButton OkBtn  
Το κουμπί Ok.
- private JButton CancelBtn  
Το κουμπί Cancel.
- private JLabel label1, label2  
Βοηθητικές ετικέτες (JLabel) για την απεικόνιση πληροφοριακού κειμένου.
- private JLabel descriptionLabel  
Βοηθητική ετικέτα (JLabel) για την απεικόνιση της περιγραφής του TSR σχήματος που έχει επιλεγεί.
- private DBSchema newDBSchema  
Αντικείμενο της κλάσης DBSchema.
- private XMLSchema newXMLSchema  
Αντικείμενο της κλάσης XMLSchema.
- private JComboBox choosetsr  
JComboBox για την επιλογή του TSR προς διαγραφή.
- private Font labelFont, lightlabelFont  
Fonts για τις ετικέτες.
- private String selectedfile  
Το όνομα του αρχείου του TSR σχήματος που επιλέχθηκε.
- private String selectedtsr  
Το όνομα του TSR σχήματος που επιλέχθηκε.
- private String selection  
Η επιλογή από το JComboBox. Αποτελείται από το όνομα του αρχείου και το όνομα του TSR σχήματος.
- private GridBagLayout blayout  
Αντικείμενο της κλάσης GridBagLayout (διάταξη δομημένων πλεγμάτων).



- private Document source  
Αντικείμενο της κλάσης Document για το XML αρχείο που θα διαχειριστεί.
- private SAXReader parser  
Ο parser που θα προσπελάσει τα στοιχεία του XML αρχείου.

### Μέθοδοι

- public String[] findAllTSR()  
Μέθοδος για την εύρεση και την αποθήκευση σε πίνακα όλων των TSR σχημάτων που είναι αποθηκευμένα είτε στη βάση δεδομένων είτε σε XML αρχεία.
- private static ResultSet execQuery(String strQuery)  
Μέθοδος για την εκτέλεση του ερωτήματος strQuery στη βάση δεδομένων.
- private void addComponent(Component c, Container p, int row, int column, int width, int height)  
Η μέθοδος αυτή είναι υπεύθυνη για την κατάλληλη τοποθέτηση των συστατικών στο JFrame της εφαρμογής.
- public void deletefromdb(String filename,String tsrname)  
Μέθοδος για τη διαγραφή του TSR σχήματος με όνομα tsrname και όνομα αρχείου filename από τη βάση δεδομένων.
- public void deletefromfile(String filename,String tsrname)  
Μέθοδος για τη διαγραφή του TSR σχήματος με όνομα tsrname από το XML αρχείο με όνομα filename.
- public void actionPerformed(ActionEvent ae)  
Μέθοδος για το χειρισμό των ενεργειών του χρήστη.
- public deleteTSR(Connection con)  
Κατασκευαστής αντικειμένων της κλάσης.

#### 5.2.3.16 public class changeTSR

Με τη βοήθεια της κλάσης changeTSR δημιουργείται ένα βοηθητικό παράθυρο για την επιλογή ενός από τα αποθηκευμένα TSR σχήματα, για αλλαγή ενός από τα χαρακτηριστικά στοιχεία του.

### Πεδία

- private static Connection pConn  
Η σύνδεση με τη βάση δεδομένων.
- private JFrame changeFrame  
Το πλαίσιο (JFrame) το οποίο εμφανίζεται στην οθόνη.
- private JPanel changePanel  
Ο τομέας (JPanel) για την εισαγωγή των συστατικών του πλαισίου.
- private JButton OkBtn  
Το κουμπί Ok.
- private JButton CancelBtn  
Το κουμπί Cancel.
- private JLabel label1, label2  
Βοηθητικές ετικέτες (JLabel) για την απεικόνιση πληροφοριακού κειμένου.
- private JLabel descriptionLabel  
Βοηθητική ετικέτα (JLabel) για την απεικόνιση της περιγραφής του TSR σχήματος που έχει επιλεγεί.
- private DBschema newDBschema  
Αντικείμενο της κλάσης DBschema.
- private XMLschema newXMLschema  
Αντικείμενο της κλάσης XMLschema.
- private JComboBox choosetsr  
JComboBox για την επιλογή του TSR προς αλλαγή.
- private Font labelFont, lightlabelFont  
Fonts για τις ετικέτες.
- private String selectedfile

- Το όνομα του αρχείου του TSR σχήματος που επιλέχθηκε.
- private String selectedtsr  
Το όνομα του TSR σχήματος που επιλέχθηκε.
- private String selection  
Η επιλογή από το JComboBox. Αποτελείται από το όνομα του αρχείου και το όνομα του TSR σχήματος.
- private GridBagLayout blayout  
Αντικείμενο της κλάσης GridBagLayout (διάταξη δομημένων πλεγμάτων).
- String alldescriptions[]  
Πίνακας που περιέχει τις περιγραφές όλων των TSR σχημάτων που εμφανίζονται στο JComboBox.

### Μέθοδοι

- private String[] findAlldescription(String[] allTSR)  
Μέθοδος για την εύρεση και την αποθήκευση σε πίνακα των περιγραφών όλων των TSR σχημάτων, τα ονόματα των οποίων βρίσκονται στον πίνακα allTSR.
- private void addComponent(Component c, Container p, int row, int column, int width, int height)  
Η μέθοδος αυτή είναι υπεύθυνη για την κατάλληλη τοποθέτηση των συστατικών στο JFrame της εφαρμογής.
- public void actionPerformed(ActionEvent ae)  
Μέθοδος για το χειρισμό των ενεργειών του χρήστη.
- public changeTSR (Connection con)  
Κατασκευαστής αντικειμένων της κλάσης.

### 5.2.3.17 public class PermittedChanges

Με τη βοήθεια της κλάσης PermittedChanges δημιουργείται ένα βοηθητικό παράθυρο για την επιλογή μίας από τις επιτρεπτές αλλαγές στο TSR σχήμα που έχει επιλεγεί. Οι επιτρεπτές αλλαγές είναι: αλλαγή του ονόματος του TSR σχήματος, αλλαγή του ονόματος του αρχείου του TSR σχήματος, αλλαγή της περιγραφής του TSR σχήματος και αλλαγή του ονόματος του κόμβου – αντικειμένου του TSR σχήματος.

### Πεδία

- private static Connection pConn  
Η σύνδεση με τη βάση δεδομένων.
- private JFrame permchangesFrame  
Το πλαίσιο (JFrame) το οποίο εμφανίζεται στην οθόνη.
- private JPanel permchangesPanel  
Ο τομέας (JPanel) για την εισαγωγή των συστατικών του πλαισίου.
- private JButton OkBtn  
Το κουμπί Ok.
- private JButton CancelBtn  
Το κουμπί Cancel.
- private JLabel label1  
Βοηθητική ετικέτα (JLabel) για την απεικόνιση πληροφοριακού κειμένου.
- private DBschema newDBschema  
Αντικείμενο της κλάσης DBschema.
- private XMLschema newXMLschema  
Αντικείμενο της κλάσης XMLschema.
- private TSR tsrforchange  
Αντικείμενο της κλάσης TSR.
- private JComboBox choosechange  
JComboBox για την επιλογή της επιθυμητής αλλαγής στο TSR σχήμα.
- private Font labelFont  
Font για την ετικέτα label1.

- private String filename  
Το όνομα του αρχείου του TSR σχήματος που έχει επιλεγεί για αλλαγή.
- private String tsrname  
Το όνομα του TSR σχήματος που έχει επιλεγεί για αλλαγή.
- private String selection  
Η επιλογή από το JComboBox, δηλαδή η επιλογή της επιθυμητής αλλαγής.
- private GridBagLayout blayout  
Αντικείμενο της κλάσης GridBagLayout (διάταξη δομημένων πλεγμάτων).
- private JFrame changeTSRnameFrame  
Το πλαίσιο (JFrame) που εμφανίζεται στην οθόνη αν επιλεγθεί η αλλαγή του ονόματος του TSR σχήματος.
- private JFrame changeFilenameFrame  
Το πλαίσιο (JFrame) που εμφανίζεται στην οθόνη αν επιλεγθεί η αλλαγή του ονόματος του αρχείου του TSR σχήματος.
- private JFrame changedescriptionFrame  
Το πλαίσιο (JFrame) που εμφανίζεται στην οθόνη αν επιλεγθεί η αλλαγή της περιγραφής του TSR σχήματος.
- private JFrame changeresourceitemFrame  
Το πλαίσιο (JFrame) που εμφανίζεται στην οθόνη αν επιλεγθεί η αλλαγή του ονόματος του κόμβου – αντικειμένου του TSR σχήματος.

#### **Μέθοδοι**

- private TSR findTSR(String file,String tsr)  
Μέθοδος για την εύρεση και την προσπέλαση του TSR σχήματος με όνομα tsr και όνομα αρχείου file, το οποίο είναι αποθηκευμένο είτε στη βάση δεδομένων είτε στο XML αρχείο με το δοθέν όνομα.
- private void addComponent(Component c, Container p, int row, int column, int width, int height)  
Η μέθοδος αυτή είναι υπεύθυνη για την κατάλληλη τοποθέτηση των συστατικών στο JFrame της εφαρμογής.
- public void actionPerformed(ActionEvent ae)  
Μέθοδος για το χειρισμό των ενεργειών του χρήστη.
- public PermittedChanges(Connection con,String file,String tsr,JFrame frame)  
Κατασκευαστής αντικειμένων της κλάσης.

#### *5.2.3.18 public class ChangeTSRName*

Με τη βοήθεια της κλάσης ChangeTSRName δημιουργείται ένα βοηθητικό παράθυρο για την εισαγωγή του νέου ονόματος του TSR σχήματος που έχει επιλεγεί για αυτήν την αλλαγή.

#### **Πεδία**

- private JLabel TSRnameLabel, insertnewTSRnameLabel  
Βοηθητικές ετικέτες (JLabel) για την απεικόνιση πληροφοριακού κειμένου.
- private JTextField insertnewTSRnameTF  
TextField για την εισαγωγή του νέου ονόματος του TSR σχήματος που επιλέχθηκε για την αλλαγή αυτή.
- private JButton Done1Btn  
Το κουμπί Ok.
- private JButton Undo1Btn  
Το κουμπί Cancel.
- private JPanel changeTSRnamePanel  
Ο τομέας (JPanel) για την εισαγωγή των συστατικών του πλαισίου.

#### **Μέθοδοι**

- public void actionPerformed(ActionEvent ae)

- Μέθοδος για το χειρισμό των ενεργειών του χρήστη.  
public ChangeTSRName(JFrame frame1)  
Κατασκευαστής αντικειμένων της κλάσης.

### 5.2.3.19 public class ChangeFileName

Με τη βοήθεια της κλάσης ChangeFileName δημιουργείται ένα βοηθητικό παράθυρο για την εισαγωγή του νέου ονόματος του αρχείου του TSR σχήματος που έχει επιλεγεί για αυτήν την αλλαγή.

#### Πεδία

- private JLabel filenameLabel, insertnewfilenameLabel  
Βοηθητικές ετικέτες (JLabel) για την απεικόνιση πληροφοριακού κειμένου.
- private JTextField insertnewfilenameTF  
TextField για την εισαγωγή του νέου ονόματος του αρχείου του TSR σχήματος που επιλέχθηκε για την αλλαγή αυτή.
- private JButton Done2Btn  
Το κουμπί Ok.
- private JButton Undo2Btn  
Το κουμπί Cancel.
- private JPanel changefilenamePanel  
Ο τομέας (JPanel) για την εισαγωγή των συστατικών του πλαισίου.

#### Μέθοδοι

- public void actionPerformed(ActionEvent ae)  
Μέθοδος για το χειρισμό των ενεργειών του χρήστη.
- public ChangeFileName(JFrame frame2)  
Κατασκευαστής αντικειμένων της κλάσης.

### 5.2.3.20 public class ChangeDescription

Με τη βοήθεια της κλάσης ChangeDescription δημιουργείται ένα βοηθητικό παράθυρο για την εισαγωγή της νέας περιγραφής του TSR σχήματος που έχει επιλεγεί για αυτήν την αλλαγή.

#### Πεδία

- private JLabel descriptionLabel, insertnewdescriptionLabel  
Βοηθητικές ετικέτες (JLabel) για την απεικόνιση πληροφοριακού κειμένου.
- private JTextField insertnewdescriptionTF  
TextField για την εισαγωγή της νέας περιγραφής του TSR σχήματος που επιλέχθηκε για την αλλαγή αυτή.
- private JButton Done3Btn  
Το κουμπί Ok.
- private JButton Undo3Btn  
Το κουμπί Cancel.
- private JPanel changedescriptionPanel  
Ο τομέας (JPanel) για την εισαγωγή των συστατικών του πλαισίου.

#### Μέθοδοι

- public void actionPerformed(ActionEvent ae)  
Μέθοδος για το χειρισμό των ενεργειών του χρήστη.
- public ChangeDescription(JFrame frame3)  
Κατασκευαστής αντικειμένων της κλάσης.

### 5.2.3.21 *public class ChangeResourceItem*

Με τη βοήθεια της κλάσης `ChangeResourceItem` δημιουργείται ένα βοηθητικό παράθυρο για την εισαγωγή του νέου ονόματος του κόμβου - αντικειμένου του TSR σχήματος που έχει επιλεγεί για αυτήν την αλλαγή.

#### Πεδία

- `private JLabel resourceitemLabel, insertnewresourceitemnameLabel`  
Βοηθητικές ετικέτες (`JLabel`) για την απεικόνιση πληροφοριακού κειμένου.
- `private JTextField insertnewresourceitemnameTF`  
`TextField` για την εισαγωγή του νέου ονόματος του κόμβου - αντικειμένου του TSR σχήματος που επιλέχθηκε για την αλλαγή αυτή.
- `private JButton Done4Btn`  
Το κουμπί `Ok`.
- `private JButton Undo4Btn`  
Το κουμπί `Cancel`.
- `private JPanel changeresourceitemPanel`  
Ο τομέας (`JPanel`) για την εισαγωγή των συστατικών του πλαισίου.

#### Μέθοδοι

- `public void actionPerformed(ActionEvent ae)`  
Μέθοδος για το χειρισμό των ενεργειών του χρήστη.
- `public ChangeResourceItem(JFrame frame1)`  
Κατασκευαστής αντικειμένων της κλάσης.

### 5.2.3.22 *public class importfile*

Με τη βοήθεια της κλάσης `importfile` δημιουργείται ένα βοηθητικό παράθυρο για την επιλογή του `gry` αρχείου προς εκτέλεση. Το `gry` αρχείο περιέχει μία ερώτηση η οποία θα αποτιμηθεί και το αποτέλεσμα της θα παρουσιαστεί σε γραφική μορφή.

#### Πεδία

- `private static Connection pConn`  
Η σύνδεση με τη βάση δεδομένων.
- `private JFrame importfileFrame`  
Το πλαίσιο (`JFrame`) που εμφανίζεται στην οθόνη.
- `private JPanel importfilePanel`  
Ο τομέας (`JPanel`) για την εισαγωγή των συστατικών του πλαισίου.
- `private JButton OkBtn`  
Το κουμπί `Ok`.
- `private JButton CancelBtn`  
Το κουμπί `Cancel`.
- `private JLabel label1, label2`  
Βοηθητικές ετικέτες (`JLabel`) για την απεικόνιση πληροφοριακού κειμένου.
- `private DBSchema newDBSchema`  
Αντικείμενο της κλάσης `DBSchema`.
- `private XMLSchema newXMLSchema`  
Αντικείμενο της κλάσης `XMLSchema`.
- `private TSR newTSR`  
Αντικείμενο της κλάσης `TSR`.
- `private JComboBox choosegry`  
`JComboBox` για την επιλογή του επιθυμητού `gry` αρχείου προς εκτέλεση.
- `private Font labelFont, lightlabelFont`  
Fonts για τις ετικέτες.
- `private String selection`

- Η επιλογή από το JComboBox. Αποτελείται από το όνομα του qry αρχείου.
- private static int x  
Βοηθητικός ακέραιος.
- private GridBagLayout blayout  
Αντικείμενο της κλάσης GridBagLayout (διάταξη δομημένων πλεγμάτων).

### Μέθοδοι

- private void addComponent(Component c, Container p, int row, int column, int width, int height)  
Η μέθοδος αυτή είναι υπεύθυνη για την κατάλληλη τοποθέτηση των συστατικών στο JFrame της εφαρμογής.
- public void actionPerformed(ActionEvent ae)  
Μέθοδος για το χειρισμό των ενεργειών του χρήστη.
- public importfile(Connection con,int howmanyimportfile)  
Κατασκευαστής αντικειμένων της κλάσης.

### 5.2.3.23 public class importxmlfile

Με τη βοήθεια της κλάσης importxmlfile δημιουργείται ένα βοηθητικό παράθυρο για την επιλογή ενός από τα διαθέσιμα XML αρχεία προς έλεγχο. Αν το XML αρχείο είναι well formed τότε αυτό αποθηκεύεται σε κατάλληλη θέση.

### Πεδία

- private static Connection pConn  
Η σύνδεση με τη βάση δεδομένων.
- private JFrame importfileFrame  
Το πλαίσιο (JFrame) το οποίο εμφανίζεται στην οθόνη.
- private JPanel importfilePanel  
Ο τομέας (JPanel) για την εισαγωγή των συστατικών του πλαισίου.
- private JButton OkBtn  
Το κουμπί Ok.
- private JButton CancelBtn  
Το κουμπί Cancel.
- private JLabel insertnameLabel, label  
Βοηθητικές ετικέτες (JLabel) για την απεικόνιση πληροφοριακού κειμένου.
- private DBschema newDBschema  
Αντικείμενο της κλάσης DBschema.
- private XMLschema newXMLschema  
Αντικείμενο της κλάσης XMLschema.
- private JComboBox choosexmlfile  
JComboBox για την επιλογή του XML αρχείου προς έλεγχο.
- private Font labelFont, lightlabelFont  
Fonts για τις ετικέτες.
- private final String filelist[]  
Πίνακας με τα διαθέσιμα προς έλεγχο XML αρχεία.
- private String flist[]  
Πίνακας με όλα τα διαθέσιμα XML αρχεία.
- private String selection  
Η επιλογή από το JComboBox. Αποτελείται από το όνομα του XML αρχείου προς έλεγχο.
- private GridBagLayout blayout  
Αντικείμενο της κλάσης GridBagLayout (διάταξη δομημένων πλεγμάτων).
- private Document source  
Αντικείμενο της κλάσης Document για το XML αρχείο που θα διαχειριστεί.
- private SAXReader parser  
Ο parser που θα προσπελάσει τα στοιχεία του XML αρχείου.

## Μέθοδοι

- `private void addFile(String filename)`  
Αν το XML αρχείο που ελέγχθηκε είναι well formed τότε η μέθοδος αυτή αναλαμβάνει να το αποθηκεύσει στον κατάλληλο κατάλογο.
- `private void addComponent(Component c, Container p, int row, int column, int width, int height)`  
Η μέθοδος αυτή είναι υπεύθυνη για την κατάλληλη τοποθέτηση των συστατικών στο JFrame της εφαρμογής.
- `public void actionPerformed(ActionEvent ae)`  
Μέθοδος για το χειρισμό των ενεργειών του χρήστη.
- `public importxmlfile(Connection con)`  
Κατασκευαστής αντικειμένων της κλάσης.

### 5.2.3.24 *public class insertResourceItemName*

Με τη βοήθεια της κλάσης `insertResourceItemName` δημιουργείται ένα βοηθητικό παράθυρο για την εισαγωγή του ονόματος του κόμβου – αντικειμένου του TSR σχήματος που δημιουργείται μετά από ένα query.

## Πεδία

- `private static Connection pConn`  
Η σύνδεση με τη βάση δεδομένων.
- `private JFrame insertname`  
Το πλαίσιο (JFrame) το οποίο εμφανίζεται στην οθόνη.
- `private JPanel insertnamePanel`  
Ο τομέας (JPanel) για την εισαγωγή των συστατικών του πλαισίου.
- `private JButton OK`  
Το κουμπί Ok.
- `private JLabel insertnodename`  
Βοηθητική ετικέτα (JLabel) για την απεικόνιση πληροφοριακού κειμένου.
- `private JTextField insertnodenameTF`  
JTextField για την εισαγωγή του ονόματος του κόμβου – αντικειμένου του TSR σχήματος.
- `private TSR newTSR`  
Αντικείμενο της κλάσης TSR.
- `private boolean CanSave`  
Βοηθητική boolean μεταβλητή.
- `private Font labelFont`  
Font για την ετικέτα `insertnodename`.
- `private GridBagLayout blayout`  
Αντικείμενο της κλάσης `GridBagLayout` (διάταξη δομημένων πλεγμάτων).

## Μέθοδοι

- `private void addComponent(Component c, Container p, int row, int column, int width, int height)`  
Η μέθοδος αυτή είναι υπεύθυνη για την κατάλληλη τοποθέτηση των συστατικών στο JFrame της εφαρμογής.
- `public insertResourceItemName(TSR localsr, Connection conn, boolean cansave)`  
Κατασκευαστής αντικειμένων της κλάσης.

### 5.2.3.25 *public class NewQuery*

Με τη βοήθεια της κλάσης `NewQuery` δημιουργείται το βασικό παράθυρο για την επιλογή των TSRs σχημάτων που θα συμμετάσχουν στην ερώτηση.

## Πεδία

- private static Connection pConn  
Η σύνδεση με τη βάση δεδομένων.
- private JFrame firstFrame  
Το πλαίσιο (JFrame) το οποίο εμφανίζεται στην οθόνη.
- private JPanel firstPanel  
Ο τομέας (JPanel) για την εισαγωγή των συστατικών του πλαισίου.
- private JButton Done1Btn  
Το κουμπί Ok.
- private JButton Cancel1Btn  
Το κουμπί Cancel.
- private JButton nextBtn  
Το κουμπί Next.
- private JButton prevBtn  
Το κουμπί Back.
- private JLabel chooseTSR, label, label1, label2  
Βοηθητικές ετικέτες (JLabel) για την απεικόνιση πληροφοριακού κειμένου.
- private DBSchema newDBSchema  
Αντικείμενο της κλάσης DBSchema.
- private XMLSchema newXMLSchema  
Αντικείμενο της κλάσης XMLSchema.
- private TSRfunctions newTSRfunction  
Αντικείμενο της κλάσης TSRfunctions.
- private JList allTSRlist  
JList για όλα τα διαθέσιμα αποθηκευμένα TSR σχήματα.
- private JList allselectedTSRlist  
JList για τα TSR σχήματα που έχουν επιλεγθεί.
- private Vector AllTSRandFilesVector  
Διάνυσμα που περιέχει όλα τα διαθέσιμα αποθηκευμένα TSR σχήματα.
- private Vector AllselectedTSRVector  
Διάνυσμα που περιέχει όλα τα TSR σχήματα που έχουν επιλεγθεί για την ερώτηση.
- private Font labelFont, lightlabelFont  
Fonts για τις ετικέτες.
- private final JPanel panel3  
Βοηθητικός τομέας (JPanel) για την εισαγωγή των συστατικών του πλαισίου.
- private GridBagLayout blayout  
Αντικείμενο της κλάσης GridBagLayout (διάταξη δομημένων πλεγμάτων).

## Μέθοδοι

- private Vector findAllTSR()  
Μέθοδος για την εύρεση και την αποθήκευση σε διάνυσμα όλων των διαθέσιμων αποθηκευμένων TSRs σχημάτων.
- private TSR findTSR(String filename,String TSRname)  
Μέθοδος για την εύρεση και την προσπέλαση του TSR σχήματος με όνομα TSRname και όνομα αρχείου filename.
- private void addComponent(Component c, Container p, int row, int column, int width, int height)  
Η μέθοδος αυτή είναι υπεύθυνη για την κατάλληλη τοποθέτηση των συστατικών στο JFrame της εφαρμογής.
- private void CheckEndOfQuery(CreateQuery createquery)  
Η μέθοδος αυτή είναι υπεύθυνη για τον εντοπισμό της ολοκλήρωσης της ερώτησης πάνω στα TSR σχήματα που έχουν επιλεγθεί.
- public NewQuery(Connection con)  
Κατασκευαστής αντικειμένων της κλάσης.



### 5.2.3.26 *public class visualizetree*

Η κλάση `visualizetree` υλοποιεί την εφαρμογή απεικόνισης ενός TSR σχήματος σε γραφική μορφή δέντρου.

#### **Πεδία**

- `private static Connection pConn`  
Η σύνδεση με τη βάση δεδομένων.
- `private JFrame frame`  
Το πλαίσιο (`JFrame`) το οποίο εμφανίζεται στην οθόνη.
- `private JToolBar jToolBar`  
Η γραμμή εργαλείων (`JToolBar`) που ομαδοποιεί τα κουμπιά της οθόνης.
- `private JButton btExit`  
Το κουμπί `Exit`.
- `private JButton btSave2XML`  
Το κουμπί `Save2XML` για αποθήκευση του TSR σχήματος σε XML αρχείο.
- `private JButton btSave2DB`  
Το κουμπί `Save2DB` για αποθήκευση του TSR σχήματος στη βάση δεδομένων.
- `private JScrollPane scroller`  
Ο κυλιόμενος τομέας (`JScrollPane`) που περιέχει το γράφημα.
- `private Element eTSR`  
Το στοιχείο `eTSR`.
- `private JGraphModelAdapter m_jgAdapter`  
Αντικείμενο της κλάσης `JGraphModelAdapter`.
- `private TSR newTSR`  
Αντικείμενο της κλάσης `TSR`.
- `private JGraph jgraph`  
Αντικείμενο της κλάσης `JGraph`.
- `private static final Color DEFAULT_BG_COLOR`  
Βοηθητική σταθερά για το χρώμα.
- `private String file`  
Το όνομα του αρχείου του TSR σχήματος που θα απεικονιστεί.
- `private String tsr`  
Το όνομα του TSR σχήματος.
- `private String resourceitem`  
Το όνομα του κόμβου – αντικειμένου του TSR σχήματος.
- `private String description`  
Η περιγραφή του TSR σχήματος.
- `private String attributenames[]`  
Πίνακας για την αποθήκευση των ιδιοτήτων του TSR σχήματος.
- `private Object tuples[][]`  
Πίνακας για την αποθήκευση των εγγράφων του TSR σχήματος.
- `boolean independentframe`  
Βοηθητική Boolean μεταβλητή
- `private Document source`  
Αντικείμενο της κλάσης `Document` για τη διαχείριση του XML αρχείου.
- `private SAXReader parser`  
Ο parser που θα προσπελάσει τα στοιχεία του XML αρχείου.
- `private GridBagLayout blayout`  
Αντικείμενο της κλάσης `GridBagLayout` (διάταξη δομημένων πλεγμάτων).

#### **Μέθοδοι**

- `public void mouseClicked(MouseEvent event)`  
Μέθοδος για το χειρισμό των ενεργειών του ποντικιού.
- `public void mousePressed(MouseEvent event)`  
Μέθοδος για το χειρισμό των ενεργειών του ποντικιού.
- `public void mouseMoved(MouseEvent event)`  
Μέθοδος για το χειρισμό των ενεργειών του ποντικιού.

- `public void mouseReleased(MouseEvent event)`  
Μέθοδος για το χειρισμό των ενεργειών του ποντικιού.
- `public void mouseEntered(MouseEvent event)`  
Μέθοδος για το χειρισμό των ενεργειών του ποντικιού.
- `public void mouseExited(MouseEvent event)`  
Μέθοδος για το χειρισμό των ενεργειών του ποντικιού.
- `private void positionVertexAt(Object vertex,int x, int y , boolean islastVertex)`  
Μέθοδος για την τοποθέτηση των κόμβων στην κατάλληλη θέση μέσα στο παράθυρο.
- `private void adjustDisplaySettings( JGraph jg )`  
Μέθοδος για τον καθορισμό των ιδιοτήτων του γραφήματος.
- `public void init( )`  
Μέθοδος για τη δημιουργία του γραφήματος.
- `public visualizetree(TSR localtsr,Connection pConn,String nodename,boolean cansave,boolean x)`  
Κατασκευαστής αντικειμένων της κλάσης.

### 5.2.3.27 *public class BoxPressed*

Με τη βοήθεια της κλάσης `BoxPressed` δημιουργείται ένα βοηθητικό παράθυρο στο οποίο εμφανίζονται οι ιδιότητες και οι εγγραφές ενός TSR σχήματος. Το παράθυρο αυτό εμφανίζεται όταν πατηθεί ο κόμβος 'D A T A' του γραφήματος.

#### **Πεδία**

#### **Μέθοδοι**

- `public BoxPressed(Object tuples[ ][ ],String attributenames[ ])`  
Κατασκευαστής αντικειμένων της κλάσης.

### 5.2.3.28 *public class CreateQuery*

Η κλάση `CreateQuery` υλοποιεί το υποσύστημα ερωτήσεων - πράξεων.

#### **Πεδία**

- `private static Connection pConn`  
Η σύνδεση με τη βάση δεδομένων.
- `private DBSchema newDBSchema`  
Αντικείμενο της κλάσης `DBSchema`.
- `private XMLSchema newXMLSchema`  
Αντικείμενο της κλάσης `XMLSchema`.
- `private TSRfunctions newTSRfunction`  
Αντικείμενο της κλάσης `TSRfunctions`.
- `private TSR resultTSR1, resultTSR2, tsr1, tsr2, finalTSR`  
Αντικείμενα της κλάσης `TSR`.
- `private JFrame oneTSRFrame`  
Το πλαίσιο (`JFrame`) όταν έχει επιλεγθεί ένα `TSR`.
- `private JPanel oneTSRPanel`  
Τομέας (`JPanel`) για την εισαγωγή των συστατικών του πλαισίου όταν έχει επιλεγθεί ένα `TSR`.
- `private JButton oneTSROkBtn, oneTSRCancelBtn`  
Τα κουμπιά `Ok` και `Cancel` όταν έχει επιλεγθεί ένα `TSR`.
- `private JPanel twoTSRPanel`  
Τομέας (`JPanel`) για την εισαγωγή των συστατικών του πλαισίου όταν έχουν επιλεγθεί περισσότερα από ένα `TSR`.
- `private JButton Done1Btn, Undo1Btn`  
Τα κουμπιά `Ok` και `Cancel` όταν έχουν επιλεγθεί περισσότερα από ένα `TSR`.

- private JLabel label1, label2, label3, label4, label5, firstquery, secondquery, finalquery, tsrnameLabel, onelabel1, onelabel2  
Βοηθητικές ετικέτες (JLabel) για την απεικόνιση πληροφοριακού κειμένου.
- private String currentfilename1, currenttsrname1, currentfilename2, currenttsrname2, firstchoice, secondchoice, queryforfirsttsr, queryforsecondtsr, queryforonetsr  
Βοηθητικά Strings για τον έλεγχο του προγράμματος.
- private int firstTSRchanged, secondTSRchanged, thirdTSRchanged, firstchoicemade, secondchoicemade  
Βοηθητικοί ακέραιοι για τον έλεγχο του προγράμματος.
- private String selectedTSRs[]  
Πίνακας που περιέχει όλα τα TSR που έχουν επιλεγθεί για να συμμετέχουν στην ερώτηση.
- private JComboBox cbx[], binaryqueryBox, unitaryqueryBox1, unitaryqueryBox2, unitaryqueryBox3, unitaryqueryBox4  
JComboBoxes για την επιλογή κατάλληλου TSR, δυαδικής πράξης ή μοναδιαίας πράξης.
- private int NoselectedTSR  
Αριθμός των επιλεγμένων TSRs.
- private Choice attribute1list, path1list  
Choice για την επιλογή ιδιοτήτων και μονοπατιών στην πράξη της επιλογής.
- private String connection  
Το πεδίο αυτό παίρνει τις τιμές AND ή OR ανάλογα με το πώς συνδέονται μεταξύ τους οι συνθήκες.
- private int attrconditioncounter, pathconditioncounter  
Βοηθητικοί ακέραιοι μετρητές.
- private String previousconditionconnector, previouspathconditionconnector  
Βοηθητικά Strings.
- JLabel thirdTSRLabel, TSRLabel1  
Βοηθητικές ετικέτες (JLabel) για την απεικόνιση πληροφοριακού κειμένου.
- JButton Done2Btn, Undo2Btn  
Τα κουμπιά Ok και Cancel.
- private int NoremainedTSR  
Αριθμός των TSRs που δεν έχουν ακόμα συμμετάσχει στην ερώτηση.
- private String remainsTSRs[]  
Πίνακας που περιέχει τα ονόματα των TSRs που δεν έχουν συμμετάσχει στην ερώτηση.
- private JLabel TSRLabel[]  
Πίνακας που περιέχει βοηθητικές ετικέτες (JLabel) για την απεικόνιση πληροφοριακού κειμένου.
- private int TSRLabelcounter  
Μετρητής ετικετών που έχουν απεικονιστεί.
- private Font queryFont, lightlabelFont  
Fonts που χρησιμοποιούνται για τις ετικέτες.
- private int maxwidth  
Μέγιστο μήκος ονόματος TSR σχήματος.
- public int donequery  
Όταν το πεδίο αυτό πάρει την τιμή 1, τότε η ερώτηση – πράξη έχει ολοκληρωθεί.
- public int cancelquery  
Όταν το πεδίο αυτό πάρει την τιμή 1, τότε η ερώτηση – πράξη έχει ακυρωθεί.
- private int currentcbx  
Μετρητής για τον αριθμό των JComboBoxes που έχουν εμφανιστεί στην οθόνη.
- private int firstTSRchanged, secondTSRchanged, thirdTSRchanged, firstchoicemade, secondchoicemade  
Βοηθητικοί ακέραιοι.
- private int isabouttochangefirstTSR, isabouttochange secondTSR, isabouttochange thirdTSR, isabouttochange finaltsr  
Βοηθητικοί ακέραιοι για τον έλεγχο του TSR σχήματος που μεταβλήθηκε.
- private Color labelColor  
Αντικείμενο της κλάσης Color.
- private GridBagConstraints constraints  
Αντικείμενο της κλάσης GridBagConstraints.
- private GridBagLayout blayout  
Αντικείμενο της κλάσης GridBagLayout.

## Μέθοδοι

- private TSR findTSR(String filename,String TSRname)  
Εύρεση και προσπέλαση του TSR σχήματος με όνομα TSRname και όνομα αρχείου filename.
- private void addComponent( Component c, Container p,int row, int column, int width, int height )  
Η μέθοδος αυτή είναι υπεύθυνη για την κατάλληλη τοποθέτηση των συστατικών στο JFrame της εφαρμογής..
- private TSR PROJECT(Vector vAttributeList,Vector vPathList,TSR tsr)  
Μέθοδος για τον έλεγχο της πράξης της προβολής.
- private TSR SELECT(Vector vAttributeCondition,Vector vPathCondition,TSR tsr)  
Μέθοδος για τον έλεγχο της πράξης της επιλογής.
- private TSR CARTESIAN\_PRODUCT(TSR firsttsr,TSR secondtsr))  
Μέθοδος για τον έλεγχο της πράξης του καρτεσιανού γινομένου.
- private TSR UNION(TSR firsttsr,TSR secondtsr)  
Μέθοδος για τον έλεγχο της πράξης της ένωσης.
- private TSR INTERSECTION(TSR firsttsr,TSR secondtsr)  
Μέθοδος για τον έλεγχο της πράξης της τομής.
- private TSR DIFFERENCE(TSR firsttsr,TSR secondtsr)  
Μέθοδος για τον έλεγχο της πράξης της διαφοράς.
- public void actionPerformed(ActionEvent ae)  
Μέθοδος για το χειρισμό των ενεργειών του χρήστη.
- public void windowClosing( WindowEvent e )  
Μέθοδος για το χειρισμό των ενεργειών του χρήστη.
- public void windowOpened( WindowEvent e )  
Μέθοδος για το χειρισμό των ενεργειών του χρήστη.
- public void windowClosed( WindowEvent e )  
Μέθοδος για το χειρισμό των ενεργειών του χρήστη.
- public void windowIconified( WindowEvent e )  
Μέθοδος για το χειρισμό των ενεργειών του χρήστη.
- public void windowDeiconified( WindowEvent e )  
Μέθοδος για το χειρισμό των ενεργειών του χρήστη.
- public void windowActivated( WindowEvent e )  
Μέθοδος για το χειρισμό των ενεργειών του χρήστη.
- public void windowDeactivated( WindowEvent e )  
Μέθοδος για το χειρισμό των ενεργειών του χρήστη.
- public CreateQuery(Connection con , String TSRsArray[] , JFrame frame)  
Κατασκευαστής αντικειμένων της κλάσης.

### 5.2.3.29 public class DepictProjectWindow

Με τη βοήθεια της κλάσης DepictProjectWindow δημιουργείται ένα βοηθητικό παράθυρο για την επιλογή των ιδιοτήτων και των μονοπατιών στην πράξη της προβολής.

## Πεδία

- private int maxattribute, maxpath  
Αριθμός ιδιοτήτων και μονοπατιών.
- String chbattname[],pathnamestable[]  
Βοηθητικοί πίνακες.
- private JCheckBox chbattribute[],chbpath[]  
JCheckBoxes για την επιλογή ιδιοτήτων και μονοπατιών.
- private String attname[]  
Πίνακας που περιέχει τα ονόματα των ιδιοτήτων.
- JPanel projectPanel  
Τομέας (JPanel) για την εισαγωγή των συστατικών του πλαισίου στην πράξη της προβολής.
- JLabel attrchoice, pathchoice, orchoice

- Βοηθητικές ετικέτες (JLabel) για την απεικόνιση πληροφοριακού κειμένου.
- JButton Ok1Btn  
Το κουμπί Ok
- private JPanel attrconditionsPanel  
Τομέας (JPanel) για την εισαγωγή των συστατικών του πλαισίου στην πράξη της επιλογής.

#### Μέθοδοι

- public void actionPerformed(ActionEvent ae)  
Μέθοδος για το χειρισμό των ενεργειών του χρήστη.
- public DepictProjectWindow(JFrame frame , TSR tmpTSR)  
Κατασκευαστής αντικειμένων της κλάσης.

#### 5.2.3.30 public class SelectAttributeConditionsWindow

Με τη βοήθεια της κλάσης SelectAttributeConditionsWindow δημιουργείται ένα βοηθητικό παράθυρο για την εισαγωγή των συνθηκών στην πράξη της προβολής.

#### Πεδία

- private Vector vAttributeCondition  
Διάνυσμα που περιέχει τις συνθήκες ιδιοτήτων.
- private Vector vPathCondition  
Διάνυσμα που περιέχει τις συνθήκες μονοπατιών.
- private Choice attroperatorlist  
Choice με τους τελεστές των συνθηκών ιδιοτήτων.
- private Choice attrcondconnectorlist  
Περιέχει τις τιμές AND και OR για τη σύνδεση των συνθηκών ιδιοτήτων.
- private JButton addattrconditionBtn  
Το κουμπί για την εισαγωγή νέας συνθήκης ιδιοτήτων.
- private JButton endattrconditionBtn  
Το κουμπί για τον τερματισμό της εισαγωγής συνθηκών ιδιοτήτων.
- private JLabel variableLabel, operatorLabel, variableorvalueLabel, connectorLabel  
Βοηθητικές ετικέτες (JLabel) για την απεικόνιση πληροφοριακού κειμένου κατά την εισαγωγή των συνθηκών ιδιοτήτων.
- private JTextField variableorvalueTF  
JTextField για την εισαγωγή του δεύτερου όρου των συνθηκών ιδιοτήτων. Μπορεί να είναι είτε μία ιδιότητα είτε μία τιμή.
- private JLabel pathLabel1, pathoperatorLabel, pathLabel2, previouspathconnector  
Βοηθητικές ετικέτες (JLabel) για την απεικόνιση πληροφοριακού κειμένου κατά την εισαγωγή των συνθηκών μονοπατιών.
- private JTextField path2TF  
JTextField για την εισαγωγή του δεύτερου όρου των συνθηκών μονοπατιών. Μπορεί να είναι είτε ένα μονοπάτι είτε μία τιμή.
- private Choice pathoperatorlist  
Choice με τους τελεστές των συνθηκών μονοπατιών.
- private Choice pathcondconnectorlist  
Περιέχει τις τιμές AND και OR για τη σύνδεση των συνθηκών μονοπατιών.
- private JButton addpathconditionBtn  
Το κουμπί για την εισαγωγή νέας συνθήκης μονοπατιών.
- private JButton endpathconditionBtn  
Το κουμπί για τον τερματισμό της εισαγωγής συνθηκών μονοπατιών.

#### Μέθοδοι

- public void actionPerformed(ActionEvent ae)  
Μέθοδος για το χειρισμό των ενεργειών του χρήστη.
- public SelectAttributeConditionsWindow(JFrame frame , TSR tmpTSR)

Κατασκευαστής αντικειμένων της κλάσης.

### 5.2.3.31 *public class HelpTopics*

Η κλάση αυτή υλοποιεί το παράθυρο με τη βοήθεια για κάθε λειτουργία που μπορεί να επιτελέσει η εφαρμογή.

#### **Πεδία**

- private JFrame HelpFrame  
Το πλαίσιο (JFrame) το οποίο εμφανίζεται στην οθόνη.
- private JTextArea textarea  
Περιοχή κειμένου (JTextArea) στην οποία εμφανίζεται η βοήθεια.
- private JScrollPane scrollPane  
Κυλιόμενος τομέας (JScrollPane) που περιέχει την περιοχή κειμένου textarea.
- private JTree tree  
Αντικείμενο της κλάσης JTree.
- private DefaultTreeModel model  
Αντικείμενο της κλάσης DefaultTreeModel.
- private Object info[]  
Πίνακας που περιέχει τις πληροφορίες για τη βοήθεια.
- private static final ImageIcon Open  
Βοηθητική εικόνα.
- private static final ImageIcon Close  
Βοηθητική εικόνα.
- private static final ImageIcon Leaf  
Βοηθητική εικόνα.

#### **Μέθοδοι**

- public HelpTopics()  
Κατασκευαστής αντικειμένων της κλάσης.

### 5.2.3.32 *class SelectionListener*

Η κλάση αυτή υλοποιεί τη χειριστή των επιλογών του χρήστη για τη βοήθεια. Καλείται κάθε φορά που ο χρήστης επιλέγει κάποιο από τα θέματα της βοήθειας.

#### **Πεδία**

#### **Μέθοδοι**

- public void valueChanged(TreeSelectionEvent e)  
Μέθοδος που υλοποιεί την αλλαγή στις πληροφορίες που φαίνονται στο παράθυρο για τη βοήθεια.

### 5.2.3.33 *class OidNode*

Με τη βοήθεια της κλάσης αυτής δημιουργούνται οι κόμβοι στο δέντρο το οποίο εμφανίζεται στην οθόνη του χρήστη με την επιλογή της λειτουργίας της βοήθειας.

#### **Πεδία**

- protected int id  
Ο κωδικός του κόμβου που δημιουργήθηκε

- `protected String name`  
Το όνομα του κόμβου που δημιουργήθηκε

#### **Μέθοδοι**

- `public OidNode(int x,String y)`  
Κατασκευαστής αντικειμένων της κλάσης. Οι παράμετροι `x` και `y` αρχικοποιούν τα αντίστοιχα πεδία της κλάσης.
- `public int getId()`  
Η μέθοδος αυτή επιστρέφει τον κωδικό αριθμό του κόμβου που δημιουργήθηκε.
- `public String toString()`  
Η μέθοδος αυτή επιστρέφει το όνομα του κόμβου που δημιουργήθηκε.
- `public String getName()`  
Όπως και η προηγούμενη μέθοδος, επιστρέφει το όνομα του κόμβου που δημιουργήθηκε.

#### *5.2.3.34 public class About*

Η κλάση αυτή υλοποιεί το παράθυρο το οποίο εμφανίζει διάφορες πληροφορίες για το σύστημα.

#### **Πεδία**

- `JFrame AboutFrame`  
Το πλαίσιο (`JFrame`) το οποίο εμφανίζεται στην οθόνη.
- `JTextArea textarea`  
Περιοχή κειμένου (`JTextArea`) στην οποία εμφανίζονται οι πληροφορίες.
- `JScrollPane scrollPane`  
Κυλιόμενος τομέας (`JScrollPane`) που περιέχει την περιοχή κειμένου `textarea`.
- `ImageIcon image`  
Βοηθητική εικόνα.

#### **Μέθοδοι**

- `public About()`  
Κατασκευαστής αντικειμένων της κλάσης.

#### *5.2.3.35 public class ImportRDFSFile*

Με τη βοήθεια της κλάσης αυτής, δημιουργείται ένα βοηθητικό παράθυρο για την επιλογή ενός από τα διαθέσιμα RDFS αρχεία προς δημιουργία νέων TSR σχημάτων.

#### **Πεδία**

- `private static Connection pConn`  
Η σύνδεση με τη βάση δεδομένων.
- `private JFrame importtrdfsfileFrame`  
Το πλαίσιο (`JFrame`) το οποίο εμφανίζεται στην οθόνη.
- `private JPanel importtrdfsfilePanel`  
Ο τομέας (`JPanel`) για την εισαγωγή των συστατικών του πλαισίου.
- `private JButton OkBtn`  
Το κουμπί `Ok`.
- `private JButton CancelBtn`  
Το κουμπί `Cancel`.
- `private JLabel insertnameLabel, label`  
Βοηθητικές ετικέτες (`JLabel`) για την απεικόνιση πληροφοριακού κειμένου.
- `private DBSchema newDBSchema`  
Αντικείμενο της κλάσης `DBSchema`.

- private XMLschema newXMLschema  
Αντικείμενο της κλάσης XMLschema.
- private JComboBox choose rdfsfile  
JComboBox για την επιλογή του RDFS αρχείου προς έλεγχο.
- private Font labelFont, lightlabelFont  
Fonts για τις ετικέτες.
- private final String filelist[]  
Πίνακας με τα διαθέσιμα RDFS αρχεία.
- private String selection  
Η επιλογή από το JComboBox. Αποτελείται από το όνομα του RDFS αρχείου.
- private GridBagLayout blayout  
Αντικείμενο της κλάσης GridBagLayout (διάταξη δομημένων πλεγμάτων).
- private Document source  
Αντικείμενο της κλάσης Document για το XML αρχείο που θα διαχειριστεί.
- private SAXReader parser  
Ο parser που θα προσπελάσει τα στοιχεία του XML αρχείου.

#### Μέθοδοι

- private void addComponent(Component c, Container p, int row, int column, int width, int height)  
Η μέθοδος αυτή είναι υπεύθυνη για την κατάλληλη τοποθέτηση των συστατικών στο JFrame της εφαρμογής.
- public void actionPerformed(ActionEvent ae)  
Μέθοδος για το χειρισμό των ενεργειών του χρήστη.
- public ImportRDFSFile(Connection con)  
Κατασκευαστής αντικειμένων της κλάσης.

#### 5.2.3.36 *public class InsertFilename*

Με τη βοήθεια της κλάσης αυτής, δημιουργείται ένα βοηθητικό παράθυρο για την επιλογή του μέσου αποθήκευσης και την εισαγωγή του ονόματος του XML αρχείου, όπου θα αποθηκευτούν τα TSR σχήματα που δημιουργήθηκαν από το RDFS αρχείο.

#### Πεδία

- private JPanel newPanel  
Ο τομέας (JPanel) για την εισαγωγή των συστατικών του πλαισίου.
- private JButton Ok1Btn  
Το κουμπί Ok.
- private JButton Cancel1Btn  
Το κουμπί Cancel.
- private JLabel label1  
Βοηθητική ετικέτα (JLabel) για την απεικόνιση πληροφοριακού κειμένου.
- private JTextField insertnameTF  
JTextField για την εισαγωγή του ονόματος του αρχείου.
- private JRadioButton DBbutton  
JRadioButton για την επιλογή αποθήκευσης των νέων TSR σχημάτων στη βάση δεδομένων
- private JRadioButton XMLbutton  
JRadioButton για την επιλογή αποθήκευσης των νέων TSR σχημάτων σε XML αρχείο
- private ButtonGroup DBorXML  
Αντικείμενο της κλάσης ButtonGroup το οποίο ομαδοποιεί τα JRadioButtons.
- private TSR allTSRs[]  
Πίνακας στον οποίο αποθηκεύονται όλα τα TSR σχήματα τα οποία δημιουργούνται από το RDFS αρχείο.



## Μέθοδοι

- `private Vector findPaths(Vector superclass, Vector subclass)`  
Η μέθοδος αυτή είναι υπεύθυνη για την δημιουργία των μονοπατιών των νέων TSR σχημάτων.
- `public void actionPerformed(ActionEvent ae)`  
Μέθοδος για το χειρισμό των ενεργειών του χρήστη.
- `public InsertFilename(JFrame frame)`  
Κατασκευαστής αντικειμένων της κλάσης.

### 5.2.3.37 *public class rdfstest*

Με τη βοήθεια της κλάσης αυτής, υλοποιείται ένα κομμάτι του υποσυστήματος δημιουργίας TSR σχημάτων από RDFS αρχείο. Συγκεκριμένα προσπελαύνεται το RDFS αρχείο και επιλέγονται από αυτό μόνο όσα στοιχεία είναι απαραίτητα για τη δημιουργία των TSR σχημάτων.

## Πεδία

- `private String previous`  
Βοηθητική String μεταβλητή.
- `private ReadRDFFileHandler rgh`  
Αντικείμενο της κλάσης `ReadRDFFileHandler`.

## Μέθοδοι

- `public rdfstest()`  
Κατασκευαστής αντικειμένων της κλάσης (χωρίς ορίσματα). Χρησιμοποιείται για κλήσεις μέσα από την ίδια την κλάση.
- `public rdfstest(String xmlFile, String tagName, String attributeName)`  
Κατασκευαστής αντικειμένων της κλάσης. Διαβάζει από το RDFS αρχείο με όνομα `xmlFile` τα `tags` με όνομα `tagName` και τις ιδιότητες με όνομα `attributeName`.

### 5.2.3.38 *class ReadRDFFileHandler*

Η κλάση αυτή προσπελαύνει το RDFS αρχείο και διαβάζει τα `tags` του.

## Πεδία

- `private String tag`  
Βοηθητική String μεταβλητή.
- `private String attribute`  
Βοηθητική String μεταβλητή.
- `private int count = 0`  
Βοηθητική ακέραια μεταβλητή.
- `public Vector superclass`  
Διάνυσμα στο οποίο αποθηκεύονται όλες οι υπερκλάσεις που υπάρχουν στο RDFS αρχείο.
- `public Vector subclass`  
Διάνυσμα στο οποίο αποθηκεύονται όλες οι υποκλάσεις που υπάρχουν στο RDFS αρχείο.

## Μέθοδοι

- `ReadRDFFileHandler(String tagname, String attributeName)`  
Κατασκευαστής αντικειμένων της κλάσης (χωρίς ορίσματα). Τα ορίσματα `tagname` και `attributeName` αρχικοποιούν τα πεδία `tag` και `attribute` αντίστοιχα.
- `public void startElement(String uri, String localName, String qName, Attributes attributes)`

Η μέθοδος αυτή καλείται αυτόματα από το σύστημα όποτε ξεκινά η ανάγνωση ενός καινούριου tag. Ο βασικός κώδικας της μεθόδου είναι:

```
//Αν το όνομα του tag είναι 'Class' τότε αποθήκευσε στο String previous το περιεχόμενο της ιδιότητας με όνομα attribute.
if (qName.equals("rdfs:Class"))
{
    count++;
    previous = attributes.getValue(attribute);
}

//Αν το όνομα του tag είναι 'subClassOf' τότε αποθήκευσε στο Vector superclass το περιεχόμενο της ιδιότητας με όνομα rdfs:resource και στο Vector subclass το previous.
if (qName.equals("rdfs:subClassOf"))
{
    String tmpstr = attributes.getValue("rdf:resource");

    char tmpchar[] = tmpstr.toCharArray();

    tmpstr = "";
    for (int i = 1;i<tmpchar.length;i++)
        tmpstr += tmpchar[i];
    superclass.add(tmpstr);
    subclass.add(previous);
}
```

Στο τέλος υπάρχουν στο διάνυσμα superclass διάφορα ονόματα κλάσεων και στο διάνυσμα subclass τα ονόματα των υποκλάσεων τους. Με μία απλή διάσχιση και των δύο διανυσμάτων μπορούμε πολύ εύκολα να βρούμε τα μονοπάτια που σχηματίζονται.

### 5.2.3.39 *public class ImportTuples*

Με τη βοήθεια της κλάσης ImportTuples δημιουργείται ένα βοηθητικό παράθυρο για την επιλογή ενός από τα αποθηκευμένα TSR σχήματα, για προσθήκη νέων εγγραφών στον κόμβο – αντικείμενο του.

#### **Πεδία**

- private static Connection pConn  
Η σύνδεση με τη βάση δεδομένων.
- private JFrame importtuplesFrame  
Το πλαίσιο (JFrame) το οποίο εμφανίζεται στην οθόνη.
- private JPanel importtuplesPanel  
Ο τομέας (JPanel) για την εισαγωγή των συστατικών του πλαισίου.
- private JButton OkBtn  
Το κουμπί Ok.
- private JButton CancelBtn  
Το κουμπί Cancel.
- private JLabel label1, label2  
Βοηθητικές ετικέτες (JLabel) για την απεικόνιση πληροφοριακού κειμένου.
- private JLabel descriptionLabel  
Βοηθητική ετικέτα (JLabel) για την απεικόνιση της περιγραφής του TSR σχήματος που έχει επιλεγεί.
- private DBschema newDBschema  
Αντικείμενο της κλάσης DBschema.
- private XMLschema newXMLschema  
Αντικείμενο της κλάσης XMLschema.
- private JComboBox choosetsr  
JComboBox για την επιλογή του TSR στο οποίο θα προστεθούν νέες εγγραφές.
- private Font labelFont, labelFont2, lightlabelFont  
Fonts για τις ετικέτες.

- private String selectedfile  
Το όνομα του αρχείου του TSR σχήματος που επιλέχθηκε.
- private String selectedtsr  
Το όνομα του TSR σχήματος που επιλέχθηκε.
- private String selection  
Η επιλογή από το JComboBox. Αποτελείται από το όνομα του αρχείου και το όνομα του TSR σχήματος.
- private Document source  
Αντικείμενο της κλάσης Document για τη διαχείριση του XML αρχείου.
- private SAXReader parser  
Ο parser που θα προσπελάσει τα στοιχεία του XML αρχείου.
- private GridBagLayout blayout  
Αντικείμενο της κλάσης GridBagLayout (διάταξη δομημένων πλεγμάτων).
- private int error = 0  
Η μεταβλητή αυτή παίρνει την τιμή 1 όταν συμβεί οποιοδήποτε λάθος.

### Μέθοδοι

- public String[] findAllTSR()  
Μέθοδος για την εύρεση και την αποθήκευση σε πίνακα όλων των TSR σχημάτων που είναι αποθηκευμένα είτε στη βάση δεδομένων είτε σε κάποιο XML αρχείο.
- private void addComponent(Component c, Container p, int row, int column, int width, int height)  
Η μέθοδος αυτή είναι υπεύθυνη για την κατάλληλη τοποθέτηση των συστατικών στο JFrame της εφαρμογής.
- private static ResultSet execQuery(String strQuery)  
Εκτέλεση του ερωτήματος strQuery στη βάση δεδομένων.
- public void actionPerformed(ActionEvent ae)  
Μέθοδος για το χειρισμό των ενεργειών του χρήστη.
- public ImportTuples(Connection con)  
Κατασκευαστής αντικειμένων της κλάσης.

### 5.2.3.40 public class InsertTuples

Με τη βοήθεια της κλάσης InsertTuples δημιουργείται ένα βοηθητικό παράθυρο για την επιλογή από το χρήστη των νέων εγγραφών και ενδεχομένως των νέων ιδιοτήτων του TSR σχήματος το οποίο έχει επιλεγεί για αυτήν την αλλαγή.

### Πεδία

- private static Connection pConn  
Η σύνδεση με τη βάση δεδομένων.
- private JPanel insertTuplesPanel  
Ο τομέας (JPanel) για την εισαγωγή των συστατικών του πλαισίου.
- private JButton Ok1Btn  
Το κουμπί Ok.
- private JButton Cancel1Btn  
Το κουμπί Cancel.
- private JLabelinsertdatanameLabel, label1, fromfileLabel, attrLabel, datatableLabel, insertqueryLabel  
Βοηθητικές ετικέτες (JLabel) για την απεικόνιση πληροφοριακού κειμένου.
- private DBSchema newDBSchema  
Αντικείμενο της κλάσης DBSchema.
- private TSR tsr  
Αντικείμενο της κλάσης TSR.
- private XMLSchema newXMLSchema  
Αντικείμενο της κλάσης XMLSchema.
- private Font labelFont, lightlabelFont  
Fonts για τις ετικέτες.

- private JComboBox choosetsr  
JComboBox για την επιλογή του TXT αρχείου από το οποίο θα αναγνωσθούν οι νέες εγγραφές.
- private String selectedtxtfile  
Το όνομα του TXT αρχείου που επιλέχθηκε. Είναι το περιεχόμενο της επιλογής από JComboBox.
- private String filename  
Το όνομα του αρχείου του TSR σχήματος στο οποίο θα γίνει η προσθήκη των εγγραφών.
- private String tsrname  
Το όνομα του TSR σχήματος στο οποίο θα γίνει η προσθήκη των νέων εγγραφών.
- private JRadioButton yesbutton  
JRadioButton για την επιλογή ανάγνωσης των νέων εγγραφών από κάποιο TXT αρχείο.
- private JRadioButton nobutton  
JRadioButton για την επιλογή ανάγνωσης των νέων εγγραφών από κάποιο πίνακα αποθηκευμένο στη βάση δεδομένων.
- private ButtonGroup yesorno  
Αντικείμενο της κλάσης ButtonGroup το οποίο ομαδοποιεί τα JRadioButtons.
- private JSeparator separator  
JSeparator που δημιουργεί μία διαχωριστική γραμμή.
- private JTextField insertqueryTF  
JTextField για την εισαγωγή της SQL ερώτησης με βάση την οποία θα δημιουργηθούν οι νέες εγγραφές και οι νέες ιδιότητες.
- private GridBagLayout blayout  
Αντικείμενο της κλάσης GridBagLayout (διάταξη δομημένων πλεγμάτων).
- private String allAttributes[]  
Βοηθητικός πίνακας ο οποίος κρατάει τα ονόματα των ιδιοτήτων που επιλέχθηκαν από το χρήστη.

### Μέθοδοι

- private boolean isNumber(Vector v)  
Η μέθοδος αυτή ελέγχει αν τα στοιχεία του διανύσματος v είναι αριθμητικά ή αλφαριθμητικά.
- private int findType(String attributename)  
Η μέθοδος αυτή βρίσκει τον τύπο της ιδιότητας attributename (text ή float).
- private void insertTuplesinTSR(Vector attr, Vector tuple, String filename, String tsrname)  
Η μέθοδος αυτή αναλαμβάνει την εισαγωγή των νέων εγγραφών στο TSR σχήμα με όνομα tsrname και όνομα αρχείου filename.
- private int getRowSize(ResultSet rs)  
Εύρεση του μεγέθους μιας γραμμής του rs.
- public void actionPerformed(ActionEvent ae)  
Μέθοδος για το χειρισμό των ενεργειών του χρήστη.

Αν ο χρήστης επιλέξει το OkBtn πρέπει να διαβασθεί το txt αρχείο που έχει επιλέξει και να αποθηκευτεί στη βάση δεδομένων στον πίνακα data. Ο κώδικας φαίνεται παρακάτω:

```
//Ο πίνακας data διαγράφεται από τη βάση δεδομένων
execQuery("DROP TABLE data;");
//διαβάζεται το txt αρχείο που επιλέχθηκε (το όνομα του βρίσκεται στην μεταβλητή
//selectedtxtfile
DataInputStream in =new DataInputStream(new
BufferedInputStream(new
FileInputStream("org/_3pq/jgrapht/my_diplo/sources/insert_files
/"+selectedtxtfile)));
```

```
String s;
int linecounter = 0;
```

```
//Το περιέχει το περιεχόμενο της γραμμής που διαβάστηκε από το αρχείο
while((s = in.readLine()) != null)
{
```

```

//αν πρόκειται για την πρώτη γραμμή τότε διαβάζονται ιδιότητες
    if (linecounter == 0)
    {
        String query = "";
        StringTokenizer st = new StringTokenizer(s, "#");
        int counter = st.countTokens();

//στον πίνακα AllAttributes αποθηκεύονται όλες οι ιδιότητες που περιέχει το αρχείο
        allAttributes = new String[counter];

        for (int m = 0;m<counter;m++)
        {
            String last = st.nextToken();
            allAttributes[m] = last;
            if (m == 0)
                query = "CREATE TABLE data("+last+"
                    text";
            else
                query += ", "+last+" text";
        }
//εκτελείται η ερώτηση που περιέχεται στο string query και με την οποία //δημιουργείται στη
//βάση δεδομένων ο πίνακας data με γνωρίσματα τις ιδιότητες του //txt αρχείου
        execQuery(query + " ");
    }
    else
    {
//αν η γραμμή δεν είναι η πρώτη τότε πρέπει να διαβαστούν εγγραφές
        String query = "";
        StringTokenizer st = new StringTokenizer(s, "#");
        int counter = st.countTokens();

        String tmpquery = "";

        for (int m = 0;m<counter;m++)
        {
            String last = st.nextToken();
            if (m == 0)
                tmpquery = ""+last+"";
            else
                tmpquery +=", "+last+"";
        }
//εισάγετε στον πίνακα data η πλειάδα που περιέχει την εγγραφή που διαβάστηκε
        execQuery("INSERT INTO DATA VALUES
            ("+tmpquery+");");
    }
    linecounter++;
}
in.close();

```

- `public InsertTuples(Connection con,String file,String tsr,JFrame frame)
 ImportTuples(Connection con)
 Κατασκευαστής αντικειμένων της κλάσης.`

### 5.2.3.41 *public class startscreen*

Η κλάση αυτή δημιουργεί το αρχικό παράθυρο του συστήματος. Περιέχει τη μέθοδο main από την οποία ξεκινά η εκτέλεση του προγράμματος.

## Πεδία

- `private static Connection pConn`  
Η σύνδεση με τη βάση δεδομένων.
- `private JFrame startFrame`  
Το πλαίσιο (JFrame) το οποίο εμφανίζεται στην οθόνη.
- `private JPanel startPanel`  
Ο τομέας (JPanel) για την εισαγωγή των συστατικών του πλαισίου.
- `private JMenuItem newquery, viewtsr, loadtsr, loadquery, quit, change, delete, save2db, save2file, helptopics, about, importrdf, importtuples`  
Στοιχεία menu.
- `private JMenu menu1, menu2, menu3`  
Αναπτυσσόμενα menu που περιέχουν τα προηγούμενα menu.
- `private JMenuBar menuBar`  
Υποδοχέας που περιέχει τα JMenu..
- `private GridBagLayout blayout`  
Αντικείμενο της κλάσης GridBagLayout (διάταξη δομημένων πλεγμάτων).
- `private int howmanyimportquery`  
Βοηθητική ακέραια μεταβλητή.

## Μέθοδοι

- `public static void main(String args[])`  
Η αρχική μέθοδος του συστήματος.
- `private static void connectToRDBMS()`  
Με τη μέθοδο αυτή επιτυγχάνεται η σύνδεση με τη βάση δεδομένων.
- `private static ResultSet execQuery(String strQuery)`  
Εκτέλεση του ερωτήματος strQuery στη βάση δεδομένων.
- `public public startscreen()`  
Κατασκευαστής αντικειμένων της κλάσης.

Σημειώνεται η κλάση αυτή περιέχει και τις κλάσεις με ονόματα Handler1, ..., Handler13. Όποτε επιλεγθεί μία από τις εφαρμογές του συστήματος καλείται ο κατάλληλος κατασκευαστής της κατάλληλης κλάσης και ξεκινά η εκτέλεση της εφαρμογής.

# 6

## *Έλεγχος*

Στο κεφάλαιο αυτό παρουσιάζεται ο έλεγχος του συστήματος μέσω ενός λεπτομερούς σεναρίου εκτέλεσης.

### *6.1 Μεθοδολογία Ελέγχου*

Ο έλεγχος του συστήματος πραγματοποιείται με χρήση σεναρίων λειτουργίας τα οποία χρησιμοποιούν όλες τις λειτουργίες του συστήματος. Τα αποτελέσματα κρίνονται σε σχέση με τα αναμενόμενα και με αυτόν τον τρόπο αξιολογείται η συμπεριφορά και η απόδοση του συστήματος.

### *6.2 Αναλυτική παρουσίαση ελέγχου*

Στην ενότητα αυτή παρουσιάζεται ένα σενάριο χρήσης του συστήματος που υλοποιήθηκε. Σημειώνεται ότι με το συγκεκριμένο παράδειγμα δε φαίνονται όλες οι εφαρμογές του συστήματος αλλά οι πιο σημαντικές από αυτές. Για περισσότερες λεπτομέρειες σχετικά με

όλες τις εφαρμογές του συστήματος, ο αναγνώστης παραπέμπεται στο εγχειρίδιο χρήσης του συστήματος.

Αρχικά θεωρούμε ένα RDFS αρχείο με όνομα `books.rdfs` το οποίο μεταξύ άλλων περιέχει τις παρακάτω δηλώσεις κλάσεων και υπερκλάσεων:

```
<rdfs:Class rdf:ID="Computers"/>
<rdfs:Class rdf:ID="Languages"/>

<rdfs:Class rdf:ID="C++"/>
<rdfs:Class rdf:ID="Cooking"/>

<rdfs:Class rdf:ID="Beverages"/>
<rdfs:Class rdf:ID="Beermaking"/>
<rdfs:Class rdf:ID="Beer"/>

<rdfs:Class rdf:ID="Languages">
  <rdfs:subClassOf rdf:resource="#Computers"/>
</rdfs:Class>

<rdfs:Class rdf:ID="C++">
  <rdfs:subClassOf rdf:resource="#Languages"/>
</rdfs:Class>

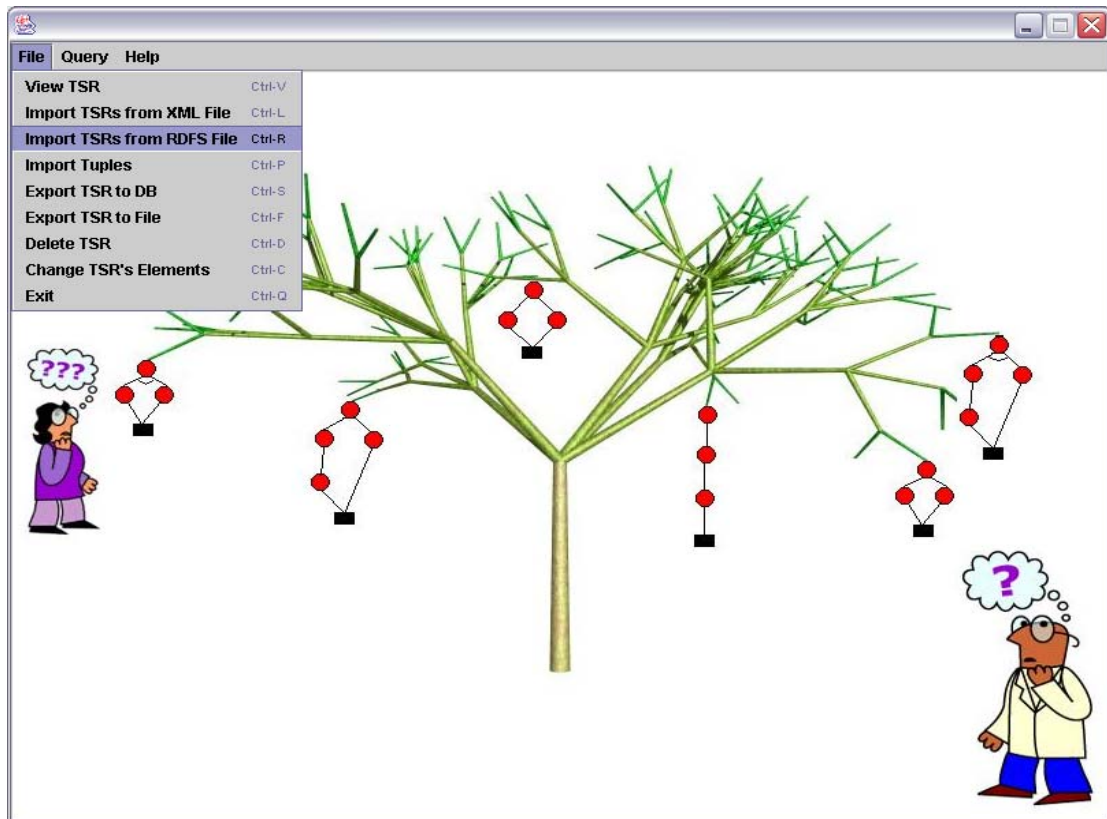
<rdfs:Class rdf:ID="Beverages">
  <rdfs:subClassOf rdf:resource="#Cooking"/>
</rdfs:Class>

<rdfs:Class rdf:ID="Beermaking">
  <rdfs:subClassOf rdf:resource="#Beverages"/>
</rdfs:Class>

<rdfs:Class rdf:ID="Beer">
  <rdfs:subClassOf rdf:resource="#Beverages"/>
</rdfs:Class>
```

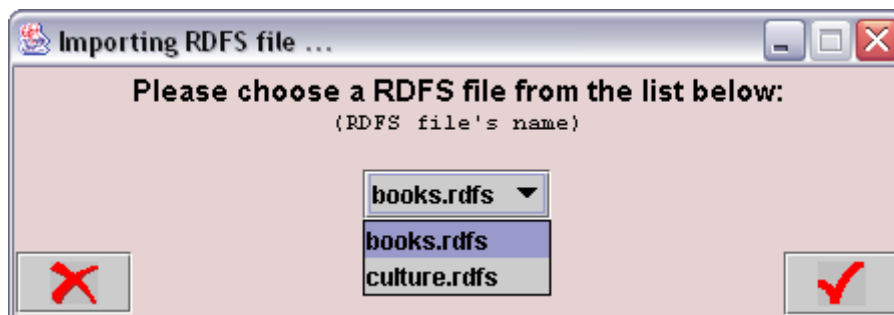
Όπως φαίνεται στο Σχήμα 6.1, από την αρχική οθόνη της εφαρμογής επιλέγουμε τη λειτουργία `'Import TSRs from RDFS file'` (δημιουργία TSR σχημάτων από RDFS αρχείο) ώστε από το προηγούμενο RDFS αρχείο να δημιουργήσουμε νέα TSR σχήματα που θα χρησιμοποιηθούν στη συνέχεια.





Σχήμα 6.1

Το παράθυρο που εμφανίζεται στη συνέχεια είναι αυτό που φαίνεται στο Σχήμα 6.2 και από το οποίο γίνεται η επιλογή του RDFS αρχείου από το οποίο θα προκύψουν τα νέα TSR σχήματα.



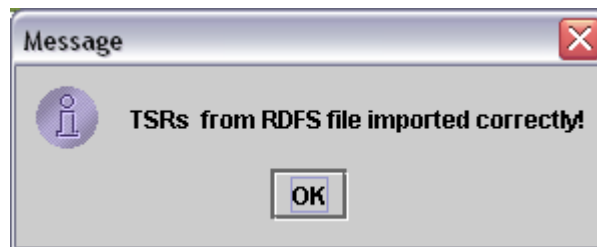
Σχήμα 6.2

Μετά την επιλογή του RDFS αρχείου εμφανίζεται το παράθυρο του Σχήματος 6.3 με το οποίο επιλέγεται το αποθηκευτικό μέσο στο οποίο θα αποθηκευτούν τα TSR σχήματα που δημιουργήθηκαν, ενώ συγχρόνως εισάγεται και το όνομα του XML αρχείου τους (αυτό στο οποίο θα αποθηκευτούν αν επιλεγθεί αποθήκευση σε XML αρχείο ή αυτό από το οποίο προέρχονται αν επιλεγθεί αποθήκευση στη βάση δεδομένων του συστήματος).



Σχήμα 6.3

Σύμφωνα με το σενάριο μας, επιλέχθηκε τα νέα TSR σχήματα να αποθηκευτούν στο XML αρχείο με όνομα powells.xml. Το επιβεβαιωτικό μήνυμα της σωστής δημιουργίας και αποθήκευσης των TSR σχημάτων είναι αυτό του Σχήματος 6.4:



Σχήμα 6.4

Ανοίγοντας το XML αρχείο powells.xml στο οποίο αποθηκεύτηκαν τα TSR σχήματα παίρνουμε το παρακάτω περιεχόμενο:

```
<root>
<tsr name="C++">
  <or>
    <and>/Computers/Languages/C++</and>
  </or>
  <description></description>
  <item name="_C++">
  </item>
</tsr>
<tsr name="Beermaking">
  <or>
    <and>/Cooking/Beverages/Beermaking</and>
  </or>
  <description></description>
  <item name="_Beermaking">
  </item>
</tsr>
```

```

<tsr name="Beer">
  <or>
    <and>/Cooking/Beverages/Beer</and>
  </or>
  <description></description>
  <item name="_Beer">
    </item>
</tsr>
</root>

```

Παρατηρούμε ότι τα TSR σχήματα που δημιουργήθηκαν αποτελούνται από μία OR συνιστώσα με ένα μονοπάτι, ενώ οι κόμβοι – αντικείμενα τους δεν περιέχουν ούτε ιδιότητες ούτε εγγραφές. Συγκρίνοντας, λοιπόν, το περιεχόμενο του XML αρχείου με τα αναμενόμενα αποτελέσματα από την εφαρμογή δημιουργίας νέων TSR σχημάτων από RDFS αρχεία παρατηρούμε ότι αυτά συμπίπτουν.

Επίσης θεωρούμε ένα XML αρχείο με όνομα half.xml το οποίο βρίσκεται αποθηκευμένο (όπως και το RDFS αρχείο) στον φάκελο 'org/\_3pq/jgraph/my\_diplo/sources/insert\_files'. Το περιεχόμενο του αρχείου αυτού φαίνεται παρακάτω:

```

<root>
<tsr name="Beer">
  <or>
    <and>/Cooking/Beverages/Beer</and>
  </or>
  <description>'Books for beer from
www.half.com'</description>
  <item name="Beer books">
    <attribute name="title" type="text"/>
    <attribute name="author" type="text"/>
    <attribute name="price" type="float"/>
    <attribute name="quality" type="text"/>
    <attribute name="year" type="int"/>
    <attribute name="discount" type="text"/>
    <tuple>'The new Complete Joy of Home Brewing', 'Charlie
Papazian', '1.25', 'Paperback', '1991', '90%'</tuple>
    <tuple>'Better Beer and How to Brew it', 'Reese',
'0.75', 'Paperback', '1981', '92%'</tuple>
    <tuple>'Under the Influence', 'Hernon/Ganey',
'0.75', 'Paperback', '1994', '95%'</tuple>
    <tuple>'Homebrewing for Dummies', 'Marty Nachel',
'12.15', 'Paperback', '1997', '39%'</tuple>
    <tuple>'Home Brewer's Gold', 'Charlie Papazian',
'5.98', 'Paperback', '1997', '50%'</tuple>
    <tuple>'More Homebrew Favorites', 'Mary Stevens',
'6.26', 'Paperback', '1997', '58%'</tuple>
  </item>
</tsr>
</root>

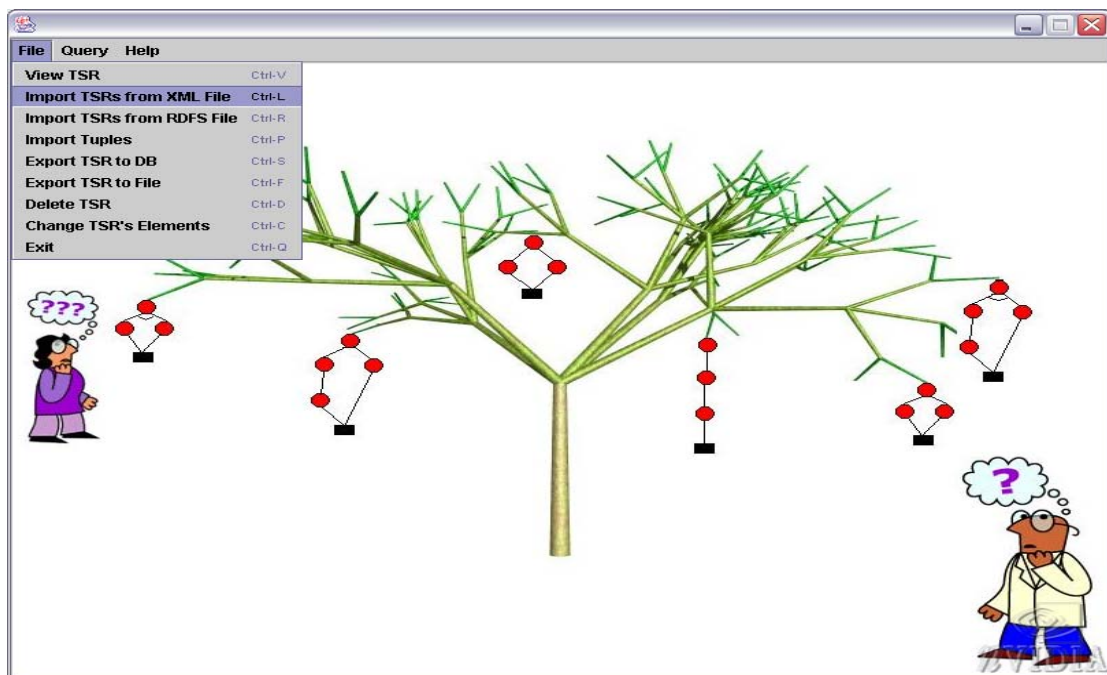
```

```

        </item>
</tsr>
<tsr name="CProgrammingLanguages">
  <or>
    <and>/Computers/Programming Languages/C,C++,Turbo
C</and>
  </or>
  <description>'Books for C,C++,Turbo C from
www.half.com'</description>
  <item name="C,C++,Turbo C books">
    <attribute name="title" type="text"/>
    <attribute name="author" type="text"/>
    <attribute name="price" type="float"/>
    <attribute name="quality" type="text"/>
    <attribute name="year" type="int"/>
    <tuple>'C How to Program', 'Harvey Deitel',
'34.50', 'Paperback', '2002'</tuple>
    <tuple>'The C Programming Language', 'Brian
Kernigham', '20.00', 'Paperback', '1988'</tuple>
  </item>
</tsr>
</root>

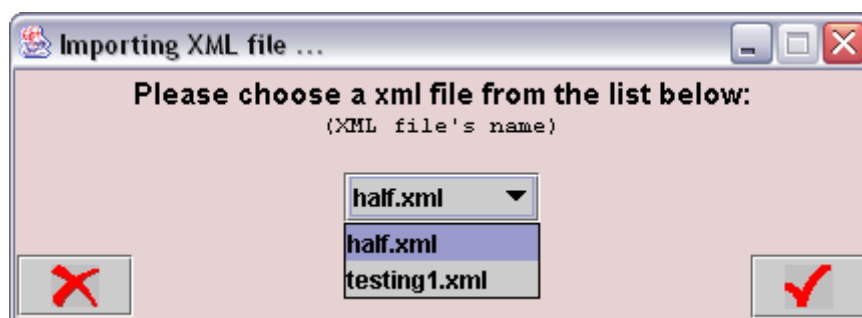
```

Αυτό το XML αρχείο περιέχει δύο TSR σχήματα. Επιλέγοντας την εφαρμογή 'Import TSRs from XML file' (εισαγωγή TSR σχημάτων από XML αρχείο) όπως φαίνεται στο Σχήμα 6.5, το αρχείο αυτό ελέγχεται για την ορθότητα του και αν δεν προκύψει κανένα σφάλμα μεταφέρεται στον φάκελο 'org/\_3pq/jgraph/ny\_diplo/sources' ώστε να μπορεί να χρησιμοποιηθεί από άλλες εφαρμογές του συστήματος.



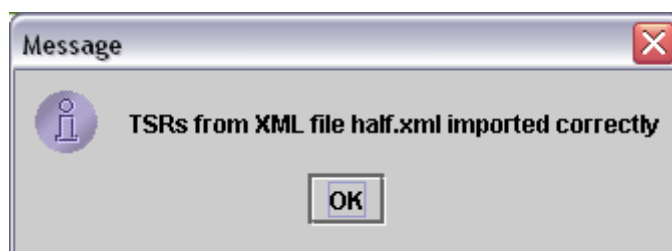
Σχήμα 6.5

Στην οθόνη εμφανίζεται το παράθυρο του Σχήματος 6.6 από το οποίο επιλέγεται το XML αρχείο το οποίο θα ελεγχθεί.



Σχήμα 6.6

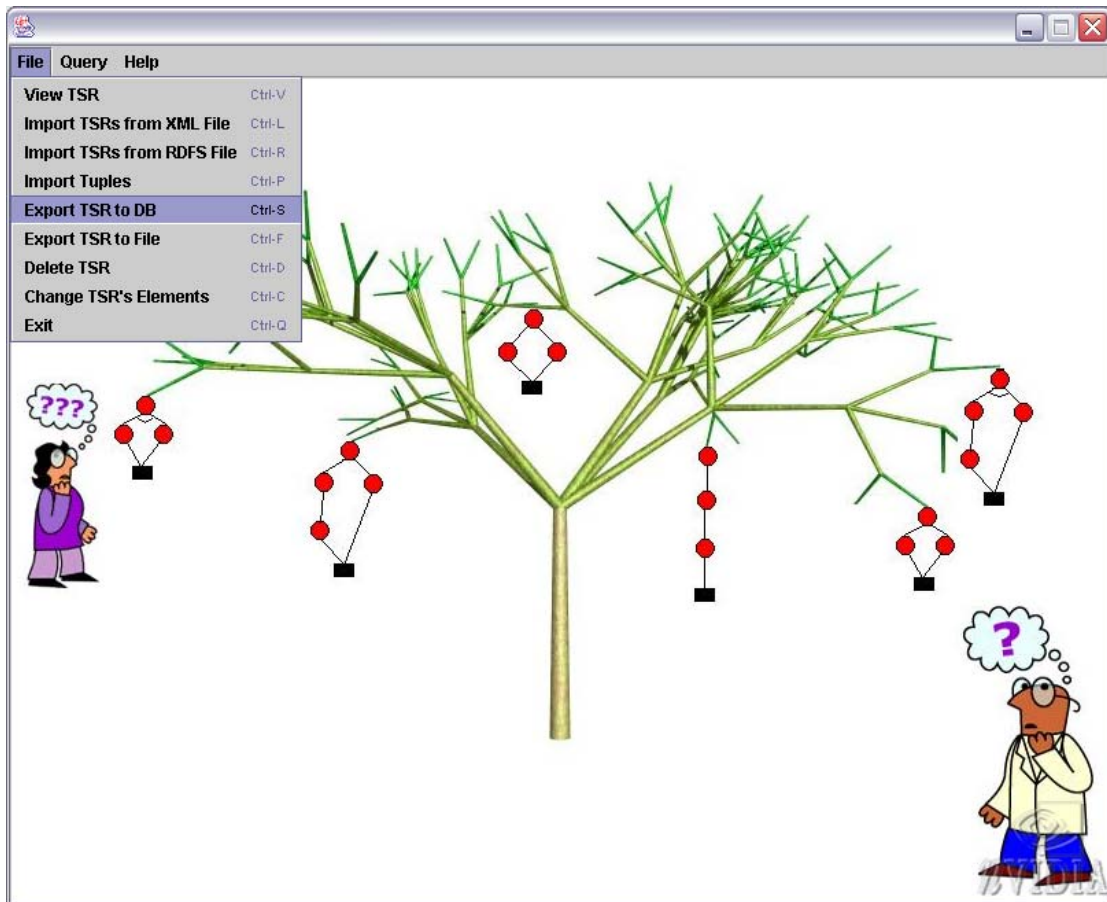
Το XML αρχείο ελέγχεται και δεν εντοπίζεται κανένα σφάλμα οπότε εμφανίζεται στην οθόνη το επιβεβαιωτικό παράθυρο του Σχήματος 6.7:



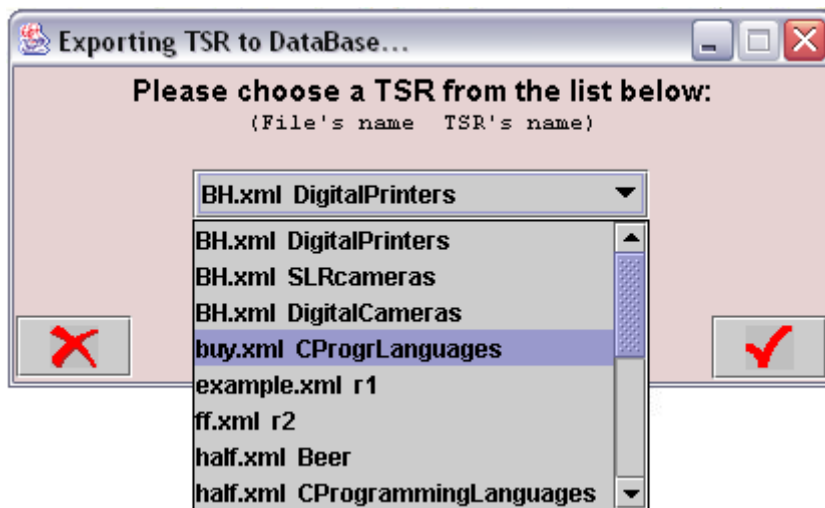
Σχήμα 6.7

Τέλος θεωρούμε ακόμα ένα TSR σχήμα με όνομα `CProgrLanguages` το οποίο βρίσκεται αποθηκευμένο στο XML αρχείο με όνομα `buy.xml`. Επιλέγοντας την εφαρμογή 'Export TSR to DB' (Μεταφορά του TSR σχήματος στη βάση δεδομένων), όπως φαίνεται στο Σχήμα 6.8, και ακολουθώντας στη συνέχεια μια συγκεκριμένη διαδικασία το TSR σχήμα αποθηκεύεται τελικά στη βάση δεδομένων.

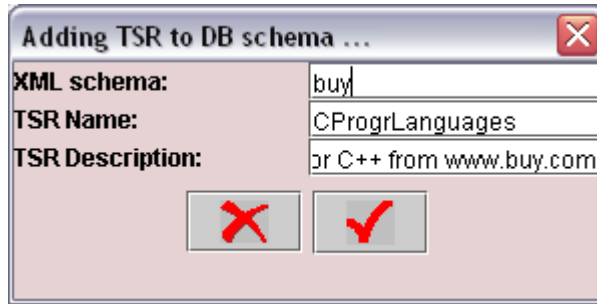
Μετά την επιλογή της συγκεκριμένης εφαρμογής εμφανίζεται το παράθυρο του Σχήματος 6.9 με τη βοήθεια του οποίου επιλέγεται το TSR σχήμα το οποίο πρέπει να αλλάξει μέσο αποθήκευσης. Μετά την επιλογή του επιθυμητού TSR σχήματος εμφανίζεται στην οθόνη το παράθυρο εισαγωγής των απαραίτητων στοιχείων για την αποθήκευση ενός TSR σχήματος στη βάση δεδομένων το οποίο φαίνεται στο Σχήμα 6.10. Τα στοιχεία αυτά είναι το όνομα του TSR σχήματος, το όνομα του αρχείου από το οποίο προέρχεται και μία σύντομη περιγραφή του. Μετά την ολοκλήρωση της εισαγωγής των στοιχείων αυτών και εάν δε συμβεί κάποιο σφάλμα, εμφανίζεται στην οθόνη το παράθυρο με το επιβεβαιωτικό μήνυμα του Σχήματος 6.11.



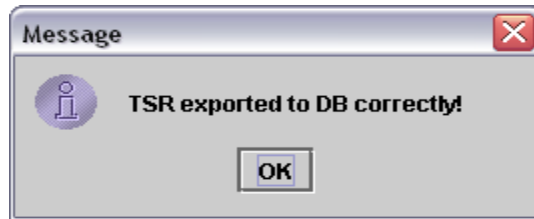
Σχήμα 6.8



Σχήμα 6.9

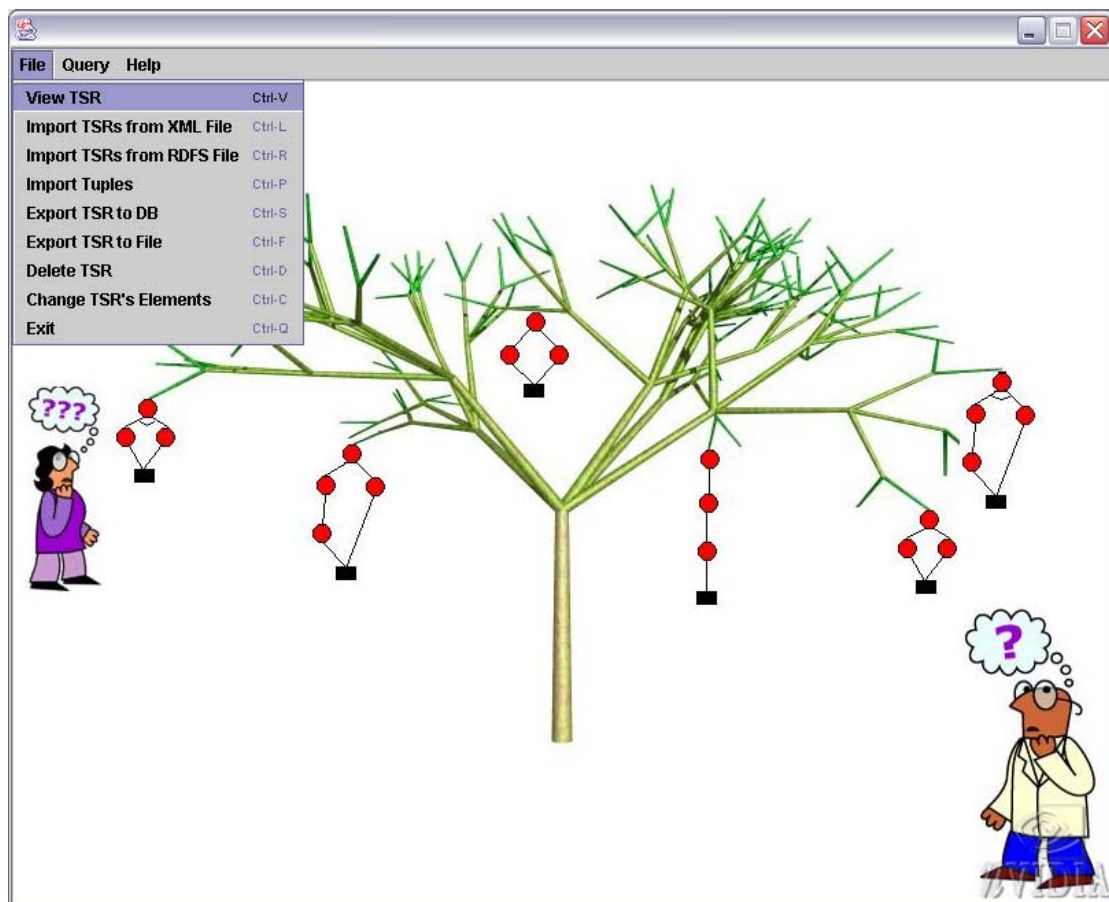


Σχήμα 6.10

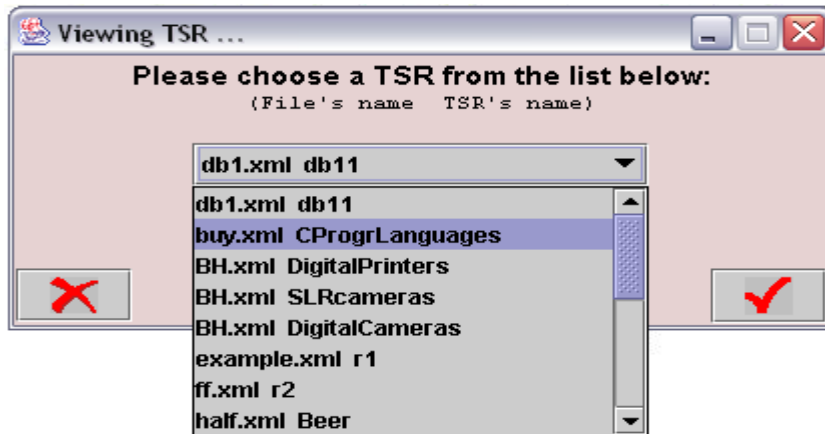


Σχήμα 6.11

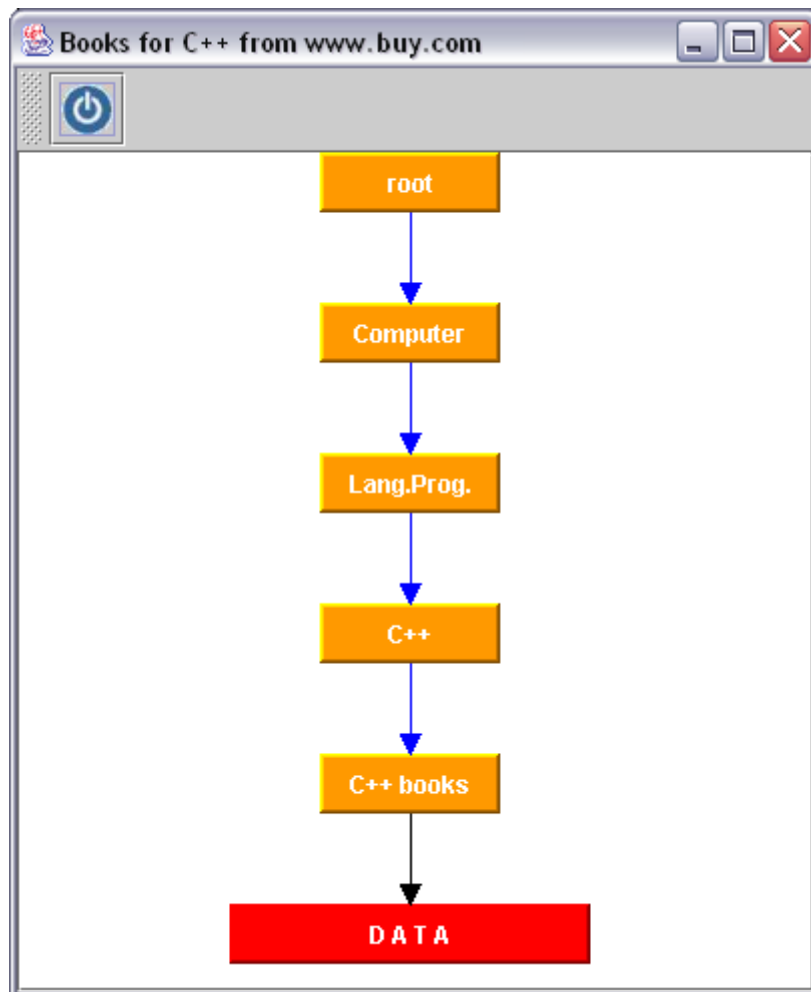
Επιλέγοντας όπως φαίνεται στο Σχήμα 6.12, την εφαρμογή 'View TSR' (Απεικόνιση TSR σχήματος), μπορούμε να επιλέξουμε το συγκεκριμένο TSR σχήμα, όπως φαίνεται στο Σχήμα 6.13, και να το δούμε σε γραφική μορφή όπως φαίνεται στο Σχήμα 6.14.



Σχήμα 6.12



Σχήμα 6.13



Σχήμα 6.14

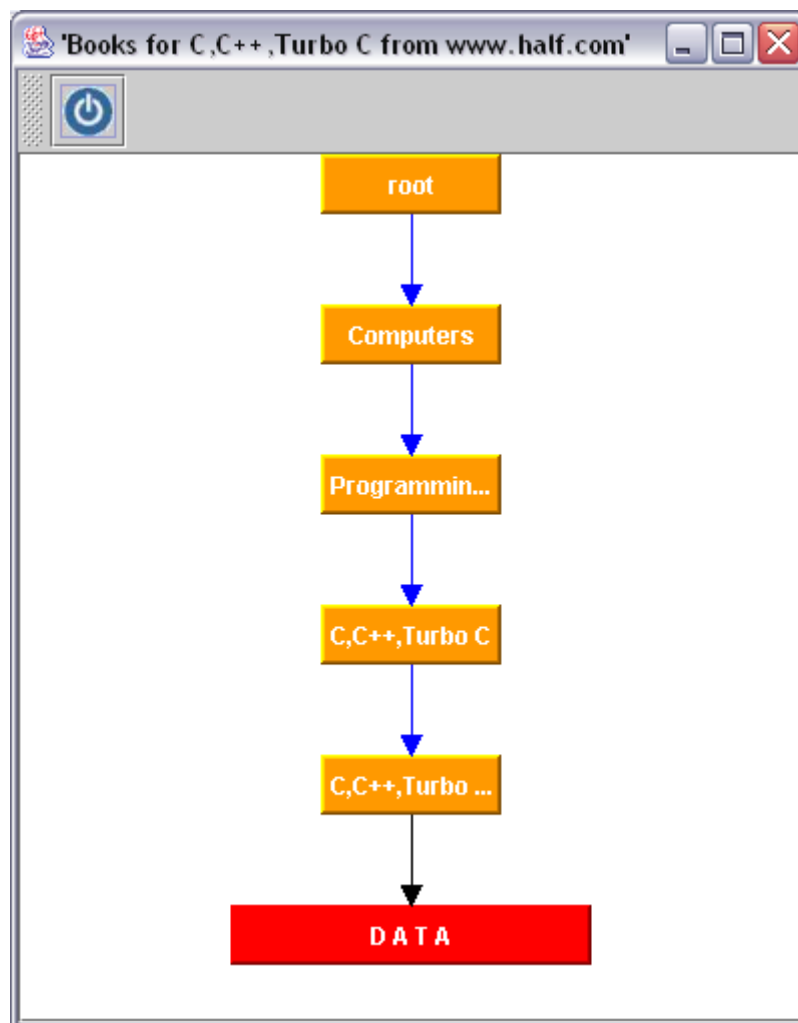
Οι ιδιότητες και οι εγγραφές αυτού του TSR σχήματος εμφανίζονται στην οθόνη του υπολογιστή αν ο χρήστης επιλέξει με το ποντίκι τον τελευταίο κόμβο με όνομα DATA, και φαίνονται στο Σχήμα 6.15:



title	author	price	quality
C++ How to Program	Harvey Deitel	58	Trade Paper
Design and Evolution of C++	Bjarne Stroustrup	45	Trade Paper
Object Oriented Prog in C++	Robert Lafore	27.95	Trade Paper
Numerical Recipes Example Book (C++)	William Vetterling	18.93	Trade Paper
C How to Program	Harvey Deitel	34.5	Paperback
Standard C++ Bible		32.44	
Data structures with C++		12.68	
The C Programming Language	Brian Kernigham	17.98	Paperback

Σχήμα 6.15

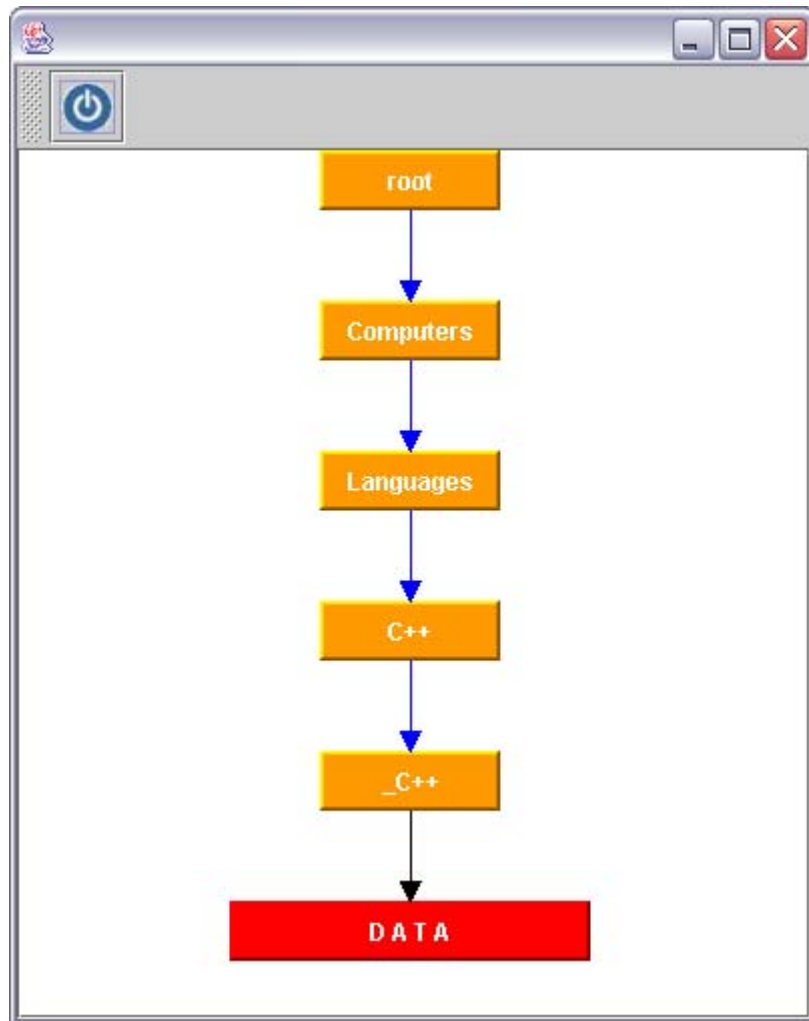
Ακολουθώντας αντίστοιχη διαδικασία, μπορούμε να επιλέξουμε να δούμε σε γραφική μορφή τόσο το TSR σχήμα που προέκυψε από τον έλεγχο του XML αρχείου `half.xml`, Σχήματα 6.16 και 6.17, όσο και το TSR σχήμα που προέκυψε από το RDFS αρχείο `books.rdfs`, Σχήμα 6.18.



Σχήμα 6.16

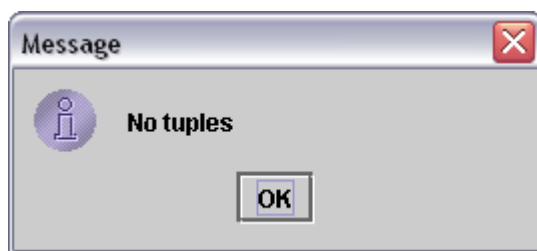
title	author	price	quality	year
C How to Program	Harvey Deitel	34.50	Paperback	2002
The C Programming Language	Brian Kernigham	20.00	Paperback	1988

Σχήμα 6.17



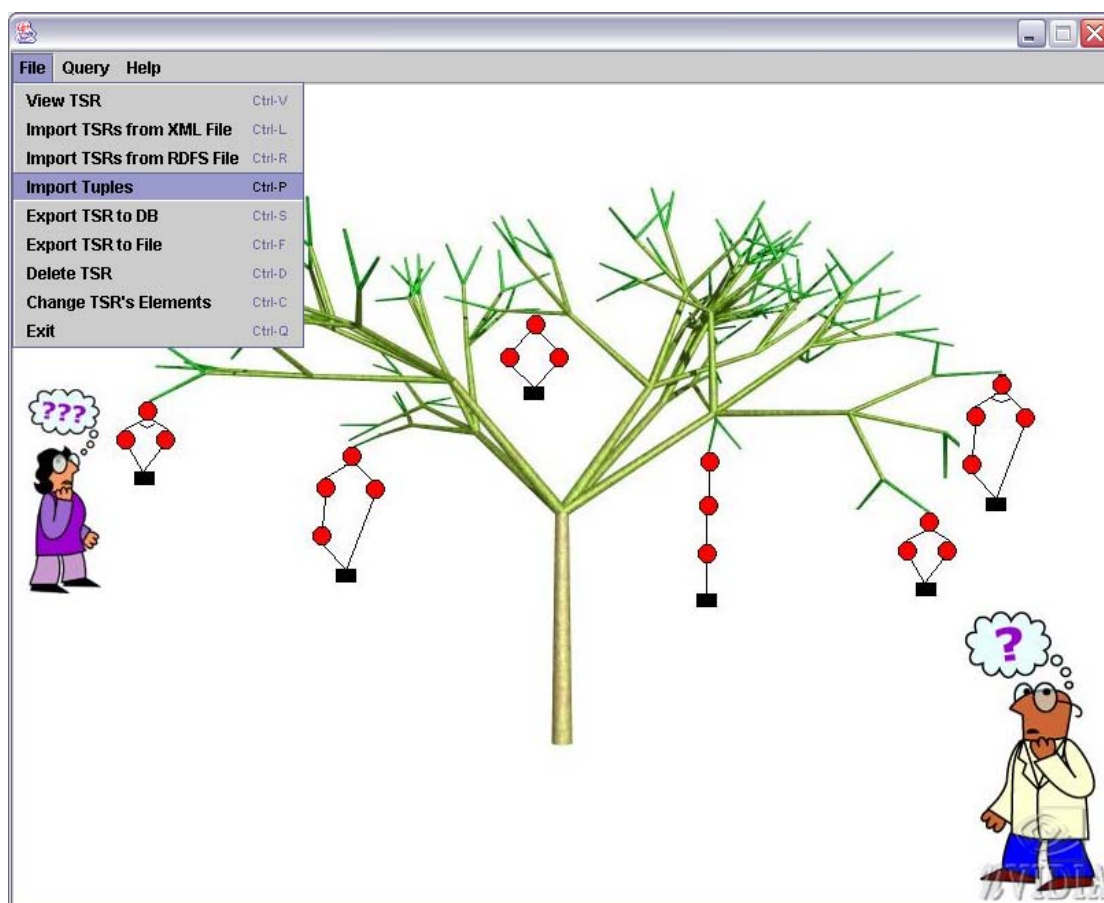
Σχήμα 6.18

Όπως σε όλα τα TSR σχήματα που δημιουργούνται από RDFS αρχεία, έτσι και στο συγκεκριμένο, ο κόμβος – αντικείμενο του δεν περιέχει ούτε ιδιότητες ούτε εγγραφές. Αυτό επιβεβαιώνεται αν ο χρήστης επιλέξει με το ποντίκι τον κόμβο με όνομα DATA οπότε και θα εμφανιστεί στην οθόνη το μήνυμα του Σχήματος 6.19.



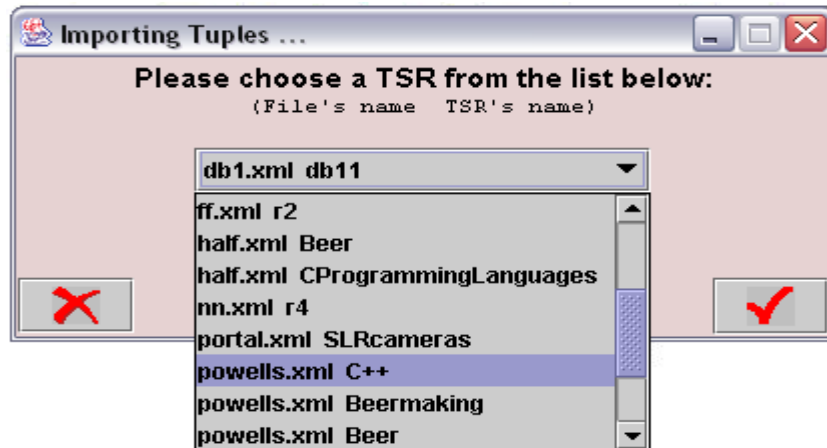
Σχήμα 6.19

Επιλέγοντας, λοιπόν την εφαρμογή 'Import Tuples' (Εισαγωγή νέων εγγραφών σε TSR σχήματα), όπως φαίνεται στο Σχήμα 6.20 μπορούμε να γεμίσουμε με ιδιότητες και εγγραφές τον κόμβο – αντικείμενο κάποιου TSR σχήματος. Υπενθυμίζεται ότι ο κόμβος – αντικείμενο του TSR σχήματος μπορεί να είναι κενός (οπότε θα δημιουργηθούν νέες ιδιότητες και νέες εγγραφές) ή μπορεί να περιέχει ήδη κάποιες ιδιότητες ή ακόμα και εγγραφές (οπότε αν οι νέες ιδιότητες είναι ίδιες με αυτές που ήδη υπάρχουν τότε θα προστεθούν οι νέες εγγραφές).



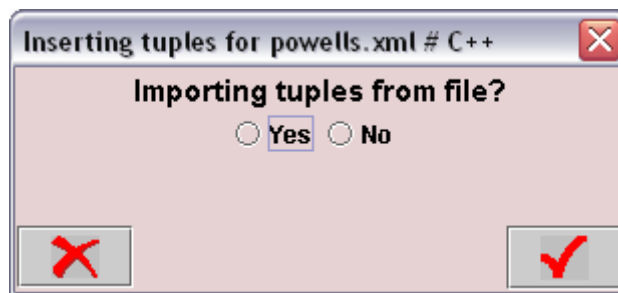
Σχήμα 6.20

Το παράθυρο που εμφανίζεται στη συνέχεια, Σχήμα 6.21, δίνει τη δυνατότητα στο χρήστη να επιλέξει σε ποιο TSR σχήμα επιθυμεί να εισάγει νέες εγγραφές και ιδιότητες.



Σχήμα 6.21

Αφού επιλεγεί το επιθυμητό TSR σχήμα εμφανίζεται στην οθόνη το παράθυρο επιλογής προέλευσης των νέων εγγραφών και ιδιοτήτων. Υπενθυμίζεται ότι υπάρχουν δύο πηγές για τις νέες εγγραφές και τις νέες ιδιότητες: ένα txt αρχείο σε κατάλληλη μορφή και ένας πίνακας της βάσης δεδομένων. Το παράθυρο επιλογής φαίνεται στο Σχήμα 6.22.



Σχήμα 6.22

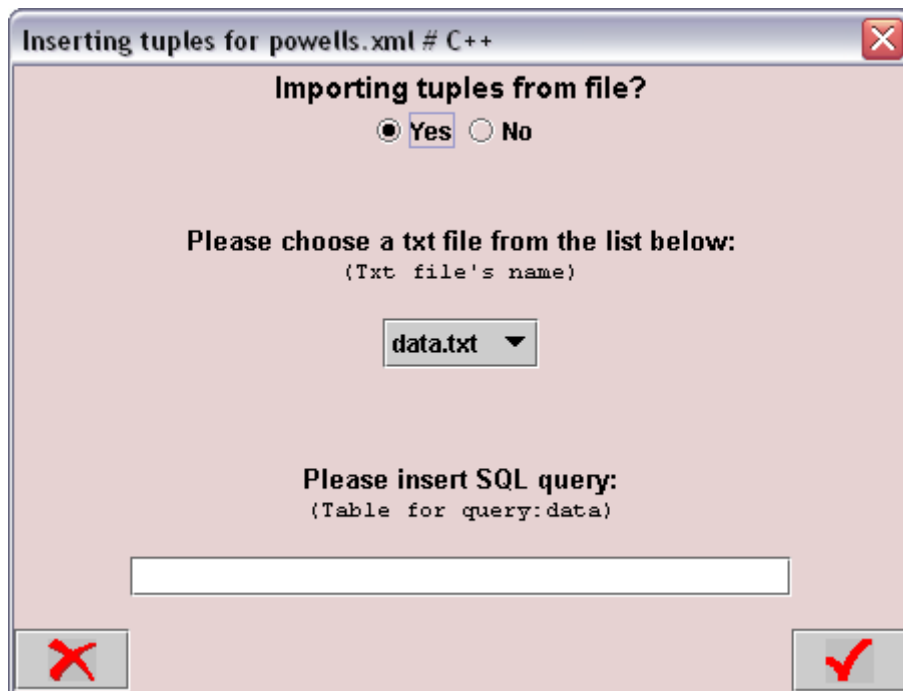
Επιλέγουμε να διαβάσουμε τα δεδομένα για τις νέες ιδιότητες και εγγραφές από το txt αρχείο με όνομα `data.txt` το περιεχόμενο του οποίου φαίνεται παρακάτω:

```

atitle#aauthor#aprice#aquality#afirst_price#category
Ultimate Beer#Michael Jackson#10.98#HardCover#29.95#beers
Beer Memorabilia#Martyn Cornell#10.50#HardCover#14.99#beers
The New Complete Joy of Home Brewing#Charlie Papazian#8.95#Trade Paper#8.95#beers
Dave Miller: HomeBrewing Guide#Dave Miller#16.95#Trade Paper#16.95#beers
Beer Lovers Rating Guide 2ND Edition#Bob Klein#8.00#Trade Paper#11.95#beers
Brew Ware#Karl Lutzen#12.50#Trade Paper#18.95#beers
Country Ales and Breweries#Roger Protz#7.98#Trade Paper#16.95#beers
Sacred and Herbal Healinh Beers#Stephen Buhner#19.95#Trade Paper#19.95#beers
C++ Programming Projects#sestak/Cep#17.50#Trade Paper#20.65#programming
Practical C++ Programming#Steve Oualline#9.50#Trade Paper#34.95#programming
C++ Plus Data Structures#Nell Dale#70.00#HardCover#87.95#programming

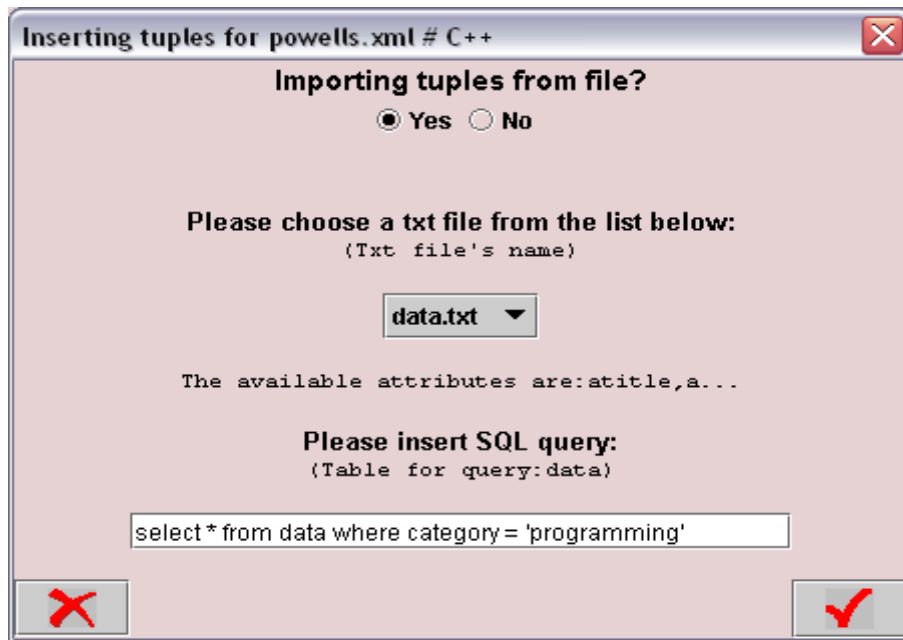
```

Effective C++#Scott Meyers#27.00#Trade Paper#39.95#programming  
C++ How to Program#Harvey Deitel#66.00#Trade Paper#82.50#programming  
Design and Evolution of C++#Bjarne Stroustrup#34.00#Trade Paper#44.99#programming  
Object Oriented Prog in C++#Robert Lafore#27.98#Trade Paper#44.99#programming  
Introduction to Computer Science With C++#Nance/Naps#78.50#HardCover#92.50#programming  
Numerical Recipes Example Book (C++)#William Vetterling#23.95#Trade Paper#35.00#programming  
C How to Program#Harvey Deitel#32.50#Paperback#65.00#programming  
The C Programming Language#Brian Kernigham#25.00#Paperback#35.70#programming  
Standard C++ Bible# #31.24# #54.98#programming  
Thinking in C++# #29.9# #35.78#programming  
Data structures with C++# #9.68# #32.12#programming  
Το παράθυρο που εμφανίζεται στην οθόνη μας είναι το εξής (Σχήμα 6.23):



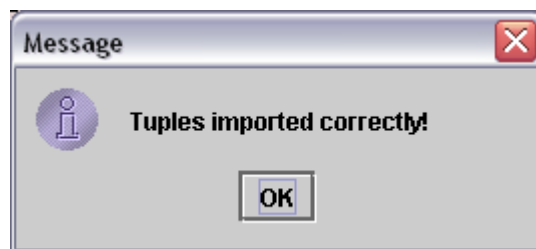
Σχήμα 6.23

Παρατηρούμε ότι ανά πάσα στιγμή μπορούμε να αλλάξουμε την πηγή των ιδιοτήτων και των εγγραφών. Από την προηγούμενη οθόνη επιλέγουμε το txt αρχείο για την ανάγνωση των νέων ιδιοτήτων και των νέων εγγραφών και εισάγουμε την SQL ερώτηση με βάση την οποία θα επιλεχθούν μερικά γνωρίσματα και μερικές πλειάδες του πίνακα data (Σχήμα 6.24):



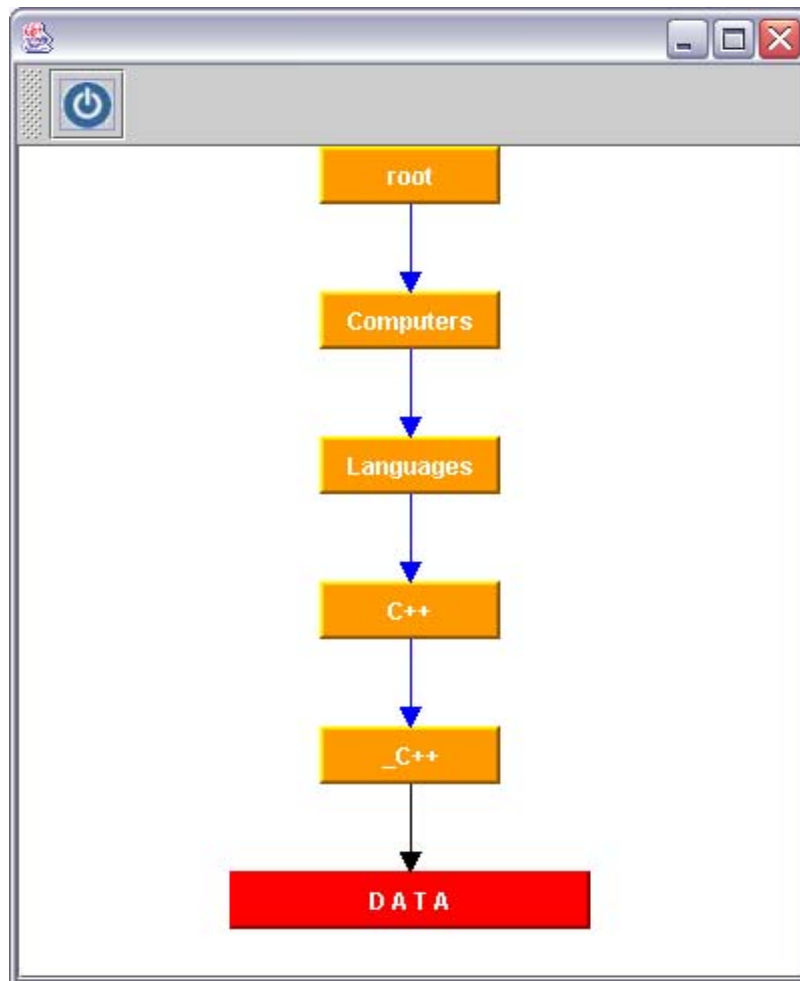
Σχήμα 6.24

Σημειώνεται ότι κάτω από το JComboBox εμφανίζονται κάθε φορά οι ιδιότητες που περιέχει το txt αρχείο που επιλέξαμε. Αν δεν προκύψει κανένα σφάλμα εμφανίζεται στην οθόνη το επιβεβαιωτικό παράθυρο του Σχήματος 6.25:



Σχήμα 6.25

Για επιβεβαίωση της ορθής εισαγωγής των νέων εγγραφών στον κόμβο – αντικείμενο του TSR σχήματος, επιλέγουμε την εφαρμογή 'View TSR' για τη γραφική απεικόνιση του TSR σχήματος (Σχήμα 6.26). Οι ιδιότητες και οι εγγραφές του κόμβου – αντικειμένου του TSR σχήματος φαίνονται στο Σχήμα 6.27.

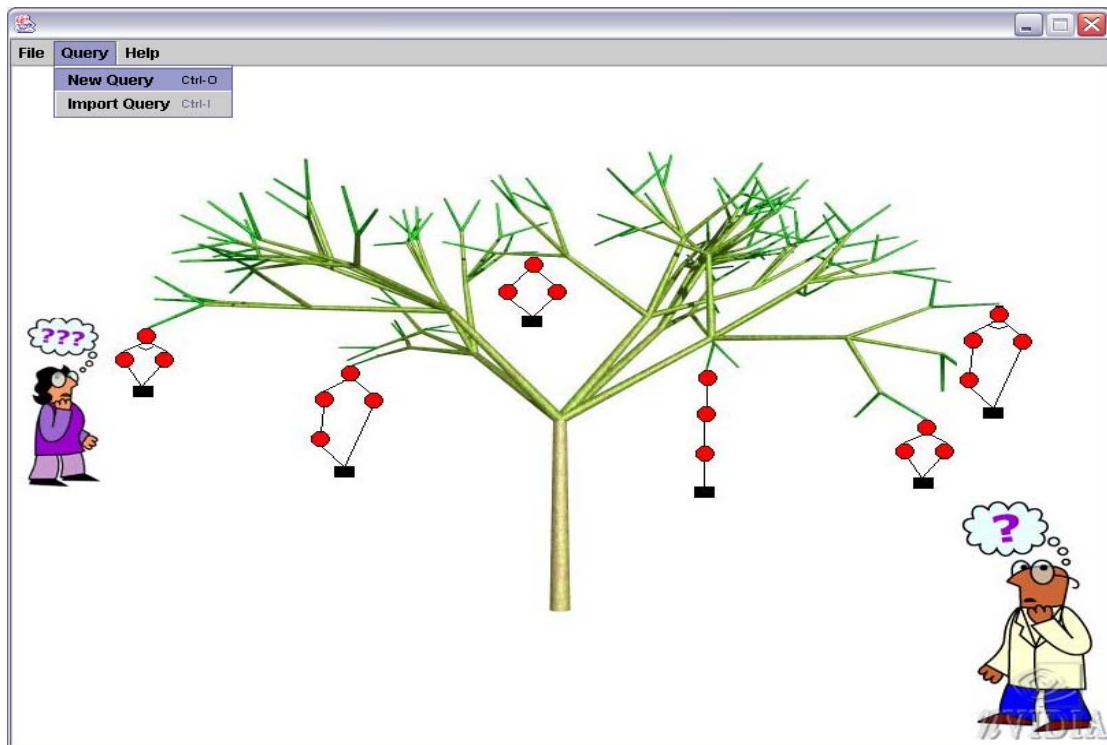


Σχήμα 6.26

atitle	aauthor	aprice	aquality	afirst_price	category
C++ Programming Projects	sestak/Cep	17.50	Trade Pap...	20.65	programming
Practical C++ Programming	Steve Oualline	9.50	Trade Pap...	34.95	programming
C++ Plus Data Structures	Nell Dale	70.00	HardCover	87.95	programming
Effective C++	Scott Meyers	27.00	Trade Pap...	39.95	programming
C++ How to Program	Harvey Deitel	66.00	Trade Pap...	82.50	programming
Design and Evolution of C++	Bjarne Stroustrup	34.00	Trade Pap...	44.99	programming
Object Oriented Prog in C++	Robert Lafore	27.98	Trade Pap...	44.99	programming
Introduction to Computer Science With C++	Nance/Naps	78.50	HardCover	92.50	programming
Numerical Recipes Example Book (C++)	William Vetterling	23.95	Trade Pap...	35.00	programming
C How to Program	Harvey Deitel	32.50	Paperback	65.00	programming
The C Programming Language	Brian Kernighan	25.00	Paperback	35.70	programming
Standard C++ Bible		31.24		54.98	programming
Thinking in C++		29.9		35.78	programming
Data structures with C++		9.68		32.12	programming

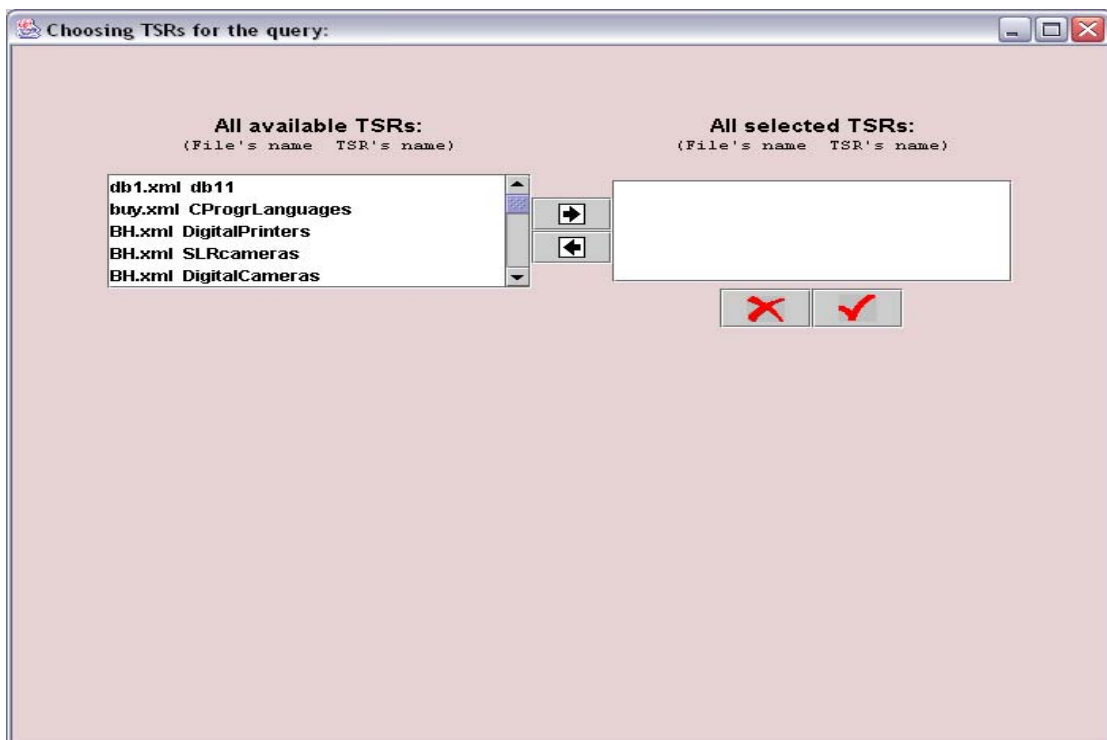
Σχήμα 6.27

Στη συνέχεια επιλέγουμε την εφαρμογή 'New Query' (Εισαγωγή ερώτησης) όπως φαίνεται στο Σχήμα 6.28:



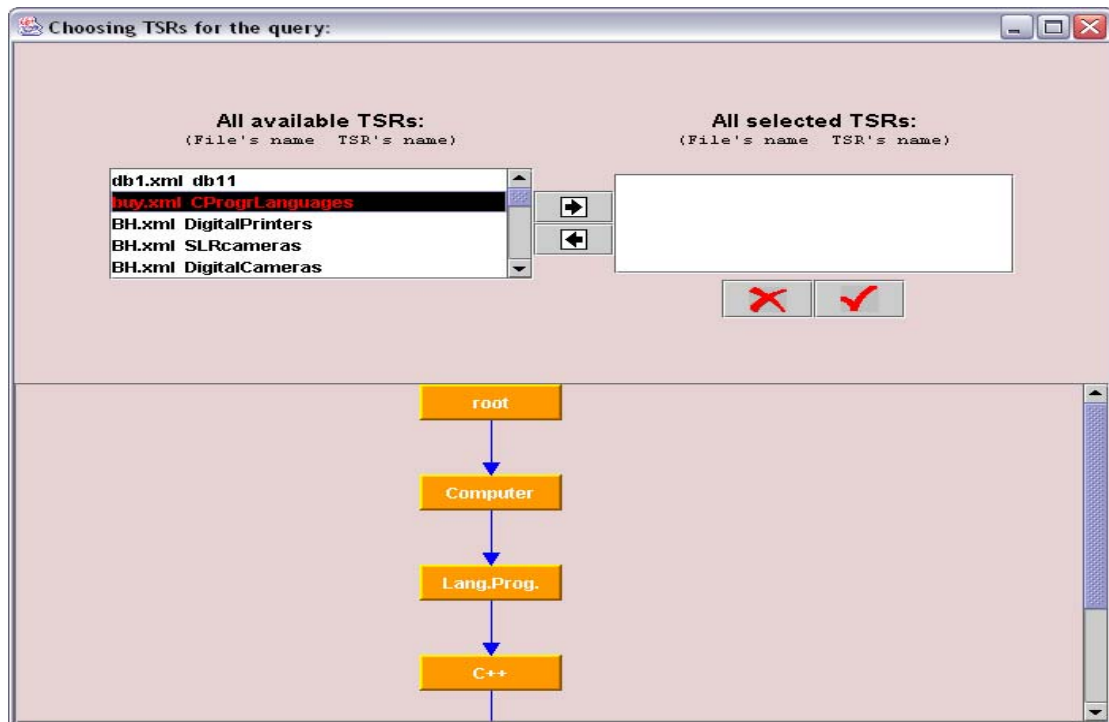
Σχήμα 6.28

Το παράθυρο που εμφανίζεται στην οθόνη του χρήστη είναι αυτό του Σχήματος 6.29. Σημειώνεται ότι στο κάτω μέρος του παραθύρου εμφανίζεται σε γραφική μορφή το TSR σχήμα που επιλέγεται από την πρώτη λίστα, όπως φαίνεται στο Σχήμα 6.30.



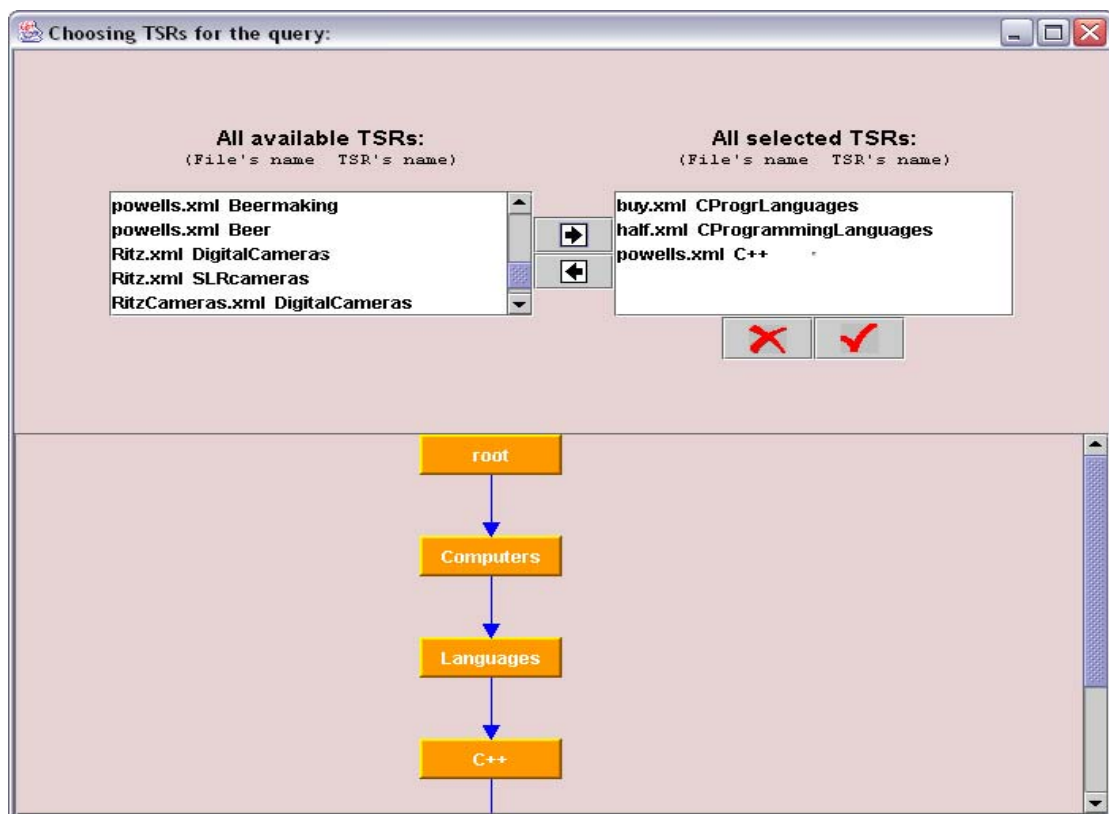
Σχήμα 6.29





Σχήμα 6.30

Αφού ολοκληρωθεί η επιλογή και των τριών TSR σχημάτων που περιγράψαμε νωρίτερα, το παράθυρο που εμφανίζεται στην οθόνη του χρήστη είναι αυτό του Σχήματος 6.31:



Σχήμα 6.31

Στη συνέχεια επιλέγοντας OK ξεκινά η κυρίως εφαρμογή ερωτήσεων – πράξεων και εμφανίζεται στην οθόνη του χρήστη το παράθυρο του Σχήματος 6.32:

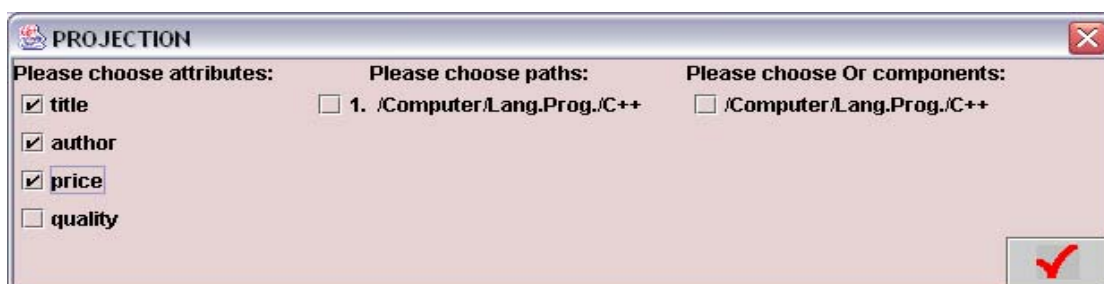


Σχήμα 6.32

Αν ο χρήστης επιλέξει με το ποντίκι το κουμπί με το μάτι, θα εμφανιστεί στην οθόνη ένα TSR σχήμα το οποίο είναι το TSR με όνομα αρχείου και όνομα σχήματος αυτά που φαίνονται δίπλα από το κουμπί (αρχικό TSR σχήμα) αν δεν έχουν γίνει πράξεις προβολής ή επιλογής. Σε αντίθετη περίπτωση εμφανίζεται το TSR σχήμα που δημιουργείται από το αρχικό αν εφαρμοστούν σε αυτό οι πράξεις προβολής και επιλογής που έχει εισάγει ο χρήστης. Από το TSR σχήμα με όνομα CProgrLanguages και όνομα αρχείου buy.xml προβάλλουμε τις ιδιότητες title, author και price όπως φαίνεται στα Σχήματα 6.33 και 6.34:



Σχήμα 6.33

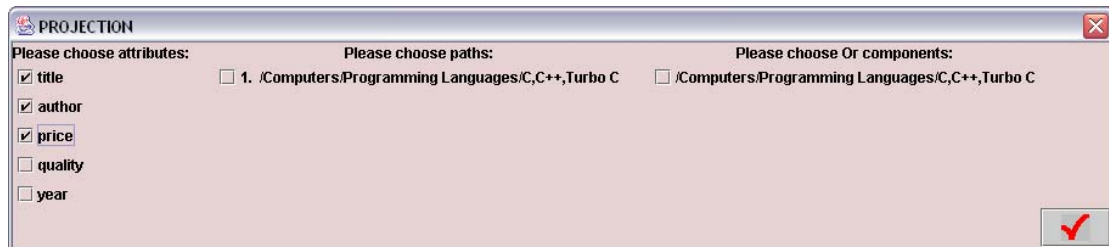


Σχήμα 6.34

Από το TSR σχήμα με όνομα CProgrammingLanguages και όνομα αρχείου half.xml προβάλλουμε και πάλι τις ιδιότητες title, author και price όπως φαίνεται στα Σχήματα 6.35 και 6.36:

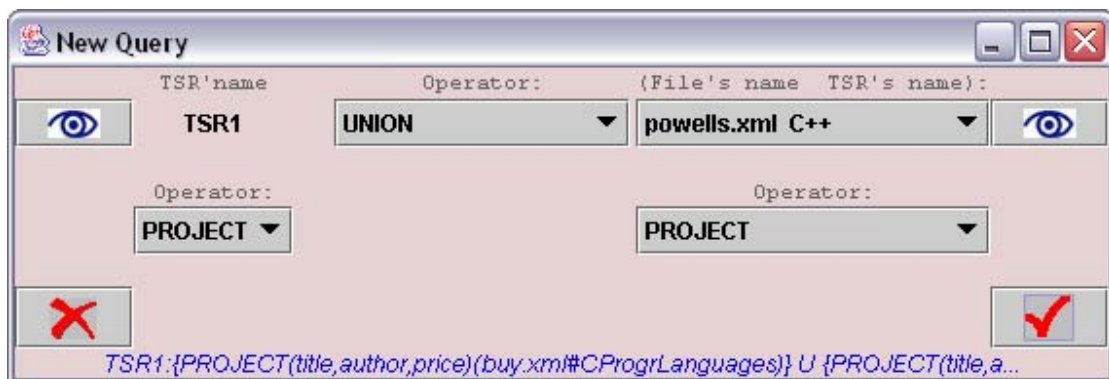


Σχήμα 6.35



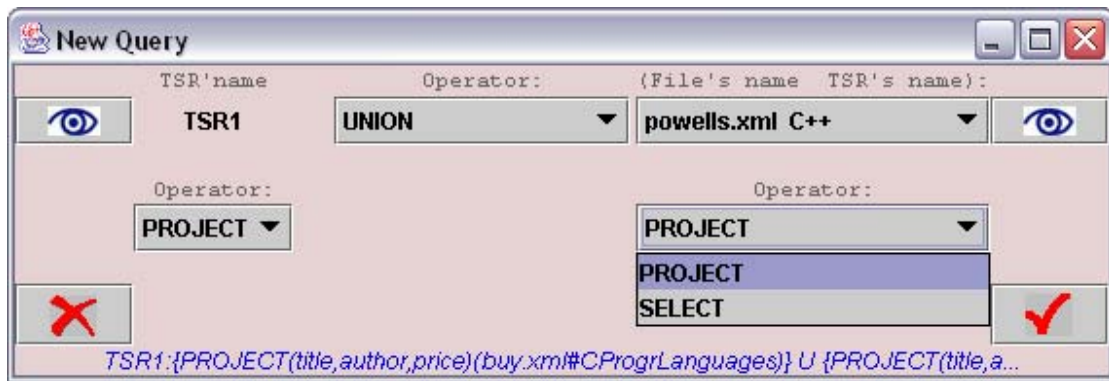
Σχήμα 6.36

Η πράξη που θα γίνει ανάμεσα σε αυτά τα δύο νέα TSR σχήματα που δημιουργούνται από αυτές τις δύο προβολές είναι η πράξη της ένωσης. Το παράθυρο που εμφανίζεται στην οθόνη του χρήστη είναι αυτό που φαίνεται στο Σχήμα 6.37:

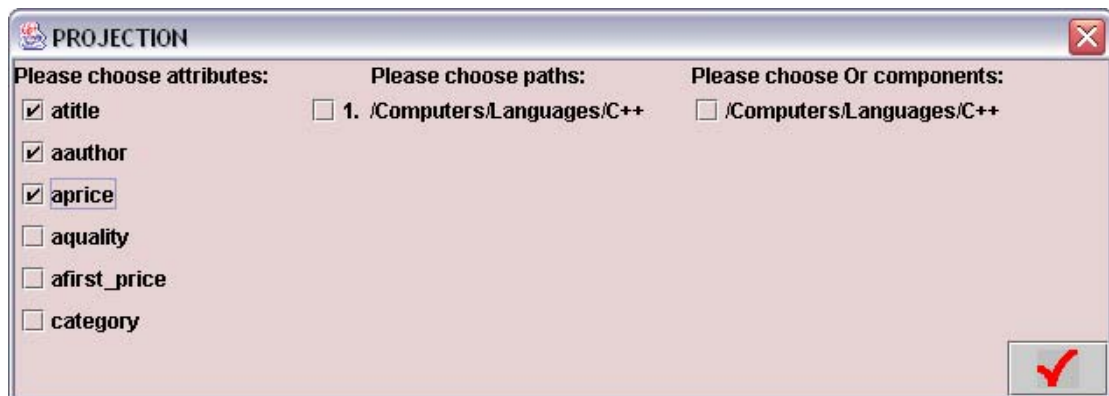


Σχήμα 6.37

Η ετικέτα TSR1 αντιπροσωπεύει το αποτέλεσμα της πράξης της ένωσης μεταξύ των δύο TSR σχημάτων. Στο κάτω μέρος της οθόνης φαίνονται πάντα οι πράξεις που έχουν γίνει ανάμεσα στα TSR σχήματα που έχουν επιλεγεί. Από το TSR σχήμα με όνομα C++ και όνομα αρχείου powells.xml προβάλλονται οι ιδιότητες atitle, aauthor και aprice όπως φαίνεται στα Σχήματα 6.38 και 6.39:

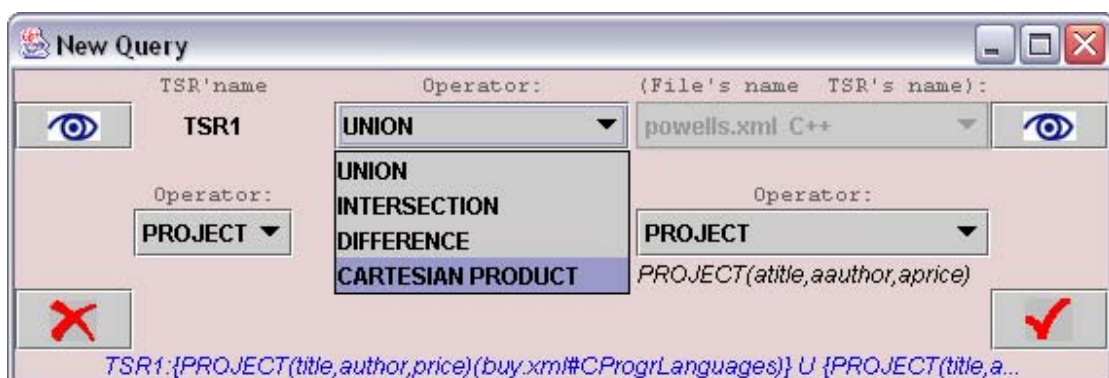


Σχήμα 6.38



Σχήμα 6.39

Ανάμεσα στο TSR σχήμα που δημιουργείται από αυτήν την προβολή και στο TSR σχήμα που είχε προκύψει από την προηγούμενη ένωση επιλέγεται η πράξη του καρτεσιανού γινομένου όπως φαίνεται στο Σχήμα 6.40:



Σχήμα 6.40

Το παράθυρο που εμφανίζεται στην οθόνη του χρήστη μετά την επιλογή της πράξης του καρτεσιανού γινομένου είναι αυτό του Σχήματος 6.41:

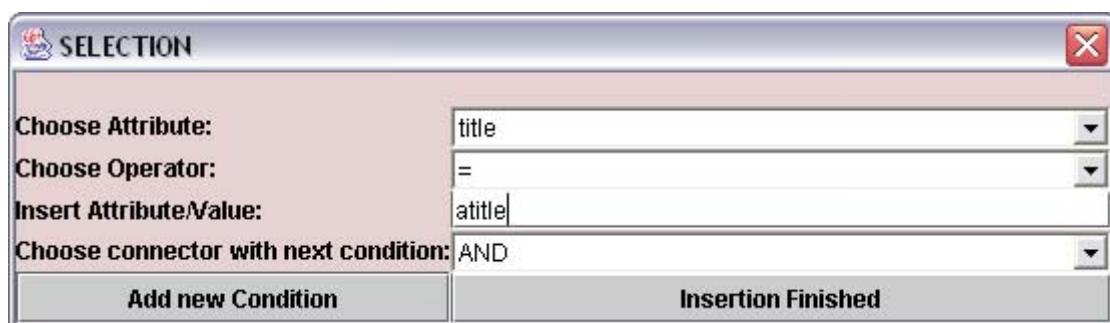


Σχήμα 6.41

Από το TSR σχήμα του αποτελέσματος επιλέγουμε τις εγγραφές για τις οποίες ισχύει η παρακάτω συνθήκη: `title = atitle`, όπως φαίνεται στα Σχήματα 6.42 και 6.43:



Σχήμα 6.42



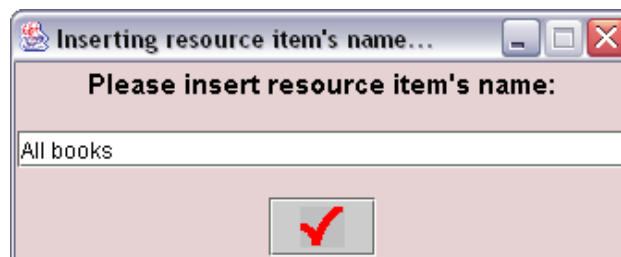
Σχήμα 6.43

Αφού ολοκληρωθεί η εισαγωγή της παραπάνω συνθήκης εμφανίζεται το παράθυρο του Σχήματος 6.44:



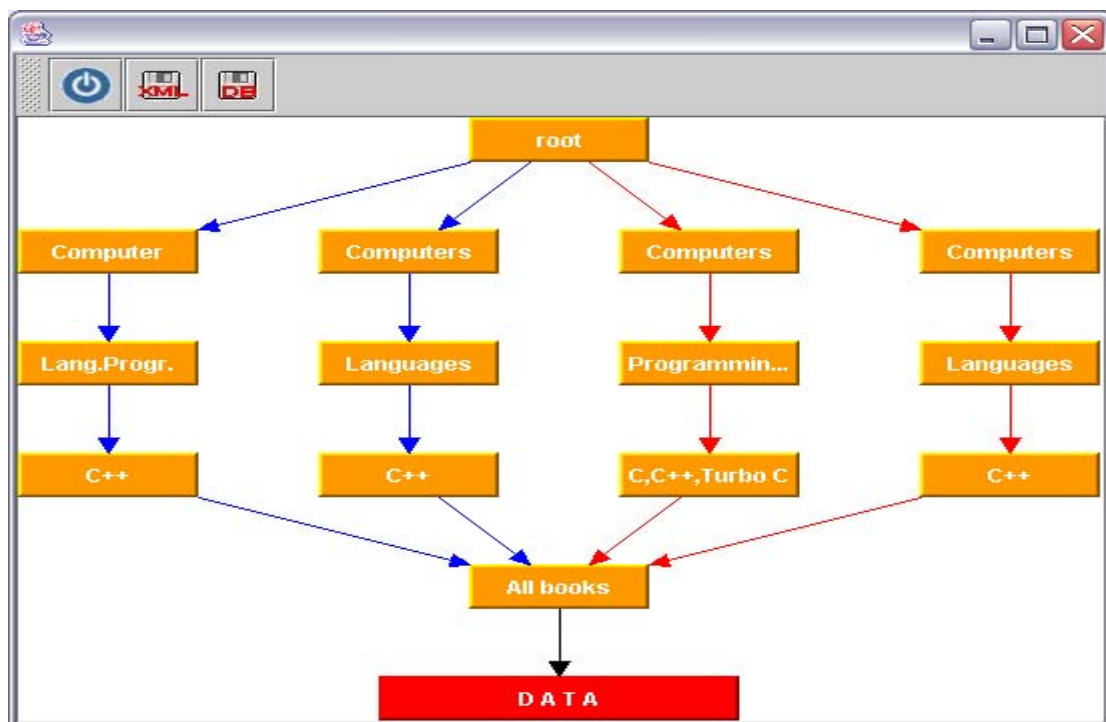
Σχήμα 6.44

Όταν ολοκληρωθεί η εφαρμογή των ερωτήσεων – πράξεων ο χρήστης πρέπει να εισάγει το όνομα του κόμβου αντικειμένου του TSR σχήματος που δημιουργήθηκε από την προηγούμενη διαδικασία με βάση την οθόνη του Σχήματος 6.45:



Σχήμα 6.45

Τέλος το TSR σχήμα του αποτελέσματος εμφανίζεται σε γραφική μορφή όπως φαίνεται στο Σχήμα 6.46:



Σχήμα 6.46

Οι ιδιότητες και οι εγγραφές του κόμβου – αντικειμένου του TSR σχήματος εμφανίζονται στην οθόνη του χρήστη μόλις αυτός επιλέξει τον κόμβο με όνομα DATA και έχει τη μορφή του Σχήματος 6.47:

title	author	price	atitle	aauthor	aprice
C++ How to Program	Harvey Deitel	58	C++ How to Program	Harvey Deitel	66.00
Design and Evolution of C++	Bjarne Stroustrup	45	Design and Evolution of C++	Bjarne Stroustrup	34.00
Object Oriented Prog in C++	Robert Lafore	27.95	Object Oriented Prog in C++	Robert Lafore	27.98
Numerical Recipes Example Book (C++)	William Vetterling	18.93	Numerical Recipes Example Book (C++)	William Vetterling	23.95
C How to Program	Harvey Deitel	34.5	C How to Program	Harvey Deitel	32.50
Standard C++ Bible		32.44	Standard C++ Bible		31.24
Data structures with C++		12.68	Data structures with C++		9.68
The C Programming Language	Brian Kernigham	17.98	The C Programming Language	Brian Kernigham	25.00
C How to Program	Harvey Deitel	34.50	C How to Program	Harvey Deitel	32.50
The C Programming Language	Brian Kernigham	20.00	The C Programming Language	Brian Kernigham	25.00

Σχήμα 6.47

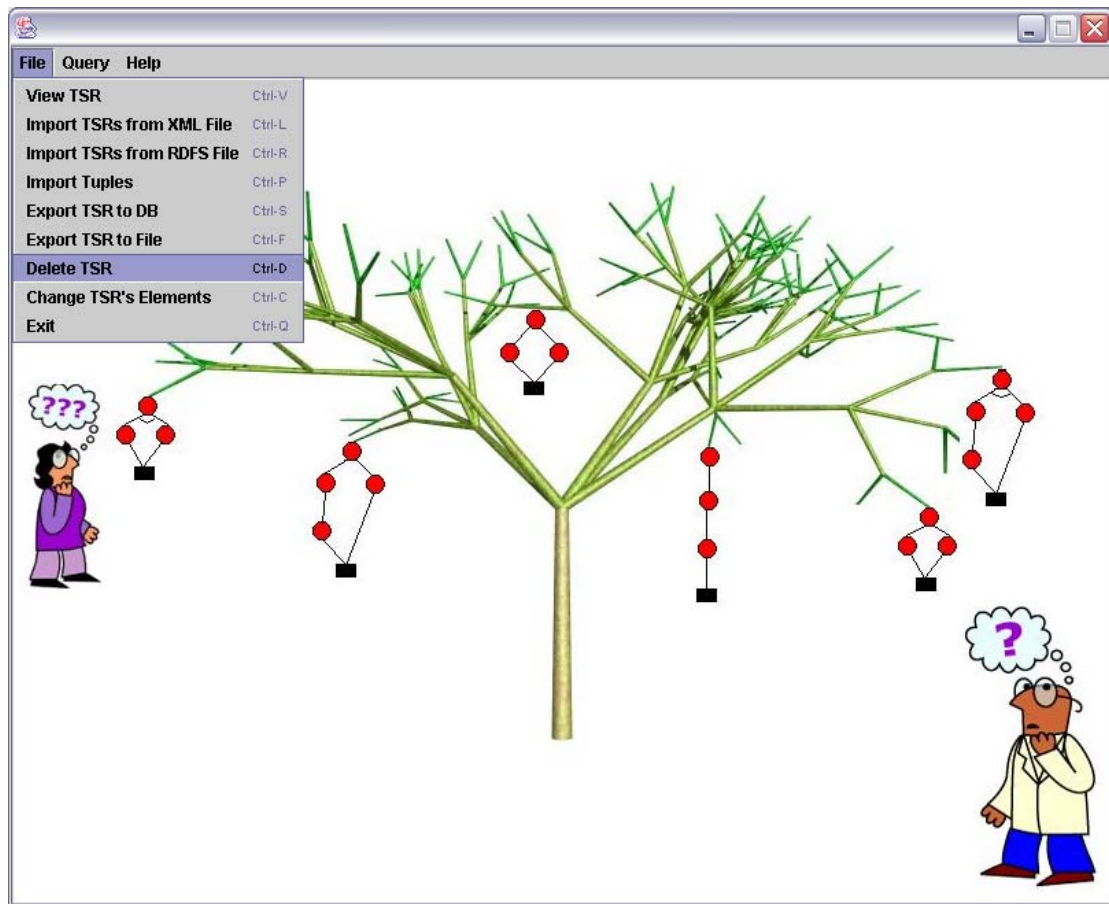
Το σύστημα παρέχει τη δυνατότητα αποθήκευσης του αποτελέσματος είτε σε κάποιο XML αρχείο είτε στη βάση δεδομένων του συστήματος. Επιλέγοντας την αποθήκευση σε XML αρχείο, απαιτείται η εισαγωγή από το χρήστη του ονόματος του TSR σχήματος, του ονόματος του XML αρχείου στο οποίο θα αποθηκευτεί και μίας σύντομης περιγραφής του. Η εισαγωγή αυτών των πληροφοριών γίνεται με τη βοήθεια του παραθύρου που φαίνεται στο Σχήμα 6.48:

Σχήμα 6.48

Με αυτόν τον τρόπο ο χρήστης μπορεί να αποθηκεύσει το TSR σχήμα είτε σε κάποιο υπάρχον XML αρχείο, προσθέτοντας το στο τέλος, είτε σε ένα καινούριο το οποίο θα δημιουργηθεί από το σύστημα. Αν ο χρήστης επιλέξει την αποθήκευση του αποτελέσματος στη βάση δεδομένων τότε εμφανίζεται στην οθόνη του ένα ανάλογο παράθυρο με το προηγούμενο το οποίο φαίνεται στο Σχήμα 6.49:

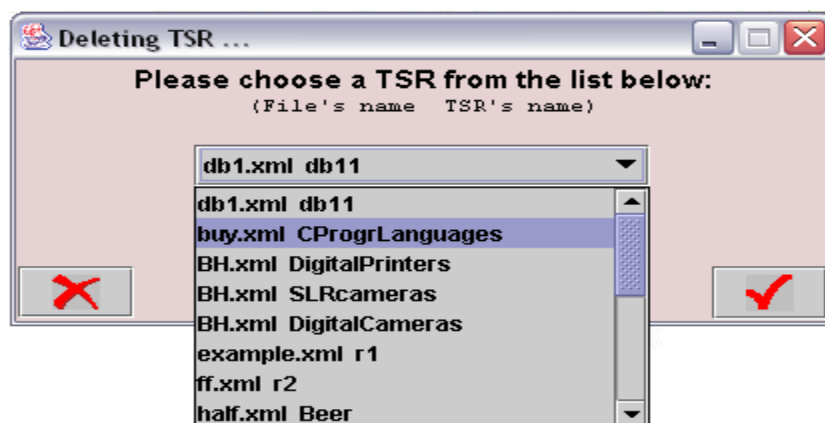
Σχήμα 6.49

Τέλος διαγράφουμε τα τρία TSR σχήματα που συμμετείχαν στην ερώτηση επιλέγοντας την εφαρμογή 'Delete TSR' (Διαγραφή TSR σχημάτων) όπως φαίνεται στο Σχήμα 6.50.



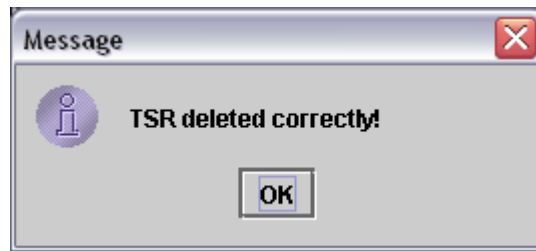
Σχήμα 6.50

Στην οθόνη εμφανίζεται το παράθυρο του Σχήματος 6.51 από το οποίο επιλέγεται το TSR σχήμα το οποίο θα διαγραφεί. Αν η διαγραφή ολοκληρωθεί επιτυχώς τότε εμφανίζεται το παράθυρο με το επιβεβαιωτικό μήνυμα του Σχήματος 6.52.



Σχήμα 6.51





Σχήμα 6.52

Η παραπάνω διαδικασία επαναλαμβάνεται και για τα TSR σχήματα με ονόματα CProgrammingLanguages και C++ και ονόματα αρχείων half.xml και powells.xml αντίστοιχα.



# 7

## *Επίλογος*

Το κεφάλαιο αυτό ολοκληρώνει την παρουσίαση της παρούσας διπλωματικής εργασίας. Στη συνέχεια παρουσιάζεται μία σύντομη ανασκόπησή της και αναφέρονται πιθανές μελλοντικές επεκτάσεις της.

### *7.1 Σύνοψη και συμπεράσματα*

Εξαιτίας της μεγάλης χρήσης του διαδικτύου και του τεράστιου όγκου δεδομένων και πληροφοριών που προωθείται και αναζητείται στον Ιστό καθημερινά, θα ήταν εξαιρετικά χρήσιμος ένας ενιαίος και αυστηρός τρόπος οργάνωσης των δεδομένων. Οι ιεραρχικές δομές και οι κατάλογοι είναι ένας τρόπος για την παροχή τέτοιας οργάνωσης και του σημασιολογικού εμπλουτισμού της διαθέσιμης πληροφορίας στον Ιστό. Σκοπός της διπλωματικής αυτής ήταν η δημιουργία ενός προτύπου συστήματος διαχείρισης μονοπατιών σε συναφείς καταλόγους και η δημιουργία μίας γλώσσας ερωτήσεων για ιεραρχίες στις οποίες οργανώνεται το πλήθος των πληροφοριών. Η γλώσσα ερωτήσεων του συστήματος χειρίζεται πολλούς καταλόγους οι οποίοι αποθηκεύουν ίδιου τύπου αντικείμενα αλλά έχουν διαφορετική οργάνωση της ιεραρχίας τους. Οι πράξεις της γλώσσας ερωτήσεων βοηθούν στον προσδιορισμό μονοπατιών που, αν και διαφορετικά, οδηγούν σε ίδια προϊόντα, και

δίνουν στο χρήστη τη δυνατότητα να επιλέξει τα μονοπάτια που θέλει ώστε να φέρει την ιεραρχία στην επιθυμητή γι' αυτόν μορφή. Η γλώσσα ερωτήσεων που υλοποιήθηκε αποτελείται από ένα σύνολο ερωτήσεων ανάλογων των ερωτήσεων της σχεσιακής άλγεβρας, οι οποίες όμως αποτιμώνται σε δενδρικές δομές δεδομένων, δίνοντας ως αποτέλεσμα ένα σύνολο μονοπατιών που οδηγούν σε εγγραφές δεδομένων.

Αρχικά στο Κεφάλαιο 2 παρουσιάστηκε το απαραίτητο θεωρητικό υπόβαθρο και ορισμένες ερευνητικές εργασίες και τεχνολογίες συναφείς με το θέμα της διπλωματικής.

Στη συνέχεια στο Κεφάλαιο 3 παρουσιάστηκε η θεωρητική μελέτη του ζητήματος διαχείρισης πληροφορίας στους δικτυακούς καταλόγους. Συγκεκριμένα ορίστηκαν οι βασικές δομές δεδομένων της γλώσσας και έγινε μία σύντομη αναφορά στο  $R_p$ - $R_i$  μοντέλο αναπαράστασης των δομών αυτών. Στη συνέχεια ορίστηκε το σύνολο των πράξεων της γλώσσας που εφαρμόζονται στα TSR σχήματα. Οι πράξεις αυτές είναι οι εξής: επιλογή (selection), προβολή (projection), καρτεσιανό γινόμενο (cartesian product), ένωση (union), τομή (intersection) και διαφορά (difference). Στην πράξη της επιλογής απαιτούνται δύο λίστες συνθηκών: μία για τις ιδιότητες και μία για τα μονοπάτια. Έτσι επιλέγονται τόσο οι OR συνιστώσες που οδηγούν στον κόμβο – αντικείμενο του TSR σχήματος, τα μονοπάτια των οποίων ικανοποιούν τις συνθήκες μονοπατιών, όσο και τα δεδομένα των εγγραφών που ικανοποιούν τις συνθήκες ιδιοτήτων. Με την πράξη της προβολής επιλέγονται μερικά από τα AND μονοπάτια κάθε OR συνιστώσας ή ολόκληρες OR συνιστώσες, καθώς και μία λίστα από τις ιδιότητες του κόμβου – αντικειμένου του TSR σχήματος, οι οποίες θα αποτελούν το σύνολο των ιδιοτήτων του κόμβου – αντικειμένου του TSR σχήματος που θα προκύψει. Το καρτεσιανό γινόμενο δέχεται σαν είσοδο δύο TSR σχήματα και παράγει μία νέα TSR. Τα μονοπάτια του αποτελέσματος είναι ο συνδυασμός κάθε OR συνιστώσας της πρώτης TSR με κάθε OR συνιστώσα της δεύτερης TSR. Οι ιδιότητες είναι το σύνολο των ιδιοτήτων των δύο TSR εισόδου, ενώ οι εγγραφές του αποτελέσματος είναι ο συνδυασμός κάθε εγγραφής του κόμβου – αντικειμένου της πρώτης TSR με κάθε εγγραφή του κόμβου – αντικειμένου της δεύτερης. Το TSR σχήμα που παράγεται από την πράξη της ένωσης αποτελείται από τις OR συνιστώσες και των δύο αρχικών TSR σχημάτων, ενώ οι εγγραφές του κόμβου – αντικειμένου του είναι η ένωση των εγγραφών των κόμβων – αντικειμένων των αρχικών TSR σχημάτων. Το TSR σχήμα που παράγεται από την πράξη της τομής αποτελείται από τις OR συνιστώσες και των δύο αρχικών TSR σχημάτων, ενώ οι εγγραφές του κόμβου – αντικειμένου του είναι η τομή των εγγραφών των κόμβων – αντικειμένων των δύο αρχικών TSR σχημάτων. Τέλος με την πράξη της διαφοράς παράγεται ένα TSR σχήμα το οποίο αποτελείται από τις OR συνιστώσες μόνο της πρώτης TSR εισόδου, ενώ περιέχει τις εγγραφές που υπάρχουν στον πρώτο κόμβο – αντικείμενο και δεν υπάρχουν στο δεύτερο. Σημειώνεται ότι στις πράξεις της ένωσης, της τομής και της διαφοράς τα δύο TSR σχήματα

εισόδου πρέπει να έχουν τις ίδιες ιδιότητες κόμβου – αντικειμένου ως προς το όνομα και τον τύπο. Αντίθετα, στην πράξη του καρτεσιανού γινομένου οι κόμβοι – αντικείμενα των δύο TSR σχημάτων πρέπει να έχουν διαφορετικές ιδιότητες.

Στα Κεφάλαια 4 και 5 παρουσιάστηκαν η ανάλυση, η σχεδίαση και η υλοποίηση του συστήματος σε γλώσσα προγραμματισμού Java. Το σύστημα διαχωρίστηκε σε επιμέρους υποσυστήματα τα οποία επεξηγήθηκαν αναλυτικά και στη συνέχεια περιγράφησαν οι εφαρμογές/ λειτουργίες που επιτελεί το σύστημα.

Στο 5<sup>ο</sup> κεφάλαιο παρουσιάστηκαν οι λεπτομέρειες της υλοποίησης και έγινε μία σύντομη περιγραφή όλων των κλάσεων που δημιουργήθηκαν για την υλοποίηση του συστήματος.

Τέλος στο 6<sup>ο</sup> Κεφάλαιο ολοκληρώθηκε η υλοποίηση του συστήματος με την αναλυτική παρουσίαση ενός σεναρίου χρήσης του.

Εν κατακλείδι, στη διπλωματική αυτή μοντελοποιήσαμε τους ιεραρχικούς καταλόγους χρησιμοποιώντας σχήματα καταλόγων και εισάγαμε τα TSR σχήματα για την αναπαράσταση των εναλλακτικών τρόπων πρόσβασης σε έναν κοινό κόμβο – αντικείμενο από ένα σύνολο σχημάτων καταλόγων. Με δεδομένο τους πολλούς τρόπους προσπέλασης ίδιων αντικειμένων στο διαδίκτυο, το σύστημα που υλοποιήθηκε προσφέρει στο χρήστη τη δυνατότητα επιλογής και επεξεργασίας των δεδομένων του αντικειμένου καθώς και των διαφορετικών τρόπων προσπέλασης του. Με τη βοήθεια των τελεστών της γλώσσας που υλοποιήθηκε ο χρήστης μπορεί να κατασκευάσει μία καινούρια ιεραρχία αλλά και να επιτελέσει ένα σύνολο από λειτουργίες πάνω σε υπάρχουσες ιεραρχίες. Το σύστημα εκτός από τη γλώσσα ερωτήσεων που παρέχει, δίνει στο χρήστη τη δυνατότητα και για άλλες εφαρμογές πάνω στα TSR σχήματα όπως είναι για παράδειγμα, η γραφική απεικόνιση αυτών των σχημάτων.

## ***7.2 Μελλοντικές επεκτάσεις***

Στη διπλωματική αυτή εργασία υλοποιήσαμε ένα σύστημα για τη διαχείριση μονοπατιών από ιεραρχίες συναφών καταλόγων. Σε γενικές γραμμές η εργασία αυτή αποτελείται από τρία συστήματα: το σύστημα για την εισαγωγή των TSRs σχημάτων, το σύστημα για την εισαγωγή των ερωτήσεων – πράξεων πάνω σε ένα ή περισσότερα TSR σχήματα και το σύστημα για την γραφική απεικόνιση των TSR σχημάτων. Για κάθε ένα από αυτά τα τρία συστήματα υπάρχουν ορισμένες προτάσεις και ιδέες για την επέκταση και αναβάθμιση του συστήματος, μερικές από τις οποίες φαίνονται στη συνέχεια:

- Αυτόματη εύρεση των εγγραφών και των ιδιοτήτων για τους κόμβους – αντικείμενα των TSR σχημάτων που προέρχονται από RDFS αρχείο. Αυτό σημαίνει ότι ο χρήστης δε θα χρειάζεται να δημιουργεί μόνος του τις εγγραφές και τις ιδιότητες του κόμβου – αντικειμένου. Το σύστημα θα αναλαμβάνει την ανάγνωση τους από το RDF αρχείο και την αποθήκευση των κατάλληλων εγγραφών και ιδιοτήτων στους κόμβους – αντικείμενα των κατάλληλων TSR σχημάτων.
- Προσθήκη εφαρμογών για την εισαγωγή και διαγραφή δεδομένων στα TSR σχήματα. Για παράδειγμα μπορούν να δημιουργηθούν εφαρμογές για την εισαγωγή και διαγραφή OR συνιστωσών ή AND μονοπατιών, καθώς επίσης και εφαρμογές για διαγραφή εγγραφών και ιδιοτήτων των κόμβων – αντικειμένων. Σημειώνεται ότι έχει ήδη υλοποιηθεί εφαρμογή για την εισαγωγή εγγραφών και ιδιοτήτων στους κόμβους – αντικείμενα των TSR σχημάτων (Import Tuples).
- Ορισμός και ενσωμάτωση νέων τελεστών διαχείρισης δενδρικών ιεραρχικών δομών των δικτυακών καταλόγων.
- Επίλυση του προβλήματος ταιριάσματος σχημάτων. Το ταίριασμα σχημάτων προσδιορίζει τις κατηγορίες σε μία ιεραρχία που είναι ίδιες σημασιολογικά με κατηγορίες σε μία άλλη ιεραρχία, έστω και αν χρησιμοποιούνται διαφορετικές λέξεις για την περιγραφή τους. Στη διπλωματική αυτή εργασία θεωρήσαμε τα θέματα ταιριάσματος σχημάτων λυμένα.

# 8

## *Βιβλιογραφία*

- [1] Serge Abiteboul, Peter Buneman, and Dan Suciu, Data on the web. From relations to semistructured data and XML, Morgan Kaufmann Publishers, San Francisco, 2002.
- [2] T. Berners-Lee, J. Handler, and O. Lassila, The semantic web, Scientific American 279 (2001), no. 5.
- [3] <http://www.semanticweb.org>.
- [4] <http://www.w3c.org/XML>.
- [5] <http://www.w3c.org/RDF>.
- [6] <http://www.w3c.org/XML/Schema>.
- [7] Daniela Florescu, Alon Levy, and Alberto Mendelzon, Database techniques for the world-wide web: A survey, ACM Sigmod Record 27 (1998), no. 3, 59-75.
- [8] David Konopnicki and Oded Shmueli, W3QS: A query system for the World - Wide Web, VLDB '95: proceedings of the 21st International Conference on Very Large Data Bases, Zurich, Switzerland (Los Altos, CA 94022, USA) (Umeshwar Dayal,

- Peter M. D. Gray, and Shojiro Nishio, eds.), Morgan Kaufmann Publishers, 1995, pp. 54-65.
- [9] A. Mendelzon, G. Mihaila, and T. Milo, Querying the world wide web, *International Journal on Digital Libraries* 1 (1997), no. 1, 54-67.
- [10] Laks V. S. Lakshmanan, Fereidoon Sadri, and Iyer N. Subramanian, A declarative language for querying and restructuring the web, *RIDE '96: proceedings of the 6th International Workshop on Research Issues in Data Engineering*, New Orleans, USA, 1996.
- [11] Gustavo O. Arocena and Alberto O. Mendelzon, WebOQL: restructuring documents, databases and webs, *ICDE'98, Proceedings ICDE International Conference on Data Engineering*, Feb, 1998, Orlando, Florida, USA, ACM Press, 1998.
- [12] Mary F. Fernandez, Daniela Florescu, Alon Y. Levy, and Dan Suciu, A query language for a web-site management system, *ACM SIGMOD Record* 26 (1997), no. 3.
- [13] Rainer Himmeroder, George Lausen, Bertam Ludascher, and Christian Schleppehorst, On a declarative semantics for web queries, *Proceedings of the Int'l Conference on Deductive and Object-Oriented Databases (DOOD)*, Singapore, 1997.
- [14] S. Abiteboul, D. Quass, J. McHugh, J. Widom, and J. Wiener, The lorel query language for semistructured data, *International Journal on Digital Libraries* 1997.
- [15] P. Buneman, S. Davidson, G. Hillebrand, and D. Suciu, A query language and optimization techniques for unstructured data, *Proceedings of the ACM SIGMOD Conference*, 1996, pp. 505-516.
- [16] Alin Deutsch, Mary Fernandez, Daniela Florescu, Alon Levy, David Maier, and Dan Suciu, Querying xml data, *IEEE Data Engineering Bulletin* 22 (1999), no. 3, 10-18.
- [17] [http://www.w3.org/TR/NOTE-xml ql/](http://www.w3.org/TR/NOTE-xml%20ql/).
- [18] Stefano Ceri, Sara Comai, Ernesto Damiani, Piero Fraternali, Stefano Paraboschi, and Letizia Tanca, Xml-gl: a graphical language for querying and restructuring xml



documents, proceedings of the 8th Int'l World Wide Web Conference, Toronto, Canada, 1999.

- [19] <http://www.w3.org/TandS/QL/QL98/pp/xql.html>.
- [20] Don Chamberlin, Jonathan Robie, and Daniela Florescu, Quilt: An XML query language for heterogeneous data sources, WebDB'00, 2000.
- [21] <http://www.w3.org/XML/Query>.
- [22] Don Chamberlin, XQuery: An XML query language, IBM Systems Journal 41, no. 4, 2002.
- [23] Catriel Beeri and Yariv Tzaban, SAL: An algebra for semistructured data and XML, WebDB'99, 1999.
- [24] H. V. Jagadish, Laks V. S. Lakshmanan, D. Srivastava, and K. Thompson, TAX: A tree algebra for XML, DBPL Conference, 2001.
- [25] Sophie Cluet, Claude Delobel, Jerome Simeon, and Katarzyna Smaga, Your mediators need data conversion, Proceedings ACM SIGMOD International Conference on Management of Data, June 2-4, 1998, Seattle, Washington, USA, 1998.
- [26] V. Christophides, S. Cluet, and J. Simeon, On wrapping query languages and efficient XML integration, Proceedings of the ACM SIGMOD Conference, 2000, pp. 141-152.
- [27] Θεόδωρος Δαλαμάγκας, Διαχείριση ιεραρχικών σχημάτων στο σημασιολογικό ιστό, Εργαστήριο ΣΒΓΔ, Ε.Μ.Π, Φεβρουάριος 2004, Διδακτορική Διατριβή.
- [28] Παναγιώτης Μπούρος, Γλώσσα ερωτήσεων για Δεδομένα Δενδρικής Δομής, Εργαστήριο ΣΒΓΔ, Ε.Μ.Π, Νοέμβριος 2003, Διπλωματική εργασία.
- [29] P. Bouros, T. Dalamagas, T. Sellis, and M. Terrovitis, PatManQL: A language to manipulate patterns and data in hierarchical catalogs, EDBT Workshop on Pattern Representation and Management (PaRMa' A04), Heraklion, Greece, 2004.

- [30] E. Rahm and P. A. Bernstein, A survey of approaches to automatic schema matching, VLDB Journal 10 (2001), no. 4, 334-350.