ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΡΗΤΗΣ
ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ
ΤΜΗΜΑ ΕΠΙΣΤΗΜΗΣ ΥΠΟΛΟΓΙΣΤΩΝ

Ενημέρωση Όψεων σε Βάσεις Γνώσης

Γιάννης Τ. Τζίτζικας

Μεταπτυχιακή Εργασία

Ηράκλειο, Οκτώβρης 1995

ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΡΗΤΗΣ
ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ
ΤΜΗΜΑ ΕΠΙΣΤΗΜΗΣ ΥΠΟΛΟΓΙΣΤΩΝ

Ενημέρωση Όψεων σε Βάσεις Γνώσης

Εργασία που υποβλήθηκε από τον
ΓΙΑΝΝΗ Τ. ΤΖΙΤΖΙΚΑ
ως μερική εκπλήρωση των απαιτήσεων
για την απόκτηση
ΜΕΤΑΠΤΥΧΙΑΚΟΥ ΔΙΠΛΩΜΑΤΟΣ ΕΙΔΙΚΕΥΣΗΣ

Συγγραφέας:

Γιάννης Τ. Τζίτζικας
Τμήμα Επιστήμης Υπολογιστών

Εισηγητική Επιτροπή:

Πάνος Κωνσταντόπουλος,Αναπληρωτής Καθηγητής, Επόπτης

Χρήστος Νικολάου,Αναπληρωτής Καθηγητής, Μέλος

Πάνος Τραχανιάς,Επίκουρος Καθηγητής, Μέλος

Δεκτή:

Πάνος Κωνσταντόπουλος
Πρόεδρος Επιτροπής Μεταπτυχιακών Σπουδών

Ηράκλειο, Οκτώβρης 1995

Ενημέρωση Όψεων σε Βάσεις Γνώσης

Γιάννης Τ. Τζίτζικας

Μεταπτυχιακή Εργασία

Τμήμα Επιστήμης Υπολογιστών
Πανεπιστήμιο Κρήτης

ΠΕΡΙΛΗΨΗ

Μια βάση γνώσης χρησιμοποιείται συνήθως από πολλούς χρήστες για την πραγματοποίηση διαφορετικών εργασιών, μερικές από τις οποίες ενημερώνουν διαφορετικές όψεις (τμήματα) της βάσης. Ένα Σύστημα Διαχείρισης Βάσεων Γνώσεων με δυνατότητα παράστασης και διαχείρισης όψεων ενημέρωσης, μπορεί να ελέγχει τις ενημερώσεις που πραγματοποιούνται στη βάση διασφαλίζοντας έτσι την ποιότητα των περιεχομένων της ως προς συγκεκριμένους περιορισμούς. Επιπλέον η ζεύξη των όψεων ενημέρωσης με ένα σύνολο γενικών λειτουργιών διαλογικής ενημέρωσης δίνει τη δυνατότητα ανάπτυξης γενικών εργαλείων διαλογικής ενημέρωσης της βάσης, τα οποία είναι φιλικά προς το χρήστη.

Η παρούσα εργασία ασχολείται με την παράσταση και τη διαχείριση όψεων ενημέρωσης σε βάσεις γνώσης. Ειδικότερα, η εργασία αυτή αφορά βάσεις γνώσης της Telos . Η Telos είναι μια γλώσσα παράστασης γνώσης η οποία υποστηρίζει ένα δομικά οντοκεντρικό μοντέλο δεδομένων και χρησιμοποιείται από το Σύστημα Σημασιολογικού Ευρετηριασμού (SIS). Μια Όψη Ενημέρωσης (ΟΕ) ορίζεται ως ένα σύνολο στοιχειωδών ενημερώσεων οι οποίες αφορούν τα δεδομένα ή το σχήμα της βάσης.

Αρχικά χρησιμοποιήθηκε ένα Μοντέλο Διεργασιών Ενημέρωσης (ΜΔΕ) στο οποίο οι δηλώσεις των ΟΕ γίνονταν με τη χρήση των ερωτηματικών συναρτήσεων της Telos . Για την αξιοποίηση των ΟΕ κατά τη διάρκεια της διερεύνησης των περιεχομένων της βάσης, απαιτείται η εξαγωγή συμπερασμάτων, πράγμα που δεν μπορεί να γίνει αποδοτικά λόγω της χρήσης των ερωτηματικών συναρτήσεων. Επιπλέον, οι δηλώσεις ΟΕ μέσω ερωτηματικών συναρτήσεων παρουσιάζουν προβλήματα ευχρηστίας, εποπτείας και συνέπειας με τα περιεχόμενα της βάσης.

Για την άρση των αδυναμιών αυτών αναπτύχθηκε ένα ΜεταΜοντέλο Όψεων Ενημέρωσης (ΜΜΟΕ). Οι ΟΕ δηλώνονται με τη βοήθεια ενός συνόλου τύπων δήλωσης, ενώ οι δηλώσεις ενσωματώνονται στο μοντέλο εφαρμογής της εκάστοτε βάσης σαν ειδικές σχέσεις της Telos . Το χαρακτηριστικό αυτό αναβαθμίζει το ρόλο των ΟΕ, αφού με τον τρόπο αυτό οι δηλώσεις τους παραμένουν συνεπείς με τα περιεχόμενα της βάσης, απαιτούν ελάχιστη συντήρηση, εξασφαλίζεται η ορθότητα τους και, το κυριότερο, αυξάνονται οι δυνατότητες και προοπτικές αξιοποίησης των ΟΕ. Επίσης, ο ορισμός ΟΕ με το ΜΜΟΕ είναι εκφραστικός και ευέλικτος, αφού υποστηρίζεται η δυνατότητα άμεσων/έμμεσων και θετικών/αρνητικών δηλώσεων, καθώς επίσης και σύνθετων δηλώσεων με τη βοήθεια των σύνθετων τύπων, οι οποίοι μπορούν να δημιουργούνται και να εξελίσσονται από τον ίδιο το χρήστη. Καθώς η δήλωση ΟΕ γίνεται με χρήση της ίδιας της Telos , δεν θέτονται επιπλέον γνωστικές απαιτήσεις στο χρήστη. Ένα επιπλέον πλεονέκτημα (υλοποίησης) είναι το ότι το ΜΜΟΕ εκμεταλεύεται τις υπάρχουσες δυνατότητες του SIS για την αποθήκευση, ανάκτηση και παρουσίαση των δηλώσεων του.

Τέλος, για τη σύζευξη διερεύνησης και ενημέρωσης σχεδιάστηκε ένα σύνολο γενικών λειτουργιών διαλογικής ενημέρωσης, το οποίο λαμβάνει υπόψη το περιβάλλον (context) στο οποίο είναι ενταγμένο ένα αντικείμενο και συνάμα αξιοποιεί τις όψεις ενημέρωσης που έχουν ορισθεί, παρέχοντας έτσι φιλική και ελεγχόμενη διαλογική εξέλιξη γνώσης.

Επόπτης : Πάνος Κωνσταντόπουλος
Αναπληρωτής Καθηγητής Επιστήμης Υπολογιστών
Πανεπιστήμιο Κρήτης

View Updates in Knowledge Bases

Yannis T. Tzitzikas

Master of Science Thesis

Department of Computer Science
University of Crete

ABSTRACT

A knowledge base is used by several users in order to perform various tasks, which may update different views of the knowledge base. A Knowledge Base Management System that represents and manages update views can control updates, ensuring the quality of the knowledge base contents with respect to specific constraints. Moreover, integration of update views with a set of general operations for interactive updates can be exploited for the development of user friendly general tools for interactive updates.

This thesis deals with the representation and management of update views in knowledge bases. Specifically, this thesis is concerned with Telos knowledge bases. Telos is a knowledge presentation language which supports a structurally object oriented data model and is used by the Semantic Index System (SIS). The Update View (UV) is defined as a set of primitive update operations which concern data and/or schema.

Initially, in order to define UV in Telos, a Model of Update Processes (MUP) was employed, in which the declaration of a UV was contacted by using the query functions of Telos. In order to integrate updates and browsing, conclusion deduction is required, which can not be effectively performed due to query functions usage. Moreover, UV declarations via query functions are hard to express, difficult to supervise and they are not consistent with the knowledge base contents.

In order to overcome the MUP weaknesses, a MetaModel of Update Views (MMUV) was developed. Update views are declared by a set of update types, while the declarations are embodied in the application model of the knowledge base under concern, by means of special Telos relationships. This feature upgrades the role of UV, because their declarations

remain consistent with the contents of the base, they require no maintenance, their correctness is ensured and, most important, their UV opportunities and exploitation perspectives are increased. In addition, the definition of UV by MMUV is expressive and flexible, because explicit/implicit and positive/negative declarations as well as composite declarations (by using composite types which can be created and evolved by the user himself) are supported. Since update views' declarations are performed using Telos, no additional knowledge is required by the user. An additional implementation advantage is that the MMUV profits from the existing capabilities of SIS for storage, retrieval and presentation of its declarations.

In order to integrate browsing and updates, a set of general update operations for interactive updates was designed, which takes into account object's context, while it exploits the defined UV, offering user friendly and controllable interactive knowledge evolution.

Supervisor : Panos Constantopoulos
Associate Professor of Computer Science
University of Crete

Ευχαριστίες

Θα ήθελα να ευχαριστήσω όλα τα μέλη (νυν και πρώην) της Ομάδας Πληροφοριακών Συστημάτων και Τεχνολογίας Λογισμικού του Ινστιτούτου Πληροφορικής του Ιδρύματος Τεχνολογίας και Έρευνας για τη συνεργασία και τη συμπαράστασή τους τα τελευταία τρία χρόνια. Επίσης ευχαριστώ το Ινστιτούτο Πληροφορικής για την υλικοτεχνική υποδομή και την οικονομική ενίσχυση που μου παρείχε κατά τη διάρκεια όλων των μεταπτυχιακών μου σπουδών.

Ιδιαίτερα θα ήθελα να ευχαριστήσω την Αθηνά Τραψιώτη η οποία με ενθάρρυνε να ξεκινήσω (και να περατώσω) τις μεταπτυχιακές μου σπουδές, καθώς και τους φίλους και συνεργάτες μου Γιώργο Γεωργιαννάκη, Μάνο Θεοδωράκη και Πολύβιο Κλημαθιανάκη για την πολύπλευρη βοήθειά τους.

Τέλος, θα ήθελα να ευχαριστήσω τον επόπτη καθηγητή μου κ. Πάνο Κωνσταντόπουλο που μου έδωσε την ευκαιρία να ασχοληθώ με την εργασία αυτή.

Περιεχόμενα

Περίληψη **i**

Abstract **iii**

Ευχαριστίες **v**

Περιεχόμενα **x**

Κατάλογος Σχημάτων **xiv**

1 Εισαγωγή **1**

1.1 Ορισμός του προβλήματος 1

1.2 Αποτελέσματα της εργασίας 2

1.3 Η οργάνωση της γραπτής εργασίας 4

2 Βιβλιογραφική Ανασκόπηση **7**

2.1 Εξέλιξη Γνώσης 7

2.2 Όψεις Ενημέρωσης 8

2.2.1 Έλεγχος Ενημερώσεων 8

2.2.2 Διευκόλυνση Ενημερώσεων 9

2.2.3 Απαιτήσεις 9

2.3 Οντοκεντρικές Βάσεις Δεδομένων 10

2.3.1 Εξέλιξη Γνώσης σε Οντοκεντρικά μοντέλα 10

2.4 Ανασκόπηση Σχετικών Μηχανισμών 11

2.4.1 Όψεις 11

2.4.2 Υποσχήματα 17

2.4.3 Μηχανισμοί Ελέγχου Πρόσβασης 18

2.5 Συμπεράσματα 20

3 Η γλώσσα Telos και το σύστημα SIS **23**

3.1 Ιδιαιτερότητες και Πλεονεκτήματα της Telos 23

3.2 Παράσταση γνώσης στη γλώσσα Telos 24

3.3 Τυπική περιγραφή της SIS-Telos 29

3.4 Το Σημασιολογικό Σύστημα Ευρετηριασμού. 32

3.4.1 Χαρακτηριστικά υλοποίησης της SIS-Telos 34

4 Εξέλιξη Γνώσης στην SIS-Telos **37**

4.1 Η Στοιχειώδης Γλώσσα Ενημερώσης 37

4.1.1 Λεξικογραφική Ανάλυση 38

4.1.2 Συντακτική - Σημασιολογική - Εκτελεστική Ανάλυση 39

4.2 Χαρακτηριστικά της Εξέλιξης Γνώσης στην SIS-Telos 45

5 Ένα Μοντέλο Διεργασιών Ενημέρωσης **47**

5.1 Γενική Περιγραφή του Μοντέλου Διεργασιών 47

5.1.1 Δήλωση Συνόλων 48

5.1.2 Ερμηνεία Δηλώσεων 49

5.1.3 Εκτέλεση των Ενημερώσεων 51

5.1.4 ‘‘Εξαγωγή’’ μιας Όψης Ενημέρωσης : *Export* 55

5.2 Οι Εργασίες Ενημέρωσης του SIB Class Management System 56

5.2.1 Χρήση του *Export* 61

5.3 Οι Όψεις Ενημέρωσης του *ΚΛΕΙΩ* 62

5.3.1 Παραδείγματα Χρήσης 63

5.3.2 Επέκταση της Describe Object 64

5.4 Αξιολόγηση 64

5.4.1 Γενικά Συμπεράσματα 68

6 Το Μεταμοντέλο Όψεων Ενημέρωσης **71**

6.1 Αιτιολόγηση 71

6.1.1 Μεταμοντέλα 73

6.2 Γενική Παρουσίαση του Μεταμοντέλου 74

6.3 Δηλώσεις Ενημέρωσης 75

6.4 Ερμηνεία Δηλώσεων Ενημέρωσης 79

6.4.1 Βήμα 1 : Κληρονόμηση Δηλώσεων 79

6.4.2 Βήμα 2 : Ανάλυση Μη-Εστιακών Δηλώσεων 82

6.4.3 Βήμα 3 : Μετάφραση σε Κατηγορήματα 85

6.4.4 Βήμα 4 : Σύνθεση Στοιχειωδών Ενημερώσεων 87
6.5 Σημεία Εκκίνησης 87
6.6 Οντολογία του MMOE 89
6.7 Παραδείγματα Χρήσης 91
6.7.1 Δηλώσεις IN 92
6.7.2 Δηλώσεις IN : Συνδυασμοί Θετικών και Αρνητικών 94
6.7.3 Δηλώσεις IN και Πεδίο Εφαρμογής onAttrs 95
6.7.4 Δηλώσεις IN : Όψεις Προβολής Κλάσεων 96
6.7.5 Πεδίο Εφαρμογής onInsts 100
6.7.6 Σχετικές κλάσεις 105
6.7.7 Δηλώσεις *SUB* 109
6.7.8 Γνωρίσματα 111
6.7.9 Δηλώσεις *CLASS* 111
6.7.10 Δηλώσεις σε Κλάσεις Συστήματος 112
6.7.11 Σύνθετες δηλώσεις 113
6.8 Όψεις επιλογής και Ενημερώσεις 116
6.8.1 Η εντολή **MakeCopy** 118
6.9 Υλοποίηση 120
6.9.1 Αλγόριθμος Εύρεσης Δηλώσεων Αντικειμένου 121
6.9.2 Αλγόριθμος Εύρεσης Κατηγορημάτων Αντικειμένου 123
6.9.3 Πολυπλοκότητα Αλγορίθμων 123
6.9.4 Βελτιστοποίηση επιδόσεων 127

7 Πρόταση Εφαρμογής του MMOE **131**
7.1 Λειτουργίες Ενημέρωσης 132
7.2 Διαχειριστικές Λειτουργίες 133
7.3 Λειτουργίες Ενημέρωσης 135
7.4 Λειτουργίες Ενημέρωσης και Όψεις Ενημέρωσης 141
7.4.1 Έλεγχος Δικαιωμάτων 141
7.4.2 Υπόδειξη Ενημερώσιμων Πληροφοριών 142
7.4.3 Εύρεση Πιθανών Ορισμάτων 142
7.5 Προτεινόμενη Επαφή Χρήσης 143
7.5.1 Δελτίο Παρουσίασης Οντότητας 145
7.5.2 Ένα Δελτίο Παρουσίασης και Ενημέρωσης Αντικειμένων 147
7.5.3 Συμπληρωματικές λειτουργίες 148

8 Συμπεράσματα και Μελλοντικές Κατευθύνσεις **155**

8.1 Αποτελέσματα της εργασίας 155

8.2 Συμπεράσματα και Σύγκριση με Σχετικές Εργασίες 157

8.3 Μελλοντική Εργασία και Κατευθύνσεις 159

8.3.1 Αξιοποίηση του MMOE 159

8.3.2 Επεκτάσεις της SIS-Telos 159

8.3.3 Επεκτάσεις του SIS 160

8.3.4 Επεκτάσεις του MMOE 160

A Συντομογραφίες **161**

B Το Μοντέλο Διεργασιών Ενημέρωσης **163**

C Οι Ερωτηματικές Εντολές του qi **165**

D Βιβλιογραφία ανά θεματική ενότητα **167**

Βιβλιογραφία **168**

Κατάλογος Σχημάτων

1.1 Η έννοια της όψης ενημέρωσης 2
1.2 Αξιοποίηση των όψεων ενημέρωσης 3

2.1 Οι εναλλακτικοί τρόποι ενημέρωσης μιας βάσης 8
2.2 Ταξινόμηση μηχανισμών διαχείρισης όψεων. 13
2.3 Παράδειγμα *μέγιστης περιοχής* οπτικού σημείου 16
2.4 Το μοντέλο **GSM**. 19
2.5 Η ιεραρχία *τύπων οντοτήτων* δικαιοδοσίας 20
2.6 Η ιεραρχία *τύπων πρόσβασης* 20
2.7 Η ιεραρχία *ρόλων* ή *υποκειμένων* 21

3.1 Η ιεραρχία των κλάσεων του συστήματος. 25
3.2 Ένα μοντέλο της Telos 27
3.3 Η επαφή χρήσεως του SIS 33
3.4 Οι δομές αποθήκευσης των αντικειμένων της SIS-Telos 34

4.1 Δημιουργία οντότητας 39
4.2 Δημιουργία γνωρίσματος 40
4.3 Δημιουργία συνδέσμου ταξινόμησης 40
4.4 Δημιουργία συνδέσμου εξειδίκευσης 41
4.5 Διαγραφή οντότητας 42
4.6 Διαγραφή γνωρίσματος 42
4.7 Διαγραφή συνδέσμου ταξινόμησης 43
4.8 Διαγραφή συνδέσμου εξειδίκευσης 44
4.9 Διαγραφή συνδέσμου εξειδίκευσης 44

5.1 Σχηματική Απεικόνιση του Μοντέλου Διεργασιών Ενημέρωσης 48
5.2 Η ροή ελέγχου των Δελτίου Εισαγωγής Δεδομένων 51

5.3 Επιλογή όψης ενημέρωσης . . . 52
5.4 Επιλογή της οντότητας *Abstrasction'SortedList* από το σύνολο οντοτήτων . . 52
5.5 Πρόσθεση υπερκλάσης . . . 54
5.6 Το πεδίο δράσης της ΟΕ **Classify Software Object** . . . 57
5.7 Το πεδίο δράσης της ΟΕ **Edit Term Hierarchy** . . . 59
5.8 Λειτουργία περιγραφής βιβλιοθήκης . . . 60
5.9 Οργάνωση και Εξαρτήσεις των αρχείων SIS-Telos της SIB . . . 62
5.10 Ορισμός πεδίου δράσεως των **DC** , **DO** . . . 64
5.11 Πιθανή κατάσταση της βάσης με χρήση των **DC** , **DO** . . . 65
5.12 Φωλιασμένες Όψεις Ένημέρωσης . . . 65
5.13 Χρήση της ταξινόμησης για την απόδοση σύνθετων ιδιοτήτων . . . 66
5.14 Παράδειγμα παράστασης ενός σύνθετου αντικειμένου . . . 67

6.1 Δύο διαφορετικές προσεγγίσεις του προβλήματος . . . 72
6.2 Αφηρημένο παράδειγμα χρήσης του ΜΜΟΕ . . . 72
6.3 Οι βασικές έννοιες του ΜΜΟΕ . . . 74
6.4 Δηλώσεις ενημέρωσης . . . 74
6.5 Τα αναγνωριστικά ενημέρωσης . . . 77
6.6 Χρήση θετικών και αρνητικών δηλώσεων . . . 78
6.7 Η μορφή των τύπων δήλωσης του ΜΜΟΕ . . . 78
6.8 Τα βήματα ερμηνείας των δηλώσεων ενημέρωσης μιας ΟΕ . . . 79
6.9 Διάταξη δηλώσεων με βάση την ισχύ τους . . . 85
6.10 Το ΜΜΟΕ σε SIS-Telos . . . 89
6.11 Σύνολα Χρηστών . . . 90
6.12 Οι Τύποι Δήλωσης του ΜΜΟΕ . . . 90
6.13 Τα Αναγνωριστικά Ενημέρωσης . . . 91
6.14 Μοντελοποίση των δηλώσεων . . . 92
6.15 Δήλωση IN σε οντότητες . . . 93
6.16 Κληρονόμηση θετικής δήλωσης IN σε οντότητα . . . 93
6.17 Κληρονόμηση αρνητικής δήλωσης IN σε οντότητα . . . 93
6.18 Δηλώσεις IN σε κατηγορίες γνωρισμάτων . . . 94
6.19 Κληρονόμηση δήλωσεων IN σε κατηγορίες γνωρισμάτων . . . 94
6.20 Επικάλυψη θετικής δήλωσης IN από αρνητική . . . 95
6.21 Επικάλυψη αρνητικής δήλωσης IN από θετική. . . . 95
6.22 Κυριαρχία αρνητικής δήλωσης IN . . . 95

6.23 Θετική δήλωση IN με πεδίο εφαρμογής onAttrs 96
6.24 Αρνητική δήλωση IN με πεδίο εφαρμογής onAttrs 96
6.25 Κληρονόμηση δήλωσης με πεδίο εφαρμογής onAttrs 97
6.26 Συνδυασμός δηλώσεων IN 97
6.27 Ορισμός όψης προβολής με απόκρυψη γνωρισμάτων 97
6.28 Ορισμός όψης προβολής με επιλογή γνωρισμάτων 98
6.29 Κληρονομούμενα γνωρίσματα και άμεσες δηλώσεις. 98
6.30 Αναδιάρθρωση ιεραρχίας εξειδίκευσης 98
6.31 Σύνθετο παράδειγμα δηλώσεων. 100
6.32 100
6.33 Θετική δήλωση IN με πεδίο εφαρμογής onInsts. 101
6.34 Αρνητική δήλωσης IN με πεδίο εφαρμογής onInsts. 101
6.35 Θετική δήλωση IN με πεδίο εφαρμογής onInsts. 102
6.36 Αρνητική δήλωσης AF με πεδίο εφαρμογής onInsts 102
6.37 Επικάλυψη άμεσης δήλωσης και δήλωσης *onInsts* 103
6.38 Επικάλυψη δηλώσεων *onInsts* 104
6.39 Επικάλυψη δήλωσης *onInsts* από κληρονομούμενη άμεση 104
6.40 Επικάλυψη δήλωσης *onAttrs* από δήλωση *onInsts* 104
6.41 Χρήση ταξινόμησης για παράσταση σύνθετων ιδιοτήτων 105
6.42 Πολλαπλή ταξινόμηση αντικειμένων. 106
6.43 Χρήση γνωρισμάτων τύπου **relatedClasses**. 107
6.44 Εξειδίκευση γνωρισμάτων τύπου **relatedClasses**. 107
6.45 Δήλωση σε γνώρισμα τύπου **relatedClasses** 108
6.46 Αρνητική δήλωση IN σε γνώρισμα τύπου **relatedClasses** 108
6.47 Θετικές και αρνητικές δηλώσεις σε γνωρίσματα τύπου **relatedClasses**. 109
6.48 Θετική δήλωση SUB σε οντότητα. 109
6.49 Αρνητική δήλωση SUB που κληρονομείται. 110
6.50 Συνδυασμός θετικών και αρνητικών δηλωσεων SUB. 110
6.51 Συνδυασμός δηλώσεων SUB. 110
6.52 Συνδυασμός δηλώσεων **AF**. 111
6.53 Δηλώσεις **AF** με πεδίο εφαρμογής onInsts. 111
6.54 Δηλώσεις CLASS 112
6.55 Ιεραρχία αναλλοίωτου πληθυσμού τιμών 114
6.56 Ιεραρχίες ταξινόμησης 115

6.57 Ιεραρχίες απόδοσης σύνθετων ιδιοτήτων 115
6.58 Σύνθετοι τύποι κλάσεων συστήματος 116
6.59 Μοντέλο εφαρμογής της όψης επιλογής *ΕργαζόμενοςΗρακλείου*. 117
6.60 Τύποι κατηγοριών γνωρισμάτων 118
6.61 Παράδειγμα εκτέλεσης εντολής **MakeCopy** 119
6.62 Παράδειγμα εκτέλεσης εντολής **MakeCopy** 120
6.63 **MakeCopy** με βάση ένα υπόδειγμα. 120
6.64 Αλγόριθμος υπολογισμού της *GetIdState* για οντότητα. 122
6.65 Αλγόριθμος υπολογισμού της *GetIdState* για γνώρισμα. 123
6.66 Πρωτότυπη εφαρμογή των αλγορίθμων του ΜΜΟΕ 124
6.67 Αλγόριθμος υπολογισμού των κατηγορημάτων με $AddIn, DelIn$. 125
6.68 Τα στοιχεία της Κρυφής Μνήμης Επαγόμενων Δηλώσεων 128
6.69 Η οργάνωση της ΚΜΕΔ 130

7.1 Αξιοποίηση των Όψεων Ενημέρωσης 131
7.2 Ενημερώσεις και Διερεύνηση : Ροή Ελέγχου 133
7.3 Διαλογική Εξέλιξη Γνώσης και Ενημέρωση Όψεων : Ροή Ελέγχου 134
7.4 Διάγραμμα ροής του ελέγχου των διαχειριστικών λειτουργιών 134
7.5 Διάγραμμα ροής ελέγχου των λειτουργιών ενημέρωσης. 135
7.6 Λειτουργίες που αφορούν τους χρήστες του συστήματος 143
7.7 Παρουσίαση καταλόγου με τους σύνθετους τύπους που έχουν ορισθεί . . . 144
7.8 Ανάλυση του σύνθετου τύπου *ClassificationHierarchy* στους συστατικούς του. 145
7.9 Παρουσίαση πληροφοριών για την όψη ενημέρωσης *Εργασία_Επιμελητή* . . 146
7.10 Ιεραρχική οργάνωση των κλάσεων δήλωσης 147
7.11 Ένα απλό μοντέλο της SIS-Telos 148
7.12 Δελτίο Παρουσίασης Οντότητας 149
7.13 Η επαφή χρήσης του προτεινόμενου ΔΠΕΑ (Α) 150
7.14 Η επαφή χρήσης του προτεινόμενου ΔΠΕΑ (Β) 151
7.15 Η επαφή χρήσης του προτεινόμενου ΔΠΕΑ (Γ) 152
7.16 Λειτουργίες της επαφής χρήσης του προτεινόμενου ΔΠΕΑ 152

8.1 Σύγκριση με σχετικές εργασίες 158
8.2 Εξειδίκευση συζυγών ζευγαριών 160

Κεφάλαιο 1

Εισαγωγή

1.1 Ορισμός του προβλήματος

Μία από τις πιο βασικές λειτουργίες ενός Συστήματος Διαχείρισης Βάσεων Δεδομένων (ΣΔΒΔ), είναι η εισαγωγή και η ενημέρωση δεδομένων. Μια βάση δεδομένων χρησιμοποιείται συνήθως από πολλούς χρήστες για την πραγματοποίηση διαφορετικών εργασιών. Κατά συνέπεια, οι ανάγκες εισαγωγής και ενημέρωσης δεδομένων διαφοροποιούνται σε μεγάλο βαθμό, ανάλογα με το χρήστη ή την εργασία που επιτελείται.

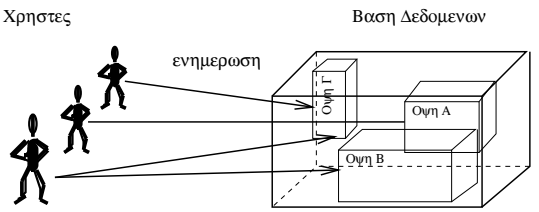
Μπορούμε γενικά να ορίσουμε ως *Όψη Ενημέρωσης* (ΟΕ) οποιοδήποτε σύνολο Στοιχειωδών Εντολών Ενημέρωσης (ΣΕΕ), οι οποίες εφαρμόζονται σε ένα μικρότερο ή απλούστερο μοντέλο της βάσης και αφορούν τα δεδομένα ή το σχήμα της βάσης (δες σχήμα 1.1). Τέτοιες ΟΕ εμφανίζονται συχνά σε εφαρμογές σχεδιαστικής ή κατασκευαστικής φύσης : περιλαμβάνουν αλλαγές στο σχήμα και δε χαρακτηρίζονται από προκαθορισμένες ακολουθίες ΣΕΕ.

Σε ΟΕ μπορεί να στηριχθεί η ανάπτυξη φιλικών και προσαρμόσιμων γενικών εργαλείων διαλογικής ενημέρωσης τα οποία, δρώντας στα πλαίσια όψεων της βάσης , διευκολύνουν το χρήστη παρέχοντας του το σύνολο των ΣΕΕ ενημέρωσης που σχετίζονται με την εργασία που επιθυμεί να διεκπεραιώσει. Επίσης ορίζοντας τις κατάλληλες ΟΕ είναι εφικτός ο έλεγχος των ενημερώσεων εμποδίζοντας έτσι την ενημέρωση ορισμένων τμημάτων της βάσης για λόγους δικαιοδοσίας ή για τη διασφάλιση της ποιότητας και της αξιοποίησης τους (δες σχήμα 1.2).

Σκοπός της παρούσας εργασίας είναι η μελέτη ΟΕ που αφορούν βάσεις γνώσης και

ειδικότερα βάσεις γνώσης της Telos . Η Telos [1] είναι μια γλώσσα παράσταση γνώσης η οποία υποστηρίζει ένα δομικά οντοκεντρικό μοντέλο δεδομένων, παρέχει πλούσιους εκφραστικούς μηχανισμούς και ομοιομορφία κατά την διαχείριση των δεδομένων και του σχήματος. Επίσης επιτρέπει αλλαγές στο σχήμα της βάσης κατά τον χρόνο εκτέλεσης. Χρησιμοποιείται από το Σύστημα Σημασιολογικού Ευρετηριασμού (Semantic Index System ή SIS) το οποίο μπορεί να διαχειρισθεί αποδοτικά μεγάλους πληθυσμούς από δεδομένα πολλαπλώς διασυνδεδεμένα. Για το λόγο αυτό η χρήση του ενδείκνυται σε εφαρμογές σχεδιαστικής ή κατασκευαστικής φύσης.

Η πραγματική ανάγκη που αποτέλεσε το έναυσμα της παρούσης εργασίας ήταν ο σχεδιασμός ενός διαλογικού δελτίου εισαγωγής δεδομένων για μία εφαμοργή του SIS, την *SIB Class Management System* [2] Σκοπός του δελτίου αυτού ήταν η παροχή στο χρήστη της δυνατότητας να πραγματοποιεί ένα σύνολο ΣΕΕ, προσφέροντας του παράλληλα την κατάλληλη καθοδήγηση. Συνάμα έπρεπε να είναι εφικτή η περιγραφή, σε μορφή εντολών της γλώσσας Telos , των περιεχομένων μιας όψης ενημέρωσης, ώστε να είναι δυνατή η μεταφορά του σε άλλες βάσεις δεδομένων. Εν συνεχεία ανέκυψε η ανάγκη διαλογικής εισαγωγής δοδομένων και για μια άλλη εφαρμογή του SIS, για το πληροφοριακό σύστημα μουσείων *ΚΛΕΙΩ* .



Σχήμα 1.1: Η έννοια της όψης ενημέρωσης

1.2 Αποτελέσματα της εργασίας

Το πρόβλημα της ενημέρωσης όψεων αρχικά προσεγγίστηκε με ένα *Μοντέλο Διεργασιών Ενημέρωσης* (ΜΔΕ). Ο σχεδιασμός και η ανάπτυξη του μοντέλου αυτού βασίστηκε στις ανάγκες της εφαρμογής **SIB Class Management System** ([14], [13]). Το σχήμα του ΜΔΕ

[1] Ακριβέστερα αναφερόμαστε στην SIS-Telos , η οποία είναι μια έκδοση της Telos η οποία αναπτύχθηκε από την Ομάδα Πληροφοριακών Συστημάτων και Τεχνολογίας Λογισμικού του ΙΠ-ΙΤΕ.

[2] Η *SIB Class Management System* είναι μια εφαρμογή που αφορά την τεκμηρίωση και την ανάλυση των στατικών ιδιοτήτων μονάδων λογισμικού.

Σχήμα 1.2: Αξιοποίηση των όψεων ενημέρωσης

περιγράφτηκε με τη βοήθεια της Telos και συγκεκριμενοποιείται προκειμένου να περιγράψει τις ΟΕ μιας συγκεκριμένης εφαρμογής. Για τη δήλωση των όψεων ενημέρωσης χρησιμοποιούνται οι ερωτηματικές συναρτήσεις της Telos . Το διαλογικό **Δελτίο Εισαγωγής Δεδομένων (ΔΕΔ)** που αναπτύχθηκε [3] διαβάζει τις δηλωμένες ΟΕ από τη βάση και προσαρμόζει ανάλογα την λειτουργία του. Επίσης μια ΟΕ μπορεί να περιγραφεί σε ένα αρχείο εντολών της Telos μέσω του εργαλείου **Export** που αναπτύχθηκε [4].

Οι δηλώσεις *όψεων ενημέρωσης* με το μοντέλο διεργασιών παρουσιάζουν αδυναμίες που το καθιστούν ακατάλληλο για εφαρμογές με συνθετότερες ανάγκες ενημέρωσης. Η χρήση ερωτηματικών συναρτήσεων δεν επιτρέπει την αποδοτική αξιοποίηση των ΟΕ, αφού δεν είναι εφικτή η αποδοτική εξαγωγή συμπερασμάτων που αφορούν τις ΟΕ. Επίσης η συνέπεια των δηλώσεων με τα περιεχόμενα της βάσης είναι δύσκολο να διασφαλισθεί, με αποτέλεσμα την ανάγκη συντήρησης των δηλώσεων καθώς η βάση εξελίσσεται. Οι διαπιστώσεις αυτές έγιναν ύστερα από την εμπειρία χρήσης του στο πληροφοριακό σύστημα μουσείων *ΚΛΕΙΩ* [26].

Για το λόγο αυτό σχεδιάστηκε ένα **ΜεταΜοντέλο Εργασιών Ενημέρωσης** (ΜΜΟΕ) [5], το οποίο άρει όλες τις αδυναμίες του προηγούμενου μοντέλου. Οι οριζόμενες με αυτό ΟΕ μπορούν να αξιοποιηθούν καλύτερα, οι δηλώσεις τους διατηρούν τη συνέπεια τους με τα περιεχόμενα της βάσης καθώς αυτή εξελίσσεται, και είναι εποπτεύσιμες από τα εργαλεία παρουσίασης του SIS. Επίσης η χρήση του ΜΜΟΕ είναι πιο εύκολη και κατανοητή για τον απλό χρήστη. Υποστηρίζονται θετικές και αρνητικές δηλώσεις (εξαιρέσεις) οι οποίες προσφέρουν ευελιξία, ενώ η δυνατότητα έμμεσων και σύνθετων δηλώσεων κάνουν τον ορισμό των ΟΕ εύκολο και ευέλικτο.

[3] Η υλοποίηση του ΔΕΔ έγινε από τον Δημήτρη Δασκαλάκη
[4] Η υλοποίηση του Export έγινε από το συγγραφέα
[5] Ο σχεδιασμός και η υλοποίηση του ΜΜΟΕ έγιναν από το συγγραφέα.

Τέλος, προτείνεται ένα δελτίο διαλογικής εξέλιξης γνώσης για το SIS το οποίο υποστηρίζει ένα σύνολο διαδικασιών διαλογικής ενημέρωσης οι οποίες αξιοποιούν τις ΟΕ του ΜΜΟΕ και το περιβάλλον (context) στο οποίο είναι ενταγμένο ένα αντικείμενο, για να παρέχουν φιλική και ελεγχόμενη εξέλιξη γνώσης. Μεταξύ των διαδικασιών αυτών περιλαμβάνεται μια σύνθετη εντολή ενημέρωσης, η εντολή αναπαραγωγής, η οποία διευκολύνει σημαντικά την εισαγωγή σύνθετων αντικειμένων. Η εντολή αυτή μπορεί να χρησιμοποιηθεί για την ημιαυτόματη δημιουργία σύνθετων αντικειμένων που ικανοποιούν εξαρχής μια συνθήκη. Η συνθήκη αυτή παριστάνεται με αντικείμενα της βάσης που την ικανοποιούν, τα οποία ονομάζονται υποδείγματα.

1.3 Η οργάνωση της γραπτής εργασίας

Μετά τη συνοπτική περιγραφή του περιεχομένου της εργασίας αυτής, στο *κεφάλαιο 2* το πρόβλημα εντάσσεται στην ευρύτερη ερευνητική περιοχή. Παρουσιάζονται και αξιολογούνται διάφορες προσεγγίσεις και παρατίθενται χαρακτηριστικά παραδείγματα.

Στο *κεφάλαιο 3* παρουσιάζεται συνοπτικά η γλώσσα παράστασης γνώσης Telos και η ειδική έκδοση της, η SIS-Telos . Επισημαίνονται οι ιδιαιτερότητες της και γίνεται αναφορά στο Σύστημα Σημασιολογικού Ευρετηριασμού (Semantic Index System ή SIS).

Στο *κεφάλαιο 4* παρουσιάζονται οι Στοιχειώδεις Εντολές Ενημέρωσης της SIS-Telos , και τα γενικά χαρακτηριστικά της εξέλιξης γνώσης που λαμβάνει χώρα στις εφαρμογές της.

Στο *κεφάλαιο 5* παρουσιάζεται το *Μοντέλο Διεργασιών Ενημέρωσης*. Παρουσιάζονται παραδείγματα από τη χρήση του και τέλος γίνεται σχολιασμός και αξιολόγηση της προσέγγισης.

Στο *κεφάλαιο 6* παρουσιάζεται το *Μεταμοντέλο Εργασιών Ενημέρωσης*, παρουσιάζονται παραδείγματα από τη χρήση του και γίνεται μελέτη της υλοποίησης του.

Στο *κεφάλαιο 7* παρουσιάζεται μια πρόταση εφαρμογής του ΜΜΟΕ στη διαλογική εξέλιξη γνώσης του SIS.

Τέλος, στο *κεφάλαιο 8* παρουσιάζονται τα αποτελέσματα και τα γενικά συμπεράσματα που προέκυψαν από την εκπόνηση της εργασίας αυτής, καθώς επίσης και κάποιες ιδέες που αφορούν την πιθανή μελλοντική επέκταση της.

Στο παράρτημα Α υπάρχει κατάλογος με τις συντομογραφίες που χρησιμοποιούνται στο κείμενο. Ακολουθεί στο παράρτημα Β η παρουσίαση του Μοντέλου Διεργασιών Ενημέρωσης σε Telos , ενώ στο παράρτημα C παρουσιάζονται οι κυριότερες ερωτηματικές συναρτήσεις της Telos . Τέλος ακολουθεί η βιβλιογραφία που χρησιμοποιήθηκε.

Κεφάλαιο 2

Βιβλιογραφική Ανασκόπηση

Στο κεφάλαιο αυτό αρχικά αναλύεται το γενικό πρόβλημα του *Ελέγχου και της Διευκόλυνσης της Εξέλιξης Γνώσης* και εν συνεχεία γίνεται μια ανασκόπηση των ερευνητικών προσπαθειών που σχετίζονται με την αντιμετώπιση του. Οι ιδέες και προτάσεις που παρουσιάζονται αποτελούν τη βάση για την ανάπτυξη της παρούσας εργασίας.

Η ανασκόπηση εντοπίζεται στο χώρο των Οντοκεντρικών Βάσεων Δεδομένων (ΟΒΔ) καθώς η συγκεκριμένη εργασία αναπτύσσεται για να στηρίξει βάσεις γνώσης της Telos η οποία υποστηρίζει ένα δομικά οντοκεντρικό μοντέλο δεδομένων [1].

2.1 Εξέλιξη Γνώσης

Ο όρος **Εξέλιξη Γνώσης** χρησιμοποιείται για να περιγράψει τις δυνατότητες ενημέρωσης των δεδομένων και του εννοιολογικού σχήματος μιας ΒΔ, συνήθως κατά τη διάρκεια της χρήσης (runtime). Οι **τύποι των εντολών ενημέρωσης**, ο **έλεγχος της εγκυρότητας** τους, καθώς και η **στρατηγική** πραγματοποίησης τους (πχ ο τρόπος διάδοσης στα δεδομένα των ενημερώσεων του σχήματος), αποτελούν αντικείμενο πολλών εργασιών στο χώρο των Οντοκεντρικών Βάσεων Δεδομένων (ΟΒΔ) ([37, 4, 28, 5]).

Κάθε ΣΔΒΔ υποστηρίζει ένα σύνολο **Στοιχειωδών Εντολών Ενημέρωσης (ΣΕΕ)** με το οποίο μπορεί να πραγματοποιηθεί κάθε δυνατή (υποστηριζόμενη) ενημέρωση στη βάση. Ο τρόπος με τον οποίο μπορεί ένας χρήστης να διατυπώσει και να εκτελέσει εντολές ενημέρωσης ποικίλει : (δες και σχήμα 2.1)

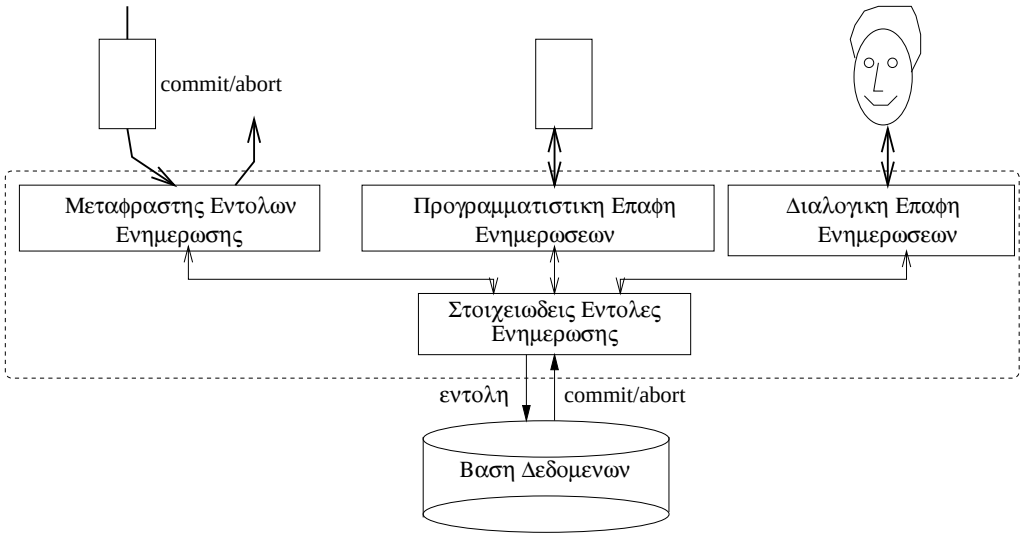
[1] Η γλώσσα Telos περιγράφεται στο κεφάλαιο 3.

1. **Γλώσσα ενημέρωσης για μαζική εισαγωγή δεδομένων**
   Παρέχονται ειδικοί μεταφραστές (parsers) οι οποίοι μεταφράζουν προγράμματα εντολών της γλώσσας αυτής και κάνουν τις κατάλληλες κλήσεις στις ΣΕΕ.

2. **Προγραμματιστική επαφής χρήσεως**
   Η λειτουργικότητα των ΣΕΕ προσφέρεται μέσω ενός συνόλου από διαδικασίες οι οποίες μπορούν να κληθούν από άλλα προγράμματα (ή να ενσωματωθούν σε αυτά).

3. **Διαλογικά Εργαλεία Ενημέρωσης**
   Παρέχονται ειδικά εργαλεία τα οποία δίνουν τη δυνατότητα στο χρήστη να ενημερώσει διαλογικά τη βάση.

Οι δυνατότητες εξέλιξης γνώσης που προσφέρει η Telos (ΣΕΕ) περιγράφονται στα κεφάλαια 3 και 4.



Σχήμα 2.1: Οι εναλλακτικοί τρόποι ενημέρωσης μιας βάσης

2.2 Όψεις Ενημέρωσης

Ορίζουμε ως **Όψη Ενημέρωσης**(ΟΕ) ένα σύνολο (μη διατεταγμένο) από πράξεις ενημέρωσης (Στοιχειώδεις Εντολές Ενημέρωσης) οι οποίες δρούν σε ένα τμήμα της βάσης και έχουν σκοπό την **διευκόλυνση** και τον **έλεγχο** των ενημερώσεων που γίνονται διαλογικά.

2.2.1 Έλεγχος Ενημερώσεων

Ο έλεγχος των ενημερώσεων είναι απαραίτητος ώστε :

- Να αποτρέπονται οι ενημερώσεις που θα αλλοίωναν τον πυρήνα του μοντέλου της βάσης και άρα θα προσέβαλαν την ομαλή λειτουργία της βάσης.
- Να προστατεύονται τα δεδομένα της βάσης για λόγους δικαιοδοσίας.
- Να επιβάλλονται περιορισμοί (στατικοί) στον τρόπο εξέλιξης της βάσης.

2.2.2 Διευκόλυνση Ενημερώσεων

Μια βάση δεδομένων συχνά χρησιμοποιείται από πολλούς χρήστες για την πραγματοποίηση διαφορετικών εργασιών. Κατά συνέπεια το μοντέλο της μπορεί να είναι πολύ μεγάλο και σύνθετο προκειμένου να καλύψει τις παραπάνω ανάγκες. Αυτό δυσκολεύει ένα χρήστη κατά την διαλογική ενημέρωση αφού :

- Είναι δύσκολος ο **εντοπισμός** των τμημάτων της βάσης που τον αφορούν, ή αφορούν μια συγκεκριμένη εργασία, την οποία θέλει να πραγματοποιήσει.
- Έχοντας εντοπίσει το τμήμα της βάσης (πχ ένα αντικείμενο) που αφορά μια εργασία συχνά είναι δύσκολη η διατύπωση (επιλογή) της κατάλληλης εντολής ενημέρωσης, αφού ο αριθμός των πιθανών ενημερώσεων ίσως είναι μεγάλος.
- Συχνά η βάση περιέχει σύνθετα αντικείμενα είτε γιατί παριστάνουν σύνθετα αντικείμενα του πραγματικού κόσμου, ή για λόγους που υπαγορεύονται από τις δυνατότητες του συστήματος και τις ανάγκες επεξεργασίας τους, παρόλο που τα αντικείμενα αυτά στον πραγματικό κόσμο δεν θεωρούνται σύνθετα. Η διαχείριση των σύνθετων αντικειμένων και κυρίως η εισαγωγή τους απαιτεί διευκόλυνση.

2.2.3 Απαιτήσεις

Για τους παραπάνω λόγους θα ασχοληθούμε με όψεις ενημέρωσης που θα αξιοποιούνται από ένα διαλογικό εργαλείο εξέλιξης γνώσης το οποίο :

1. Θα εστιάζει στις ενημερώσιμες πληροφορίες που αφορούν μια όψη ενημέρωσης.
2. Ανά αντικείμενο (του τμήματος της βάσης μιας εργασίας) θα γνωστοποιεί (προτείνει) στο χρήστη το σύνολο των σχετικών εντολών ενημερωσης.
3. Θα αποτρέπει τις ενημερώσεις που αφορούν δεδομένα που δεν αφορούν ένα χρήστη (access control).
4. Θα παρέχει στο χρήστη ένα σύνολο αυτοματισμών (σύνθετων εντολών ενημέρωσης) οι οποίοι θα βοηθούν την ενημέρωση σύνθετων αντικειμένων.

2.3 Οντοκεντρικές Βάσεις Δεδομένων

Ο ευρύτερος θεματικός χώρος της εργασίας αυτής είναι οι οντοκεντρικές βάσεις δεδομένων (ΟΒΔ) [2] ([28], [22], [10]). Πρόκειται για μια νέα γενιά ΣΔΒΔ οι οποίες πετυχαίνουν να περιγράψουν ευκολότερα και με μεγαλύτερη εκφραστικότητα τον πραγματικό κόσμο.

Βασίζονται στο οντοκεντρικό μοντέλο δεδομένων ([11], [21], [33]) το οποίο παριστάνει τη γνώση μέσω της έννοιας της **οντότητας**, και των μηχανισμών **ταξινόμησης, εξειδίκευσης** και **απόδοσης γνωρισμάτων**. Επίσης συχνά παρέχεται η δυνατότητα παράστασης και διαχείρισης *εκδόσεων*, σύνθετων οντοτήτων και διαμορφώσεων (configurations).

Οι δυνατότητες των ΟΒΔ τις κάνουν κατάλληλες για **εφαρμογές σχεδιαστικής φύσεως** (εφαρμογές CAD/CAM/CASE [3] [22]) στις οποίες τα δεδομένα παρουσιάζουν μεγάλο βαθμό ιδιομορφίας (υπάρχουν πολλοί τύποι δεδομένων) και διασύνδεσης, συχνά είναι σύνθετα (composite objects), δηλαδή αποτελούνται από άλλα αντικείμενα τα οποία με τη σειρά τους μπορεί είναι και αυτά σύνθετα, και συχνά περιγράφονται από πολλές απόψεις. Επιπλέον στις εφαρμογές αυτές είναι συχνές οι αλλαγές στο σχήμα των δεδομένων.

Ο χώρος των ΟΒΔ είναι νέος, κατά συνέπεια δεν υπάρχει τυποποίηση αλλά αντίθετα υπάρχει μεγάλη διαφοροποίηση όσον αφορά στα μοντέλα δεδομένων, και στη διαχείριση των οντοτήτων.

2.3.1 Εξέλιξη Γνώσης σε Οντοκεντρικά μοντέλα

Όπως προαναφέρθηκε βασικό στοιχείο των οντοκεντρικών μοντέλων δεδομένων είναι η *οντότητα*, η οποία είναι μια συλλογή δεδομένων που παριστάνουν μια έννοια του πραγματικού κόσμου. Κάθε οντότητα έχει μία και μοναδική ταυτότητα (ένα αναγνωριστικό) που τη διακρίνει από τις υπόλοιπες οντότητες μέσα στη βάση.

Μπορούμε να εκμεταλευτούμε την έννοια της οντότητας ως εξής :

- Κάθε οντότητα έχει ένα τύπο (ανήκει σε μια κλάση) ο οποίος περιγράφει τις ιδιότητες ενός συνόλου οντοτήτων. Το σύνολο των τύπων, το σχήμα , είναι αποθηκευμένο στη βάση και μπορεί να βοηθήσει από μόνο του σημαντικά το χρήστη κατά την εισαγωγή δεδομένων αφού του γνωστοποιεί τους τύπους των δεδομένων που συνθέτουν έναν τύπο οντοτήτων, καθώς και τις σχέσεις μεταξύ τύπων οντοτήτων. Με άλλα λόγια το σχήμα μπορεί να οδηγήσει τη λειτουργικότητα ενός διαλογικού εργαλείου εισαγωγής δεδομένων γενικής χρήσεως.

[2] Με τον όρο αυτό θα εννοούμε τα Συστήματα Διαχείρισης Οντοκεντρικών Βάσεων Δεδομένων.
[3] Computer Aided Design/Manufacturing/Software Engineering

Αντίθετα στο σχεσιακό μοντέλο δεδομένων το σχήμα της εφαρμογής δεν είναι αποθηκευμένο στη βάση. Δεδομένου οτι συχνά η παράσταση ενός αντικειμένου του πραγματικού κόσμου κατακερματίζεται σε πολλούς πίνακες της βάσης, κατανοούμε τη δυσκολία της ορθής διαλογικής εισαγωγής δεδομένων.

- Δυνατότητα ευέλικτου και αποδοτικού ελέγχου ενημερώσεων. Λόγω του ότι κάθε οντότητα έχει ταυτότητα είναι εφικτός ο έλεγχος των ενημερώσεων (πράξεων) ανά οντότητα και όχι μόνο κατά σχέση (relation) όπως συμβαίνει στα σχεσιακά μοντέλα.

2.4 Ανασκόπηση Σχετικών Μηχανισμών

Πολλά συστήματα αντιμετωπίζουν το πρόβλημα του ελέγχου και της διευκόλυνσης της εξέλιξης γνώσης με **υποσχήματα**. Ένα υποσχήμα μιας βάσης είναι ένα υποσύνολο των τύπων της, το οποίο ενδιαφέρει ένα σύνολο χρηστών ή εφαρμογών. Ολες οι πράξεις (άρα και οι ενημερώσεις) περιορίζονται στους τύπους και τις οντότητες του υποσχήματος.

Ένα υποσχήμα είναι δυνατόν να περιέχει **όψεις** (views) ή **πλασματικούς τύπους** (virtual/derived types). Οι **όψεις** ή οι πλασματικοί τύποι αποτελούν περιορισμό ή επέκταση τύπων που υπάρχουν στη βάση.

Τα υποσχήματα δεν μπορούν όμως να περιορίσουν συγκεκριμένες πράξεις σε ένα αντικείμενο και να επιτρέψουν κάποιες άλλες σε αυτό. Η δυνατότητα αυτή υποστηρίζεται όμως από **μηχανισμούς ελέγχου πρόσβασης και δικαιοδοσίας**, οι οποίοι μάλιστα εξετάζουν τις πράξεις που αφορούν το σχήμα της βάσης.

Παρακάτω γίνεται ανασκόπηση ορισμένων ενδιαφέρουσων εργασιών και προσεγγίσεων. Αρχικά μελετώνται οι μηχανισμοί διαχείρισης όψεων ή πλασματικών τύπων, εν συνεχεία παρατίθενται ορισμένες εργασίες που αφορούν υποσχήματα και τέλος παραδείγματα μηχανισμών δικαιοδοσίας. Το κεφάλαιο τελειώνει με συμπεράσματα από την ανασκόπηση.

2.4.1 Όψεις

Οι **όψεις** ή οι πλασματικοί τύποι αποτελούν περιορισμό ή επέκταση τύπων που υπάρχουν στην βάση και προσφέρουν ένα σύνολο από προοπτικές χρήσης, με τις οποίες διαφορετικές εφαρμογές και χρήστες επικοινωνούν και εργάζονται με τη ΒΔ.

Οι όψεις και οι πλασματικοί τύποι, που μας ενδιαφέρουν, είναι αυτοί, στους οποίους είναι εφικτή η ενημέρωση των περιεχομένων τους. Αντίθετα δεν ενδιαφερόμαστε για όψεις, των οποίων τα περιεχόμενα αποθηκεύονται ξεχωριστά από τη βάση (materialized views).

Εργασίες για όψεις υπάρχουν πολλές. Ο όρος άλλωστε *όψη* είναι αρκετά γενικός και συχνά χρησιμοποιείται για να περιγράψει διαφορετικά πράγματα. Στο σχήμα 2.2 παρουσιάζεται μια ταξινόμηση των μηχανισμών αυτών βασισμένη σε έξι γενικά, και ως ένα βαθμό ανεξάρτητα, κριτήρια : **ορισμό, μορφή, περιεχόμενο, αποθήκευση, πράξεις,** και **πρόθεση**.

Σε γενικές γραμμές μπορούμε να πούμε ότι είναι λίγες (σε σχέση με το σύνολο των σχετικών εργασιών) οι περιπτώσεις, κατά τις οποίες είναι εφικτή η ενημέρωση μιας βάσης μέσω όψεών της.

Στις **Σχεσιακές Βάσεις Δεδομένων** όψη είναι μια οποιαδήποτε ερώτηση. Οι δυνατότητες ενημέρωσης τους είναι περιορισμένες. Ενδεικτικά αναφέρουμε ένα παράδειγμα : η ORACLE ([8]) δεν επιτρέπει τη διαγραφή πλειάδων μέσω μιας όψης, αλλά μόνο την εισαγωγή πλειάδων αν τηρούνται οι ακόλουθες προϋποθέσεις : η όψη να περιέχει το κλειδί και ο ορισμός της να μην περιέχει join. Το πρόβλημα έγκειται στην εύρεση σαφούς αντιστοίχισης μεταξύ των πράξεων ενημέρωσης που εφαρμόζονται σε μια όψη και των ενημερώσεων που πρέπει να γίνουν στην πραγματική βάση.

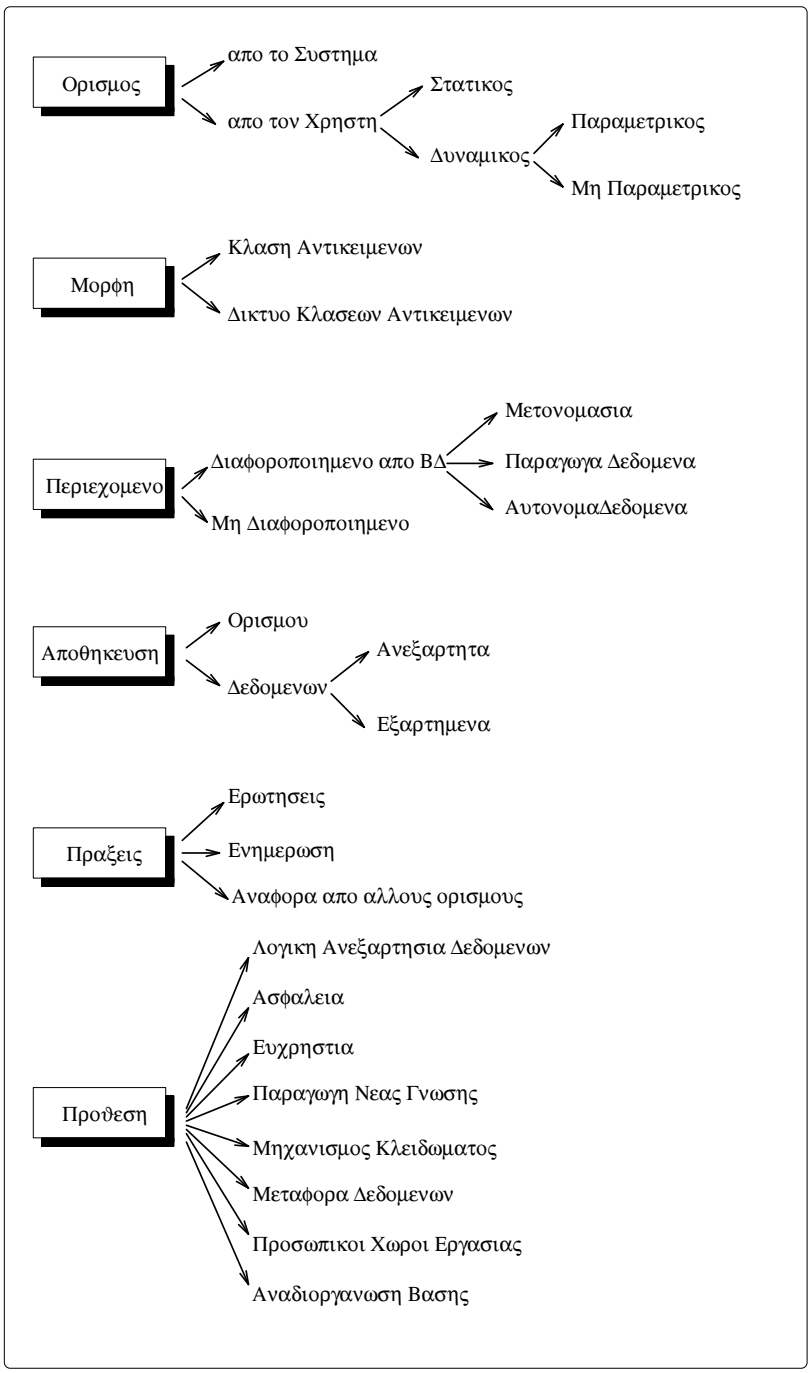
Το πρόβλημα αυτό έχει μελετηθεί θεωρητικά στο [3] και αργότερα στο [19] όπου ορίζεται η έννοια της *συνεπούς όψης* (consistent view) : όψη στην οποία μπορούμε να εφαρμόσουμε μια ενημέρωση και αιτιοκρατικά να καθορίσουμε και την αλλαγή που θα επέλθει στην καθολική ΒΔ. Οι εργασίες όμως αυτές δεν έχουν πρακτικό ενδιαφέρον αφού έχουν απαγορευτική πολυπλοκότητα ([27]). Ενδιαφέρουσα είναι η πρόταση του [27] στην οποία οι ασάφειες αναιρούνται κατά την φάση του ορισμού των όψεων ύστερα από διάλογο μεταξύ του συστήματος και του διαχειριστή της βάσης.

Στις **Οντοκεντρικές Βάσεις Δεδομένων** οι όψεις που συνήθως προτείνονται [4] ορίζονται με τη χρήση της ερωτηματικής γλώσσας. Ενημερώσεις των οντοτήτων μιας όψης είναι εφικτές αν η ταυτότητα των οντοτήτων διατηρείται και αν η ενημέρωση μπορεί να μεταφραστεί με σαφήνεια σε ενημέρωση της βάσης.

Για την ικανοποίηση του πρώτου η ερωτηματική γλώσσα που χρησιμοποιείται στον ορισμό όψεων πρέπει να παρέχει τελεστές που διατηρούν την ταυτότητα των αντικειμένων που επιστρέφονται, σε αντιδιαστολή με τους τελεστές που επιστρέφουν πλειάδες (tuples), νέα ή φανταστικά ([1]) αντικείμενα . Στην κατηγορία αυτή ανήκουν οι εργασίες [40], [18].

Το τελευταίο είναι εφικτό αν η δομή των δεδομένων της όψης ταυτίζεται με την πραγματική (τότε δεν υπάρχει θέμα μετάφρασης), ή αν είναι δυνατή η σαφής αντιστοίχιση της φαινομένης δομής με την πραγματική. Το δεύτερο δεν είναι πάντα εφικτό : μια όψη

[4] Άλλες εναλλακτικές προσεγγίσεις παρουσιάζονται στο [12].



Σχήμα 2.2: Ταξινόμηση μηχανισμών διαχείρισης όψεων.

μπορεί να περιέχει *παράγωγα γνωρίσματα*, τα οποία παίρνουν τιμή από την απάντηση μιας ερώτησης ([40], [1], [18]) ή αντιστοιχούν σε μονοπάτια γνωρισμάτων ([6]). Επιπλέον, μια όψη μπορεί να περιέχει οντότητες και σχέσεις που δεν υπάρχουν στη βάση. Σχετικά παραδείγματα βρίσκουμε σε μηχανισμούς που συμβάλλουν στην αναδιοργάνωση μιας βάσης (μέσω πράξεων σύνθεσης ή αποσύνθεσης οντοτήτων [1]), καθώς και σε μηχανισμούς υποστήριξης αφαιρετικών όψεων ([15]).

Ακολουθούν χαρακτηριστικά παραδείγματα :

1. NO^2

   Η εργασία [18] προτείνει πλασματικούς τύπους κλάσεων και αφορά το οντοκεντρικό μοντέλο δεδομένων NO^2 (New Object Oriented data model) της δομικώς οντοκεντρικής βάσης δεδομένων **CoOMS** (Combined Object Management System). Ένας πλασματικός τύπος προκύπτει με περιορισμό ενός πραγματικού τύπου μέσω **επιλογής** (selection), **προβολής** (projection), ή **προσθήκης παραγώγων (derived) γνωρισμάτων**. Για τον ορισμό, ο οποίος έχει τη μορφή :

   define view <*name*> **of** <class> **as** <*query*>

   χρησιμοποιείται η ερωτηματική γλώσσα η οποία επιστρέφει υπαρκτά αντικείμενα. Αν εξαιρέσουμε τα παράγωγα γνωρίσματα, τα οποία δεν ενημερώνονται, ένας πλασματικός τύπος είναι ενημερώσιμος.

   Για την πραγματοποίηση (instantiation) ενός πλασματικού τύπου, πρέπει να περιλαμβάνονται στον ορισμό του όλα τα υποχρεωτικά (*necessary*) ή μοναδικά (*unique*) γνωρίσματα, ενώ για τη διαγραφή μιας περίπτωσης του, πρέπει να περιλαμβάνονται στον ορισμό του όλες οι εξαρτημένες (dependent) ιδιότητες .

   Η δυνατότητα αναφοράς ενός πλασματικού τύπου από έναν άλλο δεν είναι ομοιόμορφη με τους πραγματικούς τύπους (ένας τύπος δεν μπορεί να κληρονομήσει ιδιότητες από έναν πλασματικό).

2. **COCOON**

   Η πρόταση αυτή ([40]) αφορά το οντοκεντρικό μοντέλο δεδομένων **COCOON**. Προτείνονται όψεις, οι οποίες ορίζονται με τη βοήθεια της ερωτηματικής γλώσσας *COOL* η οποία επιστρέφει πραγματικές οντότητες. Ο ορισμός μιας όψης έχει τη μορφή :

   define view <*name*> **as** <*query*>

και έχει σαν αποτέλεσμα την δημιουργία μιας πλασματικής κλάσεως, με όνομα *name* η οποία ενσωματώνεται στο σχήμα της βάσης. Αυτό επιτυγχάνεται αντιστοιχώντας σε κάθε αποτέλεσμα ερωτηματικού τελεστή ένα τύπο. Ο πληθυσμός των πλασματικών κλάσεων δεν αποθηκεύεται αλλά υπολογίζεται όταν αυτό απαιτείται.

Ανάλογα με τους τελεστές που χρησιμοποιούνται στον ορισμό μιας όψης διακρίνουμε τις όψεις σε όψεις **επιλογής, διαφοράς, ένωσης, τομής, προβολής** και **επέκτασης**.

- Μια όψη **επιλογής** είναι υποκλάση μιας πραγματικής και περιγράφει το υποσύνολο εκείνο του πληθυσμού της πραγματικής κλάσης, που ικανοποιεί μια συγκεκριμένη συνθήκη (ένα κατηγόρημα).
- Μια όψη **διαφοράς** είναι **υποκλάση** μιας πραγματικής, η οποία περιγράφει το υποσύνολο του πληθυσμού της πραγματικής κλάσης που δεν ικανοποιεί μια συγκεκριμένη συνθήκη.
- Μια όψη **ένωσης** είναι υπερκλάση δύο κλάσεων, στην οποία μπορεί να αποδοθούν (γενικευθούν) τα κοινά γνωρίσματα των δύο κλάσεων.
- Μια όψη **τομής** αποτελεί υποκλάση δύο κλάσεων και περιλαμβάνει τις κοινές οντότητες των δύο κλάσεων.
- Μια όψη **προβολής** αποτελεί υπερκλάση μιας πραγματικής κλάσης, η οποία περιέχει μόνο τα προβαλλόμενα γνωρίσματα .
- Οι όψεις **επέκτασης** είναι υποκλάσεις μιας πραγματικής κλάσης, οι οποίες περιέχουν επιπλέον ένα ή περισσότερα παράγωγα γνωρίσματα (derived) στα οποία δεν επιτρέπονται ενημερώσεις.

Οι όψεις που περιγράφηκαν χρησιμοποιούνται στον ορισμό υποσχημάτων (σύνολα πραγματικών και πλασματικών τύπων) τα οποία εξυπηρετούν λόγους ασφάλειας, δικαιοδοσίας και διαφοροποίησης μεταξύ του φαινομενικού και του πραγματικού σχήματος της βάσης.

Η εν λόγω πρόταση παρουσιάζει κάποιες **αδυναμίες** οι σημαντικότερες των οποίων είναι :

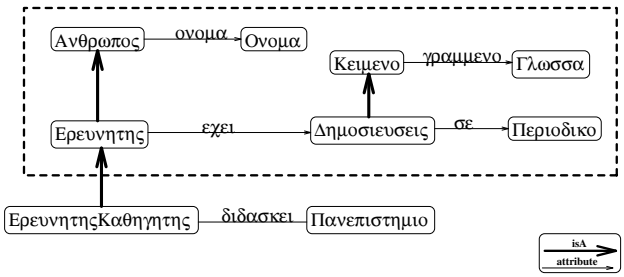
- Στις πλασματικές κλάσεις *επιλογής* υπάρχει πρόβλημα κατά την δημιουργία περιπτώσεων : πρέπει ή δεν πρέπει να γίνεται δεκτή η δημιουργία περιπτώσεων που δεν ικανοποιούν τη συνθήκη της πλασματικής κλάσης ;
- Υπάρχει ασάφεια κατά τη δημιουργία ή διαγραφή περιπτώσεων σε πλασματικές κλάσεις *ένωσης* ή *τομής* : δεν είναι ξεκάθαρη η επιλογή της πραγματικής κλάσης

ή κλάσεων στις οποίες η πράξη (δημιουργία/διαγραφή) πρέπει να εφαρμοσθεί.

- Δυσκολία συνεπούς ενσωμάτωσης (ένταξης) μιας πλασματικής κλάσης *επιλογής* σε ιεραρχία κλάσεων που περιέχει και άλλες πλασματικές κλάσεις επιλογής : το πρόβλημα αυτό ανάγεται στο πρόβλημα του *predicate subsumption* η επίλυση του οποίου, στη γενική του μορφή, είναι αμφισβητίσιμη (undecidable).
- Δύο σύνθετες ερωτήσεις μπορεί να είναι αλγεβρικά ισοδύναμες, αλλά να μεταβάλλουν, κατά την ενσωμάτωση τους, το σχήμα της βάσης με διαφορετικό τρόπο.
- Υπερβολική διεύρυνση του σχήματος της βάσης αν ο αριθμός των όψεων που έχουν ορισθεί είναι μεγάλος.

3. **Όψεις από ένα Οπτικό Σημείο**

Ο Bellahsene [6] προτείνει όψεις που έχουν τη μορφή πλασματικών κλάσεων, οι οποίες γίνονται υποκλάσεις μιας πραγματικής κλάσης που ονομάζεται *οπτικό σημείο*. Ο ορισμός της γίνεται με μια ερώτηση (προβολή, επιλογή) στην ''μέγιστη περιοχή'' του οπτικού σημείου. Μέγιστη περιοχή μιας κλάσης είναι το δίκτυο που διατρέχεται κατά την διάσχιση όλων των γνωρισμάτων της κλάσης και των υπερκλάσεων αυτής αναδρομικά (δες σχήμα 2.3) και χρησιμοποιείται για να θέσει ένα όριο στην εμβέλεια των ενημερώσεων.



Σχήμα 2.3: Παράδειγμα *μέγιστης περιοχής* οπτικού σημείου

Η μέγιστη περιοχή της κλάσης *Ερευνητής* είναι εγκλεισμένη στο πλαίσιο

Μια πλασματική κλάση μπορεί να περιέχει πλασματικά γνωρίσματα, τα οποία αντιστοιχούν σε μονοπάτια γνωρισμάτων, στα πλαίσια βέβαια της μέγιστης περιοχής. Κατά συνέπεια η απόδοση τιμής σε ένα γνώρισμα μιας περίπτωσης μιας πλασματικής κλάσεως μπορεί να πυροδοτήσει την δημιουργία πολλών οντοτήτων και σχέσεων μεταξύ τους.

Οι ενημερώσεις μέσω πλασματικών κλάσεων στην εν λόγω πρόταση είναι σαφείς γιατί τα γνωρίσματα του μοντέλου δεδομένων είναι μονότιμα.

Οι προτεινόμενες όψεις αποσκοπούν στον περιορισμό των δεδομένων μιας κλάσεως (έμφαση στα δεδομένα που ενδιαφέρουν), καθώς και στην διευκόλυνση του χρήστη κατά την εισαγωγή στοιχείων που προκύπτει λόγω των πλασματικών γνωρισμάτων.

4. Ο Hailpern [20] προτείνει ένα μοντέλο για οντοκεντρικές γλώσσες, στο οποίο οι οντότητες μπορούν να έχουν πολλές *όψεις*. Ως *όψη οντότητας* ονομάζει ένα υποσύνολο των πράξεων που έχουν ορισθεί σε αυτήν, το οποίο εκχωρείται σε χρήστες ή εφαρμογές. Οι πράξεις περιγράφονται από τριάδες της μορφής $(object, user, operation)$ και καθορίζουν τις πράξεις που ένας χρήστης μπορεί να εφαρμόσει σε μια οντότητα, άρα :

$$View \subset Object \times User \times Operation$$

Για τη διευκόλυνση του ορισμού τους προτείνεται ο ορισμός οντοτήτων και χρηστών κατά κλάσεις με τη χρήση των συμβολισμών (τελεστών) *instances(k)*, για τις άμεσες περιπτώσεις και *members(k)* για τις έμμεσες (οι άμεσες περιπτώσεις της k και όλων των υποκλάσεων αυτής).

2.4.2 Υποσχήματα

Όπως προαναφέρθηκε, υποσχήμα μιας βάσης είναι ένα υποσύνολο των τύπων της, πραγματικών ή πλασματικών (όψεων), το οποίο χρησιμοποιείται για να περιορίσει τις προσπελάσεις των χρηστών στο τμήμα εκείνο της βάσης, το οποίο τους ενδιαφέρει. Παρατίθενται ορισμένες σχετικές εργασίες:

1. **CoOMS**

   Οι πλασματικοί τύποι που προτείνονται στην πρόταση [18] παρουσιάστηκαν προηγουμένως. Χρήστες και εφαρμογές προσπελαύνουν τη βάση μέσω υποσχημάτων προκειμένου να είναι δυνατή η αναγνώριση των τμημάτων, και άρα των εφαρμογών της βάσης, που επηρεάζονται από μία ενημέρωση. Κατά συνέπεια ένα υποσχήμα πρέπει να είναι **καλώς ορισμένο** : οι βασικοί τύποι των πλασματικών τύπων που περιέχονται πρέπει να ανήκουν στο υποσχήμα. Επίσης πρέπει να είναι **κλειστό**: να μην υπάρχει αναφορά σε έναν τύπο που δεν ανήκει στο υποσχήμα.

2. **O2**

   Η **O2** [1] υποστηρίζει υποσχήματα (αναφέρονται με το όνομα όψη), τα οποία εκλαμβάνονται ως βάσεις δεδομένων. Αποτελούνται από ένα σύνολο εισαγόμενων

τύπων (από την καθολική βάση) οι οποίοι εισάγονται αυτούσιοι ή εξειδικευμένοι (με επιλογή των επιθυμητών ιδιοτήτων, ή με απόκρυψη των μη επιθυμητών ιδιότητων). Στο υποσχήμα περιλαμβάνονται αυτόματα οι υποκλάσεις των εισαγόμενων τύπων μαζί με τις ιδιότητες και τις περιπτώσεις τους. Επιπλέον υποστηρίζεται η δυνατότητα ορισμού ενός υποσχήματος ως εξειδίκευση κάποιου άλλου.

3. Η εργασία [42] προτείνει την επέκταση του οντοκεντρικού μοντέλου ορίζοντας σε κάθε κλάση πολλαπλές επαφές χρήσεως (interfaces). Κατά αυτόν τον τρόπο προτείνεται ένας στατικός ορισμός όψεων ή υποσχημάτων όπου κάθε όψη ή υποσχήμα είναι ένα σύνολο από διατεταγμένα ζεύγη <κλάση, επαφή χρήσεως>. Με τον τρόπο αυτό ορίζονται εννοιολογικές τομές (conceptual slices) οι οποίες αποτελούν αφαιρέσεις μεγάλων και πολύπλοκων δομών δεδομένων, διευκολύνοντας έτσι τη διαχείριση των δομών αυτών.

2.4.3 Μηχανισμοί Ελέγχου Πρόσβασης

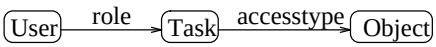
Σκοπός των μηχανισμών αυτών είναι η προστασία των δεδομένων μιας ΒΔ. Η προστασία περιλαμβάνει τον έλεγχο των προσπελάσεων (ανάγνωση,ενημέρωση) που μπορούν να κάνουν οι χρήστες στα δεδομένα ή στο σχήμα μιας ΒΔ. Επίσης υποστηρίζουν δυνατότητες απονομής και ακύρωσης των δικαιωμάτων προσπέλασης μεταξύ των χρηστών.

Μια κατηγορία τέτοιων μηχανισμών είναι οι μηχανισμοί **υποχρεωτικής** (mandatory) ή **πολυβάθμιας** (multilevel) ασφάλειας [44]. Οι μηχανισμοί αυτοί ταξινομούν ιεραρχικά τα δεδομένα (αντικείμενα) και τους χρήστες (ή εφαρμογές) σε διαφορετικές βαθμίδες δικαιοδοσίας. Η βαθμίδα δικαιοδοσίας ενός αντικειμένου και η βαθμίδα δικαιοδοσίας ενός χρήστη καθορίζουν τα δικαιώματα του χρήστη στο αντικείμενο. Οι μηχανισμοί αυτοί προϋποθέτουν ότι στη βάση υπάρχει (είναι εφικτή) μια ιεραρχική ταξινόμηση χρηστών και δεδομένων, το οποίο όμως δεν είναι συχνό φαινόμενο, αλλά εμφανίζεται κυρίως σε στρατιωτικές εφαρμογές. Αντίθετα στην παρούσα εργασία οι ενημερώσεις καθορίζονται με βάση την εργασία ενημέρωσης.

Μια άλλη κατηγορία μηχανισμών ελέγχου πρόσβασης είναι οι μηχανισμοί **διακριτικής** (discretionary) ασφάλειας. Σε αυτήν την κατηγορία τα δικαιώματα καθορίζονται από τριάδες τις μορφής : <*υποκείμενο, πράξη, αντικείμενο*>. Μια τριάδα δηλώνει το δικαίωμα του **υποκειμένου** να εφαρμόσει την **πράξη** στο **αντικείμενο**. Η προσέγγιση αυτή είναι πιο ευέλικτη και εκφραστική και σχετίζεται με το πρόβλημα που εξετάζουμε. Ακολουθούν παραδείγματα :

1. **Concept Base**

   Η εργασία [43] προτείνει το **Group Security Model (GSM)**, ένα μοντέλο για ασφάλεια σε Βάσεις Γνώσης. Βασίζεται στις αρχές *Need to Know* και *Need to Do*, δηλαδή στον περιορισμό του συνόλου των προσπελάσεων στις αναγκαίες για την επίτευξη κάποιου έργου (task). Τα συστατικά μέρη του **GSM** φαίνονται στο σχήμα 2.4.



Σχήμα 2.4: Το μοντέλο **GSM**.

   Το GSM εξειδικεύεται για την κάλυψη των αναγκών της Telos όπως αυτή έχει σχεδιαστεί και υλοποιηθεί στην *ConceptBase*. Η έκδοση αυτή της Telos διαφέρει σημαντικά από την έκδοση της Telos με την οποία ασχολείται η παρούσα εργασία, στο ότι η πρώτη υποστηρίζει παραγωγικούς μηχανισμούς (deductive rules).

2. **ORION**

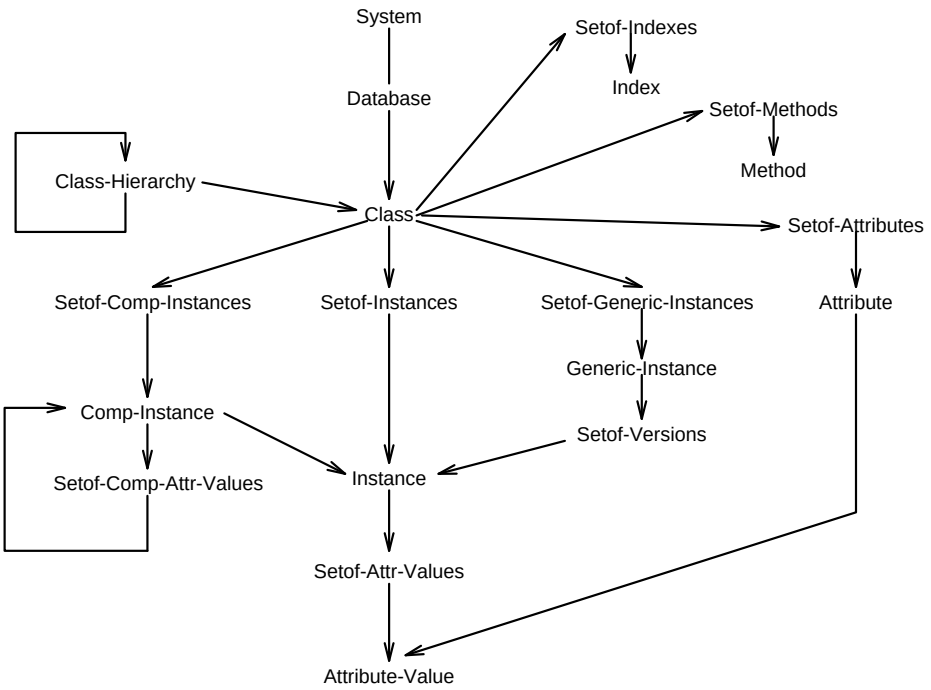
   Στην εργασία [36] προτείνεται ένα μοντέλο δικαιοδοσίας για βάσεις δεδομένων της επόμενης γενιάς. Μια δικαιοδοσία περιλαμβάνει ένα **υποκείμενο** (χρήστη), ένα **τύπο πρόσβασης** (ανάγνωση, εγγραφή, δημιουργία) και ένα **αντικείμενο**. Ιδιαίτερη έμφαση δίνεται στην ιδέα της *έμμεσης δικαιοδοσίας* (implicit authorization) και στον τρόπο χρήσης της στα σημασιολογικά μοντέλα. Η ιδέα της έμμεσης δικαιοδοσίας είναι ότι η δήλωση μιας δικαιοδοσίας μπορεί να **συνεπάγεται** έναν αριθμό από άλλες δικαιοδοσίες.

   Οι έμμεσες δικαιοδοσίες συμβάλλουν στην ελαχιστοποίηση του χώρου για την αποθήκευση τους και προσφέρουν ευκολία (συντομία) κατά τον ορισμό της δικαιοδοσίας σε μια βάση. Εισάγουν όμως ένα επιπλέον υπολογιστικό κόστος. Έμμεσες δικαιοδοσίες υποστηρίζονται και στις τρεις διαστάσεις : **υποκείμενο** (δες σχήμα 2.7), **πράξη** (δες σχήμα 2.6), **αντικείμενο** (δες σχήμα 2.5).

   Για λόγους ευελιξίας είναι εφικτή η περιγραφή εξαιρέσεων, αφού εκτός από τις **θετικές** υποστηρίζονται και **αρνητικές** δικαιοδοσίες, εν αντιθέσει με άλλα συστήματα στα οποία η απουσία μιας δικαιοδοσίας υποδηλώνει τη μη δικαιοδοσία (άρνηση). Η εισαγωγή των αρνητικών δικαιοδοσιών σε συνδυασμό με τη δυνατότητα έμμεσων δικαιοδοσιών μπορεί να δημιουργήσει ασάφειες. Για το λόγο αυτό οι δικαιοδοσίες διακρίνονται σε **ισχυρές** και **ασθενείς**. Οι ισχυρές υπερισχύουν των ασθενών : αν δηλαδή ένα αντικείμενο είναι στην εμβέλεια δύο αντίθετων δικαιοδοσιών, υπερισχύει η ισχυρή. Έτσι είναι δυνατόν μια αρνητική (και ισχυρή) δικαιοδοσία να

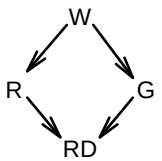
υπερισχύει θετικών (αλλά ασθενών) δικαιοδοσιών.

Ένα μέρος της πρότασης αυτής έχει υλοποιηθεί στην ΟΒΔ **ORION**.



Σχήμα 2.5: Η ιεραρχία *τύπων οντοτήτων* δικαιοδοσίας

Οι δικαιοδοσίες εκχωρούνται σε οντότητες διαφόρων τύπων. Οι τύποι αυτοί είναι οργανωμένοι σε ένα γράφο ο οποίος περιγράφει την έμμεση ισχύ των δηλώσεων δικαιοδοσίας. Η δήλωση μιας δικαιοδοσίας σε ένα αντικείμενο ενός τύπου ισχύει σε όλες τις οντότητες *χαμηλότερων* τύπων.



Σχήμα 2.6: Η ιεραρχία *τύπων πρόσβασης*

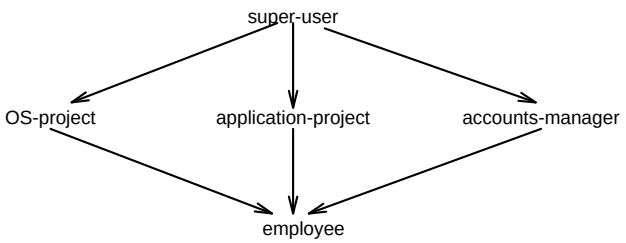
Η μετάφραση των όρων που εμφανίζονται είναι W:Write, R:Read, G:Generate, RD:ReadDefinition. Ένας τύπος πρόσβασης συνεπάγεται τους *χαμηλότερους* του.

2.5 Συμπεράσματα

Από την ανασκόπηση των ερευνητικών εργασιών προέκυψαν τα εξής γενικά συμπεράσματα :

1. **Περιεχόμενο**

   Ενημερώσεις σε όψεις υποστηρίζονται όταν τα αντικείμενα που περιέχονται σε



Σχήμα 2.7: Η ιεραρχία *ρόλων* ή *υποκειμένων*

Οι δικαιοδοσίες εκχωρούνται σε ρόλους. Οι ρόλοι είναι οργανωμένοι σε ένα γράφο. Σε ένα ρόλο αντιστοιχούν και τα δικαιώματα που έχουν εκχωρηθεί σε χαμηλότερους του.

αυτήν, διατηρούν την ταυτότητα τους και όταν αυτές έχουν την ίδια δομή με την πραγματική ή υπάρχει μια σαφής αντιστοίχιση στην πραγματική (όπως συμβαίνει με τα μονοπάτια γνωρισμάτων στην [6]).

2. **Τρόπος ορισμού**

   Η δυνατότητα *θετικών* (select) και *αρνητικών* (hide) ορισμών παρέχει ευελιξία και εκφραστικότητα ([1], [40], [36]).

   Ευκολία και ευελιξία στον ορισμό δικαιωμάτων, καθώς και οικονομία του χώρου αποθήκευσής τους, παρέχεται αν υποστηρίζονται ομαδικές ή *έμμεσες* περιγραφές. ([36], [43]). Μειονέκτημα το επιπλέον υπολογιστικό κόστος που εισάγουν. Η ευελιξία αυξάνεται ακόμα περισσότερο αν επιτρέπονται επικαλύψεις των εμβελειών τους, οι οποίες όμως δεν οδηγούν σε ασάφειες (επιλύονται).

3. **Αποθήκευση**

   Οι όψεις της βάσης μπορεί να ενσωματώνονται στο ίδιο το μοντέλο, όπως συμβαίνει συχνά με τους πλασματικούς τύπους. Αυτό δημιουργεί αρκετά προβλήματα όταν χρησιμοποιούνται για ενημέρωση. Η χρήση των πλασματικών τύπων δεν είναι ομοιόμορφη με τους πραγματικούς, υπάρχουν πολλές ασάφειες σε μερικά είδη ενημέρωσης και αν θέλουμε πολλές προοπτικές χρήσης του σχήματος, το μέγεθος και η πολυπλοκότητα του σχήματος διευρύνεται σημαντικά ([40]). Τα προβλήματα αυτά μειώνονται αν οι πλασματικοί τύποι αποτελούν περιορισμό ενός μόνο τύπου ([18]).

Κεφάλαιο 3

Η γλώσσα Telos και το σύστημα SIS

Η Telos είναι μια γλώσσα παράστασης γνώσης η οποία υποστηρίζει ένα δομικώς οντοκεντρικό μοντέλο δεδομένων. Προσφέρει τους εξής στατικούς εκφραστικούς μηχανισμούς : μη-φραγμένη ιεραρχία ταξινόμησης, πολλαπλή και αυστηρή κληρονομικότητα και πλειότιμα γνωρίσματα τα οποία μπορούν να έχουν τα δικά τους γνωρίσματα. Επίσης παρέχει μηχανισμούς εξαγωγής συμπερασμάτων και χρονικής λογικής. Η γλώσσα Telos αναπτύχθηκε από τον Μυλόπουλο και τους συνεργάτες του [32]. Μια εκδοχή του δομικού μέρους της γλώσσας, η SIS-Telos , έχει αναπτυχθεί από την Ομάδα Πληροφοριακών Συστημάτων και Τεχνολογίας Λογισμικού του Ινστιτούτου Πληροφορικής του Ιδρύματος Τεχνολογίας Έρευνας (ΙΤΕ) και αποτελεί τον πυρήνα του Συστήματος Σημασιολογικού Ευρετηριασμού (SIS, βλέπε 3.4) και το πλαίσιο της παρούσας εργασίας.

3.1 Ιδιαιτερότητες και Πλεονεκτήματα της Telos

Οι πολλές στάθμες αφαίρεσης που παρέχει η γλώσσα Telos προσφέρουν τη δυνατότητα οργάνωσης του σχήματος σύμφωνα με ένα μετασχήμα, το οποίο περιγράφει πιο αφηρημένες έννοιες και ιδιότητες, επιτρέποντας έτσι την διατύπωση και απάντηση ερωτήσεων που βασίζονται σε αφηρημένες ιδιότητες. Με τη σειρά τους, οι έννοιες του μετασχήματος μπορούν να υπάγονται σε ένα μετα-μετασχήμα, κ.ο.κ.

Ετσι η Telos παρέχει ομοιόμορφη διαχείριση δεδομένων και σχήματος τόσο στην εισαγωγή στοιχείων όσο και στον μηχανισμό ερωτήσεων. Αυτό επιτρέπει δυναμικό ορισμό και ενημέρωση του σχήματος της βάσης χωρίς χρονική επιβάρυνση.

Οι οντότητες και τα γνωρίσματα θεωρούνται και χρησιμοποιούνται σαν ισότιμα αντικείμενα, τα οποία μπορούν να έχουν λογικά ονόματα. Το λογικά ονόματα παρέχουν

επιπλέον πληροφορίες για τα αντικείμενα, κάνοντας έτσι τα περιεχόμενα μιας βάσης πιο κατανοητά στο χρήστη. Επίσης αποτελούν έναν άμεσο και κατανοητό τρόπο αναφοράς των αντικειμένων της βάσης.

Στην SIS-Telos οι σύνδεσμοι (ταξινόμησης, γενίκευσης, γνωρισμάτων) αποθηκεύονται ως σύνδεσμοι διπλής κατεύθυνσης προσφέροντας καλές επιδόσεις στις διασχίσεις και ερωτήσεις, μεγαλώνοντας όμως το μέγεθος της βάσης.

3.2 Παράσταση γνώσης στη γλώσσα Telos

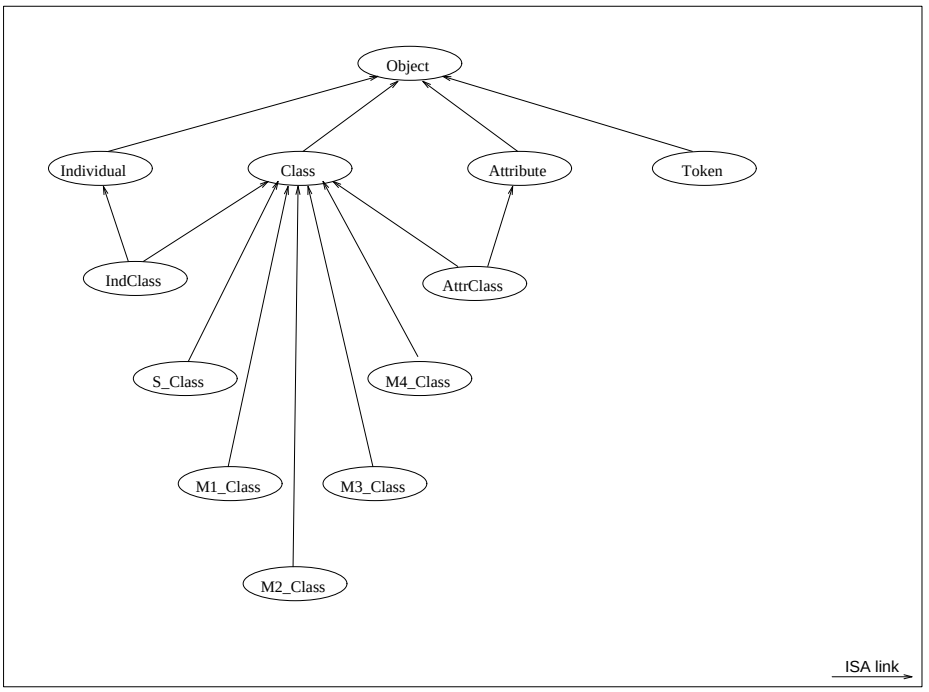
Παρακάτω περιγράφονται συνοπτικά οι εκφραστικοί μηχανισμοί που παρέχονται από τη γλώσσα Telos .

- *Ταυτότητα και Ονοματοδοσία Αντικειμένων*
  Κάθε οντότητα έχει ένα μοναδικό, εσωτερικό και παραγόμενο από το σύστημα, αναγνωριστικό όνομα (SYSID). Επιπλέον ο χρήστης έχει τη δυνατότητα να ονομάσει ο ίδιος μια οντότητα με ένα *λογικό όνομα*. Όταν πρόκειται για γνώρισμα, το λογικό όνομα του είναι της μορφής *x.y*, όπου *x* είναι το λογικό όνομα της οντότητας της οποίας αποτελεί γνώρισμα και *y* είναι τό όνομα του ιδίου. Ο μηχανισμός ονοματοδοσίας πρόκειται να επεκταθεί και η σχετική μελέτη υπάρχει στο [25].

- *Ταξινόμηση*
  Με το μηχανισμό αυτό ένα ατομικό αντικείμενο περιγράφεται ως μέλος (περίπτωση) μιας κλάσης, της οποίας κληρονομεί τα γνωρίσματα. Μια κλάση είναι και αυτή με τη σειρά της ένα αντικείμενο, άρα μπορεί να είναι περίπτωση μιας άλλης κλάσης. Η Telos απαιτεί κάθε αντικείμενο να αποτελεί περίπτωση μιας κλάσης. Έτσι δημιουργείται μια μη φραγμένη ιεραρχία από κλάσεις. Στην κατώτερη στάθμη τοποθετούνται τα *ατομικά αντικείμενα* ή *Tokens*, παραπάνω υπάρχουν οι *απλές κλάσεις* που αποτελούνται από ατομικά αντικείμενα, μετά οι *μετακλάσεις* που αποτελούνται από απλές κλάσεις, οι *μετα-μετακλάσεις*, κοκ. Η μη φραγμένη ιεραρχία είναι άπειρη αλλά όχι υποχρεωτικά. Η απαίτηση κάθε αντικείμενο να ανήκει σε μια κλάση εκπληρώνεται με την ύπαρξη ειδικών *κλάσεων του συστήματος*. Οι κλάσεις του συστήματος δεν μπορούν να αλλαχθούν από τον χρήστη. Αποτελούν τον αρχικό πληθυσμό της βάσης και οποιαδήποτε δεδομένα , που θα εισάγει στο σύστημα ο χρήστης, πρέπει να συνδεθούν έμμεσα ή άμμεσα με αυτές [32]. Ακολουθεί μια σύντομη περιγραφή τους.

Κλάσεις συστήματος

Οι κύριες κλάσεις συστήματος και η σχέση μεταξύ τους φαίνεται στο σχήμα 3.1. Όλα τα αντικείμενα που περιγράφονται στην Telos ταξινομούνται στην κλάση του συστήματος *Object*. Υποκλάσεις της *Object* είναι οι κλάσεις *Individual*, *Attribute*, *Class* και *Token*. Στην κλάση *Individual* ταξινομούνται τα αντικείμενα, οι κλάσεις αντικειμένων, οι κλάσεις από κλάσεις αντικειμένων, κοκ. Στην κλάση *Attribute* ταξινομούνται οι σχέσεις που έχουν τα αντικείμενα μεταξύ τους , οι κλάσεις αντικειμένων, οι κλάσεις από κλάσεις αντικείμενων, κοκ. Οι κλάσεις που ορίζει ο χρήστης ταξινομούνται στην κλάση του συστήματος *Class*. Τα Tokens ταξινομούνται στην κλάση *Token*. Οι απλές κλάσεις ταξινομούνται στην κλάση *S_Class*, οι μετακλάσεις στην *M1_Class*, οι μετα-μετακλάσεις στην *M2_Class*, κοκ. Στο σύνολο των κλάσεων συστήματος ανήκουν και οι κλάσεις των πρωτογενών τιμών *Integer*, *Real*, *String*. Οι πρωτογενείς τιμές δεν μπορούν να δημιουργηθούν ή να καταστραφούν αλλά μόνο να γίνει αναφορά σε αυτές.



Σχήμα 3.1: Η ιεραρχία των κλάσεων του συστήματος.

Ένα αντικείμενο μπορεί να ανήκει σε παραπάνω από μία κλάσεις. Με αυτόν τον τρόπο μπορεί η ταξινόμηση να υποκαταστήσει την απόδοση γνωρισμάτων. Με αυτόν τον τρόπο όμως πρέπει να αποδίδονται μόνο εγγενείς ιδιότητες ενός αντικειμένου ενώ

ο μηχανισμός απόδοσης γνωρίσματος είναι πιο κατάλληλος για την περιγραφή των επιφανειακών γνωρισμάτων που πιθανόν να έχει ένα αντικείμενο.

- *Γενίκευση (αντίστροφο : εξειδίκευση)*
  Ο μηχανισμός αυτός ισχύει μόνο για κλάσεις (όχι δηλαδή για ατομικά αντικείμενα) που βρίσκονται στην ίδια στάθμη αφαίρεσης. Ορίζει μια σχέση υποσυνόλου μεταξύ των κλάσεων που ονομάζεται *isA* . Αν A *isA* B (A και B κλάσεις), τότε η A ονομάζεται υποκλάση της B και η B υπερκλάση της A. Η A κληρονομεί όλα τα γνωρίσματα της B και είτε έχει επιπλέον γνωρίσματα είτε περιορίζει το σύνολο τιμών των γνωρισμάτων που κληρονομεί από την B. Μια κλάση μπορεί να έχει παραπάνω από μια υπερκλάσεις. Έτσι η σχέση *isA* υποστηρίζει πολλαπλή και αυστηρή κληρονόμηση.

  Ο μηχανισμός γενίκευσης/εξειδίκευσης επιτρέπει την οργάνωση των κλάσεων σε διάφορες ιεραρχίες γενίκευσης οι οποίες προσδίδουν οικονομία και συνέπεια στο μοντέλο, αφού δεν χρειάζεται να επαναληφθεί ο ορισμός ενός γνωρίσματος που έχει ήδη αποδοθεί σε μια γνωστή υπερκλάση.

- *Απόδοση Γνωρίσματος*
  Με το μηχανισμό αυτό αποδίδονται γνωρίσματα στα αντικείμενα. Ένα αντικείμενο κληρονομεί τα γνωρίσματα των κλάσεων στις οποίες ανήκει [1]. Στα γνωρίσματα αυτά μπορούν να δοθούν παραπάνω από μια τιμές (πλειότιμα γνωρίσματα) ή να μη δοθεί καμιά (δεν είναι υποχρεωτικά). Επίσης κάθε γνώρισμα μπορεί να έχει και αυτό γνωρίσματα καθώς επίσης να αποτελεί περίπτωση μιας κλάσης γνωρισμάτων, μετακλάσης γνωρισμάτων, κοκ. Οι δυνατότητες αυτές απορρέουν από την ισότιμη μεταχείριση οντοτήτων και γνωρισμάτων από την Telos . Μια κλάση γνωρισμάτων λέγεται και *κατηγορία.*

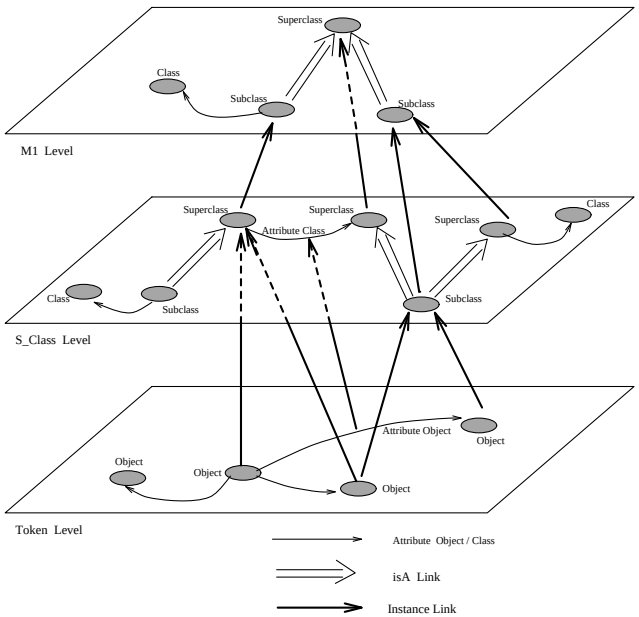
  Τα γνωρίσματα μπορούν να θεωρηθούν ως διμελείς σχέσεις με πεδίο ορισμού τα αντικείμενα, στα οποία αποδίδονται, και πεδίο τιμών τις οντότητες, που αποτελούν τιμές του γνωρίσματος. Ισχύουν οι παρακάτω περιορισμοί:

  Περιορισμός 1 *Αν ένα γνώρισμα είναι περίπτωση μιας κατηγορίας γνωρισμάτων, τότε το πεδίο ορισμού και το πεδίο τιμών του πρέπει να είναι περιπτώσεις των πεδίων ορισμού και τιμών αντιστοίχως της κλάσης γνωρισμάτων στην οποία ανήκει.*

[1] Περιλαμβάνονται και οι κλάσεις συστήματος με τις οποίες μπορούμε να δηλώνουμε αφηρημένες ιδιότητες.

Περιορισμός 2 *Έστω AC1 και AC2 κατηγορίες γνωρισμάτων. Αν AC1* isA *AC2, τότε τα πεδία ορισμού και τιμών της AC1 είναι υποκλάσεις των αντίστοιχων της AC2.*

Στο σχήμα 3.2 παρουσιάζεται ένα μοντέλο της Telos , στο οποίο φαίνεται η χρήση των μηχανισμών παράστασης γνώσης αυτής.



Σχήμα 3.2: Ένα μοντέλο της Telos .

Ακολουθεί ένα παράδειγμα του συντακτικού της γλώσσας εισαγωγής δεδομένων της Telos (Telos data entry language).

```
TELL Individual Ανθρωπος in S_Class with
    attribute
        όνομα      : Telos_String;
        επίθετο    : Telos_String
end


TELL Individual Μαθητής in S_Class isA Άνθρωπος with
    attribute
        σχολείο    : Σχολείο
        αριθμό     : Telos_Integer
end


TELL Individual ΓΤ in Token , Μαθητής with
    όνομα
                : "Γιάννης"
    επίθετο
                : "Τζίτζικας"
    σχολείο
                : ΠανεπιστήμιοΚρήτης
end
```

3.3 Τυπική περιγραφή της SIS-Telos

Μια βάση δεδομένων σε Telos παριστάνει τον πραγματικό κόσμο με ένα σύνολο στοιχείων O (Objects), και ένα σύνολο σχέσεων R (Relations).

Το σύνολο O και το σύνολο R διαμερίζονται ως εξής :

$$\begin{aligned} O &= O_u \cup O_{sys} \cup PV \\ PV &= Integer \cup Real \cup String \\ R &= R^{in} \cup R^{isA} \cup R^{from} \cup R^{to} \\ R^{in} &= R^{in}_{sys} \cup R^{in}_{u} \\ R^{isA} &= R^{isA}_{sys} \cup R^{isA}_{u} \end{aligned}$$

Το σύνολο O_{sys} περιέχει τα στοιχεία που αποτελούν τον πυρήνα του μοντέλου της Telos και τα μέλη του ονομάζονται *κλάσεις συστήματος*. Το PV διαμερίζεται στα σύνολα πρωτογενών τιμών Integer , Real , String τα οποία είναι προδηλωμένα και αντιστοιχούν στους αντίστοιχους τύπους τιμών της γλώσσας C (int, float, char*), ενώ το σύνολο O_u είναι το σύνολο των στοιχείων που έχουν ορισθεί από τον χρήστη. Τα O_{sys} και O_u, τα στοιχεία των οποίων ονομάζονται αντικείμενα, είναι ισόμορφα προς δύο ξένα υποσύνολα των φυσικών αριθμών, που περιέχουν τα αντίστοιχα αναγνωριστικά συστήματος. Επίσης μία βάση σχετίζεται με ένα σύνολο ονομάτων L, μέσω της συνάρτησης ονοματοδοσίας $\mathcal{L} : O_{sys} \cup O_u \rightarrow L$ (βλέπε [25]).

Στο σύνολο O ορίζεται η συνάρτηση R^{in}_{sys} (ταξινόμηση σε κλάση συστήματος) :

$$R^{in}_{sys} : O \rightarrow O_{sys}$$

Στο σύνολο O_{sys} ορίζονται οι ακόλουθες συναρτήσεις οι τιμές των οποίων είναι αναλλοίωτες αφού χαρακτηρίζουν τον πυρήνα του μοντέλου της Telos

$$\begin{aligned} type_{sys} &: O_{sys} \rightarrow \{Individual, Attribute\} \\ level_{sys} &: O_{sys} \rightarrow \{1, 2, 3, 4, ...\} \end{aligned}$$

Η συνάρτηση $type_{sys}$ διαχωρίζει τις κλάσεις συστήματος σε κλάσεις που περιγράφουν οντότητες ($type_{sys} = Individual$) και σε κλάσεις που περιγράφουν γνωρίσματα ($type_{sys} = Attribute$). Με βάση τη συνάρτηση $level$ οι κλάσεις συστήματος διακρίνονται σε αυτές που περιγράφουν ατομικές οντότητες ($level$ = 1), κλάσεις ($level$ = 2), μετακλάσεις, ..., κ.ο.κ.

Με τη βοήθεια της $type_{sys}$ ορίζεται για κάθε στοιχείο του O η συνάρτηση $type$ που την ονομάζουμε ‘‘τύπο’’ :

$$type : O \rightarrow \{Individual, Attribute\} \text{ όπου } type(o) = \begin{cases} type_{sys}(o) & \text{αν o} \in O_{sys} \\ type_{sys}(R^{in}_{sys}(o)) & \text{αν } o \notin O_{sys} \end{cases}$$

Η συνάρτηση $type$ διαμερίζει τα αντικείμενα σε οντότητες ($type = Individual$) και σε γνωρίσματα ($type = Attribute$).

Με τη βοήθεια της $level_{sys}$ ορίζεται για κάθε στοιχείο του O η συνάρτηση $level$, που την ονομάζουμε ‘‘στάθμη αφαίρεσης’’ :

$$level : O \rightarrow \{0, 1, 2, 3, ...\} \text{ όπου } level(o) = \begin{cases} level_{sys}(o) & \text{αν o} \in O_{sys} \\ level_{sys}(R^{in}_{sys}(o)) - 1 & \text{αν } o \notin O_{sys} \end{cases}$$

Η συνάρτηση $level$ διαμερίζει τα αντικείμενα σε ατομικές οντότητες ή Tokens ($level$ = 0), κλάσεις ή S_Classes ($level$ = 1), μετακλάσεις ..., κ.ο.κ.

Με βάση τις συναρτήσεις $type$ και $level$ μπορούμε να ορίσουμε τα παρακάτω σύνολα για ευκολία στους ορισμούς που θα ακολουθήσουν :

$$\begin{aligned} I &= \{o \mid o \in O \wedge type(o) = Individual\} \\ A &= \{o \mid o \in O \wedge type(o) = Attribute\} \\ C &= \{o \mid o \in O \wedge level(o) > 0\} \end{aligned}$$

Στα στοιχεία του A ορίζονται οι συναρτήσεις R^{from}, και R^{to} (ορίζουν τα άκρα του γνωρίσματος) : [2]

$$\begin{aligned} R^{from} &: A \rightarrow O - PV \\ R^{to} &: A \rightarrow I \end{aligned}$$

Στο σύνολο A πρέπει να ισχύει ο ακόλουθος δομικός περιορισμός (ΔΠ).

ΔΠ$_{attr}$ **1** *Η στάθμη αφαίρεσης ενός γνωρίσματος είναι μικρότερη ή ίση από τις στάθμες αφαίρεσης των άκρων του :*

$$\forall\, a \in A \; level(a) \leq min(\; level(R^{from}(a)),\; level(R^{to}(a))\;)$$

Η σχέση R^{isA} ορίζεται ως εξής :

$$\begin{aligned} R^{isA} &= R^{isA}_{sys} \cup R^{isA}_{u} \\ \text{όπου } R^{isA}_{sys} \subset O_{sys} \times O_{sys} &\quad \text{και} \quad R^{isA}_{u} \subset (O_u \cap C) \times (O_u \cap C) \end{aligned}$$

[2] Στις πρωτογενείς τιμές δεν είναι δυνατόν να αποδοθούν γνωρίσματα και τιμή ενός γνωρίσματος δεν μπορεί να είναι γνώρισμα (A) αλλά μόνο οντότητα (I).

Τα μέλη της τα ονομάζουμε συνδέσμους εξειδίκευσης. Η σχέση R_{sys}^{isA} ορίζεται μεταξύ κλάσεων συστήματος και είναι αναλλοίωτη, ενώ η σχέση R_u^{isA} ορίζεται μεταξύ κλάσεων (C) ορισμένων από το χρήστη. Οι σύνδεσμοι εξειδίκευσης υλοποιούν την αντισυμμετρική και μεταβατική σχέση **isA** , και πρέπει να ικανοποιούν τους περιορισμούς :

ΔΠ$_{isA}$ **1** *Οι σύνδεσμοι εξειδίκευσης δεν δημιουργούν κύκλο :*

$$\text{Αν} \;\; (a,b) \in R_{tr}^{isA} \text{ τότε} \;\; (b,a) \notin R_{tr}^{isA}$$

$$\text{όπου} \;\; R_{tr}^{isA} \;=\; \{(a,b) \mid ((a,b) \in R^{isA}) \vee (\exists c : (a,c) \in R^{isA} \wedge (c,b) \in R_{tr}^{isA})\}$$

ΔΠ$_{isA}$ **2** *Οι σύνδεσμοι εξειδίκευσης συνδέουν κλάσεις του ίδιου τύπου (type) και της ίδιας στάθμης αφαίρεσης (level):*

$$\forall\, (s,t) \in R_u^{isA} \text{ ισχύει} \;\; R_{sys}^{in}(s) \;=\; R_{sys}^{in}(t)$$

ΔΠ$_{isA}$ **3** *Οι σύνδεσμοι εξειδίκευσης που συνδέουν γνωρίσματα ικανοποιούν τη συνθήκη* [3] *:*

$$\begin{aligned} \text{Αν } l, m \in A \text{ και } \;\; (l,m) \in R^{isA} : & \\ (R^{from}(l), \;\; R^{from}(m)) \;\; &\in \;\; R_{tr}^{isA} \\ (R^{to}(l), \;\; R^{to}(m)) \;\; &\in \;\; R_{tr}^{isA} \end{aligned}$$

Τα μέλη της σχέσης $R^{in} \subset O \times C$ τα ονομάζουμε συνδέσμους ταξινόμησης, η οποία ορίζεται ως εξής :

$$R^{in} \;=\; R_{sys}^{in} \cup R_u^{in}$$

$$\text{όπου} \;\; R_u^{in} \subset O_u \times O_u \qquad (\text{ η } R_{sys}^{in} \text{ ορίστηκε παραπάνω })$$

ΔΠ$_{in}$ **1** *Οι σύνδεσμοι ταξινόμησης R_u^{in} μπορούν να συνδέσουν ένα αντικείμενο με μια κλάση του ίδιου τύπου της αμέσως υψηλότερης στάθμης αφαίρεσης.*

$$\forall\, (o,c) \in R_u^{in} \text{ ισχύει} \;\; R_{sys}^{in}(R_{sys}^{in}(o)) \;=\; R_{sys}^{in}(c)$$

ΔΠ$_{in}$ **2** *Οι σύνδεσμοι ταξινόμησης R_u^{in} που συνδέουν γνωρίσματα ικανοποιούν τη συνθήκη* [4] *:*

$$\begin{aligned} \text{Αν } l, L \in A \text{ και } \;\; (l,L) \in R_u^{in} : & \\ (R^{from}(l), \;\; R^{from}(L)) \;\; &\in \;\; R_{ext}^{in} \\ (R^{to}(l), \;\; R^{to}(L)) \;\; &\in \;\; R_{ext}^{in} \end{aligned}$$

[3] Ο περιορισμός αυτός αποτελεί μια τυπική διατύπωση του περιορισμού 2 (σελ 27)
[4] Ο περιορισμός αυτός αποτελεί μια τυπική διατύπωση του περιορισμού 1

$$\text{όπου} \quad R^{in}_{ext} = \{(o,c) \mid ((o,c) \in R^{in}) \vee (\exists b : (o,b) \in R^{in} \wedge (b,c) \in R^{isA}_{tr})\}$$

Για κάθε στοιχείο ο του $O_{sys} \cup O_u$ ορίζω τα ακόλουθα :

$$\begin{aligned}
Classes(o) &= \{c \mid (o,c) \in R^{in}_u\} \\
Instances(o) &= \{i \mid (i,o) \in R^{in}_u\} \\
SuperClasses(o) &= \{sup \mid (o,sup) \in R^{isA}_u\} \\
SubClasses(o) &= \{sub \mid (sub,o) \in R^{isA}_u\} \\
AttributesFrom(o) &= \{a \mid R^{from}(a) = o\} \\
AttributesTo(o) &= \{a \mid R^{to}(a) = o\}
\end{aligned}$$

3.4 Το Σημασιολογικό Σύστημα Ευρετηριασμού.

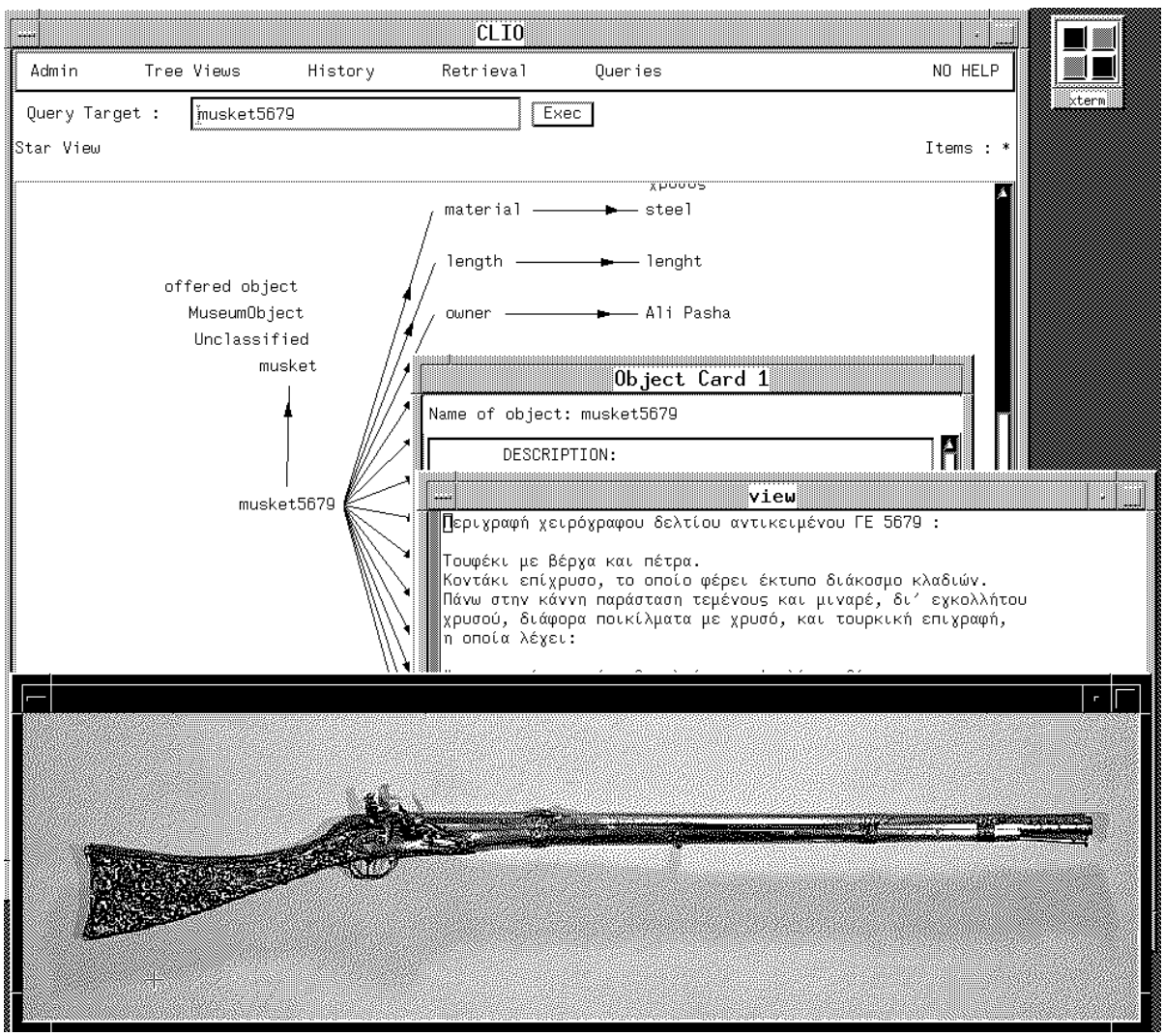
Το Σημασιολογικό Σύστημα Ευρετηριασμού (**S**emantic **I**ndex **S**ystem) είναι ένα εργαλείο για περιγραφή και τεκμηρίωση μεγάλου πληθυσμού, ιδιόμορφων, εξελισσόμενων και πολλαπλώς συνδεδεμένων δεδομένων. Για το λόγο αυτό είναι κατάλληλο για εφαρμογές σχεδιαστικής ή κατασκευαστικής φύσεως. Αναπτύχθηκε από την Ομάδα Πληροφοριακών Συστημάτων και Τεχνολογίας Λογισμικού του Ινστιτούτου Πληροφορικής του Ιδρύματος Τεχνολογίας Έρευνας.

Το SIS χρησιμοποιεί την εκδοχή της γλώσσας παράστασης γνώσης Telos , που μόλις περιγράφτηκε, η οποία χαρακτηρίζεται από τους ισχυρούς εκφραστικούς μηχανισμούς και την υψηλή απόδοση που παρέχει. Ο μηχανισμός αποθήκευσης των αντικειμένων της Telos , υποστηρίζει δοσοληψίες και ταυτόχρονη πρόσβαση από πολλούς χρήστες [24]. Το SIS συνοδεύεται από ένα μηχανισμό ερωτήσεων ([23]), ο οποίος παρέχει ένα σύνολο από ερωτηματικές εντολές, με τις οποίες είναι δυνατή η πλοήγηση στη βάση καθώς και η διατύπωση αναδρομικών ερωτήσεων με πολλαπλά κριτήρια. Οι ερωτήσεις μπορούν να κληθούν από ένα διαλογικό εργαλείο, τον answerer ή μέσα από άλλες εφαρμογές με τη βοήθεια του programmatic query interface.

Η επικοινωνία του SIS με το χρήστη υποστηρίζεται από ένα αποδοτικό και προσαρμόσιμο εργαλείο (βλέπε σχήμα 3.3) το οποίο επιτρέπει διερεύνηση σε σχήμα και δεδομένα καθώς και παρουσιάσεις πολύμορφων δεδομένων (multimedia) εξωτερικώς αποθηκευμένων. Υπάρχει η δυνατότητα ορισμού και ενεργοποίησης προκαθορισμένων ερωτήσεων,

οι οποίες μπορούν να οργανωθούν ιεραρχικά, καθώς και η δυνατότητα διατύπωσης ερωτήσεων με δελτία. Επίσης υποστηρίζονται πολλαπλές και προκαθοριζόμενες γραφικές παρουσιάσεις (όψεις) των περιεχομένων της βάσης. Η εισαγωγή δεδομένων γίνεται μαζικά από τον μεταφραστή της Telos ή διαλογικά από ένα Δελτίο Εισαγωγής Δεδομένων, του οποίου η λειτουργία μπορεί να προσαρμοσθεί ανάλογα με τις ανάγκες κάθε εφαρμογής. Η διασύνδεση και επικοινωνία του SIS με εξωτερικά εργαλεία και εφαρμογές υποστηρίζονται από ειδικούς μηχανισμούς που έχουν αναπτυχθεί.

Το SIS έχει ήδη χρησιμοποιηθεί σε ένα εργαλείο για στατική ανάλυση λογισμικού και σε ένα πληροφοριακό σύστημα μουσείων.



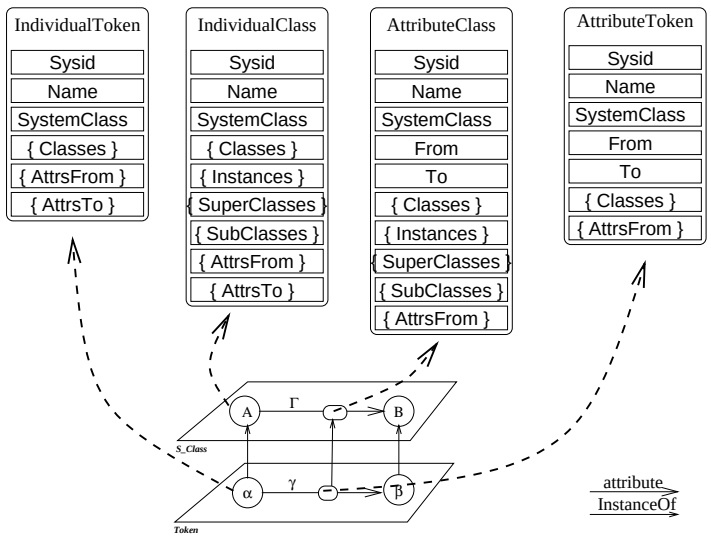
Σχήμα 3.3: Η επαφή χρήσεως του SIS

3.4.1 Χαρακτηριστικά υλοποίησης της SIS-Telos

Το σύνολο PV είναι προδηλωμένο και τα στοιχεία του δεν αποθηκεύονται, άρα ούτε ενημερώνονται. Χρησιμοποιούνται μόνο ως τιμές γνωρισμάτων.

Παρόλο που όλοι οι σύνδεσμοι της Telos είναι κατευθυνόμενοι, εντούτοις αποθηκεύονται ως αμφίδρομοι. Εξαίρεση αποτελούν οι σύνδεσμοι R^{in}_{sys} και οι R^{to} που δείχνουν σε στοιχεία του PV, των οποίων η αντίθετη κατεύθυνση δεν αποθηκεύεται.

Στο σχήμα 3.4 παρουσιάζονται σχηματικά οι δομές αποθήκευσης των αντικειμένων της SISTelos , ενώ παρακάτω περιγράφονται τα δομικά τους στοιχεία.



Σχήμα 3.4: Οι δομές αποθήκευσης των αντικειμένων της SIS-Telos .

Η SIS-Telos χρησιμοποιεί τέσσερεις διαφορετικές δομές:

- **IndividualToken** για την αποθήκευση ατομικών οντοτήτων ($I - C$).
- **IndividualClass** για την αποθήκευση κλάσεων οντοτήτων ($I \cap C$).
- **AttributeToken** για την αποθήκευση ατομικών γνωρισμάτων ($A - C$).
- **AttributeClass** για την αποθήκευση κλάσεων γνωρισμάτων ($A \cap C$).

Έστω ένα αντικείμενο ο.

- **SYSID**. Είναι το αναγνωριστικό (ο ακέραιος) το οποίο δόθηκε από το σύστημα στο o τη στιγμή που δημιουργήθηκε και το οποίο είναι μοναδικό μέσα στη βάση.

- **Name** . Πρόκειται για το λογικό όνομα του o ($\mathcal{L}(o)$) .

- **SystemClass** . Είναι ένα αναγνωριστικό το οποίο καθορίζει την κλάση συστήματος ($R^{in}_{sys}(o)$) στην οποία είναι ταξινομημένο το o, και το οποίο εκχωρήθηκε στη φάση της δημιουργίας του.

- **Classes** . Είναι ένα σύνολο το οποίο αποτελείται από τα SYSID των κλάσεων στις οποιες ανήκει το o ($Classes(o)$)

- **Instances** . Είναι ένα σύνολο το οποίο αποτελείται από τα SYSID των περιπτώσεων του o: ($Instances(o)$). Το σύνολο αυτό δεν υπάρχει αν το o είναι ένα ατομικό αντικείμενο (Token).

- **SuperClasses** . Είναι ένα σύνολο το οποίο αποτελείται από τα SYSID των υπερκλάσεων του o ($SuperClasses(o)$) . Το σύνολο αυτό δεν υπάρχει αν το o είναι ένα ατομικό αντικείμενο, αφού η σχέση γενίκευσης/εξιδείκευσης δεν ορίζεται μεταξύ ατομικών αντικειμένων.

- **SubClasses** . Είναι ένα σύνολο το οποίο αποτελείται από τα SYSID των υποκλάσεων του o ($SubClasses(o)$). Το σύνολο αυτό δεν υπάρχει αν το o είναι ένα ατομικό αντικείμενο, αφού η σχέση γενίκευσης/εξιδείκευσης δεν ορίζεται μεταξύ ατομικών αντικειμένων.

- **AttrsFrom** . Είναι ένα σύνολο το οποίο αποτελείται από τα SYSID των γνωρισμάτων τα οποία έχουν αρχή το o ($AttributesFrom(o)$).

- **AttrsTo** . Είναι ένα σύνολο το οποίο αποτελείται από τα SYSID των γνωρισμάτων τα οποία καταλήγουν στο o ($AttributesTo(o)$). Το σύνολο αυτό δεν υπάρχει αν το o είναι γνώρισμα γιατί η Telos δεν επιτρέπει ένα γνώρισμα να παίρνει τιμή ένα άλλο γνώρισμα.

- **From** . Η πληροφορία αυτή υπάρχει μόνο όταν το o είναι ένα γνώρισμα και είναι το SYSID του αντικειμένου, στο οποίο το γνώρισμα έχει αποδοθεί ($R^{from}(o)$).

- **To** . Η πληροφορία αυτή υπάρχει μόνο όταν το o είναι γνώρισμα και είναι το SYSID του αντικειμένου που αποτελεί τιμή του γνωρίσματος ($R^{to}(o)$).

Κεφάλαιο 4

Εξέλιξη Γνώσης στην SIS-Telos

Η εξέλιξη της γνώσης σε βάσεις δεδομένων της SIS-Telos μπορεί να γίνει είτε **μαζικά** (batch) μέσω του μεταφραστή (parser) με τη χρήση της γλώσσας εισαγωγής δεδομένων TELL/RETELL που παρουσιάστηκε στο κεφάλαιο 3, ή **διαλογικά** μέσω του διαλογικού Δελτίου Εισαγωγής Δεδομένων (ΔΕΔ) για το οποίο θα αναφέρουμε περισσότερα στο επόμενο κεφάλαιο [1]. Και στις δύο περιπτώσεις οι ενημερώσεις πραγματοποιούνται κατά δοσοληψίες (transactions).

Κάθε εντολή ενημέρωσης που μπορούμε να διατυπώσουμε με την γλώσσα TELL/RETELL ή με το ΔΕΔ μπορεί να αναλυθεί σε ένα σύνολο από Στοιχειώδεις Εντολές ενημέρωσης (ΣΕΕ). Το σύνολο των εντολών αυτών ορίζει την **Στοιχειώδη Γλώσσα Ενημέρωσης** της SIS-Telos , η οποία παρουσιάζεται στην επόμενη ενότητα.

Τέλος στην ενότητα 4.2 αναφέρονται ορισμένα γενικά χαρακτηριστικά της εξέλιξης γνώσης των βάσεων σε SIS-Telos .

4.1 Η Στοιχειώδης Γλώσσα Ενημερώσης

Στην ενότητα αυτή παρουσιάζεται η Στοιχειώδης Γλώσσα Ενημερώσης (ΣΓΕ) της γλώσσας SIS-Telos , με την οποία μπορούμε να διατυπώσουμε Στοιχειώδεις Εντολές Ενημέρωσης (ΣΕΕ). Η ΣΓΕ που παρουσιάζεται είναι **πλήρης**, αφού μπορεί να περιγράψει κάθε ενημέρωση, που μπορεί να γίνει σε μια βάση, και **λιτή**, αφού καμιά από τις εντολές της

[1] Μελλοντικά θα είναι επίσης εφικτή μέσω του Application Program Interface (API) της SIS-Telos που πρόκειται να αναπτυχθεί.

δεν θα μπορούσε να παραληφθεί χωρίς να χαθεί η πληρότητα της [2].

Η μελέτη της ΣΓΕ ακολουθεί την σειρά :

1. *Λεξικογραφική ανάλυση* :
   Παρουσιάζονται τα δομικά στοιχεία της γλώσσας από τα οποία συντίθενται εντολές.

2. *Συντακτική ανάλυση* :
   Παρουσιάζεται ο τρόπος σύνθεσης (συγκρότησης) εντολών μέσω των δομικών στοιχείων της γλώσσας.

3. *Σημασιολογική ανάλυση* :
   Παρουσιάζονται οι έλεγχοι που διασφαλίζουν ότι η εκτέλεση μιας συντακτικά ορθής εντολής αφήνει τη βάση σε μια συνεπή κατάσταση. Μια τέτοια εντολή τη χαρακτηρίζουμε ως σημασιολογικά ορθή. Η SIS-Telos επιτρέπει την εκτέλεση μιας ΣΕΕ μόνο αν αυτή είναι σημασιολογικά ορθή [3].

   Μια δοσοληψία (transaction) ή μια σύνθετη εντολή ενημέρωσης (πχ μια εντολή τύπου Replace) συντίθενται από ένα σύνολο από ΣΕΕ, οι οποίες εκτελούνται είτε όλες ή καμία τους (ατομικότητα). Σε αυτές τις περιπτώσεις οι σημασιολογικοί έλεγχοι, που αφορούν το σύνολο των ΣΕΕ, γίνονται όλοι μαζί στο τέλος.

4. *Ανάλυση εκτέλεσης εντολών* :
   Παρουσιάζονται οι συνέπειες της εκτέλεσης μιας σημασιολογικά ορθής εντολής στη βάση.

4.1.1 Λεξικογραφική Ανάλυση

Τα δομικά στοιχεία της γλώσσας (tokens) είναι τα εξής :

- *Λέξεις Κλειδιά* : Πρόκειται για το σύνολο :
  { **CreateIndividual** , **DeleteIndividual** , **CreateAttribute** , **DeleteAttribute** , **AddInstance** , **DeleteInstance** , **AddSubClass** , **DeleteSubClass** , **Rename** }
- *Λέξεις* : Συμβολοσειρές με μήκος μικρότερο από 256
- *Λογικά Ονόματα* : Συμβολοσειρές με μήκος μικρότερο από 96
- *Ακέραιοι* : Το σύνολο ακεραίων της γλώσσας C.
- *Πραγματικοί* : Το σύνολο πραγματικών της γλώσσας C.

[2] Η απόδειξη της πληρότητας και της λιτότητας είναι τετριμμένη.

[3] Ο έλεγχος της σημασιολογικής ορθότητας πραγματοποιείται από τον *Semantic Checker*, ο οποίος είναι μία λειτουργική μονάδα του συστήματος εκτέλεσης (run-time system) της SIS-Telos .

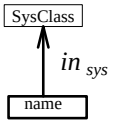
4.1.2 Συντακτική - Σημασιολογική - Εκτελεστική Ανάλυση

Ακολουθεί μια σύντομη περιγραφή του συντακτικού, της σημασιολογίας και της εκτέλεσης των ΣΕΕ.

1. **CreateIndividual** *SysClass*, *name* — SysClass : Ακέραιος, name : Λογικό Ονομα

• *Σύντομη Περιγραφή:*

Δημιουργία μιας νέας οντότητας (*Individual*) με το λογικό όνομα *name*, το οποίο γίνεται στιγμιότυπο της κλάσης συστήματος *SysClass* (δες σχήμα 4.1).



Σχήμα 4.1: Δημιουργία οντότητας

Ταυτόχρονα με τη δημιουργία της οντότητας *name*, δημιουργείται ο σύνδεσμος ταξινόμησης στην κλάση συστήματος *Sysclass*

• *Προϋποθέσεις Σημασιολογικής Ορθότητας :*

$SysClass \in O_{sys} \cap I$

Έλεγχος συνέπειας στην ονοματοδοσία

• *Συνέπειες Εκτέλεσης :*

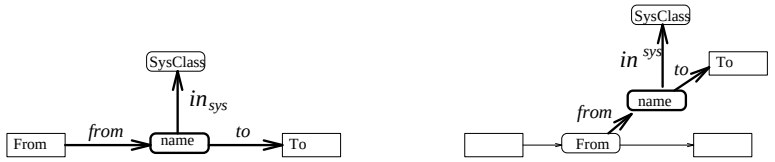
$$\begin{aligned} O'_u &= O_u \cup \{o\} \text{ όπου } o \notin O_{sys} \cup O_u \\ R^{in}_{sys}(o) &= SysClass \\ L' &= L \cup \{name\} \\ \mathcal{L}(o) &= name \end{aligned}$$

2. **CreateAttribute** *from* , *name*, *to* , *SysClass* — *from* , *SysClass* : Ακέραιοι; name : Λογικό Ονομα, *to* : Ακέραιος | Πραγματικός | Λέξη

• *Σύντομη Περιγραφή:*

Δημιουργία ενός γνωρίσματος (*Attribute*) με αρχή το αντικείμενο *from* , τέλος την οντότητα (*Individual*) *to* , λογικό όνομα το *name* και με κλάση συστήματος την *SysClass* (δες σχήμα 4.2).

• *Προϋποθέσεις Σημασιολογικής Ορθότητας :*



Σχήμα 4.2: Δημιουργία γνωρίσματος

Ταυτόχρονα με τη δημιουργία του γνωρίσματος με λογικό όνομα *name*, δημιουργείται ο σύνδεσμος ταξινόμησης στην κλάση συστήματος *Sysclass* και οι σύνδεσμοι που το συνδέουν με τα άκρα του.

$$\mathit{from} \in O - PV$$

$$\mathit{to} \in I$$

$$\mathit{SysClass} \in O_{sys} \cap A$$

$$\mathbf{\Delta\Pi}_{attr}1 : level(a) \leq min(\ level(\mathit{from}\ ),\ level(\mathit{to}\ )\ )$$

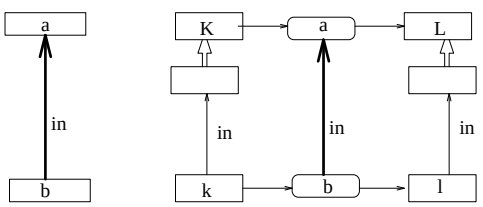
- *Συνέπειες Εκτέλεσης :*

$$\begin{aligned} O'_u &= O_u \cup \{a\} \text{ όπου } a \notin O_{sys} \cup O_u \\ R^{in}_{sys}(a) &= \mathit{SysClass} \\ R^{from}(a) &= \mathit{from} \\ R^{to}(a) &= \mathit{to} \\ L' &= L \cup \{name\} \\ \mathcal{L}(a) &= name \end{aligned}$$

3. | **AddInstance** a, b | a,b : Ακέραιοι |

- *Σύντομη Περιγραφή:*

Προσθήκη του αντικειμένου b στις περιπτώσεις του αντικειμένου a (δες σχήμα 4.3).



Σχήμα 4.3: Δημιουργία συνδέσμου ταξινόμησης

- *Προϋποθέσεις Σημασιολογικής Ορθότητας :*

$a, b \in O_u$

$\mathbf{\Delta\Pi}_{in}1 : R^{in}_{sys}(R^{in}_{sys}(b)) \ = \ R^{in}_{sys}(a)$

$\mathbf{\Delta\Pi}_{in}2$ αν $a, b \in A$.

(στο σχήμα 4.3 : $(k, K) \in R^{in}_{ext}$ και $(l, L) \in R^{in}_{ext}$)

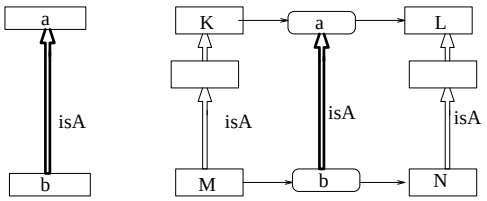
• *Συνέπειες Εκτέλεσης :*

$R^{in'}_u = R^{in}_u \cup \{(b, a)\}$

4. **AddSubClass** a, b — a,b : Ακέραιοι

• *Σύντομη Περιγραφή:*

Προσθήκη του αντικειμένου b στις υποκλάσεις του αντικειμένου a (δες σχήμα 4.4).



Σχήμα 4.4: Δημιουργία συνδέσμου εξειδίκευσης

• *Προϋποθέσεις Σημασιολογικής Ορθότητας :*

$a, b \in O_u \cap C$

$\mathbf{\Delta\Pi}_{isA}2 : R^{in}_{sys}(a) \ = \ R^{in}_{sys}(b)$

$\mathbf{\Delta\Pi}_{isA}1 : (a, b) \notin R^{isA}_{tr}$

$\mathbf{\Delta\Pi}_{isA}3$ αν $a, b \in A$.

(στο σχήμα 4.4 : $(M, K) \in R^{isA}_{tr}$ και $(N, L) \in R^{isA}_{tr}$)

• *Συνέπειες Εκτέλεσης :*

$R^{isA'}_u := R^{isA}_u \cup \{(b, a)\}$

5. **DeleteIndividual** o — o : Ακέραιος

• *Σύντομη Περιγραφή:*

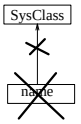
Η οντότητα o (*Individual*) διαγράφεται από την βάση. Η διαγραφή της περιλαμβάνει την διαγραφή του λογικού της ονόματος και του συνδέσμου ταξινόμησης στην κλάση συστήματος (δες σχήμα 4.5).

• *Προϋποθέσεις Σημασιολογικής Ορθότητας :*

$o \in O_u \cap I$

$| Classes(o) | \ + \ | Instances(o) | \ + \ | SuperClasses(o) | \ +$

$| SubClasses(o) | \ + \ | AttributesFrom(o) | \ + \ | AttributesTo(o) | \ = \ 0$



Σχήμα 4.5: Διαγραφή οντότητας

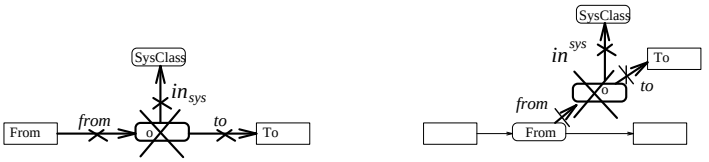
• *Συνέπειες Εκτέλεσης :*

$$\begin{aligned} O'_u &= O_u - \{o\} \\ L' &= L - \{\mathcal{L}(o)\} \end{aligned}$$

6. | **DeleteAttribute** a | a : Ακέραιος |

• *Σύντομη Περιγραφή:*

Διαγραφή του γνωρίσματος a. Η διαγραφή του περιλαμβάνει την διαγραφή του λογικού του ονόματος, του συνδέσμου ταξινόμησης στην κλάση συστήματος καθώς και τους συνδέσμους που το συνδέουν με τα άκρα του (δες σχήμα 4.6).



Σχήμα 4.6: Διαγραφή γνωρίσματος

• *Προϋποθέσεις Σημασιολογικής Ορθότητας :*

$$\begin{aligned} & a \in O_u \cap A \\ & |\, Classes(a) \,| \; + \; |\, Instances(a) \,| \; + \; |\, SuperClasses(a) \,| \; + \\ & |\, SubClasses(a) \,| \; + \; |\, AttributesFrom(a) \,| \; + \; |\, AttributesTo(a) \,| \; = \; 0 \end{aligned}$$

• *Συνέπειες Εκτέλεσης :*

$$\begin{aligned} O'_u &= O_u - \{a\} \\ L' &= L - \{\mathcal{L}(a)\} \end{aligned}$$

7. | **Rename** o , *newname* | o : Ακέραιος, *newname* : Λογικό Όνομα |

• *Σύντομη Περιγραφή:*

Το λογικό όνομα του αντικειμένου o αλλάζει σε *newname* .

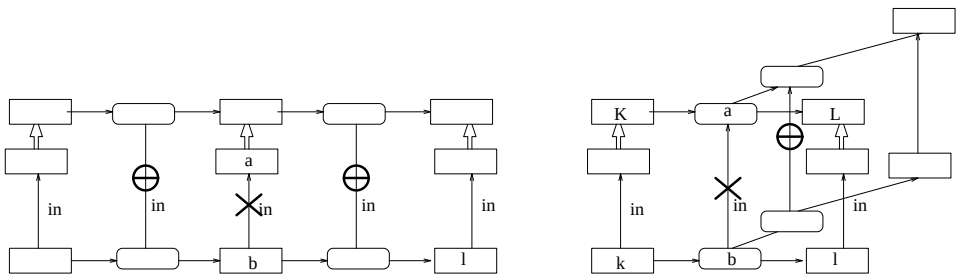
• *Συνέπειες Εκτέλεσης :*

$$L' = L - \{\mathcal{L}(o)\} \cup \{newname\}$$
$$\mathcal{L}(o) = newname$$

8. **DeleteInstance** a, b — a, b : Ακέραιοι

• *Σύντομη Περιγραφή:*

Διαγραφή του αντικειμένου b από τις περιπτώσεις του αντικειμένου a.



Σχήμα 4.7: Διαγραφή συνδέσμου ταξινόμησης

Στα παραδείγματα που παρουσιάζονται η διαγραφή ενός συνδέσμου ταξινόμησης αποτρέπεται γιατί θα είχαμε παραβίαση του $\mathbf{\Delta\Pi}_{in}2$ στα γνωρίσματα των οποίων οι σύνδεσμοι ταξινόμησης είναι σημειωμένοι με κύκλο.

• *Προϋποθέσεις Σημασιολογικής Ορθότητας :*

$a, b \in O_u$

$(b, a) \in R_u^{in}$

$\mathbf{\Delta\Pi}_{in}2$: Αρκεί να επαληθεύσουμε την ικανοποίηση του σε κάθε γνώρισμα που αρχίζει ή καταλήγει στην οντότητα b (παράδειγμα παραβίασης φαίνεται στο σχήμα 4.7).

• *Συνέπειες Εκτέλεσης :*

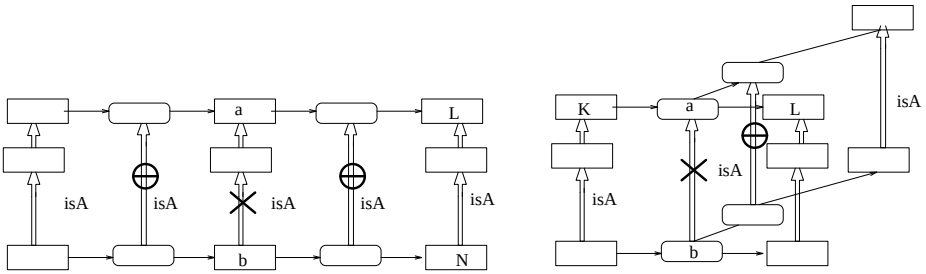
$R_u^{in'} := R_u^{in} - \{(b, a)\}$

9. **DeleteSubClass** a, b — a, b : Ακέραιοι

• *Σύντομη Περιγραφή:*

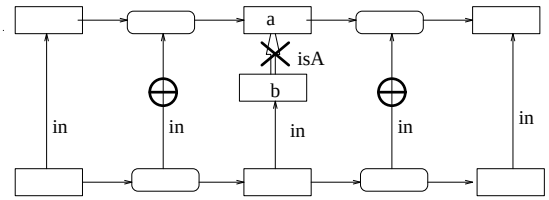
Διαγραφή του αντικειμένου b από τις υποκλάσεις του αντικειμένου a.

• *Προϋποθέσεις Σημασιολογικής Ορθότητας :*



Σχήμα 4.8: Διαγραφή συνδέσμου εξειδίκευσης

Στα παραδείγματα που παρουσιάζονται η διαγραφή ενός συνδέσμου εξειδίκευσης αποτρέπεται γιατί θα είχαμε παραβίαση του $\mathbf{\Delta\Pi}_{isA}3$ στα γνωρίσματα των οποίων οι σύνδεσμοι εξειδίκευσης είναι σημειωμένοι με κύκλο.



Σχήμα 4.9: Διαγραφή συνδέσμου εξειδίκευσης

Στα παραδείγματα που παρουσιάζονται η διαγραφή ενός συνδέσμου εξειδίκευσης αποτρέπεται γιατί θα είχαμε παραβίαση του $\mathbf{\Delta\Pi}_{in}2$ στα γνωρίσματα των οποίων οι σύνδεσμοι ταξινόμησης είναι σημειωμένοι με κύκλο.

$a, b \in O_u$

$(b, a) \in R_u^{isA}$

$\mathbf{\Delta\Pi}_{isA}3$: Αρκεί να επαληθεύσουμε την ικανοποίηση του σε κάθε γνώρισμα του αρχίζει ή καταλήγει στην οντότητα b (παράδειγμα παραβίασης φαίνεται στο σχήμα 4.8).

$\mathbf{\Delta\Pi}_{in}2$: Η μη-παραβίαση του περιορισμού αυτού αρκεί να ελεγχθεί σε κάθε γνώρισμα που αρχίζει ή καταλήγει σε μία περίπτωση της κλάσεως b ή των υποκλάσεων της (παράδειγμα παραβίασης φαίνεται στο σχήμα 4.9).

- *Συνέπειες Εκτέλεσης :*

  $R_u^{isA'} := R_u^{isA} - \{(b, a)\}$

Παρατηρούμε ότι ορισμένες ΣΕΕ πραγματοποποιούν αλλαγές, σε φυσικό επίπεδο, σε ένα μόνο αντικείμενο (**DeleteIndividual** , **Rename**), ενώ άλλες σε περισσότερα από ένα αντικείμενα.

4.2 Χαρακτηριστικά της Εξέλιξης Γνώσης στην SIS-Telos

Ορισμένα χαρακτηριστικά του τρόπου με τον οποίο εξελίσσεται μια βάση σε SIS-Telos είναι τα ακόλουθα :

- Η εισαγωγή ενός αντικειμένου στην βάση, γίνεται με τη δημιουργία ενός αντικειμένου της SIS-Telos και εν συνεχεία με τη δημιουργία ενός αριθμού συνδέσμων (ταξινόμησης, εξειδίκευσης ή γνωρισμάτων) προς άλλα αντικείμενα της βάσης. Συχνά απαιτείται η δημιουργία **επιπλέον αντικειμένων** προκειμένου να περιγραφεί ένα σύνθετο αντικειμένο (composite object). Η εισαγωγή νέας γνώσης και η εξειδίκευση ή εμπλουτισμός της υπάρχουσας αποτελούν την κυριότερη λειτουργία εξέλιξης γνώσης σε βάσεις της SIS-Telos .

- Η χρήση της SIS-Telos ενδείκνυται σε εφαρμογές σχεδιαστικής φύσεως, στις οποίες είναι αναμενόμενες και συχνές οι αλλαγές του σχήματος των δεδομένων. Η SIS-Telos επιτρέπει αλλαγές στο σχήμα σε χρόνο εκτέλεσης.

- Βάσεις δεδομένων σε SIS-Telos συχνά χρησιμοποιούνται σε συνεργασία με βάσεις δεδομένων σχεσιακών συστημάτων στις οποίες αποθηκεύονται τα περισσότερο ομοιόμορφα και λιγότερο διασυνδεδεμένα δεδομένα. Κατά συνέπεια είναι συχνές οι μαζικές εισαγωγές/ανταλλαγές δεδομένων.

Κεφάλαιο 5

Ένα Μοντέλο Διεργασιών Ενημέρωσης

Για τον ορισμό ΟΕ στην Telos αρχικά χρησιμοποιήσαμε ένα **Μοντέλο Διεργασιών Ενημέρωσης (ΜΔΕ)**. Το μοντέλο αυτό έχει παρασταθεί στη γλώσσα SIS-Telos και οι πραγματοποιήσεις του παριστάνουν τις όψεις ενημέρωσης συγκεκριμένων εφαρμογών. Οι ΟΕ αυτές ερμηνεύονται από το διαλογικό **Δελτίο Εισαγωγής Δεδομένων (ΔΕΔ)** [1], το οποίο, αφού διαβάσει τον ορισμό μιας ΟΕ από τη βάση, προσαρμόζει ανάλογα τη λειτουργικότητα του. Επίσης η δήλωση μιας ΟΕ ερμηνεύεται από το εργαλείο *Export* , το οποίο περιγράφει τα περιεχόμενα της σε ένα αρχείο εντολών της SIS-Telos .

Ο σχεδιασμός και η ανάπτυξη του ΜΔΕ βασίστηκε στις ανάγκες της εφαρμογής **SIB Class Management System** ([14],[13]), η οποία είναι μια εφαρμογή του **SIS** για την τεκμηρίωση και ανάλυση στατικών ιδιοτήτων μονάδων λογισμικού.

Το ΜΔΕ περιγράφεται στην επόμενη ενότητα, ενώ στις ενότητες 5.2 και 5.3 περιγράφεται η χρήση του για την δήλωση των όψεων ενημέρωσης των εφαρμογών της **SIB** και του πληροφοριακού συστήματος μουσείων *ΚΛΕΙΩ* [26] αντίστοιχα. Τέλος, στην ενότητα 5.4 γίνεται σχολιασμός και αξιολόγηση του ΜΔΕ.

5.1 Γενική Περιγραφή του Μοντέλου Διεργασιών

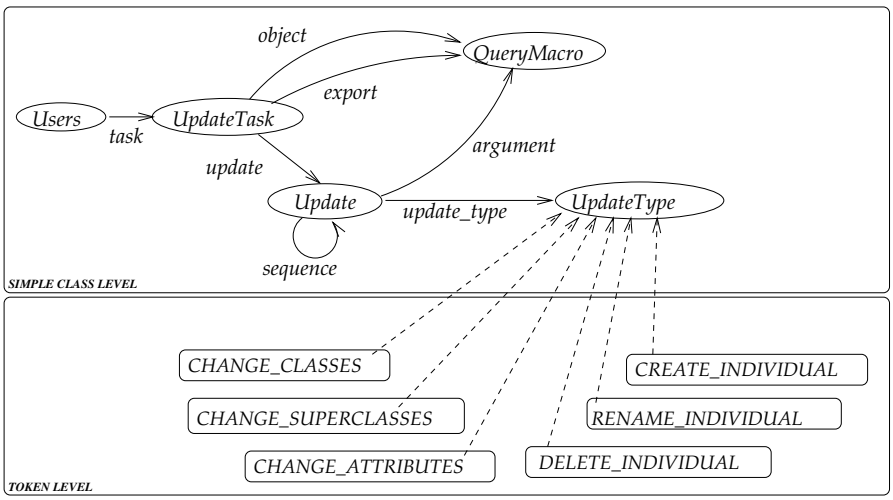
Μια γραφική απεικόνιση του ΜΔΕ παρουσιάζεται στο σχήμα 5.1, ενώ ο ορισμός στη γλώσσα SIS-Telos επισυνάπτεται στο παράρτημα Β.

[1] Η υλοποίηση του ΔΕΔ έγινε από τον Δημήτρη Δασκαλάκη.

Σε μια βάση δεδομένων μπορεί να υπάρχει ένα σύνολο από χρήστες (*Users*) οι οποίοι μπορούν να σχετίζονται με ένα σύνολο *Όψεων ή Εργασιών Ενημέρωσης (Update Tasks)*. Κάθε *όψη ενημέρωσης* περιλαμβάνει ένα *σύνολο οντοτήτων (individuals)* και ένα σύνολο *ενημερώσεων (Updates)*. Το *σύνολο οντοτήτων* περιέχει τις οντότητες στις οποίες μπορούν να εφαρμοσθούν οι *ενημερώσεις*. Κάθε *ενημέρωση* έχει ένα *τύπο (UpdateType)* και ένα *σύνολο ορισμάτων (arguments)*, το οποίο περιέχει τα αντικείμενα που μπορεί να χρησιμοποιηθούν σαν ορίσματα των ενημερώσεων που αφορούν συνδέσμους ταξινόμησης, εξειδίκευσης ή γνωρίσματα.

Μια *ενημέρωση* μπορεί να συνδέεται με μια άλλη μέσω ενός συνδέσμου *ακολουθίας (sequence)*. Η σύνδεση αυτή ερμηνεύεται (από το ΔΕΔ) ως μια ακολουθία εκτέλεσης *ενημερώσεων* που δρούν σε μία, κάθε φορά, οντότητα (του *συνόλου οντοτήτων*), η οποία μπορεί να εκτελεστεί μόνο εξ ολοκλήρου (το ΔΕΔ δεν επιτρέπει την εκτέλεση μερικών βημάτων).

Η περιγραφή των συνόλων γίνεται με *QueryMacro* τα οποία περιγράφονται στην επόμενη ενότητα, ενώ η σχέση *εξαγωγής (export)*, περιγράφεται στην ενότητα 5.1.4.



Σχήμα 5.1: Σχηματική Απεικόνιση του Μοντέλου Διεργασιών Ενημέρωσης

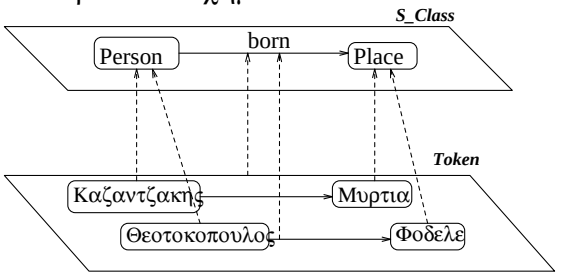
5.1.1 Δήλωση Συνόλων

Τα *σύνολα οντοτήτων* και *σύνολα ορισμάτων* δηλώνονται με τη βοήθεια της κλάσεως *QueryMacro*. Ένα *QueryMacro* είναι μια ακολουθία ερωτηματικών εντολών του *query interface (qi)* της TELOS. Οι ερωτηματικές εντολές του *QueryMacro* αποθηκεύονται σε

μορφή συμβολοσειράς και ένας κατάλογος με τις βασικότερες υπάρχει στο παράρτημα C.

Το σύνολο οντοτήτων που ορίζεται από ένα *QueryMacro*, είναι το σύνολο οντοτήτων που περιέχεται στην απάντηση της εκτέλεσης των ερωτηματικών του εντολών. Κατά συνέπεια τα σύνολα οντοτήτων που περιέχονται στον ορισμό μιας ΟΕ είναι *δυναμικά* και εξαρτώνται από τα περιεχόμενα της βάσης.

Παρακάτω παρουσιάζονται δύο παραδείγματα απλών *QueryMacro* που αφορούν το μοντέλο που φαίνεται στο παρακάτω σχήμα :



TELL Individual (All Persons) **in Token** , QueryMacro with
code
(1) : ”scn Person”;
(2) : ”gai 0”
end

TELL Individual BornIn **in Token** , QueryMacro with
code
(1) : ”scn ∗ Place”;
(2) : ”gfnc Person born 0”
end

Το *QueryMacro* **All Persons** ορίζει το σύνολο των περιπτώσεων της κλάσεως *Person*. Περιέχει δύο ερωτηματικές εντολές : την εντολή ”scn Person” η οποία θέτει ως εστιακό αντικείμενο την κλάση *Person* και την ερωτηματική εντολή ”gai 0” (**g**et **a**ll **i**nstances) η οποία επιστρέφει όλες τις περιπτώσεις του εστιακού αντικειμένου (υποδηλώνεται από το 0) δηλαδή της κλάσεως *Person*. Στο συγκεκριμένο παράδειγμα, η εκτέλεσή του επιστρέφει το σύνολο $\{ Καζαντζάκης, Θεοτοκόπουλος \}$.

Ένα *QueryMacro* μπορεί να παίρνει μία παράμετρο, της οποίας ο τύπος μπορεί να δηλώνεται. Για παράδειγμα το *QueryMacro* **BornIn** δέχεται μια παράμετρο τύπου *Place* την οποία και θέτει ως εστιακό αντικείμενο. Εν συνεχεία η ερωτηματική εντολή ”gfnc Person born 0” (**g**et **f**rom **n**ode by **c**ategory) επιστρέφει τις οντότητες από τις οποίες εκκινούν γνωρίσματα που ανήκουν στην κλάση γνωρισμάτων *born* (που έχει αποδοθεί στην κλάση *Person*) και καταλήγουν στην εστιακή οντότητα. Στο συγκεκριμένο παράδειγμα αν εκτελέσουμε το *QueryMacro* με την παράμετρο *Φόδελε* μας επιστρέφει το σύνολο $\{Θεοτοκόπουλος\}$.

5.1.2 Ερμηνεία Δηλώσεων

Στην ενότητα αυτή θα δούμε τον τρόπο, με τον οποίο ερμηνεύονται οι ενημερώσεις που

δηλώνονται μέσω μιας ΟΕ με το ΜΔΕ.

Κάθε ΟΕ ορίζει ένα σύνολο ενημερώσεων, κάθε μια από τις οποίες έχει υποχρεωτικά ένα *τύπο ενημέρωσης* που ανήκει στο σύνολο : { CREATE_INDIVIDUAL , DELETE_INDIVIDUAL , RENAME_INDIVIDUAL , CHANGE_CLASSES , CHANGE_SUPERCLASSES , CHANGE_ATTRIBUTES }.

Αν ο τύπος ανήκει στο σύνολο: { CREATE_INDIVIDUAL , CHANGE_CLASSES , CHANGE_SUPERCLASSES , CHANGE_ATTRIBUTES }, τότε η ενημέρωση έχει και ένα *σύνολο ορισμάτων*.

Η ερμηνεία των δηλώσεων σε ΣΕΕ ακολουθεί παρακάτω. Το όνομα *ObjectSet* συμβολίζει το *σύνολο οντοτήτων*. Οι συμβολισμοί $name, tname$, υποδηλώνουν ένα οποιοδήποτε λογικό όνομα, η *SysClass* υποδηλώνει μια κλάση συστήματος, τα To, $ToClass$ οντότητες, ενώ το σύμβολο $\circ$ δηλώνει ακολουθίες από ΣΕΕ οι οποίες θεωρούνται ως ατομικές ενημερώσεις (μη αναλυόμενες).

- Τύπος: DELETE_INDIVIDUAL Ορίσματα : —

  DeleteIndividual o, όπου $o \in$ *ObjectSet*

- Τύπος: RENAME_INDIVIDUAL Ορίσματα : —

  Rename o, όπου $o \in$ *ObjectSet*

- Τύπος: CREATE_INDIVIDUAL Ορίσματα : $Classes$

  $o :=$ **CreateIndividual** $name$, *SysClass* $\circ$ **AddInstance** o, c

  $\forall\, c \in Classes$

- Τύπος: CHANGE_CLASSES Ορίσματα : $Classes$

  AddInstance o, c όπου $o \in$ *ObjectSet* $, c \in Classes$

  DeleteInstance o, c όπου $o \in$ *ObjectSet* $, c \in Classes$

- Τύπος: CHANGE_SUPERCLASSES Ορίσματα : $Classes$

  AddSubClass $o, c,$ όπου $o \in$ *ObjectSet* $, c \in Classes$

  DeleteSubClass $o, c,$ όπου $o \in$ *ObjectSet* $, c \in Classes$

- Τύπος: CHANGE_ATTRIBUTES Ορίσματα : $Categories$

  $$\begin{cases} a := \textbf{CreateAttribute}\ o, name, To, \textit{SysClass} \circ \textbf{AddInstance}\ a, c \\ \qquad\qquad \text{ή} \\ To := \textbf{CreateIndividual}\ tname, \textit{SysClass} \circ \textbf{AddInstance}\ To, ToClass \circ \\ a := \textbf{CreateAttribute}\ o, name, To, \textit{SysClass} \circ \textbf{AddInstance}\ a, c \end{cases}$$

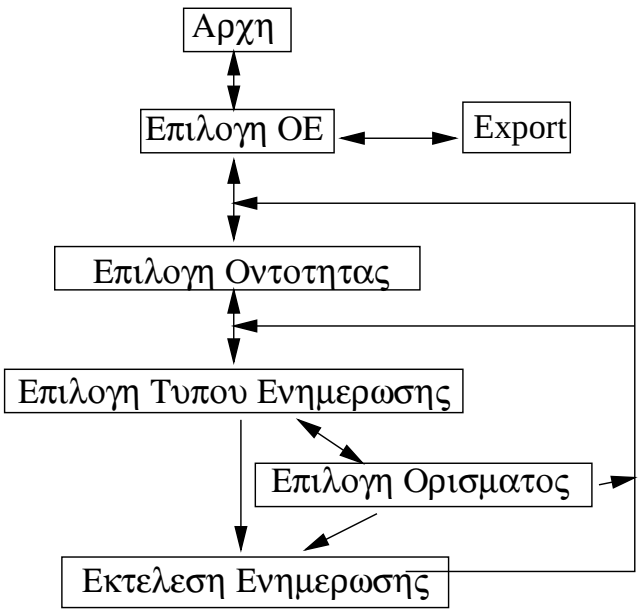
  όπου $o \in$ *ObjectSet* $, c \in Categories, R^{to}(c) = ToClass$

  DeleteInstance $a, c, \circ$ **DeleteAttribute** a

  όπου $R^{from}(a) \in$ *ObjectSet* $, c \in Categories$

5.1.3 Εκτέλεση των Ενημερώσεων

Μια ΟΕ ορίζει ένα σύνολο ΣΕΕ. Το ΔΕΔ που αναπτύχθηκε διαβάζει την περιγραφή μιας ΟΕ και παρέχει τη δυνατότητα στο χρήστη να πραγματοποιήσει ένα σύνολο από απλές διαδικασίες ενημέρωσης (*update processes*). Τα βασικά βήματα των διαδικασιών αυτών ή αλλιώς η ροή ελέγχου του ΔΕΔ παρουσιάζονται στο σχήμα 5.2 και περιγράφονται παρακάτω :



Σχήμα 5.2: Η ροή ελέγχου των Δελτίου Εισαγωγής Δεδομένων

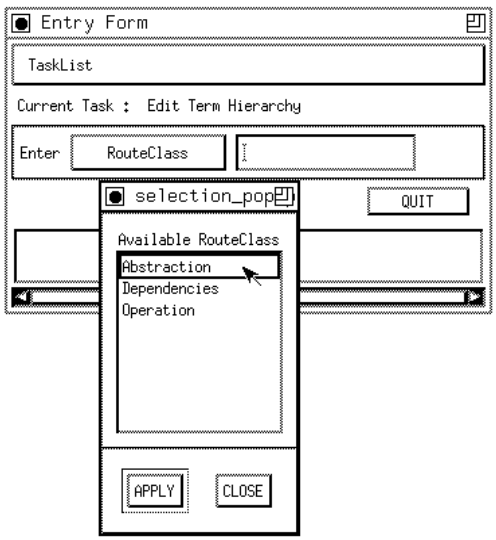
1. **Επιλογή ΟΕ**

   Ο χρήστης επιλέγει την επιθυμητή ΟΕ (τρέχουσα) από το σύνολο των ΟΕ με τις οποίες σχετίζεται (αυτές καθορίζονται από τον τύπο του χρήστη, ο οποίος ορίζεται κατά την εκκίνηση του ΔΕΔ) Αν το σύνολο οντοτήτων της ΟΕ που επιλέχθηκε είναι παραμετρικό, ο χρήστης επιλέγει την επιθυμητή παράμετρο από την λίστα των διαθέσιμων παραμέτρων (δες σχήμα 5.3).

   .

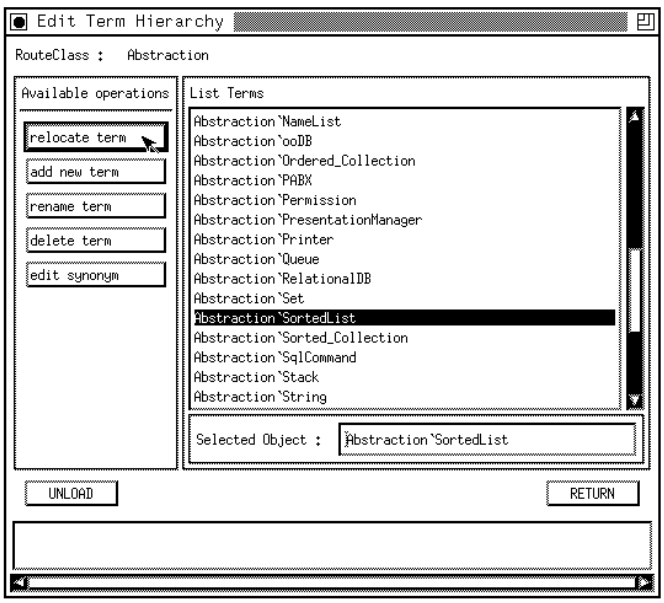
2. **Επιλογή Οντότητας**

   Ο χρήστης επιλέγει την οντότητα που θέλει να ενημερώσει (τρέχουσα), από τη λίστα που περιέχει όλες τις οντότητες του *ObjectSet* της τρέχουσας ΟΕ.



Σχήμα 5.3: Επιλογή όψης ενημέρωσης

Στην προκειμένη περίπτωση η ΟΕ *Edit Term Hierarchy* είναι παραμετρική για το λόγο αυτό ο χρήστης επιλέγει την επιθυμητή παράμετρο (περίπτωση της RouteClass)



Σχήμα 5.4: Επιλογή της οντότητας *Abstrasction'SortedList* από το σύνολο οντοτήτων

3. **Επιλογή Τύπου Ενημέρωσης**

   Ο χρήστης επιλέγει τον τύπο της ενημέρωσης που θέλει να πραγματοποιήσει στην τρέχουσα οντότητα, από ένα σύνολο κουμπιών τα οποία αντιστοιχούν στους τύπους ενημέρωσης που είναι ορισμένοι στην τρέχουσα ΟΕ (δες σχήμα 5.4). [2]

4. **Επιλογή Ορισμάτων**

   Το βήμα αυτό δεν χρειάζεται αν ο τύπος είναι ο DELETE_INDIVIDUAL .

   Αν ο τύπος είναι ο CREATE_INDIVIDUAL ή ο RENAME_INDIVIDUAL τότε ο χρήστης πρέπει να δηλώσει το επιθυμητό λογικό όνομα.

   Αν ο τύπος είναι ο CHANGE_CLASSES (ή CHANGE_SUPERCLASSES) ο χρήστης μπορεί να διαγράψει μια κλάση ή υπερκλάση (δες σχήμα 5.5) από τη λίστα με τις κλάσεις (ή υπερκλάσεις) της τρέχουσας οντότητας που ανήκουν στο σύνολο ορισμάτων, $Classes$. Επίσης μπορεί να δηλώσει μια νέα κλάση (ή υπερκλάση) διαλέγοντας την από τη λίστα με το σύνολο ορισμάτων $Classes$.

   Αν ο τύπος είναι ο CHANGE_ATTRIBUTES ο χρήστης μπορεί να διαγράψει ένα γνώρισμα από τη λίστα με τα γνωρίσματα της τρέχουσας οντότητας, που η κατηγορία τους ανήκει στο σύνολο ορισμάτων $Categories$. Επίσης μπορεί να προσθέσει ένα γνώρισμα κάτω από μια κατηγορία (τρέχουσα κατηγορία) που ανήκει στο σύνολο $Categories$, διαλέγοντας αρχικά την επιθυμητή κλάση της τιμής (τρέχουσα κλάση τιμής) του γνωρίσματος, από μια λίστα που περιέχει όλες τις υποκλάσεις της κλάσης, στην οποία η τρέχουσα κατηγορία καταλήγει. Στη συνέχεια εισάγει την τιμή του νέου γνωρίσματος, το οποίο μπορεί να είναι ένα νέο αντικείμενο (το οποίο θα δημιουργηθεί), ή ένα από τη λίστα που περιέχει τις περιπτώσεις της τρέχουσας κλάσης τιμής, που επιλέχθηκε προηγουμένως.

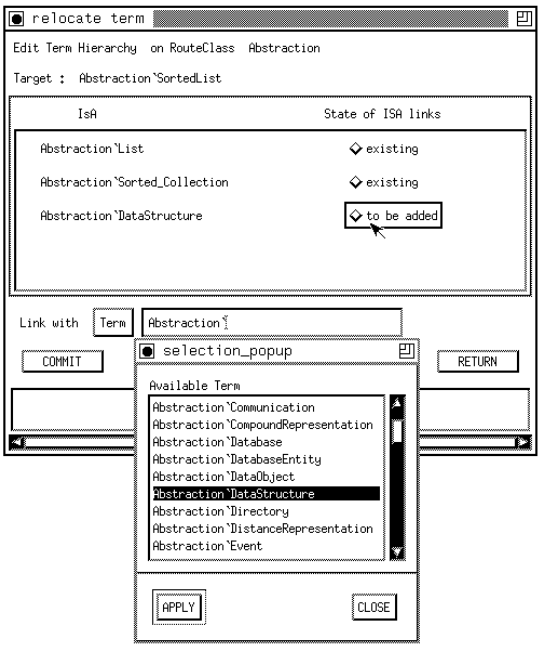
5. **Εκτέλεση Ενημέρωσης**

   Αίτηση εκτέλεσης των διατυπωμένων (από τα βήματα 3, 4) ΣΕΕ. Οι ΣΕΕ αυτές δεν είναι κατανάγκη σημασιολογικά ορθές. Το ΔΕΔ δεν ελέγχει το ίδιο τους σημασιολογικούς κανόνες (παρουσιάστηκαν στο κεφάλαιο 4) αφού ούτως η άλλως η SIS-Telos (συγκεκριμένα ο Semantic Ckecker) δεν επιτρέπει την εκτέλεση μιας ενημέρωσης που δεν είναι σημασιολογικά ορθή.

6. **Export**

   Κλήση του εργαλείου *Export* (περιγράφεται στην ενότητα 5.1.4).

[2] Το βήμα 2 δεν χρειάζεται αν ο τύπος είναι CREATE_INDIVIDUAL .



Σχήμα 5.5: Πρόσθεση υπερκλάσης

Όπως αναφέρθηκε στο βήμα 5 το ΔΕΔ δεν ελέγχει το ίδιο την τήρηση των σημασιολογικών κανόνων. Ελέγχει όμως κάποιους άλλους οι οποίοι προς το παρόν είναι επιθυμητοί (στην SIS-Telos) αλλά δεν ελέγχονται από τον Semantic Ckecker :

- Δεν επιτρέπεται η αναφορά μιας οντότητας που ήδη υπάρχει στη βάση αν η κατηγορία του γνωρίσματος είναι τύπου *non_shared.* Η κατηγορία *non_shared* είναι μια *ω-κατηγορία* [3] η οποία ερμηνεύεται (από το ΔΕΔ μόνο προς το παρόν) ως αμφιμονοσήμαντη συνάρτηση (*injection*) . Πιο απλά σημαίνει ότι μια οντότητα δεν μπορεί να αναφέρεται από παραπάνω από ένα αντικείμενα μέσω μιας κατηγορίας τύπου *non_shared.*

- Η δημιουργία νέων οντοτήτων αποτρέπεται αν η κλάση, στην οποία πρόκειται να ταξινομηθούν, είναι μια κλάση σταθερού πληθυσμού . Για να περιγράψουμε τις κλάσεις αυτές δημιουργήσαμε μια ειδική κλάση, την *RestrictedVocabulary*, στην οποία μπορούμε να ταξινομούμε τις κλάσεις που δεν επιδέχονται νέες περιπτώσεις.

[3] Οι κατηγορίες γνωρισμάτων που αποδίδονται σε κλάσεις συστήματος ονομάζονται *ω-κατηγορίες*. Οι κατηγορίες αυτές κληρονομούνται (και άρα μπορούν να χρησιμοποιηθούν) σε όλες τις οντότητες της αντίστοιχης κλάσης συστήματος.

5.1.4 "Εξαγωγή" μιας Όψης Ενημέρωσης : *Export*

Κάθε ΟΕ (όπως έχει ορισθεί) αποτελείται από ένα σύνολο ΣΕΕ οι οποίες εισάγουν, διαγράφουν ή τροποποιούν ένα τμήμα της ΒΔ, ένα σύνολο δηλαδή αντικειμένων και σχέσεων. Το εργαλείο *Export* παρέχει τη δυνατότητα περιγραφής του τμήματος αυτού σε ένα Αρχείο Εντολών Telos (ΑΕΤ) με εντολές TELL/RETELL .

Η δυνατότητα αυτή είναι χρήσιμη αφού μπορεί να χρησιμοποιηθεί ποικιλοτρόπως :

- **Μεταφορά ή ανταλλαγή δεδομένων μεταξύ διαφορετικών βάσεων**.
  Τα ΑΕΤ ενδείκνυνται αφού επιτρέπουν την μεταφορά ή ανταλλαγή δεδομένων μεταξύ βάσεων σε μη συμβατά υπολογιστικά συστήματα : αρκεί η μετάφρασή τους, από τον μεταφραστή (parser) του εκάστοτε συστήματος.

  Βέβαια η μεταφορά δεδομένων στη γενική της περίπτωση έχει αρκετές δυσκολίες όπως το πρόβλημα της *ομωνυμίας*, *συνωνυμίας* και *ανεξαρτησίας* (ένα τμήμα μπορεί να περιέχει λογικές αναφορές σε αντικείμενα που δεν υπάρχουν στη βάση παραλήπτη). Για να επιτύχουμε την ενημέρωση (μεταφορά δεδομένων νέας έκδοσης και αντικατάσταση των δεδομένων προγενέστερης έκδοσης) απαιτείται αφενός μεν η δυνατότητα περιγραφής της βάσης με ένα σύνολο ΑΕΤ (με τέτοιο τρόπο ώστε το προς αντικατάσταση τμήμα να είναι ένα ΑΕΤ), καθώς και επίλυση τους προβλήματος της ανεξαρτησίας : το νεό τμήμα προς εισαγωγή στη βάση παραλήπτη , μπορεί να περιέχει αντικείμενα , που αναφέρουν αντικείμενα που δεν περιέχονται στη βάση, ή αντίστροφα το προς αντικατάσταση τμήμα μπορεί να περιέχει αντικείμενα, τα οποία δεν υπάρχουν στο νέο τμήμα αλλά αναφέροναι από άλλα άντικείμενα της βάσης.

  Μια τέτοια απαίτηση (αντικατάσταση τμήματος της βάσης) εμφανίστηκε στην SIB και ικανοποιήθηκε όπως περιγράφεται στην ενότητα 5.2.1.

- **Αναβάθμιση έκδοσης**
  Αν τα περιεχόμενα μιας βάσεις έχουν παρασταθεί με ΑΕΤ είναι εφικτή η αλλαγή έκδοσης (αναβάθμιση/υποβάθμιση) της δυαδικής παράστασης της βάσης, με ανα-μετάφραση των ΑΕΤ από τον κατάλληλο μεταφραστή (προγενέστερο/νεότερο). Το *Export* έχει τη δυνατότητα περιγραφής σε ένα ΑΕΤ όλων των περιεχομένων της βάσης.

- **Δημιουργία υποβάσεων**
  Η παράσταση των περιεχομένων της βάσης με ΑΕΤ δίνει τη δυνατότητα δημιουργίας "υποβάσεων" αν μεταφράσουμε ένα υποσύνολο των ΑΕΤ.

- **Διατήρηση αντιγράφων**
  Ίσως η καλύτερη μορφή για την αποθήκευση αντιγράφων (backups) : ανεξάρτητα δυαδικής έκδοσης, αναγνώσιμα, απαιτούν λιγότερο χώρο αποθήκευσης (σε σύγκριση με την παρούσα έκδοση).

Το εργαλείο *Export* μπορεί να δημιουργήσει ένα ΑΕΤ που εκφράζει ολόκληρη τη βάση ή το πεδίο δράσης μιας ΟΕ. Στη δεύτερη περίπτωση απαιτεί ορίσματα τα οποία δεν είναι άλλα από την δήλωση της ΟΕ. Όμως για τεχνικούς λόγους που αφορούν τον κώδικα που εκτελεί ένα *QueryMacro*, οι οποίοι δεν κρίνονται ουσιαστικοί για να αναλυθούν εδώ, απαιτήθηκε η σύνδεση κάθε ΟΕ με γνώρισμα κατηγορίας *export* με ένα επιπλέον *QueryMacro*.

5.2 Οι Εργασίες Ενημέρωσης του SIB Class Management System

Το σύστημα **SIB Class Management System** είναι μια εφαρμογή του **SIS** για τεκμηρίωση και ανάλυση των στατικών ιδιοτήτων μονάδων λογισμικού. Παρέχει τη δυνατότητα περιγραφής και ανάκλησης των μονάδων λογισμικού με βάση τη λειτουργία τους, με σκοπό την αναχρησιμοποίησή τους.

Το σύστημα δουλεύει ως εξής : προγράμματα και βιβλιοθήκες λογισμικού μεταφράζονται από ειδικούς μεταφραστές (parsers) ειδικευμένους στη γλώσσα προγραμματισμού των μονάδων λογισμικού. Το αποτέλεσμα της μετάφρασης είναι περιγραφές των στατικών ιδιοτήτων των μονάδων λογισμικού στη γλώσσα SIS-Telos .

Κατόπιν οι μονάδες λογισμικού ταξινομούνται διαλογικά από τον χρήστη με βάση την λειτουργία τους. Η ταξινόμηση αυτή ονομάζεται *Πολυεδρική Ταξινόμηση* και αποτελείται από τέσσερεις ιεραρχίες εξειδίκευσης ή αλλιώς *Όψεις/Έδρες Ταξινόμησης* (*Classification Facets* [39]) : *Abstraction, Operation, OperatesOn* και *Dependencies*, οι οποίες μάλιστα μπορούν να εξελίσσονται.

Οι απαιτήσεις διαλογικής εισαγωγής και ενημέρωσης της εφαρμογής ενός χρήστη του συστήματος καλύπτονται από τρείς ΟΕ :

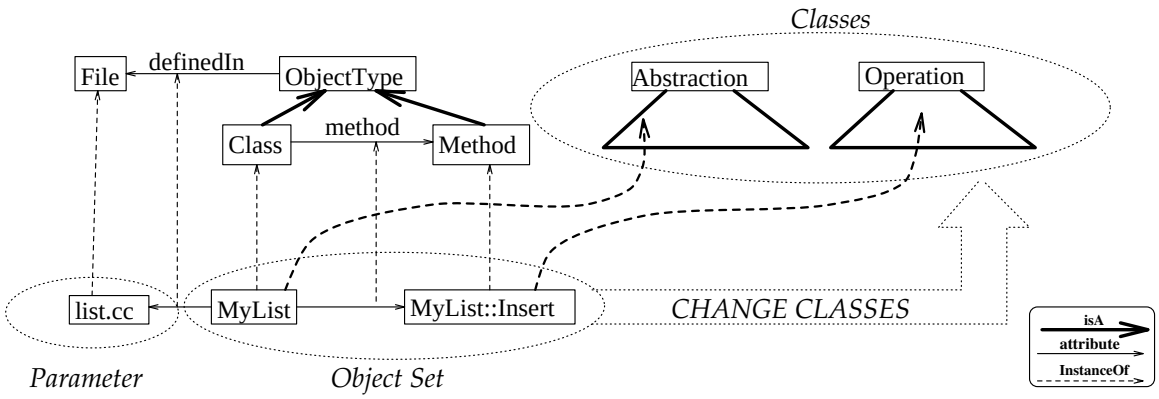
1. **Classify Software Object**
2. **Edit Term Hierarchy**
3. **Describe Library**

οι οποίες ορίζονται με τη βοήθεια του ΜΔΕ ως εξής :

```
TELL Individual SIBuser in Token , Users
    with tasks
        (Classify Software Object)    : (ClassifyS/O);
        (Edit Term Hierarchy)         : EditTermHierarchy;
        (Describe Library)            : DescribeLibrary
end
```

1. Η ΟΕ **Classify Software Object** επιτρέπει στο χρήστη την **πολυεδρική ταξινόμηση** των μονάδων λογισμικού που είναι δηλωμένες σε ένα αρχείο, σε ένα ή περισσότερους όρους ανάλογα με τον λειτουργικό ρόλο κάθε μονάδας λογισμικού. Ο κώδικας SIS-Telos παρουσιάζεται παρακάτω.

   Το QueryMacro που υπολογίζει το *ObjectSet* (*ClassifiablesInFile*) δέχεται μία παράμετρο τύπου *ImplementationFile* και επιστρέφει τις ταξινομήσιμες μονάδες λογισμικού (*classes, member_functions, friend, operators, public*) που είναι δηλωμένες στο αρχείο, του οποίου το όνομα δόθηκε σαν παράμετρος. Στην εργασία αυτή είναι ορισμένη μια πράξη (*ClassifyObject*) τύπου (CHANGE_CLASSES) με σύνολο ορισμάτων το σύνολο που ορίζει το *QueryMacro Available_Related_Terms*. Το σύνολο αυτό δέχεται σαν παράμετρο την οντότητα που ο χρήστης θέλει να ταξινομήσει και αν αυτή είναι περίπτωση (άμεση ή έμμεση) της κλάσης *Class*, τότε επιστρέφει τους όρους της ιεραρχίας **Abstraction**, ενώ αν είναι περίπτωση της κλάσης *Method* τότε επιστρέφει τους όρους της ιεραρχίας ειδίκευσης **Operation** (δες σχήμα 5.6). Το σύνολο αυτό υπολογίζεται κατά τη χρήση από το ΔΕΔ και προτείνεται στο χρήστη για την επιλογή του επιθυμητού όρου.



Σχήμα 5.6: Το πεδίο δράσης της ΟΕ **Classify Software Object**

Οι περιπτώσεις της κλάσης *Class* ταξινομούνται στην ιεραρχία της κλάσης *Method*, ενώ οι περιπτώσεις της κλάσης *Routine* ταξινομούνται στην ιεραρχία της κλάσης *Operation*.

Η περιγραφή σε SIS-Telos της πράξεως αυτής ακολουθεί :

```
TELL Individual (ClassifyS/O) in Token , UpdateTask
    with object
        list_classifiables   : ClassifiablesInFile
    update_operation
        Classify             : ClassifyObject
end

TELL Individual ClassifiablesInFile in Token , QueryMacro
    with code
        (1)   : "scn * ImplementationFile" ;
        (2)   : "gfnc Structure defined_in 0 ";
        (3)   : "gtnc C++'Structure member_functions 1 ";
        (4)   : "gtnc C++'Structure friend 1 ";
        (5)   : "gtnc C++'Structure operators 1 ";
        (6)   : "gtnc C++'Structure public 1 ";
        (7)   : "su 2 3 ";
        (8)   : "su 2 4 ";
        (9)   : "su 2 5 ";
        (10)  : "su 1 2 "
end

TELL Individual ClassifyObject in Token , Update
    with update_type
                : CHANGE_CLASSES
    argument
        Term    : Available_Related_Terms
end
```

Τα ονόματα των γνωρισμάτων που εμφανίζονται στις εντολές TELL/RETELL εντολές χρησιμοποιούνται από το ΔΕΔ κατά τη διάρκεια του διαλόγου του με το χρήστη.

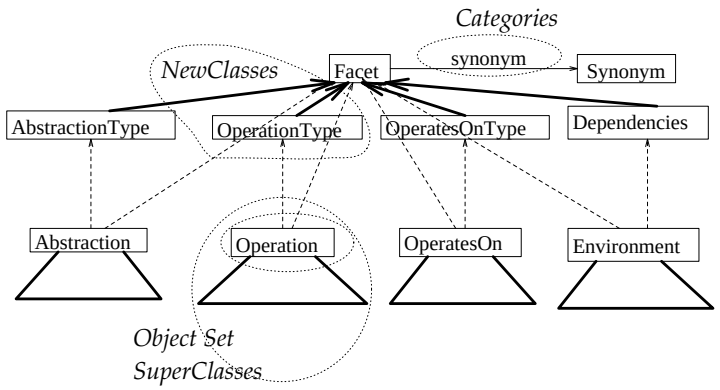
2. Η ΟΕ **Edit Term Hierarchy** επιτρέπει στο χρήστη την εξέλιξη των *Όψεων Ταξινόμησης* λογισμικού. Παρέχει ένα σύνολο πράξεων με τις οποίες είναι δυνατή η εξέλιξη μιας ιεραρχίας γενίκευσης : εισαγωγή όρου (πράξη *Add Term* τύπου CREATE_INDIVIDUAL), μετονομασία όρου (πράξη *Rename Term* τύπου RENAME_INDIVIDUAL), διαγραφή όρου (πράξη *Delete Term* τύπου DELETE_INDIVIDUAL), ανατοποθέτηση όρου στην ιεραρχία (πράξη *Relocate Term* τύπου CHANGE_SUPERCLASSES) και προσθήκη συνωνύμων σε έναν όρο (πράξη *EditSynonym* τύπου CHANGE_SUPERCLASSES) :

UpdateType	ArgSet Name
RENAME_INDIVIDUAL	-
DELETE_INDIVIDUAL	-
CREATE_INDIVIDUAL	$NewClasses$
CHANGE_SUPERCLASSES	$SuperClasses$
CHANGE_ATTRIBUTES	$Categories$

Το *ObjectSet* αυτής δέχεται ως παράμετρο την κορυφή της ιεραρχίας εξειδίκευσης. Με τον τρόπο αυτό μπορεί να δράσει και στις τέσσερεις έδρες ταξινόμησης. Για το λόγο αυτό οι κλάσεις *Abstraction, Operation, OperatesOn* και *Dependencies*, δηλώθηκαν ως άμεσες περιπτώσεις της κλάσεως *Facet* και το *QueryMacro* του *ObjectSet* έχει ως εξής :

TELL Individual Class_Set **in Token** , QueryMacro with
code
(1) : "scn * Facet";
(2) : "gasb 0";
(3) : "sput 1"
end

Στο σχήμα 5.7 φαίνονται σχηματικά τα περιεχόμενα των υπολοίπων συνόλων της ΟΕ.



Σχήμα 5.7: Το πεδίο δράσης της ΟΕ **Edit Term Hierarchy**

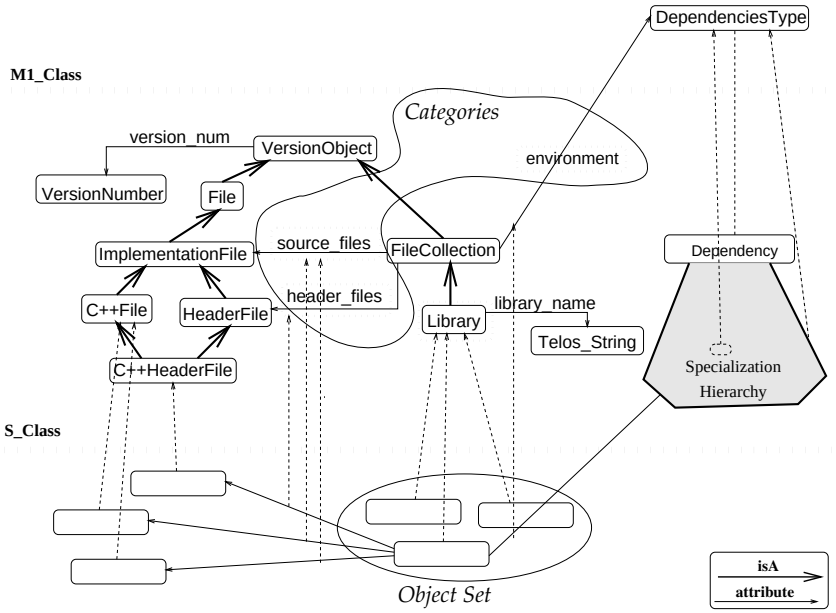
Το σχήμα αυτό περιγράφει τις δηλώσεις της όψης ενημέρωσης *Edit Term Hierarchy* με παράμετο την κλάση *Operation*.

Η ενημέρωση *Add Term* συνδέεται με ένα σύνδεσμο **ακολουθίας (sequence)** με την ενημέρωση *Relocate Term*. Η δήλωση αυτή υποδεικνύει στο ΔΕΔ να απαιτήσει από το χρήστη την ένταξη στην ιεραρχία κάθε νέου όρου που δημιουργεί [4].

[4]Οι δύο αυτές ΣΕΕ πραγματοποιούνται μαζί σε μία δοσοληψία. Αν η δημιουργία του όρου γινόταν

3. Η ΟΕ **Describe Library** παρέχει τη δυνατότητα στο χρήστη να προσθέσει και να περιγράψει νέες βιβλιοθήκες λογισμικού. Η έννοια της βιβλιοθήκης μοντελοποιείται με την κλάση *Library* που φαίνεται στο σχήμα 5.8. Η ΟΕ αυτή σε γενικές γραμμές επιτρέπει την διαχείριση των περιπτώσεων μιας προβολής της κλάσεως *Library*:

   Ορίστηκε μια ενημέρωση *AddLibrary* (τύπου CREATE_INDIVIDUAL) , με την οποία ο χρήστης μπορεί να δημιουργεί νέες περιπτώσεις της κλάσεως *Library* και μια ενημέρωση τύπου *CHANGE_ATTRIBUTE*, η οποία επιτρέπει την προσθήκη/διαγραφή γνωρισμάτων που η κατηγορία τους ανήκει στο υποσύνολο των κατηγοριών της κλάσεως *Library* : {*header_files, source_files, environment*}. Το *ObjectSet* αποτελείται από όλες τις περιπτώσεις της κλάσης *Library*.



Σχήμα 5.8: Λειτουργία περιγραφής βιβλιοθήκης

Το σχήμα περιγράφει τις δηλώσεις της όψης ενημέρωσης *Describe Library* , οι οποίες ορίζουν τις ενημερώσεις που προσθέτουν ή διαγράφουν περιπτώσεις στην κλάση *Library*, και στις κατηγορίες γνωρισμάτων *header_files, source_files, και environment*. Κατά την απόδοση των γνωρισμάτων *header_files, source_files*, μπορούν να συγκεκριμενοποιηθούν οι υποκλάσεις της κλάσης *File*

ξεχωριστά τότε θα ήταν αδύνατη η εκ των υστέρων ένταξη του στην ιεραρχία μέσω της παρούσης ΕΕ, γιατί δεν θα ανήκε στο *ObjectSet* αυτής, αφού δεν θα ήταν υποκλάση της αρχικής κλάσης.

5.2.1 Χρήση του *Export*

Στην SIB η περιγραφή της δομής μιας μονάδας λογισμικού σε SIS-Telos , πρέπει να ενημερώνεται (να αντικαθιστά την παλαιά) όταν ο πηγαίος κώδικας της μονάδας αλλάζει. Το πρόβλημα της αντικατάστασης επιλύθηκε διατηρώντας τα περιεχόμενα της SIB σε ένα αριθμό από ΑΕΤ (Αρχεία Εντολών Telos). Η αντικατάσταση επιτυγχάνεται αναδημιουργώντας τη βάση από τα ΑΕΤ έχοντας αντικαταστήσει τα παλαιά ΑΕΤ στατικής ανάλυσης με τα νέα.

Τα αρχεία στα οποία η SIB διαμερίζεται είναι τα εξής :

- **SIS Kernel Model, C++Model, Facet Kernel Model**
  Αμετάβλητα αφού περιγράφουν τον πυρήνα του μοντέλου της βάσης ο οποίος δεν επιδέχεται αλλαγές.

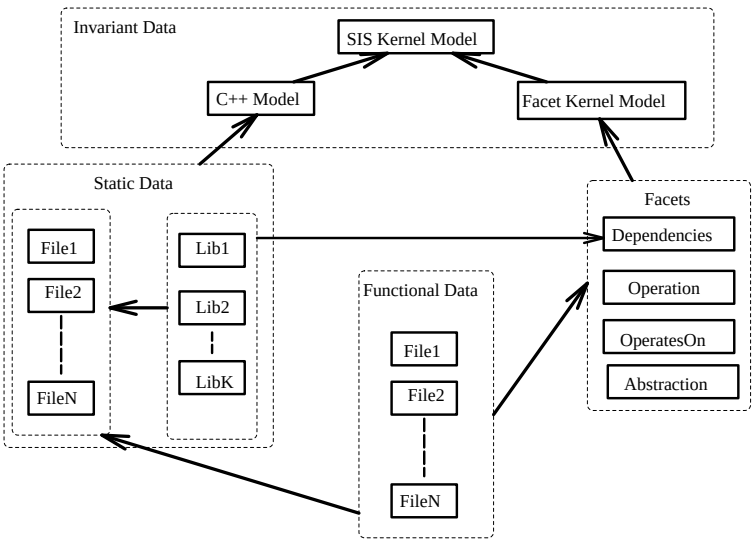
- **File1, File2, .. FileN** : αρχεία στατικής ανάλυσης
  Μεταβλητά ως συνέπεια αλλαγών στον κώδικα των μονάδων λογισμικού. Παράγονται από ειδικούς μεταφραστές.

- **Lib1, .., LibK**
  Μεταβλητά αφού περιγράφουν τα δεδομένα που έχουν εισαχθεί διαλογικά μέσω της ΟΕ *Describe Library*. Παράγονται με κλήση του *Export* .

- **Abstraction,Operation,OperatesOn, Dependencies**
  Μεταβλητά αφού περιγράφουν τις διαλογικά εξελισσόμενες έδρες ταξινόμησης, μέσω των ΟΕ *Edit Term Hierarchy*. Παράγονται με κλήση του *Export* .

- **Functional**{**File1, File2, .. FileN**} : αρχεία λειτουργικής ταξινόμησης
  Μεταβλητά αφού περιγράφουν τους συνδέσμους της πολυεδρικής ταξινόμησης των μονάδων λογισμικού. Παράγονται με κλήση του *Export* μέσω της ΟΕ *Classify Software Object*.

Τα παραπάνω αρχεία δεν είναι ανεξάρτητα αλλά περιέχουν λογικές αναφορές (με βάση το λογικό όνομα) σε άλλα αντικείμενα. Οι εξαρτήσεις αυτές φαίνονται στο σχήμα 5.9 και καθορίζουν τη σειρά μετάφρασης τους κατά την αναδημιουργία της βάσης (μια ορθή σειρά είναι η εξής : *Invariant Data* → *Facets* → *Static Data* → *Functional Data*).



Σχήμα 5.9: Οργάνωση και Εξαρτήσεις των αρχείων SIS-Telos της SIB

Ένα βέλος από ένα αρχείο (ή σύνολο αρχείων) $\mathcal{A}$ σε ένα αρχείο (ή σύνολο αρχείων) $\mathcal{B}$ δηλώνει ότι το $\mathcal{A}$ πιθανόν αναφέρει λογικά αντικείμενα που δηλώνονται στο $\mathcal{B}$.

5.3 Οι Όψεις Ενημέρωσης του *ΚΛΕΙΩ*

Μετά την εφαρμογή του ΜΔΕ στο **SIB Class Management System**, έγινε προσπάθεια χρήσης του στο σύστημα *ΚΛΕΙΩ* [26] ώστε να είναι εφικτή η διαλογική εισαγωγή και ενημέρωση της βάσης. Το σύστημα *ΚΛΕΙΩ* είναι ένα πληροφοριακό σύστημα μουσείων το οποίο μπορεί να χρησιμοποιηθεί σαν επιστημονικός κατάλογος μουσειακών αντικειμένων και καταχωρεί φυσικές, μορφολογικές, χρηστικές, ιστορικές, γεωγραφικές και χρονικές πληροφορίες.

Χαρακτηριστικά μοντέλου

Το μοντέλο του είναι αρκετά σύνθετο (περιέχει παραστάσεις σύνθετων οντοτήτων) και μπορεί να εμπλουτίζεται διαρκώς όσο χρησιμοποιείται.

Τρόπος χρήσης του ΜΔΕ

Ο ορισμός εξειδικευμένων ΟΕ θα ήταν **επίπονος** αφού το μοντέλο είναι αρκετά ευρύ, και θα απαιτούσε **συνεχή ενημέρωση** καθώς το σχήμα της βάσης εξελίσσεται διαρκώς. Για το λόγο αυτό έγινε προσπάθεια ορισμού γενικών και παραμετρικών ΟΕ που βασίζονται στον συνήθη τρόπο που μια βάση εξελίσσεται. Αποτέλεσμα της προσπάθειας αυτής ήταν ο σχεδιασμός δύο ‘‘γενικών’’ ΟΕ :

- **Describe Class** (**DC**) : Επιτρέπει την εξέλιξη ιεραρχιών εξειδίκευσης.

Describe Class		
UpdateType	*Όνομα*	*Περιεχόμενα*
-	*ObjectSet*	υποκλάσεις $\mathcal{R}$
RENAME_INDIVIDUAL	-	-
DELETE_INDIVIDUAL	-	-
CREATE_INDIVIDUAL	$NewClasses$	οι κλάσεις του $\mathcal{R}$
CHANGE_CLASSES	$Classes$	όλες οι υποκλάσεις των κλάσεων του $\mathcal{R}$
CHANGE_SUPERCLASSES	$Classes$	το $\mathcal{R}$ και όλες οι υποκλάσεις του
CHANGE_ATTRIBUTES	$Categories$	τα γνωρίσματα όλων των υποκλάσεων των κλάσεων του $\mathcal{R}$

Πίνακας 5.1: Περιγραφή της ΟΕ Describe Class

Describe Object		
UpdateType	*Όνομα*	*Περιεχόμενα*
-	*ObjectSet*	όλες οι περιπτώσεις της $\mathcal{R}$
RENAME_INDIVIDUAL	-	-
DELETE_INDIVIDUAL	-	-
CREATE_INDIVIDUAL	$NewClasses$	η $\mathcal{R}$
CHANGE_CLASSES	$Classes$	όλες οι υποκλάσεις της $\mathcal{R}$
CHANGE_ATTRIBUTES	$Categories$	τα γνωρίσματα όλων των υποκλάσεων της $\mathcal{R}$

Πίνακας 5.2: Περιγραφή της ΟΕ Describe Object

- **Describe Object** (**DO**): Επιτρέπει την εξέλιξη του πληθυσμού ιεραρχιών εξειδίκευσης.

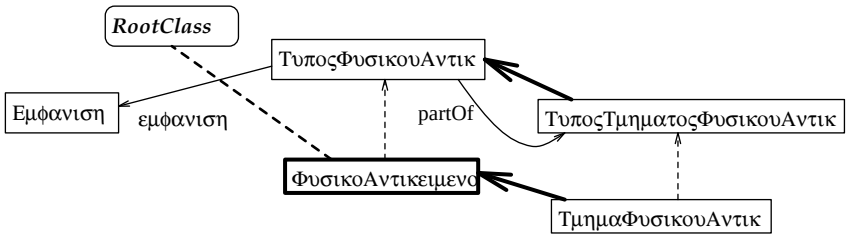
Οι ιεραρχίες στις οποίες εδράζονται οι δύο ΟΕ καθορίζονται με την ταξινόμηση των κορυφαίων τους υπερκλάσεων [5] στην κλάση **RootClass** η οποία δημιουργήθηκε για αυτό το σκοπό. Τις κλάσεις αυτές θα τις ονομάζουμε *εστιακές κλάσεις*. Η δήλωση μιας κλάσης $\mathcal{R}$ ως εστιακής παρέχει τη δυνατότητα εξέλιξης της υποιεραρχίας εξειδίκευσης της $\mathcal{R}$ (μέσω της ΟΕ **DC**), και των περιπτώσεων της υποιεραρχίας της $\mathcal{R}$ (μέσω της ΟΕ **DO**). Οι **DC** και **DO** περιγράφονται με περισσότερη λεπτομέρεια στους πίνακες 5.1 και 5.2.

5.3.1 Παραδείγματα Χρήσης

Ένα παράδειγμα της λειτουργικότητας των δύο ΟΕ παρουσιάζεται στα σχήματα 5.10 και 5.11.

[5] Σε μια ιεραρχία ή υποιεραρχία εξειδίκευσης, ονομάζουμε *κορυφαία υπερκλάση* την κλάση της εκείνη, που είναι υπερκλάση κάθε άλλης κλάσης της ιεραρχίας.

Στο σχήμα 5.10 βλέπουμε ένα τμήμα του πυρήνα του μοντέλου του *ΚΛΕΙΩ* , στο οποίο παρατηρούμε ότι η κλάση *ΦυσικόΑντικείμενο* είναι δηλωμένη ως *εστιακή*, ενώ στο σχήμα 5.11 βλέπουμε ένα πιθανό στιγμιότυπο της βάσης το οποίο μπορεί να προκύψει με τη χρήση των **DC** , **DO** .



Σχήμα 5.10: Ορισμός πεδίου δράσεως των **DC** , **DO** .

Δύο ΟΕ μπορεί να είναι φωλιασμένες, δηλαδή για δύο *εστιακές κλάσεις* $\mathcal{A}$, $\mathcal{B}$ να ισχύει $(\mathcal{B}, \mathcal{A}) \in R_u^{isA}$. Η σκοπιμότητα τέτοιων δηλώσεων είναι ο περιορισμός μιας ΟΕ, ο οποίος συμβάλλει στη ευκολία του χρήστη.

5.3.2 Επέκταση της Describe Object

Η πράξη ταξινόμησης (τύπου CHANGE_CLASSES) της **DO** επιτρέπει την ενημέρωση της ταξινόμησης των οντοτήτων στα πλαίσια μόνο της ιεραρχίας εξειδίκευσης της εστιακής κλάσεως. Αυτό όμως ήταν αρκετά περιοριστικό αφού συχνά η ταξινόμηση χρησιμοποιείται για την απόδοση ιδιαίτερων ή σύνθετων ιδιοτήτων σε μια οντότητα : ταξινόμηση *Τεχνητών Αντικειμένων* σε *Τεχνοτροπία*, *Προσώπων* σε *Επάγγελμα* (δες σχήμα 6.41) κλπ.

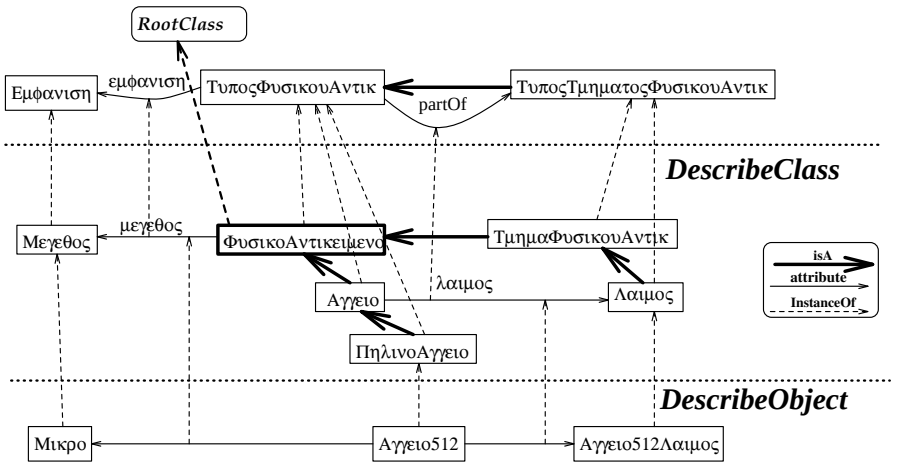
Για το λόγο αυτό ορίσθηκε μια νέα πράξη, η **Give Aspect**, με τύπο CHANGE_CLASSES , της οποίας το σύνολο ορισμάτων περιέχει όλες τις *εστιακές κλάσεις* (περιπτώσεις της *RootClass*). Η πράξη αυτή κάνει δυνατή την ταξινόμηση μιας οντότητας σε μια άλλη ιεραρχία.

5.4 Αξιολόγηση

Θετικά Σημεία

- **Δυναμικός Ορισμός**

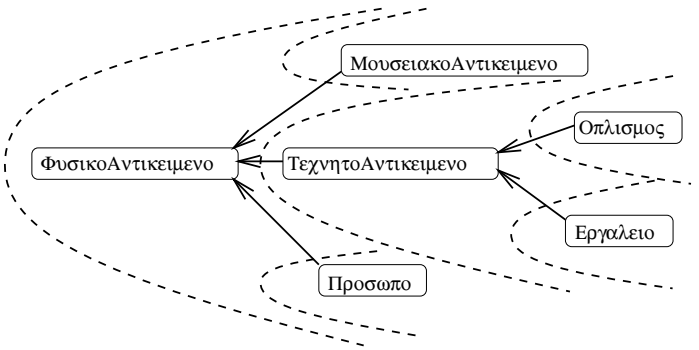
  Το σύνολο των ΣΕΕ που μια ΟΕ ορίζει είναι δυναμικό αφού τα σύνολα από τα οποία ορίζεται, δηλώνονται περιγραφικά (*QueryMacro*).



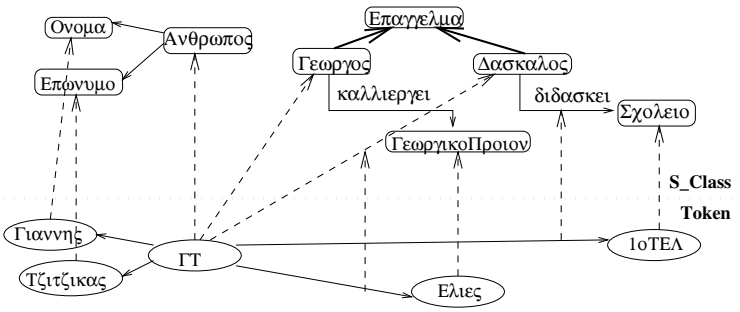
Σχήμα 5.11: Πιθανή κατάσταση της βάσης με χρήση των **DC** , **DO** .

Πιθανά βήματα εκτέλεσης του ΔΕΔ :

1. Επιλογή ΟΕ **DC**
2. Εισαγωγή παραμέτρου (*ΦυσικόΑντικείμενο*)
3. Δημιουργία γνωρίσματος *Μέγεθος* (CHANGE_ATTRIBUTES)
4. Δημιουργία *Αγγείου* (CREATE_INDIVIDUAL)
5. Δήλωση *Αγγείου* ως εξειδίκευση του *ΦυσικόΑντικείμενο* (CHANGE_SUPERCLASSES)
6. Ταξινόμηση *Αγγείου* σε *ΤύποΦυσικούΑντικειμένου* (CHANGE_CLASSES)
7. Επανάληψη βημάτων 4,5 για δημιουργία *Λαιμού*
8. Ταξινόμηση *Λαιμού* σε *ΤύποΤμήματοςΦυσικούΑντικειμένου* (CHANGE_CLASSES)
9. Δημιουργία γνωρίσματος *λαιμός* (CHANGE_ATTRIBUTES)
10. Επανάληψη βημάτων 5,6 για δημιουργία *Πήλινου Αγγείου*



Σχήμα 5.12: Φωλιασμένες Όψεις Ένημέρωσης



Σχήμα 5.13: Χρήση της ταξινόμησης για την απόδοση σύνθετων ιδιοτήτων

- **Παραμετροποίηση**
  Υπάρχει η δυνατότητα δήλωσης παραμετρικών ΟΕ που δρούν σε διαφορετικά τμήματα της βάσης. Αυτό συμβάλλει στην αναχρησιμοποίηση των δηλώσεων.

- **Εκφραστική ικανότητα ΟΕ.**
  Η χρήση των ερωτηματικών συναρτήσεων του **qi** στα *QueryMacro* παρέχει τη δυνατότητα ορισμού εξειδικευμένων όψεων ενημέρωσης.

Αρνητικά Σημεία

- **Μελέτη ιδιοτήτων και Εξαγωγή Συμπερασμάτων**
  Δύσκολη η εξαγωγή συμπερασμάτων τις μορφής :

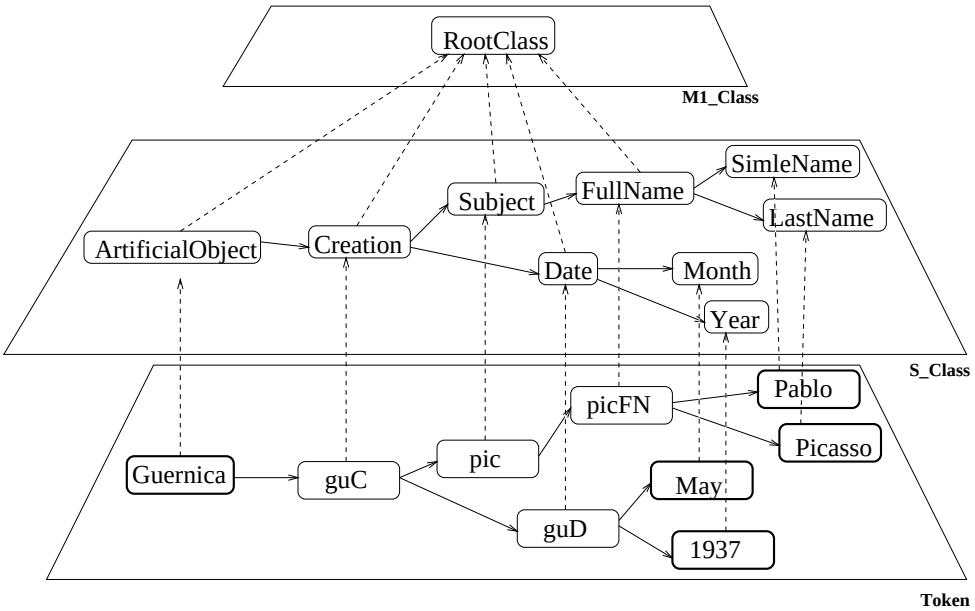
  -- *Ποιές ΣΕΕ μπορώ να εκτελέσω (έχουν οριστεί) στο αντικείμενο $\mathcal{A}$?*

  ή

  Ποιές ΣΕΕ μπορώ να εκτελέσω (έχουν οριστεί) στο αντικείμενο $\mathcal{A}$ στα πλαίσια της ΟΕ $\mathcal{E}$?

  Η απάντηση στην ερώτηση αυτή απαιτεί την αναζήτηση του $\mathcal{A}$ σε όλα τα *ObjectSet* των ΟΕ που είναι ορισμένες, το οποίο με τη σειρά του απαιτεί την εκτέλεση των ερωτηματικών εντολών των αντίστοιχων *QueryMacro*. Αν μάλιστα λάβουμε υπόψη ότι συνήθως τα *ObjectSet* ορίζονται παραμετρικά, καταλαβαίνουμε ότι η αναζήτηση αυτή δεν μπορεί να γίνει αποδοτικά.

  Η δυνατότητα απάντησης τέτοιου είδους ερωτήσεων θα ήταν πολύ χρήσιμη αφού θα επέτρεπε την ανάπτυξη ενός νέου ΔΕΔ το οποίο δεν θα υποχρέωνε το χρήστη να ακολουθήσει συγκεκριμένα βήματα όπως συμβαίνει τώρα (επιλογή ΟΕ → επιλογή οντότητας → επιλογή τύπου ενημέρωσης → εκτέλεση) οπως αναφέρθηκε και στην παράγραφο 5.1.3.



Σχήμα 5.14: Παράδειγμα παράστασης ενός σύνθετου αντικειμένου

Η εισαγωγή στη βάση του πίνακα της *Guernica*, για τον οποίο το μόνο που γνωρίζουμε είναι ότι ζωγραφίστηκε από τον *Pablo Picasso* τον Μάιο του 1920, απαιτεί την εξής ακολουθία βημάτων :

1. Επιλογή **DO** με παράμετρο *ArtificialObject* για τη δημιουργία της οντότητας *Guernica* και του γνωρίσματος *guCr*.
2. Επιλογή **DO** με παράμετρο *Creation* , εύρεση της οντότητας *guCr* και δημιουργία του γνωρίσματος *guD*, καθώς και δημιουργία ή αναφορά του γνωρίσματος *pic*.
3. Επιλογή **DO** με παράμετρο *Subject* , εύρεση της οντότητας *pic* και δημιουργία του γνωρίσματος *picFN*.
4. Επιλογή **DO** με παράμετρο *FullName* , εύρεση της οντότητας *picFN* και δημιουργία των γνωρισμάτων *Pablo* και *Picasso*.
5. Επιλογή **DO** με παράμετρο *Date* , εύρεση της οντότητας *guD* και δημιουργία των γνωρισμάτων *May* και *1920*.

Η απαίτηση αυτή είναι καίρια σε εφαρμογές που διαχειρίζονται σύνθετα αντικείμενα, όπως στο σύστημα *ΚΛΕΙΩ* . Η εισαγωγή σύνθετων αντικειμένων περιλαμβάνει εισαγωγή αντικειμένων και σχέσεων τα οποία σχηματίζουν μονοπάτια. Λόγω του ότι τα αντικείμενα που βρίσκονται κατά μήκος ενός μονοπατιού ανήκουν συνήθως σε *ObjectSet* διαφορετικών ΟΕ, η εισαγωγή ενός σύνθετου αντικειμένου απαιτεί από τον χρήστη την εναλλαγή πολλών ΟΕ το οποίο είναι ιδιαίτερα ενοχλητικό. Παράδειγμα παρουσιάζεται στο σχήμα 5.14.

-- *Ασυμμετρία Εκτέλεσης των ΣΕΕ*

Λόγω της αδυναμίας εξαγωγής συμπερασμάτων σχετικά με τις δηλώσεις μιας ΟΕ, το ΔΕΔ υποστηρίζει την εκτέλεση των ΣΕΕ που ορίζονται από μια ΟΕ

κατά την μια φορά μόνο. Η αδυναμία αυτή κάνει ορισμένες πράξεις όπως για παράδειγμα τη διαγραφή ενός αντικειμένου που αναφέρεται από άλλα ως τιμή γνωρίσματος, πολύ επίπονες, αφού απαιτείται η μετάβαση σε κάθε αντικείμενο που αναφέρει το εν λόγω αντικείμενο και διαγραφή του σχετικού συνδέσμου.

- **Ακεραιότητα**
  Ο κώδικας ενός QueryMacro αναφέρει αντικείμενα με το λογικό τους όνομα. Η μετονομασία ή η διαγραφή ενός τέτοιου αντικειμένου προκαλεί ασυνέπεια στις ΟΕ που χρησιμοποιούν αυτό το *QueryMacro* στη δήλωσης τους. Άρα υπάρχει ανάγκη συντήρησης.

- **Εποπτικότητα**
  Η δυνατότητα κατανόησης των ΣΕΕ που ορίζει μια ΟΕ είναι δύσκολη (όχι άμεση) αφού η κατανόηση του συνόλου που ορίζεται από ένα *QueryMacro* είναι αρκετά δύσκολη.

- **Επαλήθευση ορθότητας**
  Η σημασιολογική ορθότητα των ΣΕΕ, που ορίζονται μέσω των τύπων CHANGE_CLASSES , CHANGE_SUPERCLASSES , CHANGE_ATTRIBUTES , δεν είναι εύκολο να ελεγχθεί κατά τη φάση δήλωσης (η κατανόηση των ερωτηματικών συναρτήσεων δεν είναι έυκολη). Συνήθως ελέγχεται δοκιμάζοντάς τις με το ΔΕΔ.

- **Η χρήση του pqi**
  Η χρήση του pqi είναι κοπιαστική και εφικτή μόνο από έμπειρους χρήστες.

5.4.1 Γενικά Συμπεράσματα

Το ΜΔΕ κάλυψε πλήρως τις ανάγκες της διαλογικής εξέλιξης γνώσης της εφαρμογής **SIB** αφού ο τρόπος ερμηνείας του από το ΔΕΔ οδήγησε σε ένα φιλικό διάλογο με το χρήστη. Σημειώνουμε βέβαια ότι οι ΟΕ της SIB είναι επακριβώς προκαθορισμένες, απλές και δεν εισάγουν (ενημερώνουν) σύνθετα αντικείμενα.

Στο σύστημα *ΚΛΕΙΩ* οι ανάγκες εξέλιξης γνώσης είναι μεγαλύτερες. Τα τμήματα του μοντέλου που εξελίσσονται είναι πολλά και όχι καθορισμένα (σταθερά) εξ αρχής. Το πρόβλημα αυτό αντιμετωπίστηκε με τη σχεδίαση δύο γενικών ΟΕ οι οποίες δέχονται μία παράμετρο η οποία καθορίζει το πεδίο δράσης τους. Με τον τρόπο αυτό μπορούμε εύκολα να ορίζουμε νέες όψεις ενημέρωσης (αρκεί να δηλώσουμε το πεδίο δράσης τους). Στο *ΚΛΕΙΩ* επίσης υπάρχει η ανάγκη διαλογικής εισαγωγής και ενημέρωσης σύνθετων

αντικειμένων. Η παρούσα προσέγγιση μειονεκτεί στο σημείο αυτό γιατί ο τρόπος ερμηνείας των ΟΕ από το ΔΕΔ (ο οποίος καθορίζεται από τον τρόπο περιγραφής των ΟΕ) δεν επιτρέπει την άμεση πρόσβαση και εν συνεχεία την ενημέρωση, ενός αντικειμένου, αλλά απαιτεί μια ακολουθία κοπιαστικών, για το χρήστη, βημάτων (διαδικασιών).

Κεφάλαιο 6

Το Μεταμοντέλο Όψεων Ενημέρωσης

6.1 Αιτιολόγηση

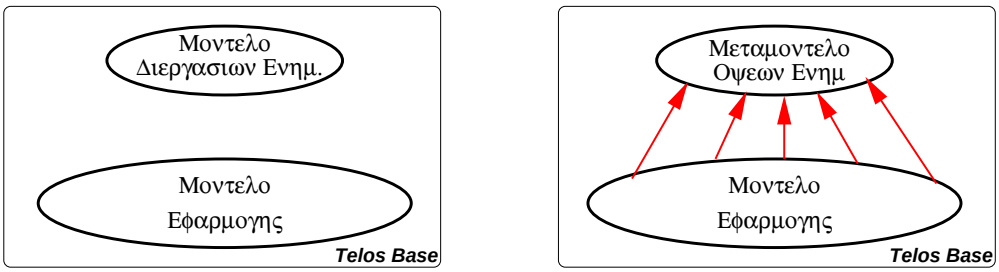
Για την άρση των αδυναμιών του προηγούμενου μοντέλου διεργασιών, επιχειρείται η **μοντελοποίηση των περιεχομένων** των ΟΕ και η **ενσωμάτωση** των δηλώσεων αυτών (των μεταδεδομένων), στο μοντέλο της εκάστοτε εφαρμογής. Με άλλα λόγια προτείνεται η χρήση ενός **ΜεταΜοντέλου Όψεων Ενημέρωσης** (ΜΜΟΕ) .

Η προηγούμενη προσέγγιση, το Μοντέλο Διεργασιών Ενημέρωσης, δεν μπορεί να χαρακτηριστεί ως μεταμοντέλο. Αν και η δήλωση μιας όψης ενημέρωσης αποθηκεύεται στην ίδια τη βάση, η σύνδεση της με τις οντότητες που αφορά (το μοντέλο εφαρμογής) δεν είναι μοντελοποιημένη με τους μηχανισμούς της SIS-Telos . Η προσέγγιση αυτή δεν επιτρέπει την αποδοτική αξιοποίηση των ΟΕ αφού είναι δύσκολη η εξαγωγή συμπερασμάτων που αφορούν τα περιεχόμενα των ΟΕ (όπως αναφέρθηκαν κατά την αξιολόγηση του ΜΔΕ στην τελευταία ενότητα του προηγούμενου κεφαλαίου).

Αντίθετα, η μοντελοποίηση των δηλώσεων που ορίζουν μια ΟΕ και η ενσωμάτωση τους στο μοντέλο εφαρμογής, υπόσχεται μια πιο αποδοτική αξιοποίηση τους αφού, όπως θα δούμε, είναι δυνατή η εξαγωγή των απαραίτητων συμπερασμάτων (κάνοντας τις κατάλληλες ερωτήσεις με το **qi**). Όσο για την ταχύτητα εξαγωγής συμπερασμάτων, σημειώνουμε ότι η SIS-Telos έχει υψηλές επιδόσεις στη διάσχιση συνδέσμων.

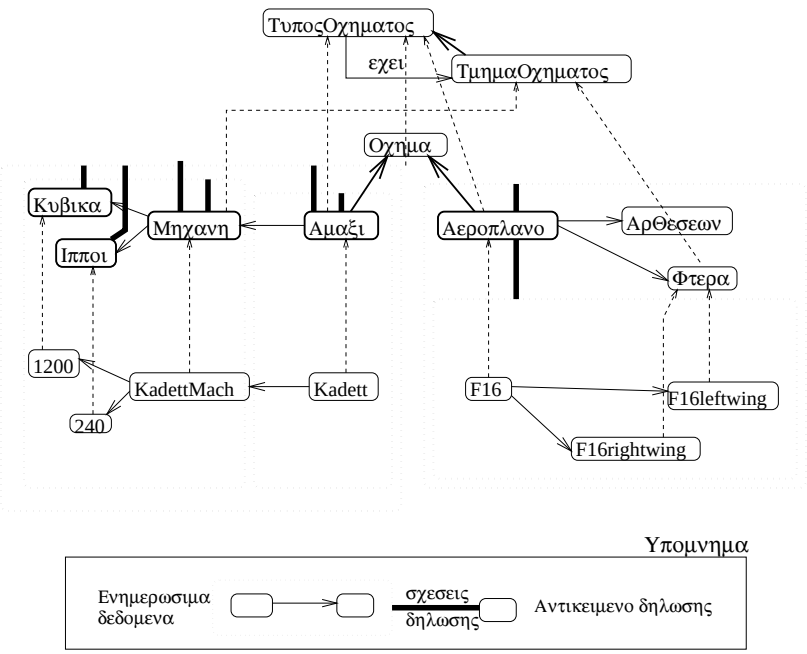
Μια σχηματική απεικόνιση των δύο διαφορετικών προσεγγίσεων παρουσιάζεται στο σχήμα 6.1, ενώ ένα πιο συγκεκριμένο παράδειγμα της προτεινόμενης προσέγγισης παρουσιάζεται στο σχήμα 6.2.

Η χρήση ενός μεταμοντέλου κάνει τη δήλωση των ΟΕ: **α) πιο κατανοητή** στο χρήστη ή στο σχεδιαστή, αφού χρησιμοποιούνται οι γνωστοί μηχανισμοί παράστασης γνώσης της SIS-Telos , **β) πιο απλή** εν συγκρίσει με την πολυπλοκότητα της χρήσης του μηχανισμού



Σχήμα 6.1: Δύο διαφορετικές προσεγγίσεις του προβλήματος

Το αριστερό σχήμα παρουσιάζει την χρήση του ΜΔΕ ενώ το δεξιό το προτεινόμενο ΜΜΟΕ



Σχήμα 6.2: Αφηρημένο παράδειγμα χρήσης του ΜΜΟΕ

Οι παχιές γραμμές αντιπροσωπεύουν τις δηλώσεις του ΜΜΟΕ . Μια σχέση δήλωσης συνδέει ένα αντικείμενο (το αντικείμενο δήλωσης) με ένα πλαίσιο το οποίο περικλείει τα ενημερώσιμα αντικείμενα λόγω της δήλωσης. Το αντικείμενο δήλωσης μπορεί να είναι μέσα ή έξω από το πλαίσιο.

ερωτήσεων (*query interface*), **γ) εποπτική** αφού είναι δυνατή η εποπτεία των δηλώσεων που έχουν γίνει μέσω των εργαλείων παρουσίασης του SIS (μηχανισμός προκαθορισμένων ερωτήσεων, γραφικός διερευνητής), και **δ) σταθερότερη** αφού οι ανάγκες **συντήρησης** των δηλώσεων είναι μικρότερες : η μετονομασία ενός αντικειμένου στο οποίο έχει γίνει μια δήλωση, δεν αλλοιώνει την επικαιρότητα της δήλωσης (αφού η ταυτότητα του αντικειμένου δεν εξαρτάται από το λογικό του όνομα), μία αίτηση διαγραφής ενός αντικειμένου γίνεται αποδεκτή από τον Semantic Checker μόνο αφού πρώτα διαγραφούν όλες οι δηλώσεις που έχουν γίνει σε αυτό (όπως και κάθε άλλη σχέση του αντικειμένου). Άρα τελικά η ακεραιότητα των δηλώσεων διατηρείται. Αντίθετα μια δήλωση του ΜΔΕ χάνει την επικαιρότητα της κατά την μετονομασία ή διαγραφή ενός αντικειμένου που αναφέρεται σε ένα *QueryMacro* της.

Μερικά γενικά πλεονεκτήματα της χρήσης μεταμοντέλων παρουσιάζονται στην επόμενη παράγραφο.

6.1.1 Μεταμοντέλα

Η χρήση μεταμοντέλων για την περιγραφή της λειτουργίας ενός ΣΔΒΔ έχει μελετηθεί και μελετείται ([29, 5, 16]) στον ερευνητικό χώρο υπό το όνομα *επεκτάσιμες βάσεις δεδομένων* (extensible database systems) [1].

Για παράδειγμα στην εργασία [16] προτείνεται η χρήση μεταμοντέλων για την επέκταση των λειτουργιών ενός Συστήματος Διαχείρισης Βάσεων Δεδομένων (ΣΔΒΔ). Η πρόταση χρησιμοποιείται στο Οντοκεντρικό ΣΔΒΔ **Adam**. Μια από αυτές τις επεκτάσεις είναι και ο έλεγχος της εξέλιξης γνώσης και η περιγραφή περιορισμών. Οι *δομικοί περιορισμοί* ελέγχονται από το σύστημα αλλά υπάρχει ανάγκη περιγραφής και ελέγχου *στατικών περιορισμών* που περιορίζουν τις καταστάσεις στις οποίες μπορεί να βρίσκεται μια ΒΔ. Αυτοί οι περιορισμοί μπορούν να περιγραφούν από ένα μεταμοντέλο. Έτσι ο έλεγχος και η εφαρμογή των περιορισμών αυτών γίνεται από το ίδιο το ΣΔΒΔ χωρίς να απαιτούνται ιδιαίτεροι έλεγχοι από την εκάστοτε εφαρμογή που χρησιμοποιεί τη βάση. Τα οφέλη είναι η **εύκολη επέκταση** αφού αυτή γίνεται με τη χρήση του μεταμοντέλου και όχι με το γράψιμο κώδικα. Επίσης η **ομοιομορφία** και η **οικονομία υλοποίησης** αφού χρησιμοποιούνται οι ίδιοι μηχανισμοί και εργαλεία για την αποθήκευση, εξιδείκευση και ανάκτηση δεδομένων και μεταδεδομένων.

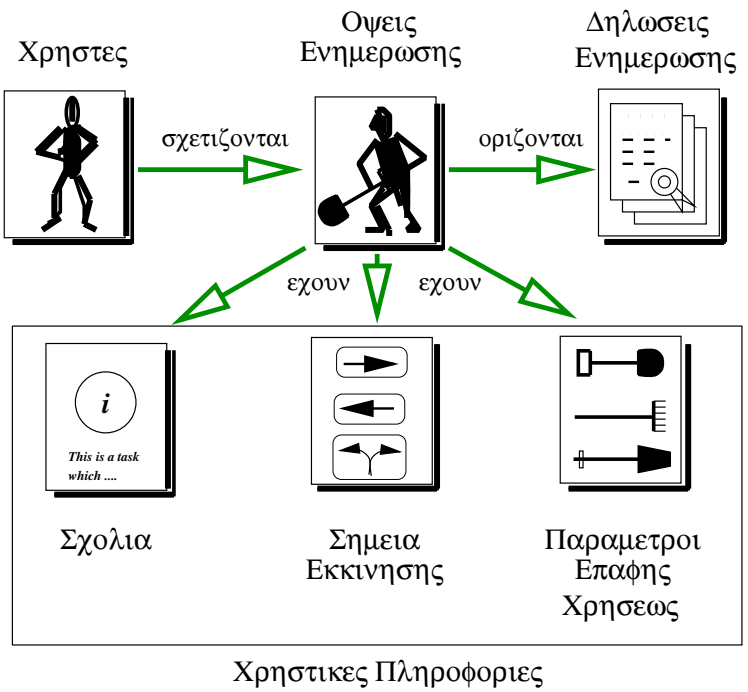
Σχετική είναι και η πρόταση [5] στην οποία προτείνεται ένα μεταμοντέλο για την

[1] Στόχος των *επεκτάσιμων βάσεων δεδομένων* είναι η εύρεση των τεχνικών εκείνων κατασκευής συστημάτων βάσεων δεδομένων ούτως ώστε η λειτουργία τους να διευρύνεται ή τροποποιείται εύκολα. ([28]).

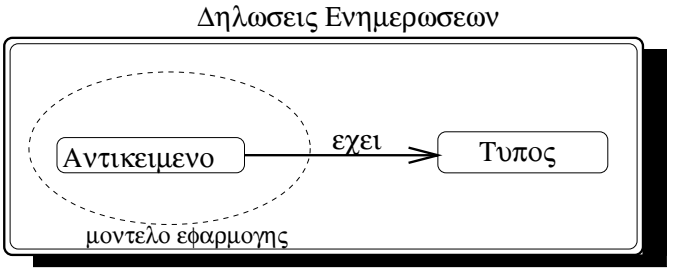
περιγραφή της εξέλιξης γνώσης σε οντοκεντρικά ΣΔΒΔ. Με το προτεινόμενο μεταμοντέλο είναι δυνατή η επέκταση ή τροποποίηση των ίδιων των εκφραστικών μηχανισμών αυτών καθεαυτών.

6.2 Γενική Παρουσίαση του Μεταμοντέλου

Μια επισκόπηση των βασικών εννοιών του ΜΜΟΕ παρουσιάζεται στο σχήμα 6.3, ενώ μία από αυτές, η *Δήλωση Ενημερώσεων*, αναλύεται περισσότερο στο σχήμα 6.4.



Σχήμα 6.3: Οι βασικές έννοιες του ΜΜΟΕ



Σχήμα 6.4: Δηλώσεις ενημέρωσης

Μια ΟΕ αφορά ένα σύνολο **χρηστών** και περιλαμβάνει ένα σύνολο **χρηστικών πληροφοριών** και ένα σύνολο **Δηλώσεων Ενημέρωσης**. Οι χρηστικές πληροφορίες βοηθούν το χρήστη στη διεκπεραίωση των ΣΕΕ που ορίζονται από την ΟΕ, παρέχοντάς του τις απαραίτητες πληροφορίες και καθορίζοντας τις παραμέτρους των εργαλείων που χρησιμοποιεί.

Οι χρηστικές πληροφορίες αναλύονται σε :

- **Σημεία Εκκίνησης**
  Πρόκειται για κάποια ενδεικτικά αντικείμενα της βάσης από τα οποία ο χρήστης ξεκινά την ενημέρωση μιας όψης της βάσης (περισσότερα στην ενότητα 6.5).

- **Σχόλια**
  Πρόκειται για μια άτυπη περιγραφή των περιεχομένων μιας ΟΕ σε μορφή απλού κειμένου. Θα μπορούσε να χρησιμοποιηθεί και μη-γραμμικό κείμενο με αναφορές σε αντικείμενα της βάσης ή άλλες (σχετικές) ΟΕ.

- **Παράμετροι Επαφής Χρήσεως**
  Πρόκειται για ένα σύνολο παραμέτρων οι οποίες καθορίζουν τη λειτουργικότητα των διαλογικών εργαλείων του SIS (predefined queries,graph views, retrieval queries) κατά τέτοιο τρόπο ώστε να βοηθούν το χρήστη όσο το δυνατό περισσότερο στη διεκπεραίωση των ΣΕΕ μιας συγκεκριμένης ΕΕ.

Κάθε ΟΕ ορίζει ένα σύνολο ΣΕΕ με τη βοήθεια των **Δηλώσεων Ενημέρωσης (ΔΕ)**. Όπως φαίνεται και στο σχήμα 6.4, μια δήλωση είναι στην ουσία μια σχέση μεταξύ ενός αντικειμένου της βάσης, ενός **Τύπου** ενημέρωσης και μιας *ΟΕ*. Πρόκειται δηλαδή για μια διατεταγμένη τριάδα :

<αντικείμενο, τύπος, όψη ενημέρωσης>

η οποία αποθηκεύεται στη βάση με τη χρήση των μηχανισμών παράστασης γνώσης της SIS-Telos . Οι τύποι δήλωσης που υποστηρίζονται και η ερμηνεία τους, παρουσιάζονται στις επόμενες ενότητες, ενώ στην ενότητα 6.6 παρουσιάζεται όλο το MMOE σε SIS-Telos .

6.3 Δηλώσεις Ενημέρωσης

Το ουσιαστικότερο μέρος του μεταμοντέλου είναι οι υποστηριζόμενες δηλώσεις ενημέρωσης και **ο τρόπος ερμηνείας** τους, ή αλλιώς, η σημασιολογία τους. Θεωρώντας ότι αναφερόμαστε σε ένα στιγμιότυπο μιας βάσης κάποια χρονική στιγμή, το σύνολο των ΔΕ στη βάση ορίζει τη σχέση D :

$$D \subseteq O \times Type \times TaskIDs$$
$$Type \subseteq ID \times Target \times State$$

η αλλιώς:

$$D \subseteq \{(o, id, sc, st, t) \mid o \in O, id \in ID, sc \in Target, st \in State, t \in TaskIDs\}$$

όπου :

$$\begin{aligned} O &: \text{το σύνολο αντικειμένων της βάσης} \\ TaskIDs &\subset N \\ Target &= \{onObject, onAttrs, onInsts\} \\ State &= \{\text{POS}, \text{NEG}\} \\ ID &= \{CrObj, DelObj, REN, DEL, AddAF, DelAF, AddAT, DelAT, \\ &\quad AddIn, DelIn, AddSub, DelSub, AddClass, DelClass, AddSup, DelSup\} \end{aligned}$$

• **TaskIDs**

Σε κάθε ΟΕ αντιστοιχεί ένας διαφορετικός ακέραιος. Οι ακεραίοι αυτοί ορίζουν το σύνολο αναγνωριστικών ΟΕ :

$$TaskIDs \subset N$$

• **ID**

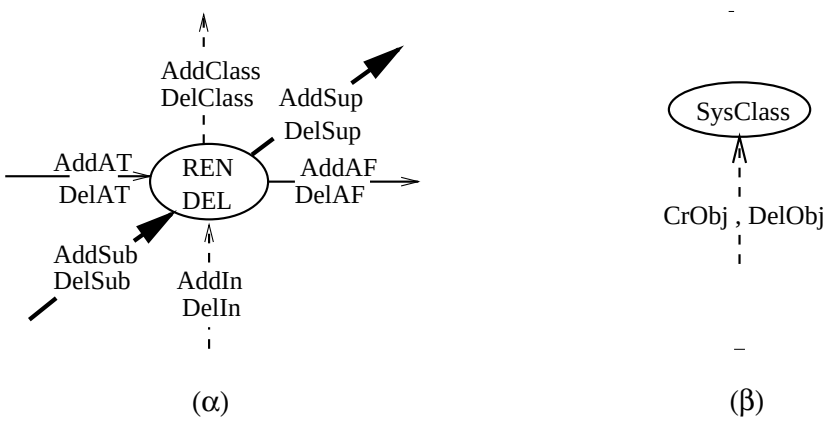
Ορίζουμε το σύνολο Αναγνωριστικών Ενημερώσεων (ΑΕ) ID ως εξής :

$$ID = \{CrObj, DelObj, REN, DEL, AddAF, DelAF, AddAT, DelAT, AddIn, \\ DelIn, AddSub, DelSub, AddClass, DelClass, AddSup, DelSup\}$$

Τα ΑΕ καθορίζουν τις ΣΕΕ που ορίζει μία δήλωση και παρουσιάζονται σχηματικά στο σχήμα 6.5. Για παράδειγμα το ΑΕ $AddClass$ αφορά την ενημέρωση των κλάσεων ενός αντικειμένου, το REN αφορά την μετονομασία ενός αντικειμένου, κ.ο.κ. Τα ΑΕ που παρουσιάζονται στο σχήμα 6.5(β) αφορούν μόνο τις κλάσεις συστήματος και με αυτά ελέγχονται οι ΣΕΕ που δημιουργούν/διαγράφουν αντικείμενα.

• **Target**

Το αντικείμενο στο οποίο έχει γίνει μια δήλωση το ονομάζουμε Εστιακό Αντικείμενο Δήλωσης (ΕΑΔ). Το ΜΜΟΕ επιτρέπει **εστιακές** και **μη εστιακές** δηλώσεις. Οι εστιακές



Σχήμα 6.5: Τα αναγνωριστικά ενημέρωσης

δηλώσεις ορίζουν ΣΕΕ που ‘‘αφορούν’’ άμεσα το ΕΑΔ, στην ουσία ‘‘εκχωρούν’’ το ΑΕ στο ΕΑΔ. Αντίθετα οι **μη-εστιακές δηλώσεις** αφορούν ένα σύνολο αντικειμένων το οποίο ορίζεται έμμεσα και δυναμικά με βάση το ΕΑΔ. Για αυτό κάθε τύπος ενημέρωσης (*Type*) περιέχει ένα στοιχείο του συνόλου $Target = \{onObject, onAttrs, onInsts\}$ το οποίο καθορίζει το πεδίο εφαρμογής της δήλωσης και για τις εστιακές δηλώσεις έχει την τιμή $onObject$, ενώ για τις μη εστιακές μπορεί να πάρει την τιμή $onAttrs$ ή $onInsts$. Οι εστιακές δηλώσεις ορίζουν το σύνολο *FD*[2] :

$$FD = \{(o, id, sc, st, t) \in D \mid sc = onObject\}.$$

ενώ οι μη-εστιακές είναι οι δηλώσεις του συνόλου $D - FD$.

- **State**

Το ΜΜΟΕ για λόγους ευελιξίας υποστηρίζει **θετικές** και **αρνητικές** δηλώσεις (δες σχήμα 6.6). Για αυτό κάθε δήλωση περιέχει ένα ειδικό αναγνωριστικό το οποίο καθορίζει το είδος της. Τα αναγνωριστικά αυτά ορίζουν το σύνολο $State$:

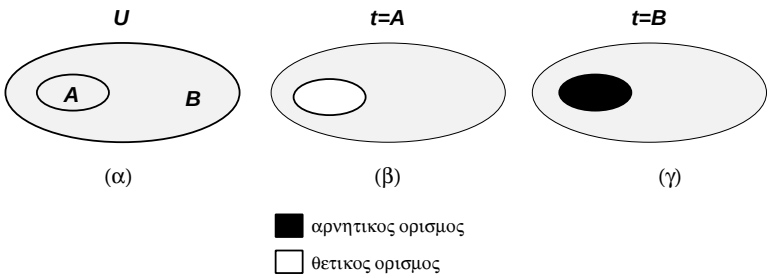
$$State = \{\text{POS}, \text{NEG}\}$$

Ορίζω επίσης τις ακόλουθες σχέσεις :

- Το σύνολο δηλώσεων μιας ΟΕ $t, t \in TaskIDs$:

$$D_t = \{(o, id, sc, st, ta) \in D \mid ta = t\}$$

[2]Η ονομασία προέρχεται από το Focus Declarations.



Σχήμα 6.6: Χρήση θετικών και αρνητικών δηλώσεων

Το σύνολο U συμβολίζει όλες τις ΣΕΕ μιας βάσης, του οποίου μια διαμέριση είναι τα σύνολα A και B. Ο ορισμός μιας ΟΕ t που περιέχει τις ΣΕΕ του συνόλου A γίνεται χρησιμοποιώντας θετικές δηλώσεις (δες διάγραμμα (β)), ενώ ο ορισμός μιας ΟΕ t με τις ΣΕΕ του συνόλου B γίνεται ευκολότερα με χρήση αρνητικών δηλώσεων : $t = U - A$ (δες διάγραμμα (γ)).

- Το σύνολο εστιακών δηλώσεων μιας ΟΕ t :

$$FD_t = \{(o, id, sc, st, ta) \in D \mid sc = onObject \wedge ta = t\}$$

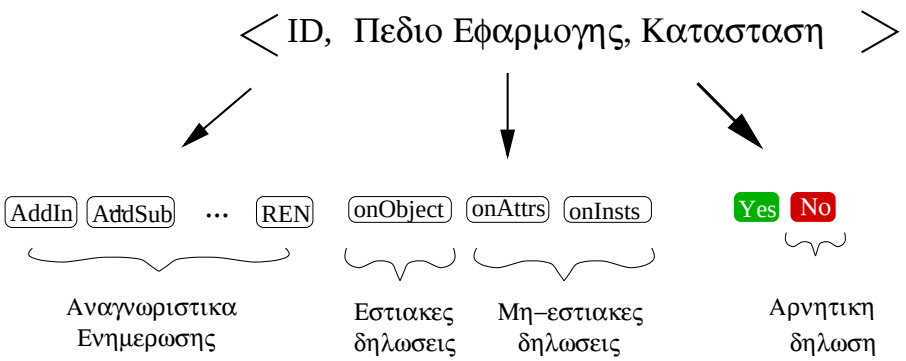
- Το σύνολο θετικών εστιακών δηλώσεων μιας ΟΕ t :

$$FD_t^{\text{POS}} = \{(o, id, sc, st, ta) \in D \mid sc = onObject \wedge st = \text{POS} \wedge ta = t\}$$

- Το σύνολο αρνητικών εστιακών δηλώσεων μιας ΟΕ t :

$$FD_t^{\text{NEG}} = \{(o, id, sc, st, ta) \in D \mid sc = onObject \wedge st = \text{NEG} \wedge ta = t\}$$

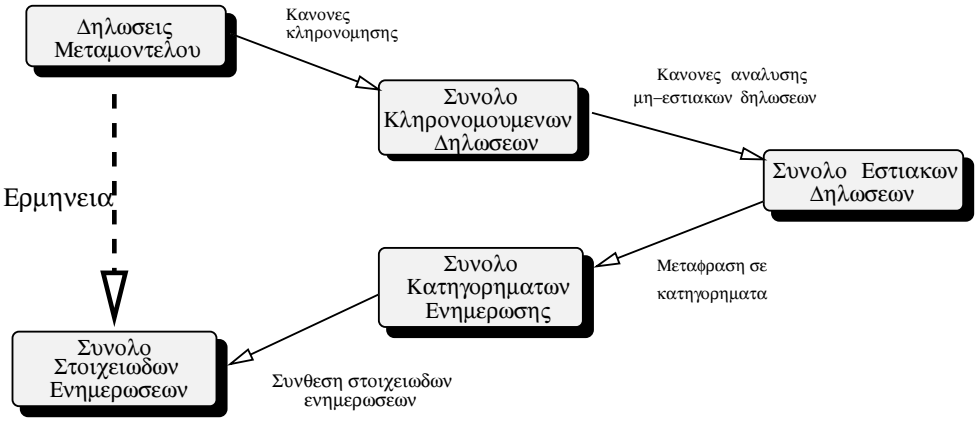
Τα συστατικά ενός τύπου παρουσιάζονται σχηματικά στο σχήμα 6.7.



Σχήμα 6.7: Η μορφή των τύπων δήλωσης του ΜΜΟΕ

6.4 Ερμηνεία Δηλώσεων Ενημέρωσης

Τα βήματα που απαιτούνται για την ερμηνεία των ΔΕ παρουσιάζονται σχηματικά στο σχήμα 6.8 και περιγράφονται αναλυτικά στις επόμενες ενότητες.



Σχήμα 6.8: Τα βήματα ερμηνείας των δηλώσεων ενημέρωσης μιας ΟΕ

Η ανάλυση της ερμηνείας των δηλώσεων σε βήματα επιτρέπει την εύκολη τροποποίηση του μηχανισμού και συμβάλλει στην κατανόηση της.

6.4.1 Βήμα 1 : Κληρονόμηση Δηλώσεων

Το ΜΜΟΕ υποστηρίζει δύο μορφές κληρονόμησης δηλώσεων. Θα ορίσουμε *το σύνολο όλων των δηλώσεων* (κληρονομημένων και μη) D_t^{inh} :

$$D_t^{inh} \subset O \times ID \times Target \times State \times TaskIDs$$

με την βοήθεια των σχέσεων $\xrightarrow{isA}$ και $\xrightarrow{isAsys}$ (οι οποίες είναι δύο δυαδικές σχέσεις στο $O \times ID \times Target \times State \times TaskIDs$) ως εξής :

$$D_t^{inh} = D_t \cup \{d \mid \exists a \in D_t : a \xrightarrow{isA/isAsys} d\}$$

Οι σχέσεις $\xrightarrow{isA}$, $\xrightarrow{isAsys}$ περιγράφονται παρακάτω [3] :

- Σχέση $\xrightarrow{isA}$ (κληρονόμηση δηλώσεων από υπερκλάση)

$$\text{Αν } (o, id, sc, st, t) \in D_t \wedge o \in C \text{ τότε } \forall s \in gasb(o) \text{ ισχύει :}$$
$$(o, id, sc, st, t) \xrightarrow{isA} (s, id, sc, st, t)$$

[3] Οι ερωτηματικές εντολές (του **qi**) που εμφανίζονται στους παρακάτω ορισμούς επεξηγούνται στο παράρτημα C.

- Σχέση $\xrightarrow{isAsys}$ (κληρονόμηση δηλώσεων από κλάσεις συστήματος)

$$\text{Αν } (s,id,sc,st,t) \in D_t \ \wedge \ s \in O_{sys} \text{ τότε } \forall i : R^{in}_{sys}(i) \in gasb(s) \cup \{s\} \text{ ισχύει } :$$
$$(s,id,sc,st,t) \xrightarrow{isAsys} (i,id,sc,st,t)$$

Οι δύο σχέσεις $\xrightarrow{isA}, \xrightarrow{isAsys}$ ενδέχεται να δημιουργήσουν ασάφειες στον ορισμό του D^{inh}_t ως συνέπεια της δυνατότητας χρήσης θετικών και αρνητικών δηλώσεων. Το D^{inh}_t θεωρείται ασαφές αν $\exists \ \{(o,id,s,sc,t),(o,id,\bar{s},sc,t)\} \in D^{inh}_t$. Η μεταβλητή s συμβολίζει μια τιμή από το σύνολο $State$, ενώ το $\bar{s}$ συμβολίζει μια τιμή συμπληρωματική της τιμής της s. Ειδικότερα ασάφεια μπορεί να προκύψει :

1. μεταξύ άμεσων και κληρονομούμενων δηλώσεων :

$$\left.\begin{array}{r} (o,id,sc,s,t) \in D_t \\ \exists a \in D_t : a \xrightarrow{isA/isAsys} (o,id,sc,\bar{s},t) \end{array}\right\} \rightarrow (o,id,sc,\mathbf{?},t) \in D^{inh}_t$$

2. μεταξύ κληρονομούμενων δηλώσεων :

$$\left.\begin{array}{r} \exists a \in D_t : a \xrightarrow{isA/isAsys} (o,id,sc,s,t) \\ \exists b \in D_t : b \xrightarrow{isA/isAsys} (o,id,sc,\bar{s},t) \end{array}\right\} \rightarrow (o,id,sc,\mathbf{?},t) \in D^{inh}_t$$

Δηλώσεις που προκαλούν ασάφειες επιτρέπονται αλλά χρησιμοποιείται το παρακάτω σύνολο κανόνων για την επίλυση τους, ώστε να προκύψει ένας σαφής ορισμός του D^{inh}_t. Οι παρακάτω κανόνες βασίζονται στη γενική πολιτική να θεωρούμε ισχυρότερη μια δήλωση που εκφράζει ειδικότερη γνώση. Μια δήλωση Α εκφράζει ειδικότερη γνώση από μια δήλωση Β, αν το σύνολο των αντικειμένων που αφορά η Α είναι υποσύνολο των αντικειμένων που αφορά η Β.

1. Ασάφεια μεταξύ άμεσων και κληρονομούμενων δηλώσεων :

$$\left.\begin{array}{r} (o,id,sc,s,t) \in D_t \\ \exists a : a \xrightarrow{isA/isAsys} (o,id,sc,\bar{s},t) \end{array}\right\} \rightarrow (o,id,sc,s,t) \in D^{inh}_t$$

   Η άμεση δήλωση υπερισχύει.

2. Ασάφεια μεταξύ κληρονομούμενων δηλώσεων από υπερκλάσεις:

$$\left.\begin{array}{r} (a,id,sc,s,t) \xrightarrow{isA} (o,id,sc,s,t) \\ (b,id,sc,\bar{s},t) \xrightarrow{isA} (o,id,sc,\bar{s},t) \end{array}\right\} \rightarrow \left\{\begin{array}{ll} (o,id,sc,s,t) \in D^{inh}_t & \text{αν } a \ isA \ b \\ (o,id,sc,\bar{s},t) \in D^{inh}_t & \text{αν } b \ isA \ a \\ (o,id,sc,\text{NEG},t) \in D^{inh}_t & \text{αλλιώς} \end{array}\right.$$

Σε αυτού του είδους τις ασάφειες υπερισχύει η δήλωση που κληρονομείται από την ειδικότερη κλάση. Αυτός ο κανόνας επίλυσης ασαφειών αναφέρεται στην βιβλιογραφία με τον όρο *inferential distance ordering* ([45], [41]) και χρησιμοποιείται συχνά για την εύρεση των ιδιοτήτων που κληρονομούνται, σε συστήματα που υποστηρίζουν θετικές και αρνητικές ιδιότητες.

Σε περίπτωση που δεν υπάρχει μοναδική ειδικότερη κλάση, τότε υπερισχύει η αρνητική δήλωση. Πρόκειται για έναν απλό κανόνα ο οποίος εφαρμόζεται όταν δεν μπορούμε να εφαρμόσουμε τον προηγούμενο, δηλαδή δεν μπορούμε να χαρακτηρίσουμε τη μια δήλωση ειδικότερη της άλλης. Τέτοιες περιπτώσεις εμφανίζονται σε συστήματα που υποστηρίζουν πολλαπλή κληρονόμηση (όπως η Telos).

Θα γίνει τώρα μια σύντομη αναφορά στον τρόπο με τον οποίο παρόμοια προβλήματα αντιμετωπίζονται από άλλα συστήματα. Στο σύστημα ORION ([36]) κάθε δήλωση χαρακτηρίζεται από το χρήστη εξαρχής ως *ισχυρή* ή *ασθενής*. Μια ισχυρή δήλωση είναι ισχυρότερη από κάθε ασθενή. Το σύστημα απαγορεύει τη δημιουργία ασαφειών (επικαλυπτόμενες και αντίθετες ισχυρές ή επικαλυπτόμενες και αντίθετες ασθενείς) για αυτό και διενεργεί τους κατάλληλους ελέγχους. Όπως καταλαβαίνουμε ένα μόνο ‘‘φώλιασμα’’ δηλώσεων είναι εφικτό. Συγκρίνοντας τους κανόνες αυτούς με τους κανόνες του MMOE μπορούμε να πούμε ότι οι κανόνες του MMOE είναι απλούστεροι, εκφραστικότεροι και επιλύουν κάθε ασάφεια που προκύπτει για το λόγο αυτό δεν απαιτείται η διενέργεια ειδικών ελέγχων.

3. Ασάφεια μεταξύ κληρονομούμενων δηλώσεων από κλάσεις συστήματος:

$$\left.\begin{array}{l}(p,id,sc,s,t) \overset{isAsys}{\longrightarrow} (o,id,sc,s,t)\\ (r,id,sc,\bar{s},t) \overset{isAsys}{\longrightarrow} (o,id,sc,\bar{s},t)\end{array}\right\} \rightarrow \begin{cases}(o,id,sc,s,t) \in D_t^{inh} & \text{αν } p\ isA\ r\\ (o,id,sc,\bar{s},t) \in D_t^{inh} & \text{αν } r\ isA\ p\\ (o,id,sc,\text{NEG},t) \in D_t^{inh} & \text{αλλιώς}\end{cases}$$

Υπερισχύει η δήλωση που κληρονομείται από την ειδικότερη κλάση συστήματος στην οποία ανήκει. Αν δεν υπάρχει μοναδική ειδικότερη, υπερισχύει η αρνητική δήλωση.

Πρόκειται για τους κανόνες που επεξηγήθηκαν προηγουμένως, που απλά εφαρμόζονται σε κλάσεις συστήματος.

4. Ασάφειες λόγω κληρονόμησης δηλώσεων από υπερκλάση και από κλάση συστήματος :

$$\left.\begin{array}{l}(a, id, sc, s, t) \xrightarrow{isA} (o, id, sc, s, t) \\ (s, id, sc, \bar{s}, t) \xrightarrow{isAsys} (o, id, sc, \bar{s}, t)\end{array}\right\} \rightarrow (o, id, sc, s, t) \in D_t^{inh}$$

Σε αυτού του είδους την ασάφεια υπερισχύει η δήλωση που κληρονομείται από υπερκλάση.

6.4.2 Βήμα 2 : Ανάλυση Μη-Εστιακών Δηλώσεων

Όπως προαναφέραμε μια μη εστιακή δήλωση αφορά ένα σύνολο αντικειμένων το οποίο ορίζεται έμμεσα και δυναμικά με βάση το ΕΑΔ. Στο βήμα αυτό, από το σύνολο των κληρονομημένων δηλώσεων, D_t^{inh}, θα ορίσουμε το σύνολο *όλων*[4] *των εστιακών δηλώσεων*, FD_t^{all} :

$$FD_t^{all} \subset O \times ID \times Target \times State \times TaskIDs$$

Το σύνολο αυτό αποτελείται από το σύνολο των εστιακών δηλώσεων του D_t^{inh}, δηλαδή το FD_t^{inh} :

$$FD_t^{inh} = \{(o, id, sc, st, t) \in D_t^{inh} \mid sc = onObject\}$$

και ένα σύνολο από δηλώσεις που προκύπτουν *έμμεσα* από την ανάλυση των μη-εστιακών δηλώσεων. Η ανάλυση αυτή γίνεται με τη βοήθεια των σχέσεων $\xrightarrow{onAttrs}$, $\xrightarrow{onInsts}$. Ειδικότερα :

$$FD_t^{all} = FD_t^{inh} \cup \{d \mid \exists a \in D_t^{inh} : a \xrightarrow{onAttrs/onInsts} d\}$$

Οι σχέσεις $\xrightarrow{onAttrs}$ και $\xrightarrow{onInsts}$ περιγράφονται παρακάτω :

- Σχέση $\xrightarrow{onAttrs}$:

$$\text{Αν } (o, id, onAttrs, s, t) \in D_t^{inh} \text{ τότε } \forall l \in gilf(o) \text{ ισχύει} :$$
$$(o, id, onAttrs, s, t) \xrightarrow{onAttrs} (l, id, onObject, s, t)$$

- Σχέση $\xrightarrow{onInsts}$:

$$\text{Αν } (o, id, onInsts, s, t) \in D_t^{inh} \text{ τότε } \forall i \in gi(o) \text{ ισχύει:}$$
$$(o, id, onInsts, s, t) \xrightarrow{onInsts} (i, id, onObject, s, t)$$

[4] Άμεσων και έμμεσων

Οι δύο σχέσεις $\overset{onAttrs}{\longrightarrow}$, $\overset{onInsts}{\longrightarrow}$ ενδέχεται να δημιουργήσουν ασάφειες στον ορισμού του FD_t^{all}. Το FD_t^{all} είναι ασαφές αν $\exists \{(o, id, onObect, s, t), (o, id, onObect, \bar{s}, t)\} \in FD_t^{all}$. Δηλώσεις που προκαλούν ασάφειες επιτρέπονται αλλά χρησιμοποιείται το παρακάτω σύνολο κανόνων για την επίλυσης τους. Στους ακόλουθους κανόνες επίλυσης ασαφειών ισχύουν οι ακόλουθοι συμβολισμοί: $a, b \in D_t^{inh}, d \in FD_t$, και $f = (o, id, onObject, s, t)$ και $g = (o, id, onObject, \bar{s}, t)$.

Οι παρακάτω κανόνες βασίζονται επίσης στη γενική πολιτική να θεωρούμε ισχυρότερη μια δήλωση που εκφράζει ειδικότερη γνώση. Υπενθυμίζουμε ότι μια δήλωση Α εκφράζει ειδικότερη γνώση από μια δήλωση Β, αν το σύνολο των αντικειμένων που αφορά η Α είναι υποσύνολο των αντικειμένων που αφορά η Β.

Ασάφεια μεταξύ :

- άμεσης εστιακής και μη-εστιακής *onInsts*

$$\left.\begin{array}{r} f \in FD_t \\ a \overset{onInsts}{\longrightarrow} g \end{array}\right\} \rightarrow f \in FD_t^{all}$$

- άμεσης εστιακής και μη-εστιακής *onAttrs*

$$\left.\begin{array}{r} f \in FD_t \\ a \overset{onAttrs}{\longrightarrow} g \end{array}\right\} \rightarrow f \in FD_t^{all}$$

- μη-εστιακής *onInsts* και μη-εστιακής *onAttrs*

$$\left.\begin{array}{r} a \overset{onInsts}{\longrightarrow} f \\ b \overset{onAttrs}{\longrightarrow} g \end{array}\right\} \rightarrow f \in FD_t^{all}$$

  Σε αυτήν την περίπτωση δεν μπορούμε να χαρακτηρίσουμε μια δήλωση ως ειδικότερη της άλλης. Η επιλογή της *onInsts* ως ισχυρότερη μας ικανοποιεί περισσότερο (σχετικά παραδείγματα παρουσιάζονται στην ενότητα 6.7.5) και μπορεί να δικαιολογηθεί αν θεωρήσουμε ότι η σχέση (ταξινόμησης) ενός γνωρίσματος με μια κλάση του είναι ισχυρότερη από τη σχέση του με την οντότητα στην οποία έχει αποδοθεί.

- δύο μη-εστιακών *onAttrs*

$$\left.\begin{array}{r} a \overset{onAttrs}{\longrightarrow} f \\ b \overset{onAttrs}{\longrightarrow} g \end{array}\right\} \rightarrow f/g \text{ ανάλογα με την κλάση εξέτασης}$$

Η ασάφεια αυτή προκύπτει λόγω της κληρονόμησης των γνωρισμάτων της SIS-Telos και επιλύεται υπέρ της δήλωσης που υπερισχύει στην κλάση από την οποία χρησιμοποιείται το γνώρισμα (κλάση εξέτασης). Σχετικό παράδειγμα παρουσιάζεται στην ενότητα 6.7.4.

- δύο μη-εστιακών *onInsts*

$$\left.\begin{array}{r} a \xrightarrow{onInsts} f \\ b \xrightarrow{onInsts} g \end{array}\right\} \rightarrow (o, id, onObject, \text{NEG}, t) \in FD_t^{all}$$

Ο κανόνας αυτός, η επιλογή δηλαδή της αρνητικής δήλωσης ως ισχυρότερη, γίνεται γιατί καμία από τις δύο δηλώσεις δεν είναι ειδικότερη της άλλης (παρόμοιος κανόνας περιγράφηκε για την κληρονόμηση των δηλώσεων).

- κληρονομούμενης από υπερκλάση και μη-εστιακής *onAttrs*

$$\left.\begin{array}{r} d \xrightarrow{isA} f \\ b \xrightarrow{onAttrs} g \end{array}\right\} \rightarrow f \in FD_t^{all}$$

- κληρονομούμενης από κλάση συστήματος και μη-εστιακής *onAttrs*

$$\left.\begin{array}{r} d \xrightarrow{isAsys} f \\ b \xrightarrow{onAttrs} g \end{array}\right\} \rightarrow g \in FD_t^{all}$$

- κληρονομούμενης από υπερκλάση και μη-εστιακής *onInsts*

$$\left.\begin{array}{r} d \xrightarrow{isA} f \\ b \xrightarrow{onInsts} g \end{array}\right\} \rightarrow f \in FD_t^{all}$$

- κληρονομούμενης από κλάση συστήματος και μη-εστιακής *onInsts*

$$\left.\begin{array}{r} d \xrightarrow{isAsys} f \\ b \xrightarrow{onInsts} g \end{array}\right\} \rightarrow g \in FD_t^{all}$$

Ορίζουμε τις έννοιες της *πληρότητας* και της *σαφήνειας* στο FD_t^{all}, αφού πρώτα εισάγουμε τους συμβολισμούς :

$$FD_{t^{\text{POS}}}^{all} = \{(o, id, sc, st, t) \in FD_t^{all} \mid st = \text{POS}\}$$
$$FD_{t^{\text{NEG}}}^{all} = \{(o, id, sc, st, t) \in FD_t^{all} \mid st = \text{NEG}\}$$

- **Πληρότητα**

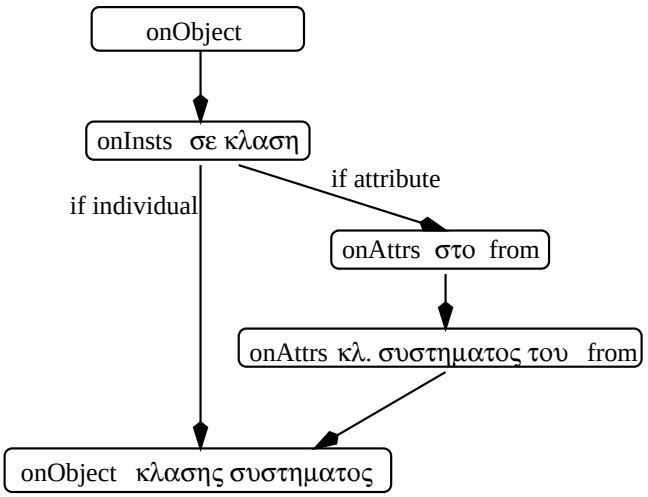
$$\forall o \in O, \forall id \in ID \text{ πρέπει:} \quad (o, id) \in FD^{all}_{t^{\text{POS}}} \cup FD^{all}_{t^{\text{NEG}}}$$

- **Σαφήνεια**

$$FD^{all}_{t^{\text{POS}}} \cap FD^{all}_{t^{\text{NEG}}} = \emptyset$$

Η δήλωση μιας ΟΕ πρέπει να είναι πλήρης και σαφής. Αυτό σημαίνει ότι το FD^{all}_t πρέπει να είναι πλήρες και σαφές. Η σαφήνεια διασφαλίζεται μέσω των κανόνων επίλυσης ασαφειών που παρουσιάστηκαν. Η πληρότητα μπορεί να επιτευχθεί εύκολα κάνοντας μια δήλωση για κάθε $id \in ID$ στην κλάση συστήματος *Telos_Object*. Οι δηλώσεις αυτές κληρονομούνται σε κάθε αντικείμενο της βάσης μέσω της σχέσης $\xrightarrow{isAsys}$ αφού η *Telos_Object* είναι κλάση συστήματος κάθε αντικειμένου της SIS-Telos .

Ένα σχήμα το οποίο παρουσιάζει συνοπτικά τους παραπάνω κανόνες επίλυσης ασαφειών παρουσιάζεται στο σχήμα 6.9.



Σχήμα 6.9: Διάταξη δηλώσεων με βάση την ισχύ τους

6.4.3 Βήμα 3 : Μετάφραση σε Κατηγορήματα

Συχνά στην Telos χρησιμοποιείται ο μηχανισμός της ταξινόμησης για την περιγραφή αντικειμένων από πολλές απόψεις, καθώς και για την απόδοση σύνθετων γνωρισμάτων. Η ανάγκη ορισμού ΟΕ που ορίζουν την ενημέρωση συνδέσμων ταξινόμησης αυτού του είδους, απαιτεί την εισαγωγή στην Telos μιας ειδικής κατηγορίας γνωρισμάτων, της κατηγορίας *relatedClasses*. Η ιδέα αυτή παρουσιάζεται εκτενώς στην ενότητα 6.7.6, συνοδευόμενη από παραδείγματα.

Προκειμένου λοιπόν να εισάγουμε κάποιους επιπλέον κανόνες που αφορούν την ταξινόμηση (και άρα τα αναγνωριστικά ενημέρωσης $AddIn, DelIn$) θα προσθέσουμε ένα βήμα στη διαδικασία ερμηνείας των δηλώσεων, την μετάφραση σε κατηγορήματα.

Ορίζουμε το σύνολο $Predicates \subseteq (O \times ID) \cup (O \times O \times ID)$ ή καλύτερα :

$$Predicates = \{< o, id > | \ id \in \{CrObj, DelObj, REN, DEL, AddAF, DelAF, AddAT, DelAT, AddSub, DelSub, AddSup, DelSup, AddClass, DelClass\}\}$$
$$\cup \{< a, b, id > | \ id \in \{AddIn, DelIn\}\}$$

Για κάθε ΟΕ t θα ορίσουμε το σύνολο ; $Pred_t$ ($Pred_t \subset Predicates$) με τη βοήθεια μιας συνάρτησης $pred$:

$$pred : 2^{O \times ID \times Target \times State \times TaskIDs} \rightarrow 2^{Predicates}$$

ως εξής :

$$Pred_t = pred(FD_t^{all})$$

Η $pred$ ορίζεται με τη βοήθεια των παρακάτω κανόνων :

1. Αν$(o, id, onObject, \text{POS}, t) \in FD_t^{all}$ και $id \in ID - \{AddIn, DelIn\}$ τότε
   $< o, id > \in Pred_t$

2. Αν$(c, id, onObject, \text{POS}, t) \in FD_t^{all}$ και $id \in \{AddIn, DelIn\}$ τότε
   $< o, c, id > \in Pred_t \ \forall o \in O$

3. Αν $\exists \ categ \in gilf(gc(a)) \wedge$
   $relatedClasses \in gc(categ) \wedge$
   $(categ, id, onObject, \text{POS}, t) \in FD_t^{all} \wedge$
   $a \in I, b \in gasb(gtv(categ)) \wedge$
   $id \in \{AddIn, DelIn\}$ τότε $< a, b, id > \in Pred_t$

4. Αν $\exists \ categ \in gilf(gc(a)) \wedge$
   $relatedClasses \in gc(categ) \wedge$
   $(categ, id, onObject, \text{NEG}, t) \in FD_t^{all} \wedge$
   $a \in I, b \in gasb(gtv(categ)) \wedge$
   $id \in \{AddIn, DelIn\}$ τότε $< a, b, id > \notin Pred_t$

Ο αλγόριθμος που επιλύει τις ασάφειες που δημιουργούνται από τους κανόνες 2,3 και 4, παρουσιάζεται στην ενότητα 6.9.2.

6.4.4 Βήμα 4 : Σύνθεση Στοιχειωδών Ενημερώσεων

Από το σύνολο $Pred_t$ θα ''συνθέσουμε'' το σύνολο των ΣΕΕ που ορίζει μια ΟΕ, το οποίο θα το συμβολίζουμε U_t ($U_t \subseteq U_{stx}$, όπου U_{stx} είναι το σύνολο των συντακτικά ορθών ΣΕΕ της βάσης). Το σύνολο αυτό ορίζεται με τη βοήθεια της συνάρτησης $updatePred$ ως εξής :

$$U_t = \{u \in U_{stx} \mid updatePred(u) \in Pred_t\}$$

Η συνάρτηση $updatePred$ αντιστοιχεί σε κάθε στοιχειώδη εντολή ενημέρωσης ένα ή περισσότερα κατηγορήματα (μέλη του $Predicates$) :

$$updatePred : U_{stx} \rightarrow 2^{Predicates}$$

Η συνάρτηση αυτή περιγράφεται από στον παρακάτω πίνακα :

$u \in U_{stx}$	$updatePred(u)$
CreateIndividual n, S	$< S, CrObj >$
Rename o , $newname$	$< o , REN >$
DeleteIndividual o	$< o , DEL >, < R^{in}_{sys}(o), DelObj >$
CreateAttribute $from$, to , n, S	$< from , AddAF >, < to , AddAT >, < S, CrObj >$
DeleteAttribute o , $from$, to , n, S	$< o , DEL >, < from , DelAF >,$ $< to , DelAT >, < S, DelObj >$
AddInstance a , b	$< b , a , AddIn >, < b , AddClass >$
DeleteInstance a , b	$< b , a , DelIn >, < b , DelClass >$
AddSubClass a , b	$< a , AddSub >, < b , AddSup >$
DeleteSubClass a , b	$< a , DelSub >, < b , DelSup >$

Διαισθητικά μια συγκεκριμένη ΣΕΕ u ανήκει σε μία ΟΕ t, αν όλα τα κατηγορήματα της ($updatePred(u)$) ανήκουν στο σύνολο κατηγορημάτων της ΟΕ ($Pred_t$).

6.5 Σημεία Εκκίνησης

Η δημιουργία μιας ατομικής οντότητας συνήθως γίνεται αφού πρώτα εντοπισθεί η κλάση της [5]. Η εύρεση της κατάλληλης κλάσης από το χρήστη δεν είναι εύκολο πράγμα,

[5]Με τον όρο κλάση εννοούμε μια κλάση που έχει ορισθεί από το χρήστη : $O_u \cap C$

όταν μάλιστα το μοντέλο της βάσης είναι μεγάλο (όπως συμβαίνει για παράδειγμα στο σύστημα *ΚΛΕΙΩ*).

Για το λόγο αυτό το ΜΜΟΕ έχει τη δυνατότητα παράστασης και αποθήκευσης των πιθανών κλάσεων νέων οντοτήτων . Στο σύνολο αυτό δεν ανήκουν όλες οι κλάσεις του μοντέλου αφού :

- Ορισμένες κλάσεις χρησιμοποιούνται για την απόδοση σύνθετων γνωρισμάτων σε οντότητες (πχ οι κλάσεις *Style, Period*) και όχι για την αρχική ταξινόμηση μιας οντότητας.

- Ορισμένες κλάσεις χρησιμοποιούνται για την ταξινόμηση των ‘‘ενδιάμεσων ’’ ή βοηθητικών οντοτήτων που συχνά απαιτούνται για την περιγραφή μιας σύνθετης οντότητας (πχ οι κλάσεις *Creation, Date, Acquisition, SimpleName, FullName*)

- Ορισμένες κλάσεις χρησιμοποιούνται για την ταξινόμηση των τιμών των γνωρισμάτων (πχ οι κλάσεις *Weight, Distance, Appearance, Action, SimpleName, FullName, Part, Place*).

- Ορισμένες κλάσεις αποτελούν γενικεύσεις άλλων κλάσεων και δεν δημιουργήθηκαν για την άμεση ταξινόμηση οντοτήτων, αλλά για οικονομία κατά την ανάπτυξη του μοντέλου.

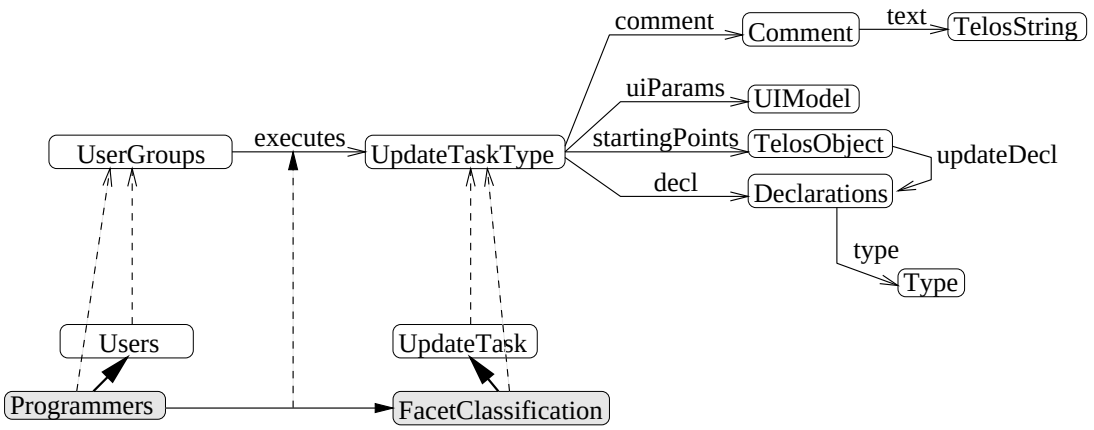
Μια κλάση η οποία χρησιμοποιείται για την αρχική ταξινόμηση νέων οντοτήτων θα την ονομάζουμε *Κλάση Δημιουργίας Οντοτήτων (ΚΔΟ).* Κάθε υποκλάση μιας ΚΔΟ θεωρείται και αυτή ΚΔΟ. Για παράδειγμα στο σύστημα *ΚΛΕΙΩ* μπορούμε να θεωρήσουμε ΚΔΟ τις : *MuseumObject, ArtificialObject, ConceptualObject, Inscription (isA ArtificialObject), Group, Collection, Event, Place, Person, Tool, Part.*

Επίσης το ΜΜΟΕ μπορεί να παραστήσει *Κλάσεις Εξειδίκευσης.* Πρόκειται για τις κορυφαίες υπερκλάσεις ιεραρχιών που εξελίσσονται και χρησιμοποιούνται για την καθοδήγηση του χρήστη.

6.6 Οντολογία του ΜΜΟΕ

Στην ενότητα αυτή παρουσιάζεται το ΜΜΟΕ όπως αυτό μοντελοποιήθηκε σε SIS-Telos . Ο τρόπος μοντελοποίησης δεν είναι μοναδικός. Οι εναλλακτικές προτάσεις αξιολογήθηκαν με βάση κάποιους μη λειτουργικούς παράγοντες όπως η ευκολία (συντομία) ορισμού, η εποπτικότητα των δηλώσεων με τα εργαλεία του SIS, και η δυνατότητα γρήγορης απάντησης των ερωτήσεων που απαιτούνται για την αξιοποίηση των ΟΕ.

Στο σχήμα 6.10 παρουσιάζεται το κύριο μέρος του ΜΜΟΕ .



Σχήμα 6.10: Το ΜΜΟΕ σε SIS-Telos

Ενδεικτικά παρουσιάζονται με σκούρο χρώμα κάποιες περιπτώσεις των κλάσεων των ΜΜΟΕ .

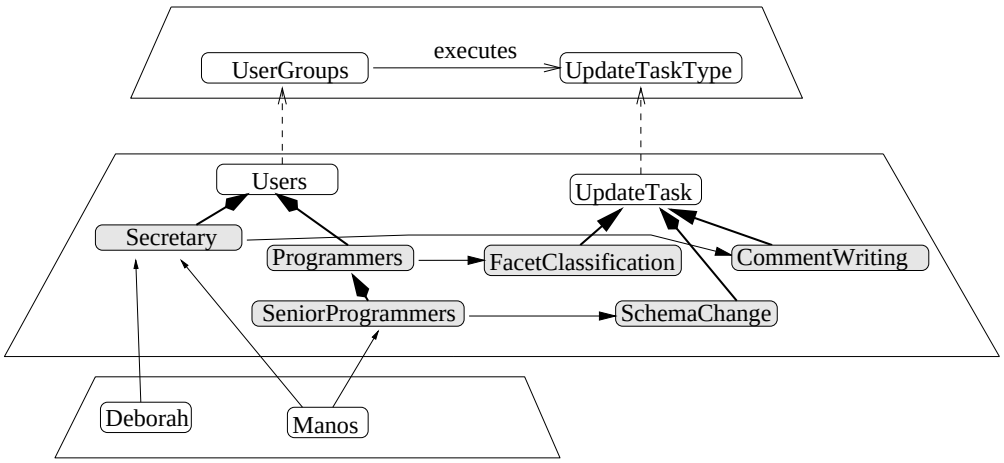
- **Χρήστες**

Οι πραγματικοί χρήστες οργανώνονται σε σύνολα (groups). Οι ΟΕ εκχωρούνται σε σύνολα χρηστών (δες σχήμα 6.11) . Ένας χρήστης μπορεί να ανήκει σε πολλά σύνολα και να κληρονομεί τις σχετικές ΟΕ. Επίσης ένα σύνολο χρηστών μπορεί να δηλωθεί σαν εξειδίκευση ενός ή περισσοτέρων συνόλων. Κατά αυτόν τον τρόπο το σύνολο αυτό κληρονομεί τις ΟΕ των υπερσυνόλων του

- **UIModel**

Μια ΟΕ συνδέεται με το *UIModel*. Το μοντέλο αυτό καθορίζει τη λειτουργικότητα της διαλογικής επαφής χρήσεως του SIS (*predefined queries, graph views, retrieval queries*).

- **Σημεία εκκίνησης**



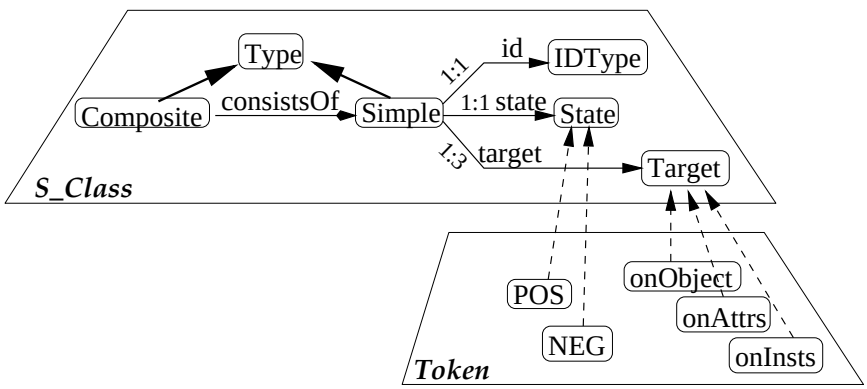
Σχήμα 6.11: Σύνολα Χρηστών

Τα σύνολα χρηστών μοντελοποιήθηκαν σαν μια ιεραρχία εξειδίκευσης. Έτσι είναι δυνατή η δημιουργία υποομάδων οι οποίες κληρονομούν τις ΟΕ που έχουν εκχωρηθεί στις υπερκλάσεις τους.

Η κατηγορία συνδέσμων *startingPoints* χρησιμοποιείται για να υποδείξει τα σημεία εκκίνησης μιας ΟΕ. Η κατηγορία αυτή μπορεί να αναλυθεί σε δύο : *κλάσεις δημιουργίας οντοτήτων* και *κλάσεις εξειδίκευσης*.

- **Types**

Ο τρόπος παράστασης των *τύπων δήλωσης* παρουσιάζεται στο σχήμα 6.12. Η κλάση των απλών τύπων έχει τρία γνωρίσματα *id*, *state* και *target*. Όπως παρατηρούμε παρέχεται η δυνατότητα παράστασης σύνθετων τύπων οι οποίοι αποτελούνται από απλούς.



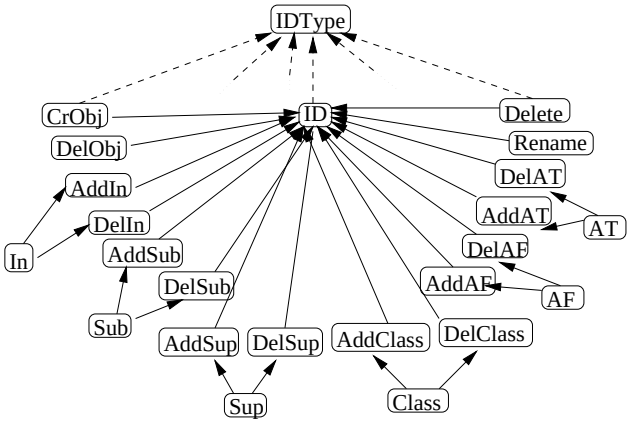
Σχήμα 6.12: Οι Τύποι Δήλωσης του ΜΜΟΕ

• **Target & State**

Τα αναγνωριστικά κατάστασης *State* και πεδίου εφαρμογής *Target*, έχουν μοντελοποιηθεί σαν δύο κλάσεις οντοτήτων με περιπτώσεις τα μέλη των συνόλων αυτών (δες σχήμα 6.12).

• **IDs**

Ο τρόπος μοντελοποίησης των *Αναγνωριστικών Ενημέρωσης* (ID) παρουσιάζεται στο σχήμα 6.13. Παρατηρούμε ότι έχουν παρασταθεί κάποια αναγνωριστικά ως υποκλάσεις κάποιων άλλων. (*In, Sub,Class, Sup*) με τη χρήση της σχέσης isA. Μια δήλωση με αναγνωριστικό *In* ισοδυναμεί με δύο δηλώσεις με αναγνωριστικά $AddIn, DelIn$ αντίστοιχα.



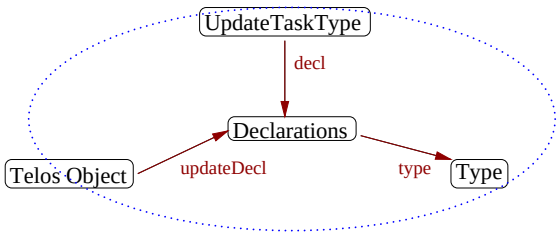
Σχήμα 6.13: Τα Αναγνωριστικά Ενημέρωσης

• **Δηλώσεις**

Οι δηλώσεις αποτελούν μια τριαδική σχέση : <TelosObject, Type, UpdateTaskType>. Ο τρόπος μοντελοποίησης της σχέσεως αυτής παρουσιάζεται καλύτερα στο σχήμα 6.14. Όπως παρατηρούμε στην κλάση *TelosObject* έχει αποδοθεί η κατηγορία γνωρισμάτων *updateDecl* ώστε να μπορεί να χρησιμοποιηθεί από κάθε αντικείμενο της βάσης.

6.7 Παραδείγματα Χρήσης

Ακολουθεί μια σειρά παραδειγμάτων δηλώσεων ενημέρωσης που παρουσιάζουν τον



Σχήμα 6.14: Μοντελοποίση των δηλώσεων

τρόπο χρήσης των τύπων του μεταμοντέλου.

Τα παραδείγματα των επόμενων ενοτήτων παρουσιάζονται σχηματικά για να είναι πιο κατανοητά. Οι δηλώσεις συμβολίζονται βάσει των ακόλουθων συμβάσεων. Θεωρούμε ότι οι δηλώσεις που παρουσιάζονται ανήκουν σε μια ΟΕ για αυτό παραλείπεται το μέλος αυτό των δηλώσεων. Κάθε δήλωση συμβολίζεται με ένα κουτάκι το οποίο συνδέεται με το αντικείμενο δήλωσης .Μέσα σε κάθε κουτάκι υπάρχει ένα ID. Αν πρόκειται για αρνητική δήλωση το ID είναι γραμμένο σε σκούρο φόντο. Αν η δήλωση είναι μη-εστιακή τότε δίπλα στο ID αναγράφεται το πεδίο εφαρμογής της δήλωσης (*Target*).

6.7.1 Δηλώσεις IN

Το αναγνωριστικό **IN** συμβολίζει το ζευγάρι αναγνωριστικών {AddIn, DelIn}. Παράδειγμα **θετικής** δήλωσης παρουσιάζεται στο σχήμα 6.15 (α) όπου η εμφανιζόμενη δήλωση ορίζει τις ΣΕΕ που προσθέτουν/διαγράφουν περιπτώσεις στην κλάση *Άνθρωπος*.

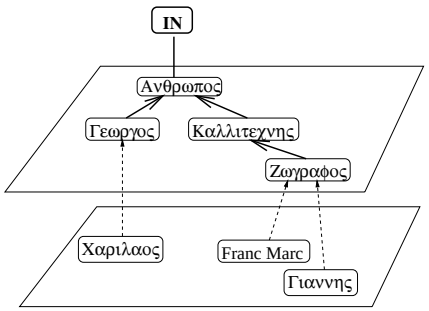
Παράδειγμα **αρνητικής** δήλωσης παρουσιάζεται στο σχήμα 6.15(β) όπου δηλώνεται η απαγόρευση αλλαγής του πληθυσμού της κλάσεως *Φύλο*. Σημειώνουμε εδώ ότι η απαγόρευση αλλαγής του πληθυσμού μιας κλάσεως δηλώνεται είτε γιατί η κλάση αυτή μοντελοποιεί ένα μη μεταβαλλόμενο πληθυσμό του πραγματικού κόσμου (πχ οι κλάσεις *Φύλο*, *Μήνας*, *Ημέρα*, *Μέγεθος*, *Χρώμα*) είτε για να περιορίσει τις ΣΕΕ που αφορούν μια ΟΕ.

Η δήλωση του σχήματος 6.16 κληρονομείται σε όλες τις υποκλάσεις της κλάσης *Άνθρωπος* και άρα ορίζει τις αντίστοιχες ΣΕΕ προσθήκης/διαγραφής περιπτώσεων σε αυτές.

Παράδειγμα κληρονομούμενης αρνητικής δήλωσης παρουσιάζεται στο σχήμα 6.17. Η δήλωση αυτή αποτρέπει αλλαγή στις περιπτώσεις της κλάσεως *Εμφάνιση* και σε όλες τις υποκλάσεις αυτής. Η σκοπιμότητα μιας τέτοιας δήλωσης είναι η διασφάλιση του βαθμού

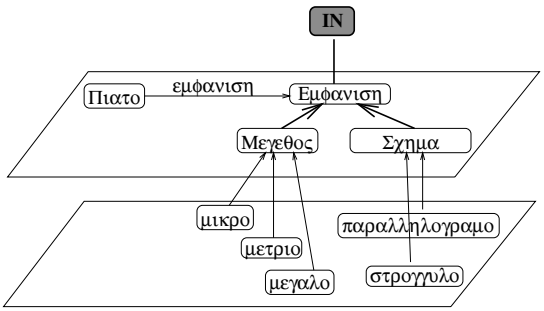
(α) (β)

Σχήμα 6.15: Δήλωση IN σε οντότητες



Σχήμα 6.16: Κληρονόμηση θετικής δήλωσης IN σε οντότητα

ανάκλησης *πιάτων* [6].



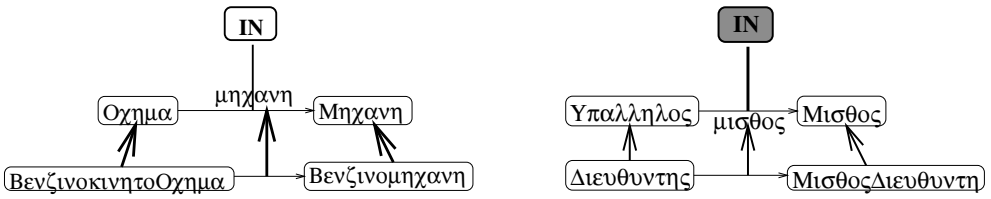
Σχήμα 6.17: Κληρονόμηση αρνητικής δήλωσης IN σε οντότητα

Το αναγνωριστικό **IN** μπορεί να χρησιμοποιηθεί σε δηλώσεις γνωρισμάτων. Με τον τρόπο αυτό ορίζουμε (ή αποτρέπουμε) τις ΣΕΕ που αφορούν τις περιπτώσεις τους (πρόσθεση/διαγραφή γνωρισμάτων) : η δήλωση του σχήματος 6.18(α) ορίζει τις ΣΕΕ που προσθέτουν ή διαγράφουν γνωρίσματα κατηγορίας *όνομα* από τις περιπτώσεις της κλάσεως *Άνθρωπος*, ενώ η δήλωση του σχήματος 6.18(β) αποτρέπει την προσθήκη/διαγραφή

[6]Θεωρείστε ότι η ανάκληση γίνεται μέσω ενός δελτίου αναζήτησης *πιάτων* με κριτήριο το *μέγεθος* τους (*μικρό, μέτριο, μεγάλο*) και το *σχήμα* τους (*στρογγυλό, παραλληλόγραμμο*).

(α) (β)

Σχήμα 6.18: Δηλώσεις IN σε κατηγορίες γνωρισμάτων



(α) (β)

Σχήμα 6.19: Κληρονόμηση δήλωσεων IN σε κατηγορίες γνωρισμάτων

γνωρισμάτων κατηγορίας *μισθός* από τις περιπτώσεις της κλάσεως *Υπάλληλος*.

Οι δηλώσεις σε κατηγορίες κληρονομούνται επίσης στις υποκατηγορίες τους όπως φαίνεται στα παραδείγματα του σχήματος 6.19.

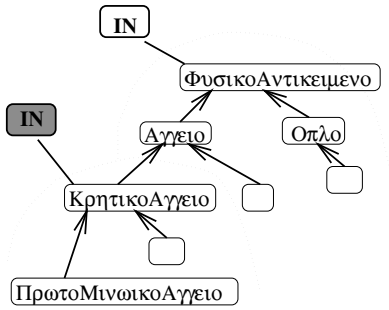
6.7.2 Δηλώσεις IN : Συνδυασμοί Θετικών και Αρνητικών

Η δυνατότητα επικαλυπτόμενων θετικών και αρνητικών δηλώσεων παρέχει ευελιξία στον ορισμό ΟΕ, όπως προκύπτει από τα παρακάτω παραδείγματα.

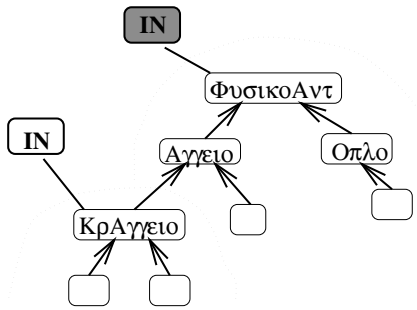
Οι δηλώσεις του σχήματος 6.20 ορίζουν τις ΣΕΕ που ενημερώνουν τις περιπτώσεις της ιεραρχίας εξειδίκευσης της κλάσης *Φυσικό Αντικείμενο* εκτός της υποϊεραρχίας με κορυφή την κλάση *Κρητικό Αγγείο*. Από το σχήμα φαίνεται ότι η οντότητα *ΠρωτοΜινωικόΑγγείο* κληρονομεί τη δήλωση της οντότητας *ΦυσικόΑντικείμενο* και τη δήλωση της κλάσης *ΚρητικόΑγγείο*. Υπερισχύει η δήλωση της κλάσης *ΚρητικόΑγγείο* αφού αυτή είναι ειδικότερη της κλάσης *ΦυσικόΑντικείμενο*.

Μια ‘‘αντίθετη’’ κατάσταση παρουσιάζεται στο σχήμα 6.21. Οι δηλώσεις του σχήματος αυτού ορίζουν την προσθήκη/διαγραφή οντοτήτων μόνο στην υποϊεραρχία εξειδίκευσης της κλάσης *ΚρητικόΑγγείο*.

Στο σχήμα 6.22 παρουσιάζονται δύο ‘‘αντίθετες’’ δηλώσεις οι οποίες κληρονομούνται στην κλάση *Μαχαίρι*. Η ασάφεια που προκύπτει δεν μπορεί να επιλυθεί με τον προηγούμενο τρόπο. Σε τέτοιες περιπτώσεις υπερισχύουν οι αρνητικές (πιο περιοριστικές) δηλώσεις όπως αναλύθηκε στην ενότητα 6.4.1. Στο εν λόγω παράδειγμα δεν ορίζονται οι ΣΕΕ αλλαγής

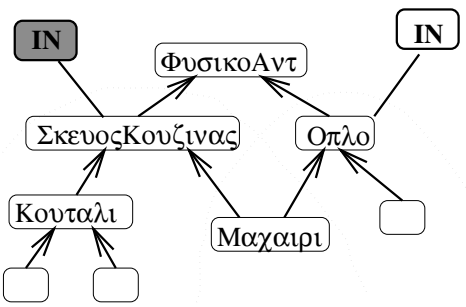


Σχήμα 6.20: Επικάλυψη θετικής δήλωσης IN από αρνητική



Σχήμα 6.21: Επικάλυψη αρνητικής δήλωσης IN από θετική.

του πληθυσμού της κλάσεως *Μαχαίρι*.



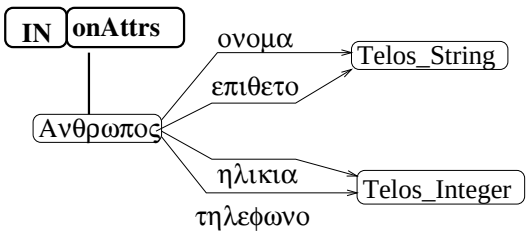
Σχήμα 6.22: Κυριαρχία αρνητικής δήλωσης IN

6.7.3 Δηλώσεις IN και Πεδίο Εφαρμογής onAttrs

Το αναγνωριστικό **IN** | **onAttrs** συμβολίζει μια δήλωση η οποία αποτελείται από το ζευγάρι αναγνωριστικών {AddIn, DelIn} με πεδίο εφαρμογής τα γνωρίσματα (συμπεριλαμβανομένων και των κληρονομούμενων) του αντικειμένου δήλωσης. Η χρήση της παρέχει **ευκολία**(συντομία) κατά τον ορισμό ΟΕ και αποτελεί μια **δυναμική δήλωση** (ακολουθεί

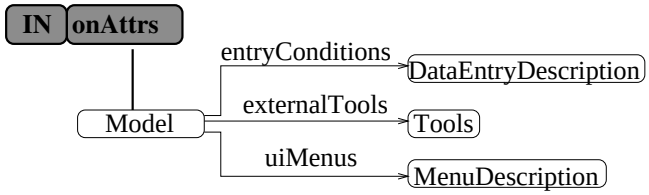
την εξέλιξη του σχήματος). Παρακάτω παρουσιάζονται παραδείγματα χρήσης της με το αναγνωριστικό **IN**.

Η δήλωση που εικονίζεται στο σχήμα 6.23 ορίζει τις ΣΕΕ πρόσθεσης/διαγραφής περιπτώσεων σε όλα τα γνωρίσματα της κλάσης *Άνθρωπος*.



Σχήμα 6.23: Θετική δήλωση IN με πεδίο εφαρμογής onAttrs

Αντίθετα η αρνητική δήλωση που εικονίζεται στο σχήμα 6.24 απαγορεύει την ενημέρωση των περιπτώσεων των γνωρισμάτων της κλάσεως *Model*. Η σκοπιμότητα της δήλωσης αυτής είναι η απαγόρευση των ΣΕΕ που θα αλλοίωναν τη διαμόρφωση (configuration) της επαφής χρήσεως του SIS.



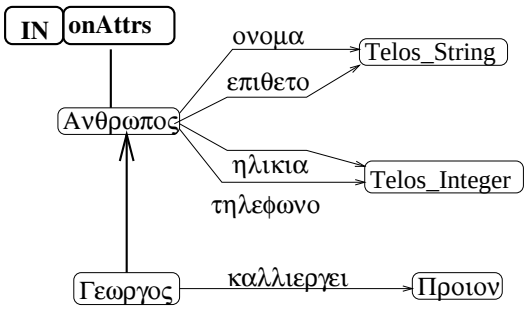
Σχήμα 6.24: Αρνητική δήλωση IN με πεδίο εφαρμογής onAttrs

Οι δηλώσεις με πεδίο εφαρμογής onAttrs κληρονομούνται επίσης. Για παράδειγμα η δήλωση που εικονίζεται στο σχήμα 6.25 κληρονομείται στην κλάση *Γεωργός* και άρα ορίζει τις ΣΕΕ αλλαγής των περιπτώσεων της κατηγορίας *καλλιεργεί*.

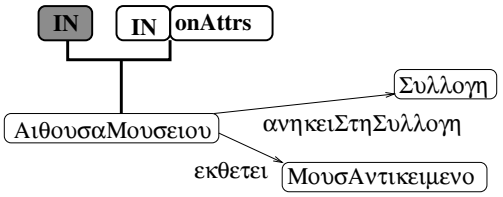
Στο σχήμα 6.26 ορίζονται οι ΣΕΕ αλλαγής του πληθυσμού των γνωρισμάτων της κλάσης *Αίθουσας Μουσείου* (οι τιμές των οποίων είναι χρονικά μεταβαλλόμενες) αλλά αποτρέπεται την αλλαγή του πληθυσμού της κλάσεως *Αίθουσα Μουσείου* (συνήθως είναι σταθερές).

6.7.4 Δηλώσεις IN : Όψεις Προβολής Κλάσεων

Η χρήση θετικών και αρνητικών δηλώσεων και πεδίου εφαρμογής onAttrs δίνει τη δυνατότητα ορισμού όψεων που ενημερώνουν προβολές κλάσεων.

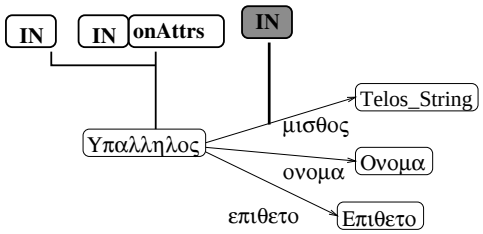


Σχήμα 6.25: Κληρονόμηση δήλωσης με πεδίο εφαρμογής onAttrs



Σχήμα 6.26: Συνδυασμός δηλώσεων IN

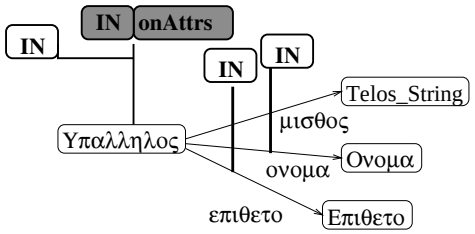
Στο σχήμα 6.27 ορίζεται μια όψη προβολής της κλάσεως *Υπάλληλος* με ‘‘απόκρυψη’’ του γνωρίσματος *μισθός*.



Σχήμα 6.27: Ορισμός όψης προβολής με απόκρυψη γνωρισμάτων

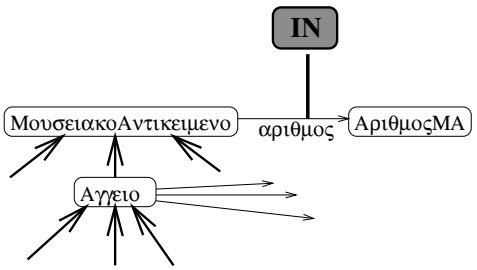
Στο σχήμα 6.28 ορίζεται μια κλάση προβολής της κλάσεως *Υπάλληλος* με ‘‘επιλογή’’ των γνωρίσματων *όνομα, επίθετο*.

Ένα πλεονέκτημα του μηχανισμού είναι ότι οι δηλώσεις κληρονομούνται άρα ορίζονται έμμεσα προβολές όλης της υποϊεραρχίας εξειδίκευσης (άλλα συστήματα όπως το [40] αδυνατούν να περιγράψουν κάτι τέτοιο). Αυτό οφείλεται στο ότι τα γνωρίσματα που κληρονομεί μια κλάση της SIS-Telos δεν αποθηκεύονται και σε αυτήν αλλά υπολογίζονται κατά το χρόνο εκτέλεσης. Ως αποτέλεσμα μια άμεση δήλωση ενημέρωσης σε ένα γνώρισμα, το συνοδεύει σε όλες τις κλάσεις στις οποίες αυτό κληρονομείται. Αυτή η ιδιαιτερότητα έχει κάποια πλεονεκτήματα (βλέπε σχήμα 6.29) και κάποιες αδυναμίες :



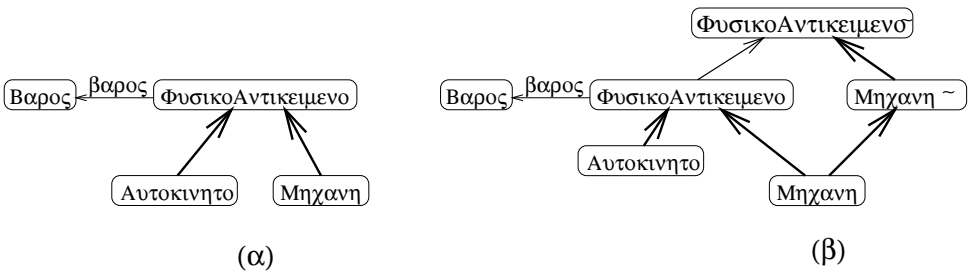
Σχήμα 6.28: Ορισμός όψης προβολής με επιλογή γνωρισμάτων

δεν μπορούμε να ορίσουμε τις ΣΕΕ ενημέρωσης ενός γνωρίσματος ανάλογα με την κλάση από την οποία χρησιμοποιείται (δες σχήμα 6.30(α)).



Σχήμα 6.29: Κληρονομούμενα γνωρίσματα και άμεσες δηλώσεις.

Η αρνητική δήλωση IN του γνωρίσματος *αριθμός*, το συνοδεύει σε όλες τις κλάσεις στις οποίες κληρονομείται. Η δήλωση αυτή ανήκει σε μια ΟΕ η οποία απευθύνεται σε έναν αρχαιολόγο ο οποίος δεν ενδιαφέρεται για τις διαχειριστικές πληροφορίες των μουσειακών αντικειμένων.



Σχήμα 6.30: Αναδιάρθρωση ιεραρχίας εξειδίκευσης

Στο σχήμα (α) το γνώρισμα *βάρος* κληρονομείται στις κλάσεις *Αυτοκίνητο* και *Μηχανή* για αυτό δεν μπορούμε να ορίσουμε μια ΟΕ η οποία να ορίζει τις ΣΕΕ πρόσθεσης γνωρισμάτων τύπου *βάρος* μόνο στα αυτοκίνητα και όχι στις μηχανές.
Στο σχήμα (β) παρατηρούμε ότι με την προσθήκη των κλάσεων *ΦυσικόΑντικείμενο~* και *Μηχανή~* και με την αναδιάρθρωση της ιεραρχίας εξειδίκευσης, αυτό είναι εφικτό αφού η κλάση *Μηχανή~* δεν κληρονομεί το γνώρισμα *βάρος*.

Μια λύση στο πρόβλημα θα ήταν η κατάλληλη αναδιάρθρωση της ιεραρχίας εξειδίκευσης (δες σχήμα 6.30(β)). Ο τρόπος αυτός έχει υιοθετηθεί από αρκετά συστήματα [40], στα οποία κάθε όψη προβολής αποτελεί ένα νέο τύπο ο οποίος ενσωματώνεται

κατάλληλα στο σχήμα. Υπάρχουν στην βιβλιογραφία ([2], [31]) αλγόριθμοι οι οποίοι παράγουν τους απαραίτητους τύπους προβολής και τους ενσωματώνουν στην ιεραρχία, αναδιοργανώνοντας κατάλληλα το σχήμα, ώστε να μην δημιουργείται κανένα πρόβλημα στους υπάρχοντες τύπους. Αυτοί οι αλγόριθμοι θα μπορούσαν να χρησιμοποιηθούν και στην SIS-Telos [7] , με στόχο την πληρέστερη υποστήριξη κλάσεων προβολής. Η χρήση τους όμως δεν ενδείκνυται αφού παρουσιάζουν τις παρακάτω αδυναμίες :

- Προκαλούν πολλές αλλαγές στο σχήμα. Επομένως αν οι ΟΕ είναι πολλές, τότε το σχήμα της βάσης διευρύνεται σημαντικά.

- Δεν υποστηρίζεται η δυνατότητα αναίρεσης (undo), η οποία θα μπορούσε να επαναφέρει το σχήμα στην αρχική του κατάσταση.

- Αλλάζουν τη θέση (της κλάσης εκκίνησης) πολλών γνωρισμάτων και άρα είναι πιθανόν να προκύψουν προβλήματα κατά την αναφορά τους μέσω των λογικών τους ονομάτων. [8]

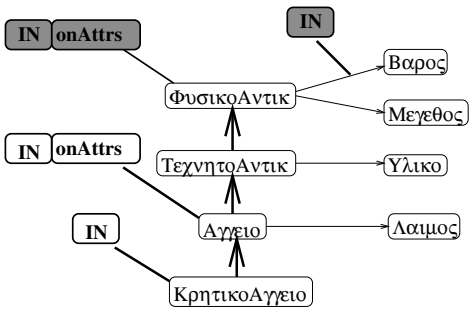
- Δημιουργούνται ασάφειες σχετικά με τις κλάσεις στις οποίες πρέπει να δημιουργούνται περιπτώσεις.

Για τους παραπάνω λόγους μια τέτοια λύση δεν κρίνεται κατάλληλη. Αντίθετα καλύτερη λύση θα ήταν η αλλαγή της πολιτικής της SIS-Telos όσον αφορά τα κληρονομούμενα γνωρίσματα. Αν η SIS-Telos αποθήκευε αντίγραφα των κληρονομουμένων γνωρισμάτων τότε θα ήταν δυνατόν να δηλώσουμε τη δυνατότητα ενημέρωσης τους ανάλογα με την κλάση στην οποία αυτά χρησιμοποιούνται. Μια τέτοια αλλαγή πολιτικής θα επιτάχυνε τις ερωτήσεις στη βάση (αφού τα κληρονομούμενα γνωρίσματα δεν θα υπολογίζονταν κάθε φορά), αλλά θα επέφερε μεγαλύτερο υπολογιστικό κόστος στις ενημερώσεις που γίνονται.

Για την καλύτερη κατανόηση των κανόνων επίλυσης ασαφειών που χρησιμοποιούνται σε περιπτώσεις αντίθετων και επικαλυπτόμενων δηλώσεων, παρουσιάζεται το παράδειγμα του σχήματος 6.31. Στο σχήμα αυτό ορίζονται οι ΣΕΕ που προσθέτουν/διαγράφουν *Κρητικά Αγγεία* και ενημερώνουν τα γνωρίσματα όλων των *Αγγείων* (και των *Κρητικών Αγγείων*) εκτός των γνωρισμάτων κατηγορίας *βάρους*.

[7] Απαιτούν μικρές τροποποιήσεις ώστε να λαμβάνονται υπόψη και τα γνωρίσματα που καταλήγουν σε μια κλάση και τις σχέσεις εξειδίκευσης που μπορεί να ισχύουν μεταξύ γνωρισμάτων.

[8] Η αναφορά ενός γνωρίσματος της SIS-Telos , μέσω του λογικού του ονόματος απαιτεί και το όνομα της κλάσης στην οποία έχει αποδοθεί).

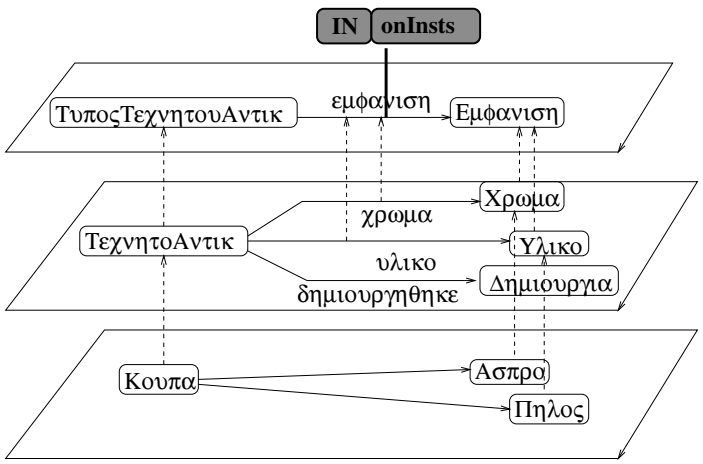


Σχήμα 6.31: Σύνθετο παράδειγμα δηλώσεων.

6.7.5 Πεδίο Εφαρμογής onInsts

Μια δήλωση με πεδίο εφαρμογής onInsts αφορά όλες τις περιπτώσεις του ΕΑΔ (εστιακού αντικειμένου δήλωσης), άμεσες και έμμεσες (λόγω κληρονόμησης στις υποκλάσεις του ΕΑΔ). Η χρήση της παρέχει **ευκολία** (συντομία) στον ορισμό ΟΕ, αφού εκμεταλεύεται τη δυνατότητα που προσφέρει η SIS-Telos της ταξινόμησης του σχήματος σε μετασχήμα. Με αυτόν τον τρόπο, όπως θα δούμε, υποστηρίζονται δυναμικές όψεις προβολής κλάσεων.

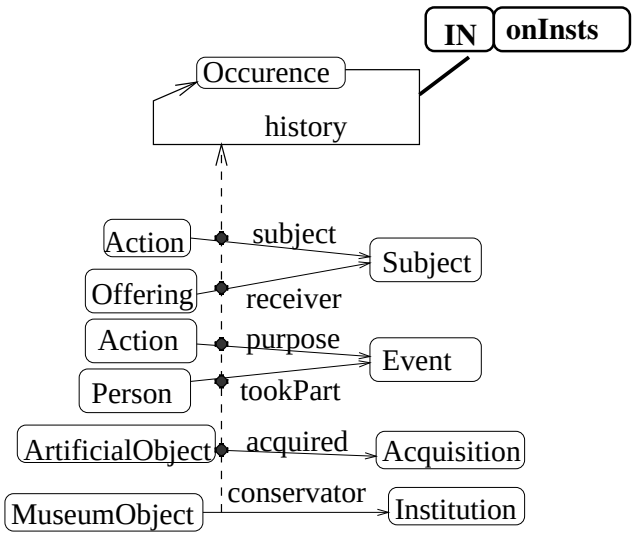
Η δήλωση που εικονίζεται στο σχήμα 6.32 ορίζει τις ΣΕΕ που ενημερώνουν τα γνωρίσματα που έχουν μετακατηγορία την κλάση γνωρισμάτων *εμφάνιση.* Στη συγκεκριμένη κατάσταση της βάσης, η δήλωση αφορά της κατηγορίες *χρώμα* και *υλικό.*



Σχήμα 6.32:

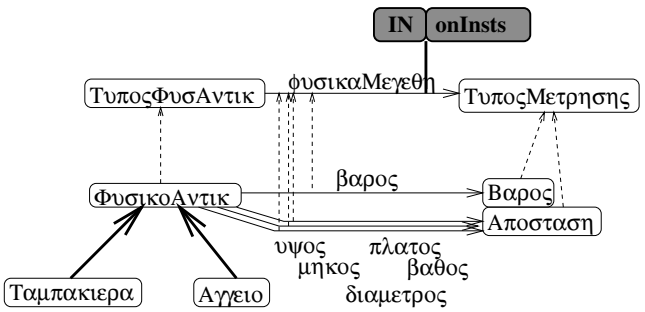
Θετική δήλωση IN με πεδίο εφαρμογής onInsts.

Η δήλωση που εικονίζεται στο σχήμα 6.33 ορίζει ένα σύνολο από ΣΕΕ που ενημερώνουν τις περιπτώσεις των κατηγοριών τύπου *history*. Η σκοπιμότητα μιας τέτοιας δήλωσης είναι ο ορισμός μιας ΟΕ που απευθύνεται σε έναν ιστορικό.



Σχήμα 6.33: Θετική δήλωση IN με πεδίο εφαρμογής onInsts.

Η δήλωση στο παράδειγμα του σχήματος 6.34 αποτρέπει την ενημέρωση των *φυσικών μεγεθών* των *Φυσικών Αντικειμένων*. Σημειώνουμε εδώ ότι επειδή τα γνωρίσματα που αφορά η δήλωση κληρονομούνται στις υποκλάσεις της *ΦυσικόΑντικείμενο*, η δήλωση αυτή ορίζει μια προβολή των κλάσεων όλης της ιεραρχίας εξειδίκευσης της κλάσης *ΦυσικόΑντικείμενο*.

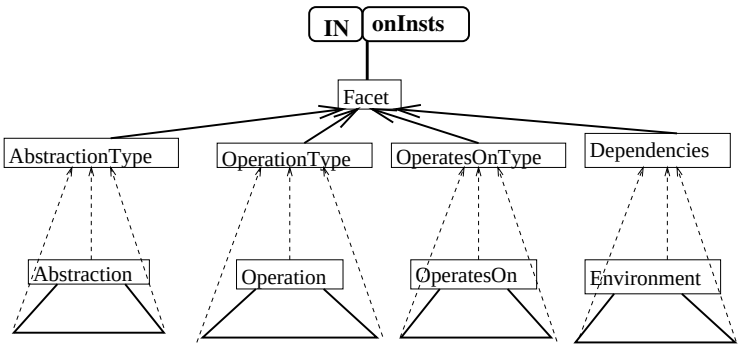


Σχήμα 6.34: Αρνητική δήλωσης IN με πεδίο εφαρμογής onInsts.

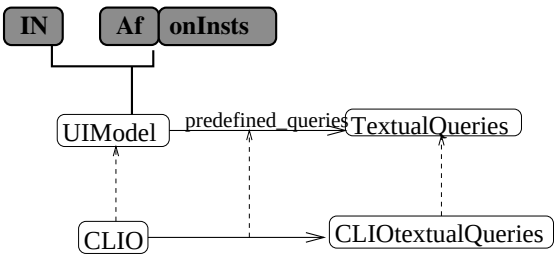
Η δήλωση στο παράδειγμα του σχήματος 6.35 ορίζει τις ΣΕΕ για πολυεδρική ταξινόμηση. Η εικονιζόμενη δήλωση κληρονομείται στις τέσσερεις υποκλάσεις της. Λόγω του ότι κάθε όρος ταξινόμησης είναι ταξινομημένος σε μία από αυτές τις κλάσεις η δήλωση αυτή δηλώνει τις ΣΕΕ ταξινόμησης αντικειμένων στις έδρες ταξινόμησης.

Η δήλωση στο παράδειγμα του σχήματος 6.36 αποτρέπει την ενημέρωση των γνωρισμάτων που έχουν αποδοθεί στις περιπτώσεις της κλάσεως *UIModel*.

Συμπερασματικά μπορούμε να πούμε ότι οι δηλώσεις με πεδίο εφαρμογής onInsts είναι



Σχήμα 6.35: Θετική δήλωση IN με πεδίο εφαρμογής onInsts.



Σχήμα 6.36: Αρνητική δήλωσης AF με πεδίο εφαρμογής onInsts

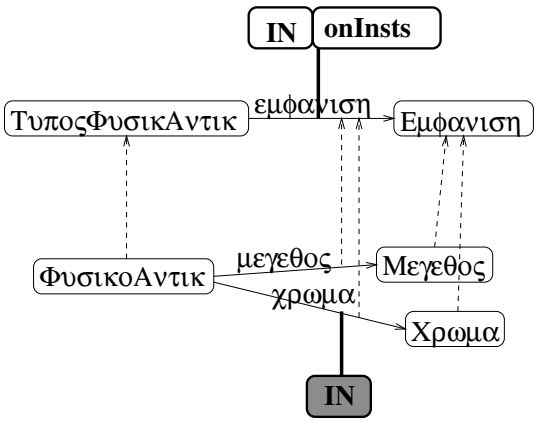
Το αναγνωριστικό ενημέρωσης AF αφορά την ενημέρωση των γνωρισμάτων που ξεκινάνε από ένα αντικείμενο. Η εικονιζόμενη μη-εστιακή δήλωση αφορά την οντότητα *CLIO* και αποτρέπει την προσθήκη και τη διαγραφή γνωρισμάτων σε αυτήν.

σύντομες και επιτρέπουν το δυναμικό ορισμό ΟΕ. Για παράδειγμα η δήλωση στο σχήμα 6.33 ορίζει την ενημέρωση προβολών κλάσεων που ανήκουν σε πολλές και διαφορετικές ιεραρχίες εξειδίκευσης. Αυτό έχει το πλεονέκτημα ότι καθώς το σχήμα εξελίσσεται (πχ προσθήκη νέων κατηγοριών, αλλαγή ταξινόμησης των κατηγοριών σε μετακατηγορίες) εξελίσσονται αυτόματα και οι ΟΕ που ορίζονται από αυτές τις δηλώσεις.

Πεδίο Εφαρμογής onInsts και Ασάφειες

Στην υποενότητα αυτή παρουσιάζονται παραδείγματα αντίθετων και επικαλυπτόμενων δηλώσεων για την καλύτερη κατανόηση των κανόνων που χρησιμοποιούνται για την επίλυση των ασαφειών που προκύπτουν.

Στο σχήμα 6.37 η δήλωση στην κατηγορία *χρώμα* αποτρέπει την ενημέρωση των περιπτώσεων της, ενώ η δήλωση στην μετακατηγορία *εμφάνιση* ορίζει το αντίθετο. Σε αυτές τις περιπτώσεις υπερισχύουν οι άμεσες δηλώσεις (η δήλωση στην κατηγορία *χρώμα*).

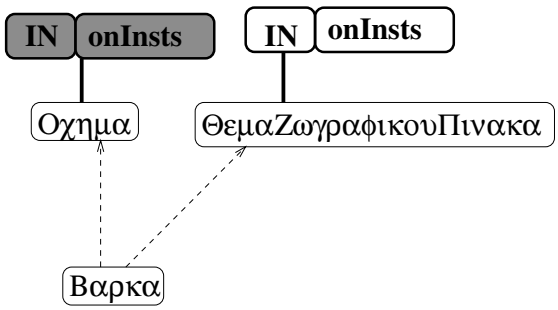


Σχήμα 6.37: Επικάλυψη άμεσης δήλωσης και δήλωσης *onInsts*

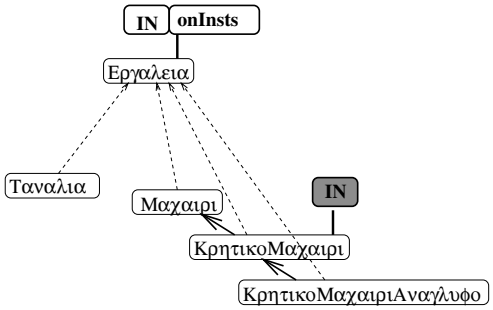
Στο σχήμα 6.38 η κλάση *Βάρκα* είναι στην εμβέλεια δύο αντίθετων δηλώσεων με πεδίο εφαρμογής onInsts. Σε αυτές τις περιπτώσεις, όπως αναλύθηκε στην ενότητα 6.4.2, υπερισχύουν οι αρνητικές δηλώσεις.

Στο σχήμα 6.39 η κλάση *ΚρητικόΜαχαίριΑνάγλυφο* κληρονομεί μια άμεση δήλωση και είναι στο πεδίο εφαρμογής της δήλωσης στην κλάση *Εργαλεία*. Σε αυτές τις περιπτώσεις , όπως αναλύθηκε στην ενότητα 6.4.2, υπερισχύουν οι κληρονομούμενες δηλώσεις.

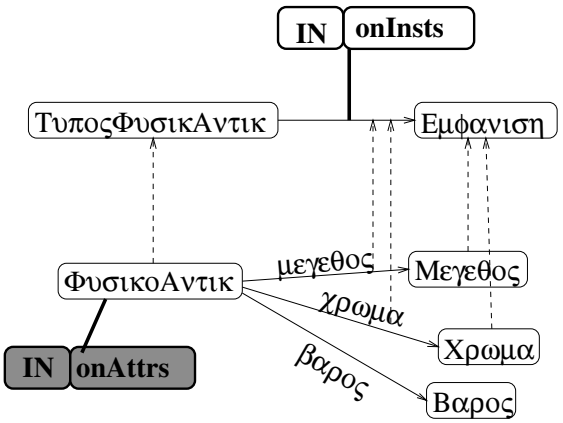
Στο σχήμα 6.40 η κατηγορία *μέγεθος* είναι στο πεδίο εφαρμογής μιας δήλωσης onAttrs και μιας δήλωσης onInsts. Σε αυτές τις περιπτώσεις υπερισχύουν οι δηλώσεις onInsts.



Σχήμα 6.38: Επικάλυψη δηλώσεων *onInsts*



Σχήμα 6.39: Επικάλυψη δήλωσης *onInsts* από κληρονομούμενη άμεση

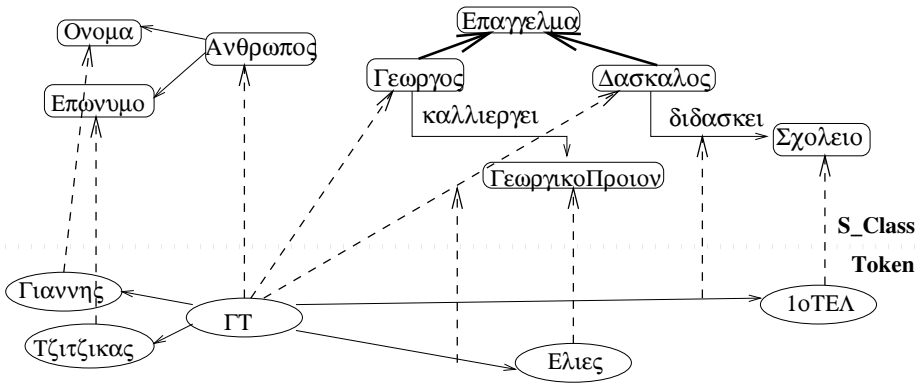


Σχήμα 6.40: Επικάλυψη δήλωσης *onAttrs* από δήλωση *onInsts*

6.7.6 Σχετικές κλάσεις

Η SIS-Telos υποστηρίζει τη δυνατότητα πολλαπλής ταξινόμησης των αντικειμένων της (οντότητες και γνωρίσματα). Η δυνατότητα αυτή συχνά αξιοποιείται για την περιγραφή αντικειμένων από διαφορετικές απόψεις. Ως παραδείγματα μπορούμε να παραθέσουμε την *μορφολογική* περιγραφή (ταξινόμηση στην κλάση *PhysicalObject*) και *τεχνοτροπική* περιγραφή (ταξινόμηση στην κλάση *Style*) ενός μουσειακού αντικειμένου. Άλλο παράδειγμα αποτελεί η *δομική* περιγραφή (ταξινόμηση στην κλάση *Declarator*) και η *λειτουργική* περιγραφή (ταξινόμηση στις έδρες ταξινόμησης ή *Facets*) των μονάδων λογισμικού.

Επίσης συχνά η πολλαπλή ταξινόμηση χρησιμοποιείται για την περιγραφή σύνθετων ιδιοτήτων. Τέτοια παραδείγματα παρουσιάζονται στα σχήματα 6.41 και 6.42.



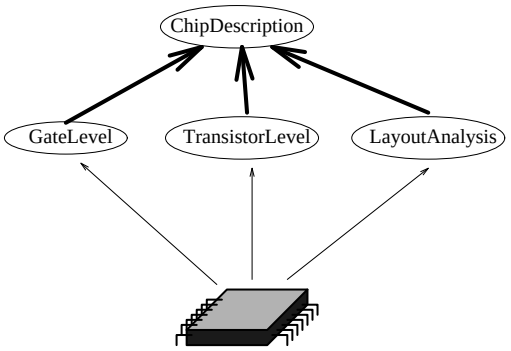
Σχήμα 6.41: Χρήση ταξινόμησης για παράσταση σύνθετων ιδιοτήτων

Η έννοια του επαγγέλματος δεν μοντελοποιήθηκε ως γνώρισμα της κλάσεως *Άνθρωπος* αλλά σαν μια ξεχωριστή ιεραρχία εξειδίκευσης. Οι περιπτώσεις της κλάσης *Άνθρωπος* ταξινομούνται στην κατάλληλη κλάση επαγγέλματος. Αυτό δίνει τη δυνατότητα απόδοσης τιμής σε γνωρίσματα που αφορούν το κάθε επάγγελμα.

Στο οντοκεντρικό μοντέλο με την έννοια του αντικειμένου επιτυγχάνεται η ομαδοποίηση και η σύνδεση των σχετικών πληροφοριών. Αυτό βοηθάει πολύ τη διαχείριση και την αξιοποίηση των πληροφοριών. Με την ίδια λογική προτείνεται η σύνδεση (μέσω του μοντέλου της SIS-Telos) των εν λόγω συγγενικών κλάσεων.

Οι συνδέσεις αυτές θα συνέβαλαν στην κατανόηση της χρήσης του μοντέλου από το χρήστη, αφού θα του υποδείκνυαν την ύπαρξη σχετικών κλάσεων και θα μπορούσαν να αξιοποιηθούν άμεσα από ένα διαλογικό δελτίο ενημέρωσης δεδομένων.

Η ανάγκη αυτή αντιμετωπίσθηκε από το ΜΔΕ με τον ορισμό της πράξεως *GiveAspect* η οποία όριζε τις ΣΕΕ ταξινόμησης ενός αντικειμένου σε οποιαδήποτε κλάση η οποία είχε ταξινομηθεί ως *RootClass*. Η λύση αυτή δεν είναι πολύ καλή αφού για παράδειγμα περιπτώσεις περιπτώσεις *RootClass* μπορεί να είναι οι κλάσεις { *Άνθρωπος*, *Επάγγελμα*,



Σχήμα 6.42: Πολλαπλή ταξινόμηση αντικειμένων.

Ενα chip μπορεί να περιγραφεί με τη βοήθεια των κλάσεων *GateLevel*, *TransistorLevel* και *LayoutAnalysis*. Κάθε μία από αυτές περιγράφει από διαφορετική άποψη (με τη βοήθεια ενός συνόλου γνωρισμάτων) ένα chip. Η κλάση *GateLevel* το περιγράφει από λογικής απόψεως προκειμένου να γίνει η επαλήθευση της λογικής του (logic verification). Η *TransistorLevel* περιγράφει ένα chip από χρονικής απόψεως προκειμένου να γίνει ανάλυση του χρονισμού του (timing analysis) , ενώ η *LayoutAnalysis* περιγράφει την τοποθέτηση των τμημάτων του προκειμένου να ελεγθεί η τήρηση των σχεδιαστικών κανόνων (design-rule checking).

Αγγείο, Τεχνοτροπία }. Με τον τρόπο αυτό μπορεί να ταξινομηθεί μια οντότητα στις κλάσεις *Άνθρωπος* και *Αγγείο*, ενώ κανονικά αυτό που έπρεπε να ορισθεί είναι οι συγγένεια των ζευγαριών (*Άνθρωπος, Επάγγελμα*) και (*Αγγείο, Τεχνοτροπία*) [9].

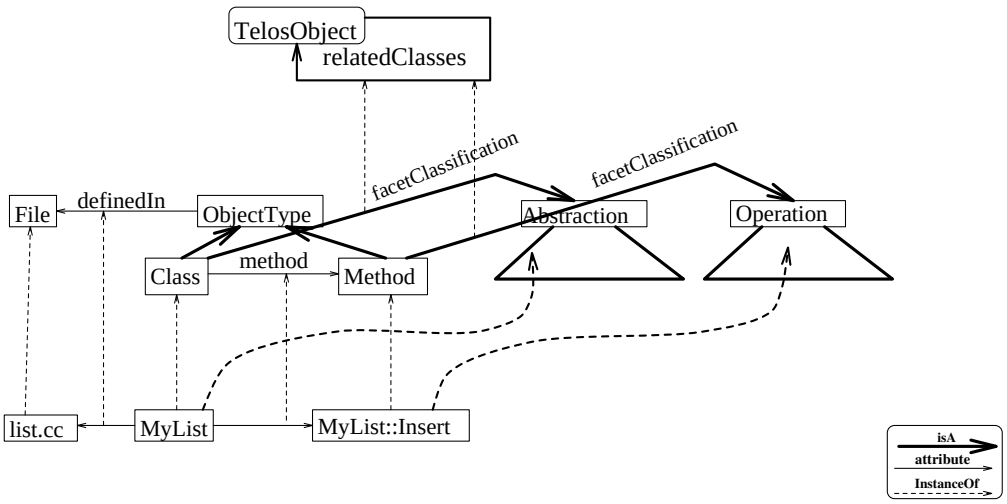
Για το λόγο αυτό προτείνεται ένας νέος τύπος γνωρισμάτων, του οποίου ένα πιθανό όνομα είναι **relatedClasses**. Η σύνδεση μιας κλάσης Α με μια κλάση Β μέσω ενός γνωρίσματος αυτού του τύπου δηλώνει ένα σύνθετο γνώρισμα της κλάσεως οντοτήτων Α. Αν σε μια περίπτωση της κλάσεως Α, θελήσουμε να αποδώσουμε τιμή σε ένα τέτοιο γνώρισμα , αυτό που πρέπει να κάνουμε είναι να την ταξινομήσουμε σε μια υποκλάση της κλάσης Β (δες σχετικό παράδειγμα στο σχήμα 6.43).

Ένα γνώρισμα τέτοιου τύπου, όπως και κάθε άλλο γνώρισμα της SIS-Telos κληρονομείται και μάλιστα μπορεί να εξειδικεύεται (δες σχήμα 6.44). Κατά την εξειδίκευση πρέπει να επιβάλλονται οι ίδιοι δομικοί περιορισμοί με αυτούς της εξειδίκευσης των κανονικών γνωρισμάτων.

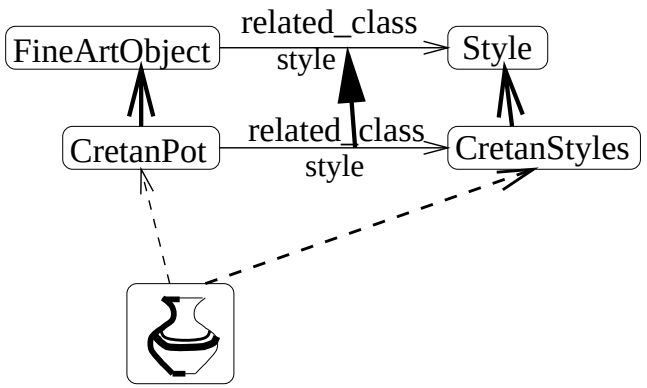
Στην ουσία περιπτώσεις των γνωρισμάτων αυτών είναι σύνδεσμοι ταξινόμησης (*instOf*). Επειδή όμως αυτό άπτεται του γενικότερου θέματος της ταξινόμησης και κατηγοριοποίησης των συνδέσμων *instOf* (το οποίο δεν υποστηρίζεται από την SIS-Telos), δεν προτείνεται αρχικώς καμια σχετική αλλαγή στην υλοποίηση της SIS-Telos .

Ο προτεινόμενος τύπος γνωρισμάτων χρησιμοποιείται μόνο για την διευκόλυνση της διαλογικής ενημέρωσης της βάσης (και όχι για παράδειγμα στην διατύπωση ερωτήσεων).

[9] Η σύνδεση των συγγενών κλάσεων θα μπορούσε να γίνει μέσω του μεταεπιπέδου (ένα επίπεδο αφαίρεσης παραπάνω) δηλαδή μέσω μιας σύνδεσης των μετακλάσεων τους. Αυτό θα απαιτούσε την ταξινόμηση των κλάσεων σε μεταεπίπεδο προκειμένου να περιγραφεί η σχέση συγγένειας μεταξύ τους.



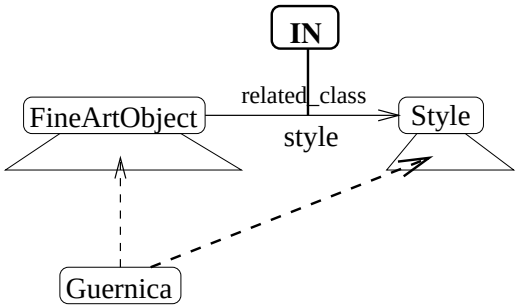
Σχήμα 6.43: Χρήση γνωρισμάτων τύπου **relatedClasses**.



Σχήμα 6.44: Εξειδίκευση γνωρισμάτων τύπου **relatedClasses**.

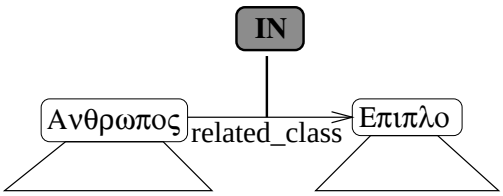
Επομένως στα γνωρίσματα αυτά δεν είναι εφικτή η ταξινόμηση τους, η συγκεκριμενοποίηση τους ή η απόδοση γνωρισμάτων σε αυτά, αλλά μόνο η εξειδίκευση τους από γνωρίσματα του ίδιου τύπου (αυτός ο περιορισμός μπορεί να επιβληθεί εύκολα από τον Semantic Checker).

Στα γνωρίσματα αυτού του τύπου μπορούν να γίνουν δηλώσεις του μεταμοντέλου με ΑΕ $AddIn, DelIn$ ώστε να ορίζονται εξειδικευμένες ΣΕΕ ταξινόμησης. Για παράδειγμα η δήλωση του σχήματος 6.45 ορίζει τις ΣΕΕ που προσθέτουν/αφαιρούν περιπτώσεις από την κλάση *Style* μόνο αν οι περιπτώσεις αυτές είναι και περιπτώσεις της κλάσεως *FineArtObject*.



Σχήμα 6.45: Δήλωση σε γνώρισμα τύπου **relatedClasses**

Παράδειγμα αρνητικής δήλωσης σε τέτοιο γνώρισμα παρουσιάζεται στο σχήμα 6.46, η οποία απαγορεύει την ταξινόμηση μιας περίπτωσης της κλάσης *Άνθρωπος* στην κλάση *Έπιπλο*. Παρόμοια δήλωση θα μπορούσε να γίνει στο αμοιβαίως αποκλειόμενο ζευγάρι κλάσεων *Άνδρας, Γυναίκα*. Στις περιπτώσεις αυτές ένα γνώρισμα τύπου *relatedClasses* χρησιμοποιείται καταχρηστικά για την παράσταση ενός απλού στατικού περιορισμού της βάσης ο οποίος μπορεί να χρησιμοποιηθεί για την ελαχιστοποίηση των πιθανών λαθών κατά την διαλογική εισαγωγή δεδομένων.

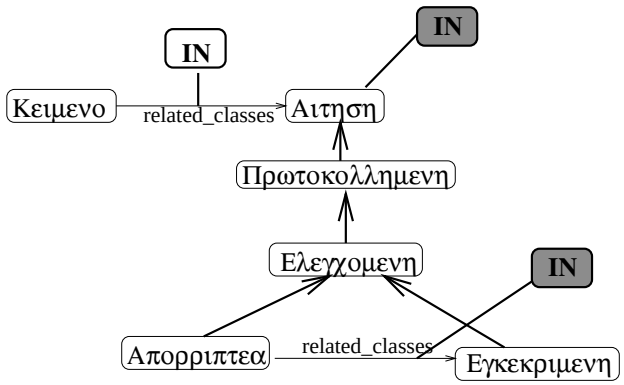


Σχήμα 6.46: Αρνητική δήλωση IN σε γνώρισμα τύπου **relatedClasses**

.

Ένα σύνθετο παράδειγμα παρουσιάζεται στο σχήμα 6.47. Η αρνητική δήλωση IN στην κλάση *Αίτηση* αποτρέπει την αλλαγή του πληθυσμού της, αλλά η θετική δήλωση IN στο γνώρισμα τύπου relatedClasses που καταλήγει σε αυτήν, επιτρέπει την προσθήκη και τη

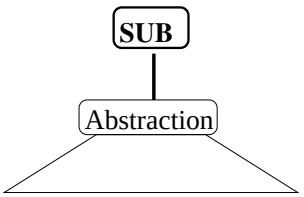
διαγραφή συνδέσμων ταξινόμησης σε αυτήν (και τις υποκλάσεις της), μόνο αν το άλλο άκρο των συνδέσμων αυτών είναι περιπτώσεις της κλάσης *Κείμενο*. Τέλος η αρνητική δήλωση μεταξύ των κλάσεων *Απορριπτέα*, *Εγκεκριμένη* αποτρέπει την ταξινόμηση των *Απορριπτέων* αιτήσεων στην κλάση *Εγκεκριμένη*.



Σχήμα 6.47: Θετικές και αρνητικές δηλώσεις σε γνωρίσματα τύπου **relatedClasses**.

6.7.7 Δηλώσεις *SUB*

Το αναγνωριστικό **SUB** συμβολίζει το ζευγάρι αναγνωριστικών {AddSub, DelSub}. Το παράδειγμα θετικής δήλωσης που παρουσιάζεται στο σχήμα 6.48, ορίζει τις ΣΕΕ εξειδίκευσης των κλάσεων που ανήκουν στην ιεραρχία εξειδίκευσης με κορυφή την κλάση *Abstraction*.

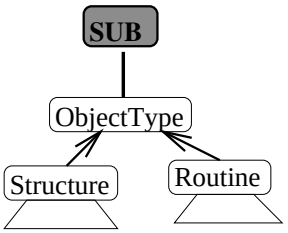


Σχήμα 6.48: Θετική δήλωση SUB σε οντότητα.

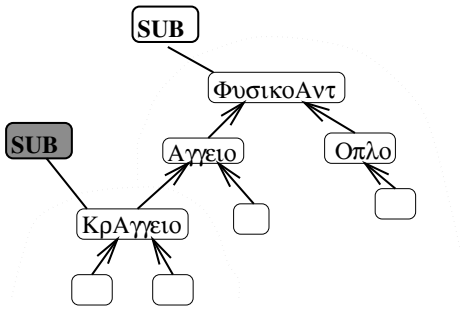
Το παράδειγμα αρνητικής δήλωσης που παρουσιάζεται στο σχήμα 6.49, απαγορεύει τις ΣΕΕ εξειδίκευσης των κλάσεων που ανήκουν στην ιεραρχία εξειδίκευσης με κορυφή την κλάση *ObjectType*.

Στο παράδειγμα του σχήματος 6.50 δηλώνονται οι ΣΕΕ που εξειδικεύουν την ιεραρχία εξειδίκευσης της κλάσεως *ΦυσικόΑντικείμενο* εκτός της υποϊεραρχίας της κλάσεως *ΚρητικόΑγγείο*. Η σκοπιμότητα της συγκεκριμένης δήλωσης μπορεί να είναι η συντήρηση μιας τυπικής ιεραρχίας ταξινόμησης κρητικών αγγείων.

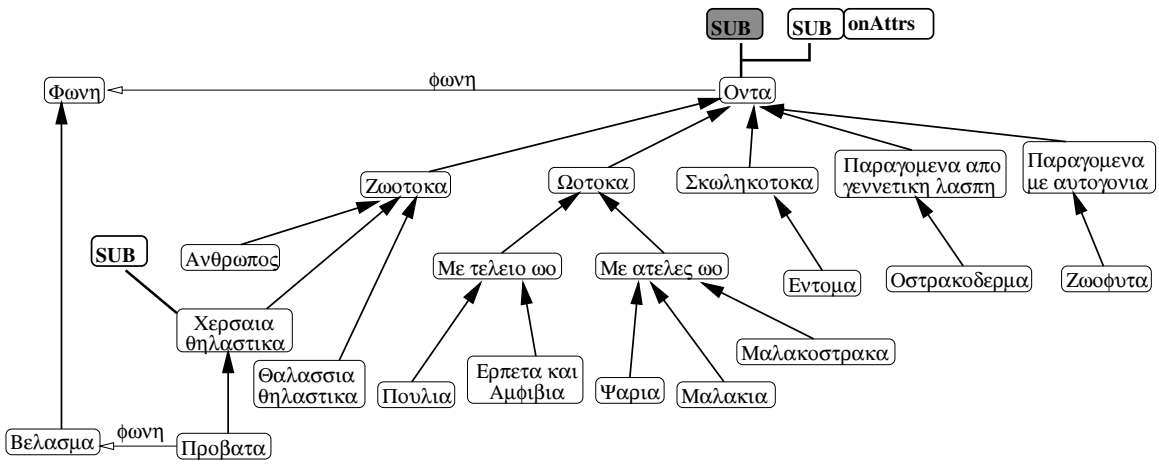
Άλλο ένα σύνθετο παράδειγμα δηλώσεων παρουσιάζεται στο σχήμα 6.51.



Σχήμα 6.49: Αρνητική δήλωση SUB που κληρονομείται.



Σχήμα 6.50: Συνδυασμός θετικών και αρνητικών δηλωσεων SUB.

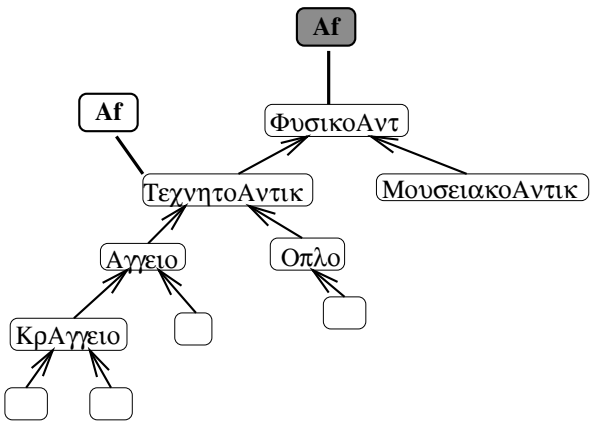


Σχήμα 6.51: Συνδυασμός δηλώσεων SUB.

Στο σχήμα αυτό παρουσιάζεται η ταξινόμηση των φυσικών ειδών (μια ‘‘scala naturae’’ δηλαδή) κατά τον Αριστοτέλη ([38]) με βάση τον τρόπο γένεσης του κάθε γένους. Οι δηλώσεις ενημέρωσης που φαίνονται, επιτρέπουν την εξέλιξη της ιεραρχίας μόνο κάτω από τα *Χερσαία Θηλαστικά*, αλλά επιτρέπει την εξειδίκευση των γνωρισμάτων του αναλλοίωτου τμήματος της ιεραρχίας (όπως το γνώρισμα *φωνή*).

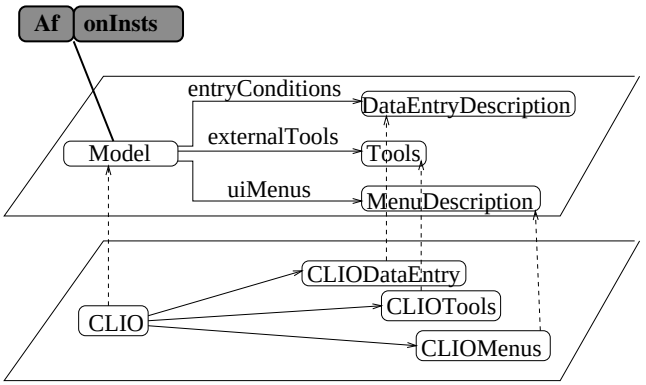
6.7.8 Γνωρίσματα

Το αναγνωριστικό **AF** συμβολίζει το ζευγάρι αναγνωριστικών {AddAF, DelAF}. Ένα σύνθετο παράδειγμα χρήσης του παρουσιάζεται στο σχήμα 6.52. Οι εικονιζόμενες δηλώσεις απαγορεύουν τη ενημέρωση των γνωρισμάτων της κλάσεως *ΦυσικόΑντικείμενο* αλλά την επιτρέπουν στην κλάση *ΤεχνητόΑντικείμενο* και τις υποκλάσεις της.



Σχήμα 6.52: Συνδυασμός δηλώσεων **AF**.

Παράδειγμα χρήσης του αναγνωριστικού **AF** με πεδίο εφαρμογής onInsts παρουσιάζεται στο σχήμα 6.53. Η εικονιζόμενη δήλωση εμποδίζει την ενημέρωση των γνωρισμάτων των περιπτώσεων της κλάσεως *Model*.

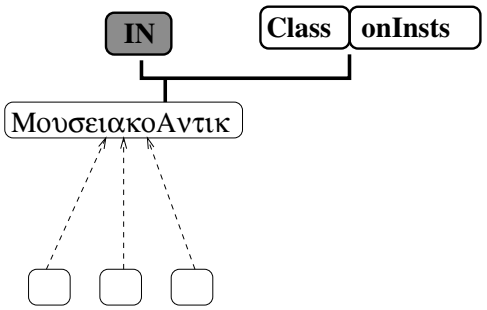


Σχήμα 6.53: Δηλώσεις **AF** με πεδίο εφαρμογής onInsts.

6.7.9 Δηλώσεις *CLASS*

Το αναγνωριστικό **CLASS** συμβολίζει το ζευγάρι αναγνωριστικών {addclass, delclass}. Ένα παράδειγμα χρήσης του παρουσιάζεται στο σχήμα 6.54. Η αρνητική δήλωση IN

αποτρέπει την αλλαγή του πληθυσμού της κλάσης *ΜουσειακόΑντικείμενο.* Η θετική δήλωση CLASS με πεδίο εφαρμογής onInsts, ορίζει τις ΣΕΕ πρόσθεσης/διαγραφής κλάσεων στις περιπτώσεις της κλάσεως *ΜουσειακόΑντικείμενο.* Ο συνδιασμός αυτός ορίζει τις ΣΕΕ αλλαγής της ταξινόμησης των μουσειακών αντικειμένων σε κλάσεις πλην της κλάσης *ΜουσειακόΑντικείμενο.*



Σχήμα 6.54: Δηλώσεις CLASS

6.7.10 Δηλώσεις σε Κλάσεις Συστήματος

Όπως προαναφέραμε, μια δήλωση σε μια κλάση συστήματος κληρονομείται σε όλες τις περιπτώσεις της (άμεσες και έμμεσες), αλλά όπως προκύπτει από τους κανόνες επίλυσης ασαφειών, η δήλωση αυτή είναι η λιγότερο ισχυρή.

Με τέτοιες δηλώσεις μπορούμε να ορίζουμε εύκολα την αρχική (default) κατάσταση ενημέρωσης της βάσης, και μάλιστα επιλεκτικά για κάθε κλάση συστήματος (αν επιθυμούμε κάτι τέτοιο) . Εν συνεχεία μπορούμε, με τις κατάλληλες δηλώσεις, να ορίζουμε τις επιθυμητές εξαιρέσεις στην κατάσταση αυτή. Ακολουθούν παραδείγματα :

- Ορισμός όλων των ενημερώσεων που αφορούν μόνο τις οντότητες επιπέδου $Token$ και των γνωρισμάτων που έχουν αποδοθεί σε αυτές :

$$(Telos_Object, id, \text{NEG}, onObject)$$
$$(Individual_Token, id, \text{POS}, sc)$$
$$\text{όπου } id \in ID \text{ και } sc \in \{onObject, onAttrs\}$$

  Οι παραπάνω δηλώσεις δεν ορίζουν (και αυτό είναι επιθυμητό) τις ΣΕΕ που αφορούν ατομικά γνωρίσματα, αλλά έχουν αποδοθεί σε μη-ατομικά αντίκείμενα (κλάσεις,μετακλάσεις).

- Ορισμός των ενημερώσεων που αφορούν τις οντότητες των επιπέδων $Token$ και S_Class ,των γνωρισμάτων που έχουν αποδοθεί σε αυτές, καθώς και τις ΣΕΕ ταξινόμησης του σχήματος στο μετασχήμα.

$$(Telos_Object, id, \text{NEG}, onObject)$$
$$(M1_Class, AddIn, \text{POS}, onObject)$$
$$(M1_Class, DelIn, \text{POS}, onObject)$$
$$(Individual_Class, id, \text{POS}, sc)$$
$$(Individual_Token, id, \text{POS}, sc)$$
$$\text{όπου } id \in ID \text{ και } sc \in \{onObject, onAttrs\}$$

- Ορισμός των ενημερώσεων δημιουργίας σχέσεων και όχι οντοτήτων.

$$(Individual, id, \text{NEG}, onObject)$$
$$(Attribute, id, \text{POS}, onObject)$$
$$\text{όπου } id \in \{CrObj, DelObj\}$$

Σημειώνουμε επίσης ότι τα αναγνωριστικά ενημέρωσης $CrObj, DelObj$ αφορούν δηλώσεις που γίνονται μόνο σε κλάσεις συστήματος και ελέγχουν τις ΣΕΕ δημιουργίας και διαγραφής αντικειμένων.

6.7.11 Σύνθετες δηλώσεις

Συχνά σε ένα αντικείμενο γίνονται πολλές δηλώσεις. Προκειμένου να εκμεταλευτούμε το γεγονός αυτό το ΜΜΟΕ υποστηρίζει **σύνθετες δηλώσεις**. Μια **απλή δήλωση** έχει τη μορφή :

Δήλωση : <Αντικείμενο, ΤύποςΔήλωσης>
ΤύποςΔήλωσης : <ΑναγνωριστικόΕνημ, Κατάσταση,ΠεδίοΕφαρμογής>

ενώ μια **σύνθετη δήλωση** είναι της μορφής :

ΣύνθετηΔήλωση : <Αντικείμενο, ΣύνθετοςΤύποςΔήλωσης>
ΣύνθετοςΤύποςΔήλωσης : $\{s \mid s \in$ ΤύποςΔήλωσης$\}$

Το ΜΜΟΕ έχει τη δυνατότητα αποθήκευσης των **σύνθετων τύπων**. Αυτό επιτρέπει την αναχρησιμοποίηση τους και την ελαχιστοποίηση του κόπου ορισμού νέων ΟΕ. Πρόκειται στην ουσία για δηλώσεις υψηλότερου επιπέδου.

Ακολουθούν μερικά παραδείγματα **σύνθετων τύπων** :

1. **Ιεραρχία αναλλοίωτου πληθυσμού τιμών**
   Ο τύπος αυτός μαρκάρει ιεραρχίες εξειδίκευσης με αναλλοίωτο σύνολο περιπτώσεων το οποίο χρησιμοποιείται ως τιμές γνωρισμάτων. Η σύνθετη δήλωση περιγράφεται στο σχήμα 6.55

   Παραδείγματα : *Appearance*, *UniversityDepartment*.

2. **Ιεραρχίες ταξινόμησης**.

   Ορίζει τις ΣΕΕ ταξινόμησης σε μια ιεραρχία εξειδίκευσης και αποτρέπει τις ΣΕΕ που αφορούν το σχήμα της βάσης. Η δήλωση αυτή μπορεί να χρησιμοποιηθεί σε μεταεπίπεδο είτε σε επίπεδο σχήματος. Η σύνθετη δήλωση περιγράφεται στο σχήμα 6.56

3. **Ιεραρχίες απόδοσης σύνθετων ιδιοτήτων**

   Η σύνθετη αυτή δήλωση χρησιμοποιείται για τον ορισμό ιεραρχιών εξειδίκευσης οι οποίες δεν χρησιμοποιούνται για την αρχική ταξινόμηση οντοτήτων (δεν είναι κλάσεις δημιουργίας οντοτήτων, ΚΔΟ). Για το λόγο αυτό δεν δηλώνεται το *(IN,onObject,*POS*)* αλλά μόνο το *(IN,onAttrs,*POS*)*. Οι ΣΕΕ ταξινόμησης στην ιεραρχία αυτή δηλώνονται με ένα γνώρισμα τύπου **relatedClasses** το οποίο καταλήγει στην κορυφή της ιεραρχίας. Η σύνθετη δήλωση περιγράφεται στο σχήμα 6.57

   Παραδείγματα : *Style, Abstraction, Operation, Profession*

4. **Σύνθετοι τύποι που αφορούν κλάσεις συστήματος**
   Παράδειγμα θετικής σύνθετης δήλωσης που αφορά κλάσεις συστήματος παρουσιάζεται στο σχήμα 6.58.

Controlled Values			
id	*obj*	*onAttrs*	*onInsts*
IN	NEG	NEG	
AT	POS		POS
AF	NEG	NEG	NEG
SUB	NEG	NEG	NEG
SUP	NEG	NEG	NEG
CLASS	NEG	NEG	NEG
REN	NEG	NEG	NEG
DEL	NEG	NEG	NEG

Σχήμα 6.55: Ιεραρχία αναλλοίωτου πληθυσμού τιμών

Η δυνατότητα **AT** ορίζεται και στο επίπεδο της δήλωσης (νέες κλάσεις μπορούν να αναφέρουν αυτήν την ιεραρχία) και στο επίπεδο των περιπτώσεων τους.

ClassificationHierarchy			
id	*onObject*	*onAttrs*	*onInsts*
IN	POS	POS	
AT	NEG		POS
AF	NEG	NEG	POS
SUB	NEG	NEG	POS
SUP	NEG	NEG	POS
CLASS	NEG	NEG	POS
REN	NEG	NEG	POS
DEL	NEG	NEG	POS

Σχήμα 6.56: Ιεραρχίες ταξινόμησης

ComplexAttributeHierarchy			
id	*onObject*	*onAttrs*	*onInsts*
IN		POS	
AT	NEG		
AF	NEG	NEG	
SUB	NEG	NEG	
SUP	NEG	NEG	
CLASS	NEG	NEG	
REN	NEG	NEG	
DEL	NEG	NEG	

Σχήμα 6.57: Ιεραρχίες απόδοσης σύνθετων ιδιοτήτων

Σημειώνουμε επίσης ότι το σύνολο των σύνθετων τύπων δεν είναι κλειστό αλλά μπορεί να επεκτείνεται και να εξελίσσεται ανάλογα με τις απαιτήσεις που προκύπτουν.

PositiveSysClass			
id	*onObject*	*onAttrs*	*onInsts*
CrObj	POS		
DelObj	POS		
IN	POS	POS	
AT	POS		
AF	POS	POS	
SUB	POS	POS	
SUP	POS	POS	
CLASS	POS	POS	
REN	POS	POS	
DEL	POS	POS	

Σχήμα 6.58: Σύνθετοι τύποι κλάσεων συστήματος

Ο τύπος *PositiveSysClass* αποτελείται από ένα σύνολο θετικών δηλώσεων και περιλαμβάνει δηλώσεις με αναγνωριστικά ενημέρωσης $CrObj$, $DelObj$.

6.8 Όψεις επιλογής και Ενημερώσεις

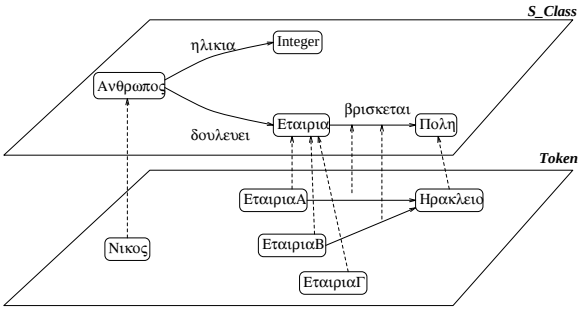
Ένα υποσύνολο των περιπτώσεων μιας κλάσης το οποίο ικανοποιεί μία συνθήκη συχνά αναφέρεται με το όνομα *όψη επιλογής* (selection view). Το MMOE που παρουσιάζεται δεν υποστηρίζει όψεις επιλογής. Η εισαγωγή δεδομένων μέσω όψεων επιλογής, όπως αναφέρθηκε και στο κεφάλαιο 2, θέτει το ερώτημα για το αν θα πρέπει η νέα οντότητα που εισάγεται, να ικανοποιεί την συνθήκη, να είναι δηλαδή μέλος της όψεως επιλογής. Οι όψεις αυτές θα διευκόλυναν την εξέλιξη γνώσης αν το σύστημα κατά την εισαγωγή ενός αντικειμένου μπορούσε αυτόματα (ή ημιαυτόματα) να δημιουργήσει τις απαραίτητες αναφορές ή άλλες οντότητες προκειμένου το νέο στοιχείο να ικανοποιεί την συνθήκη.

Αν η συνθήκη περιέχει έναν τελεστή διάζευξης ή διάταξης ή είναι σύνθετη δεν μπορούμε να καθορίσουμε με σαφήνεια τις ενέργειες που πρέπει να πραγματοποιηθούν σε μια οντότητα ώστε η συνθήκη επιλογής να ικανοποιείται για αυτήν την οντότητα Έστω το σχήμα που φαίνεται στο σχήμα 6.59 και έστω οι εξής όψεις επιλογής της κλάσης *Άνθρωπος* :

SELECTION VIEW *ΠροστατεύομενοΜέλος*
FROM α **IN** *Άνθρωπος*
WHERE α.*ηλικία* $<$ *18* **OR** α.*ηλικία* $>$ *75*

Η ηλικία που πρέπει να αποδοθεί σε μια νέα περίπτωση της *ΠροστατεύομενοΜέλος* δεν μπορεί να αποφασιστεί αυτόματα.

Έστω επίσης η όψη επιλογής *ΕργαζόμενοςΗρακλείου* του μοντέλου που παρουσιάζεται στο σχήμα 6.59 :



Σχήμα 6.59: Μοντέλο εφαρμογής της όψης επιλογής *ΕργαζόμενοςΗρακλείου.*

SELECTION VIEW *ΕργαζόμενοςΗρακλείου*
FROM α **IN** *Άνθρωπος*
WHERE $\alpha.$*δουλεύει.βρίσκεται* $=$ *Ηράκλειο*

Στην περίπτωση αυτή δεν μπορούμε να αποφασίσουμε την τιμή που πρέπει να αποδώσουμε στο γνωρίσμα *δουλεύει* μια νέας περίπτωσης της *ΕργαζόμενοςΗρακλείου.* Μπορεί να είναι μία από τις υπάρχουσες εταιρίες (ποιά όμως ;) ή καμία από αυτές (μια εταιρία που δεν έχει ακόμα περιγραφεί στη βάση) .

Η μελέτη του θέματος απαιτεί την ύπαρξη μιας ερωτηματικής γλώσσας (η SIS-Telos δεν διαθέτει ακόμα) η οποία θα μετέφραζε την συνθήκη σε ένα σύνολο από κατηγορήματα. Κατά την εισαγωγή δεδομένων και αν υπήρχε ασάφεια, το σύστημα θα ρωτούσε το χρήστη παραπέρα πληροφορίες όσπου να ικανοποιηθούν τα κατηγορήματα που είναι απαραίτητα για την ικανοποίηση της συνθήκης της κλάσεως επιλογής.

Πιστεύουμε όμως ότι το πρόβλημα αντιμετωπίζεται σε ικανοποιητικό βαθμό με την έννοια του **υποδείγματος** και την **εντολή παραγωγής αντιγράφων** ή **αναπαραγωγής** (**MakeCopy**). Το υπόδειγμα είναι ένα αντιπροσωπευτικό αντικείμενο (απλό ή σύνθετο) το οποίο ικανοποιεί μια συνθήκη επιλογής. Μια κλάση μπορεί να έχει περισσότερα από ένα υποδείγματα, από τα οποία ο χρήστης μπορεί να επιλέξει το επιθυμητό.

Η **MakeCopy** μπορεί να δημιουργεί αντίγραφα αντικειμένων, ακόμα και σύνθετων. Είναι πολύ χρήσιμη κατά την εισαγωγή ατομικών αντικειμένων σε μια βάση αφού συχνά είναι ευκολότερη η διαφοροποίηση μιας οντότητας από μια άλλη από την πλήρη δήλωση της. Ο τρόπος αυτός πολλαπλασιασμού χρησιμοποιείται άλλωστε και στη φύση (μιτωτική διαίρεση κυττάρων).

Για να δουλέψει η **MakeCopy** ικανοποιητικά σε σύνθετα αντικείμενα απαιτείται η εισαγωγή στην SIS-Telos των **ειδικών κατηγοριών γνωρισμάτων** που προτείνονται και

περιγράφονται στο [25]. Αν και αυτό δεν έχει γίνει ακόμα, στην παρακάτω ενότητα γίνεται μελέτη της **MakeCopy** λαμβάνοντας υπόψη τις ειδικές κατηγορίες γνωρισμάτων.

6.8.1 Η εντολή MakeCopy

Εκτελώντας την **MakeCopy** για μια οντότητα Α, δημιουργείται μια νέα οντότητα Α′ της ίδιας κλάσης συστήματος, στην οποία εκχωρούνται οι κλάσεις και οι υπερκλάσεις της Α. Η μελέτη της αναπαραγωγής των γνωρισμάτων του A, προϋποθέτει την ύπαρξη των ειδικών κατηγοριών γνωρισμάτων που προτείνει ο Θεοδωράκης [25] :

	Τυποι Κατηγοριων Γνωρισματων	Παραδειγματα
1	f συναρτηση	Ανθρωπος $\xrightarrow[\text{επιθετο}]{\text{1:1 f 0:ν}}$ Επιθετο
2	f 1–1	ΑδειαΟδηγησης $\xrightarrow[\text{ανηκει}]{\text{1:1 f 0:1}}$ Ανθρωπος
3	f Επι	Βουλευτης $\xrightarrow[\text{εκπροσωπει}]{\text{1:1 f 1:ν}}$ ΕκλογικηΠεριφερεια
4	f 1–1 και Επι	Ανθρωπος $\xrightarrow[\text{καρδια}]{\text{1:1 f 1:1}}$ Καρδια
5	f^{-1} συναρτηση	Επιθετο $\xrightarrow[\text{αφορα}]{\text{0:ν f 1:1}}$ Ανθρωπος
6	f^{-1} 1–1	Ανθρωπος $\xrightarrow[\text{αδειαΟδ}]{\text{0:1 f 1:1}}$ ΑδειαΟδηγησης
7	f^{-1} Επι	ΕκλογικηΠεριφερεια $\xrightarrow[\text{εκλεγει}]{\text{1:ν f 1:1}}$ Βουλευτης
8	απλη σχεση	Ανθρωπος $\xrightarrow[\text{ασκει}]{\text{0:ν f 0:ν}}$ Επαγγελμα

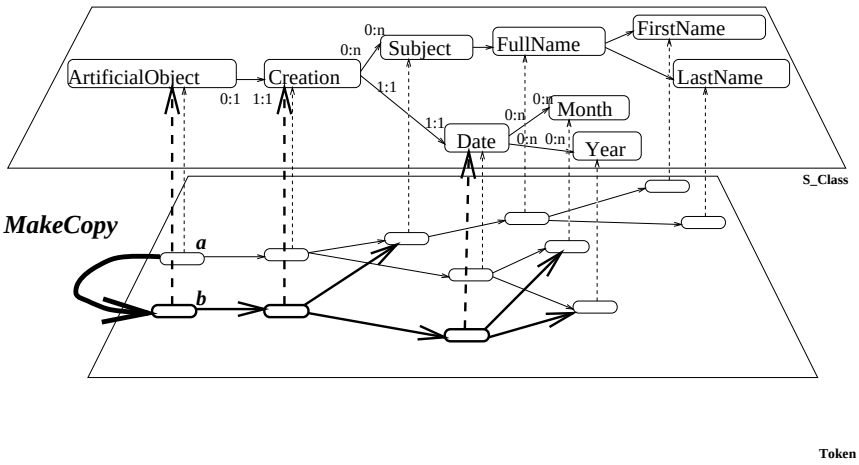
Σχήμα 6.60: Τύποι κατηγοριών γνωρισμάτων

Έστω οτι η οντότητα Α συνδέεται μέσω ενός γνωρίσματος l, με μια οντότητα B, και έστω f το είδος της σχέσης του γνωρίσματος l. Το ερώτημα που τίθεται είναι αν θα πρέπει να δημιουργηθεί ένα αντίγραφο του l, δηλαδή ένα νέο γνώρισμα l', και αν ναι, τότε ποιά θα πρέπει να είναι η τιμή του, δηλαδή το B'. Η απάντηση παρουσιάζεται στον πίνακα που ακολουθεί. Παρατηρούμε ότι η εντολή αναπαραγωγής είναι ημιαυτόματη αφού ο χρήστης σε ορισμένες περιπτώσεις πρέπει να επιλέξει τι επιθυμεί : την μη-αντιγραφή ενός γνωρίσματος (συμβολίζεται με $\nexists\, l'$) ή την αντιγραφή του οπότε σε αυτή την περίπτωση θα πρέπει να ονομάσει την τιμή του γνωρίσματος (συμβολίζεται με $new(username)$). Ο συμβολισμός $new(noname)$ συμβολίζει ένα νέο αντικείμενο το οποίο δημιουργείται

αυτόματα και στο οποίο δεν είναι ανάγκη να αποδοθεί κανένα λογικό όνομα (αυτό θα είναι εφικτό όταν υλοποιηθεί ο μηχανισμός ονοματοδοσίας που προτείνει ο Θεοδωράκης στο [25]).

Τύπος f	**Τιμή B' του l'**
1	B
2	$\nexists\ l'$ ή $new(username)$
3	B
4	$new(noname)$
5	$\nexists\ l'$ ή $new(username)$
6	$new(noname)$
7	$\nexists\ l'$ ή $new(username)$
8	B

Εν συνεχεία η **MakeCopy** καλείται αναδρομικά για κάθε νέο αντικείμενο l', B' που δημιουργείται. Παρουσιάζονται ενδεικτικά δύο απλά παραδείγματα στα σχήματα 6.61 και 6.62.

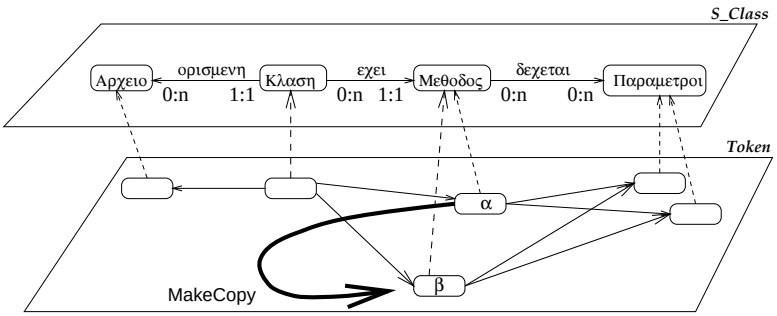


Σχήμα 6.61: Παράδειγμα εκτέλεσης εντολής **MakeCopy** .

Παρατηρούμε ότι δημιουργούνται νέοι κόμβοι τύπου *Creation*, *Date* λόγω των ειδικών περιορισμών των αντίστοιχων γνωρισμάτων.

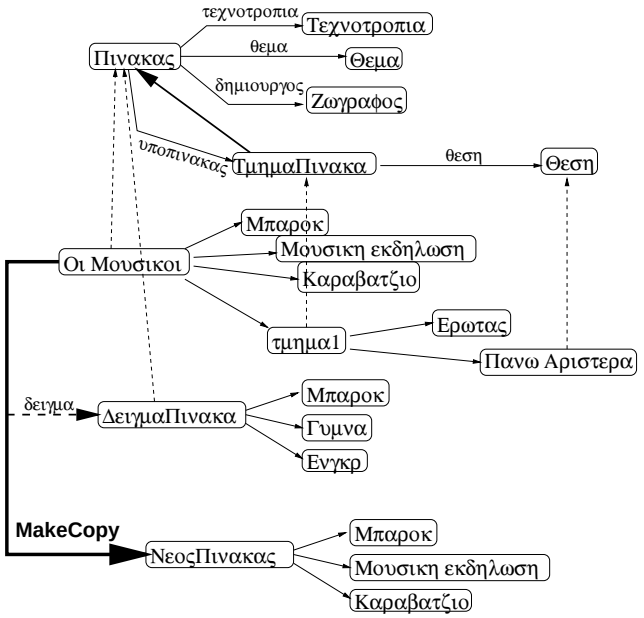
.

Η εντολή της αναπαραγωγής μπορεί να επεκταθεί ώστε να αναπαράγει ένα αντικείμενο (το πρωτότυπο) λαμβάνοντας υπόψη ένα υπόδειγμα. Το υπόδειγμα υποδεικνύει τις πληροφορίες που πρέπει να αναπαραχθούν: αναπαράγονται οι πληροφορίες του πρωτοτύπου που έχουν κοινό τύπο (κατηγορία) με τις πληροφορίες που έχει το υπόδειγμα (δες σχήμα 6.63). Έτσι ελέγχεται το είδος και το εύρος των ενημερώσεων που γίνονται με μια εντολή



Σχήμα 6.62: Παράδειγμα εκτέλεσης εντολής **MakeCopy** .

αναπαραγωγής. Σε άλλα συστήματα και προτάσεις (όπως στην [6]), αυτό επιτυγχάνεται με τον ορισμό εικονικών κλάσεων στις οποίες αποδίδονται εικονικά μονοπάτια γνωρισμάτων.



Σχήμα 6.63: **MakeCopy** με βάση ένα υπόδειγμα.

Αν η **MakeCopy** δεν γινόταν μέ βάση το υπόδειγμα τότε θα δημιουργόταν επίσης ένας υποπίνακας για τον νέο πίνακα.

6.9 Υλοποίηση

Στην ενότητα αυτή μελετώνται θέματα υλοποίησης που αφορούν την αξιοποίηση του MMOE . Η χρήση της SIS-Telos για την παράσταση και την αποθήκευση του μεταμοντέλου, μας απαλάσσει από το κόστος ανάπτυξης ειδικών μηχανισμών αποθήκευσης και ανάκλησης του MMOE και των δηλώσεων του.

Οι τύποι δήλωσης αποθηκεύονται στη βάση σαν αντικείμενα της SIS-Telos , ενώ

οι δηλώσεις σαν σύνδεσμοι μεταξύ των αντικειμένων της εφαρμογής και των τύπων δήλωσης. Αυτό μας απαλάσσει από το χρόνο υλοποίησης σε χαμηλό επίπεδο άλλου είδους σχέσεων. Επίσης χρησιμοποιώντας τον μηχανισμό ερωτήσεων της SIS-Telos (**qi**) μπορούμε να ανακαλέσουμε αποδοτικά [10] (από οποιαδήποτε κατεύθυνση) τα μεταδεδομένα (δηλώσεις του MMOE) και να χρησιμοποιήσουμε την επαφή χρήσεως του SIS για την εποπτεία των δηλώσεων.

Στις επόμενες ενότητες μελετάται η διαδικασία ερμηνείας των δηλώσεων του MMOE , η οποία είναι αναπόσπαστο τμήμα κάθε μηχανισμού αξιοποίησης των OE.

Στο επόμενο κεφάλαιο παρουσιάζεται μια συγκεκριμένη και ολοκληρωμένη πρόταση για την διαλογική ενημέρωση δεδομένων της SIS-Telos και της χρήσης των OE σε αυτήν.

6.9.1 Αλγόριθμος Εύρεσης Δηλώσεων Αντικειμένου

Στην ενότητα αυτή παρουσιάζεται ο αλγόριθμος ο οποίος υπολογίζει και επιστρέφει τις δηλώσεις που ανήκουν στο FD_t^{all} και αφορούν ένα συγκεκριμένο αντικείμενο. Με άλλα λόγια παρουσιάζεται ο τρόπος υπολογισμού της τιμής της συνάρτησης :

$$GetIdState : O \times ID \rightarrow State$$

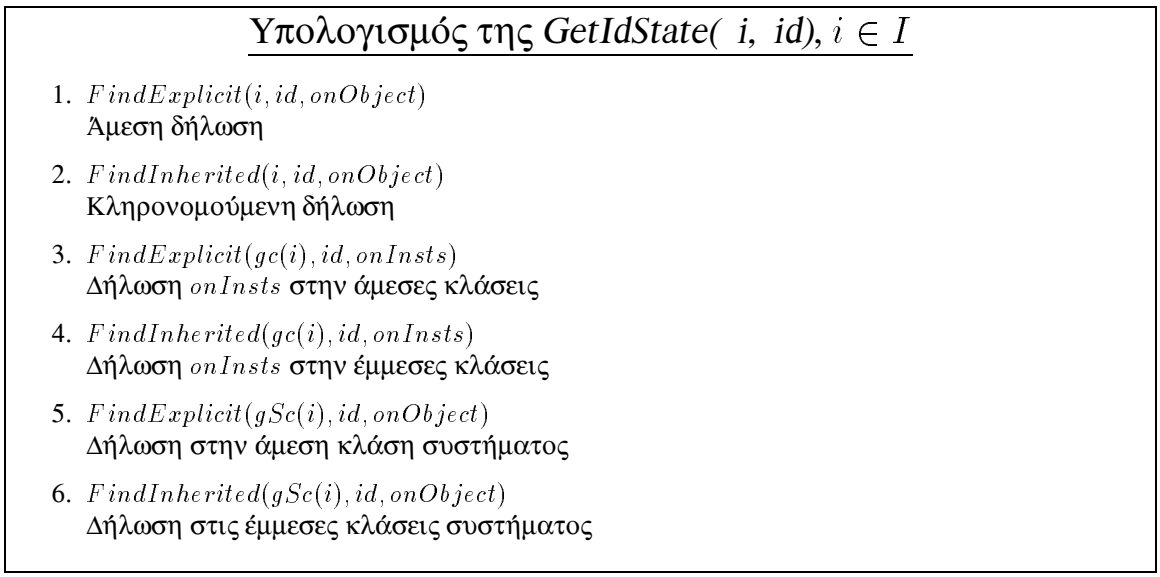
η οποία δέχεται σαν όρισμα ένα αντικείμενο και ένα ID και επιστρέφει την κατάσταση (state) της σχετικής δήλωσης του FD_t^{all}. Για την παρουσίαση του τρόπου υπολογισμού της αρχικά περιγράφονται δύο συναρτήσεις :

$$FindExplicit : 2^O \times ID \times Target \rightarrow \{\text{POS}, \text{NEG}, \text{NONE}\}$$
$$FindInherited : 2^O \times ID \times Target \rightarrow \{\text{POS}, \text{NEG}, \text{NONE}\}$$

Η τιμή της συνάρτησης $FindExplicit(objectSet, id, sc)$ υπολογίζεται από μια διαδικασία η οποία ψάχνει σε κάθε αντικείμενο του $objectSet$ για μια δήλωση με αναγνωριστικό id και πεδίο εφαρμογής sc. Αν δεν υπάρχει **καμία τέτοια δήλωση** επιστρέφει την τιμή NONE. Αν βρεθεί **μια δήλωση** επιστρέφει την $state$ της, αν βρεθούν περισσότερες από μία και σύμφωνες επιστρέφει την $state$ τους, ενώ αν βρεθούν περισσότερες από μία και **συγκρουόμενες** δηλώσεις τότε επιστρέφει την τιμή NEG.

Η τιμή της συνάρτησης $FindInherited(objectSet, id, sc)$ υπολογίζεται από μια διαδικασία η οποία ψάχνει να βρεί τις κληρονομούμενες (λόγω εξειδίκευσης) δηλώσεις σε κάθε αντικείμενο του $objectSet$ με αναγνωριστικό id και πεδίο εφαρμογής sc. Αν δεν υπάρχει **καμία τέτοια δήλωση** επιστρέφει την τιμή NONE. Αν βρεθεί κληρονομούμενη **μια δήλωση**

[10] Μια αναδρομική διάσχιση 3000 συνδέσμων της Telos διαρκεί 2 δευτερόλεπτα.



Σχήμα 6.64: Αλγόριθμος υπολογισμού της *GetIdState* για οντότητα.

επιστρέφει την $state$ της, αν βρεθούν περισσότερες από μια και σύμφωνες τότε επιστρέφει την $state$ τους, ενώ αν βρεθούν περισσότερες από μια και **συγκρουόμενες** κληρονομούμενες δηλώσεις τότε επιστρέφει την κατάσταση της **πιο ειδικής κλάσης**. Αν δεν υπάρχει μία μόνο ειδικότερη αλλά περισσότερες και συγκρουόμενες τότε επιστρέφει την τιμή NEG.

Με τη βοήθεια των δύο παραπάνω συναρτήσεων στα σχήματα 6.64 και 6.65 παρουσιάζεται ο αλγόριθμος υπολογισμού της τιμής της $GetIdState$ ο οποίος μάλιστα διαφοροποιείται ανάλογα με τον τύπο του αντικειμένου που παρέχεται ως όρισμα (οντότητα, γνώρισμα). Ο αλγόριθμος τερματίζει μόλις η εκτέλεση κάποιου βήματος επιστρέψει μια τιμή διαφορετική της NONE.

Με το όνομα $fromClass(a)$ συμβολίζουμε την κλάση υπό την οποία εξετάζουμε το γνώρισμα a (πρόκειται για την κλάση στην οποία έχει αποδοθεί το γνώρισμα a ή μια υποκλάση αυτής).

Οι αλγόριθμοι αυτοί υλοποιήθηκαν και μια πρωτότυπη εφαρμογή τους παρουσιάζεται στο σχήμα 6.66.

Υπολογισμός της *GetIdState(a, id)* , $a \in A$

1. $FindExplicit(a, id, onObject)$
   Άμεση δήλωση
2. $FindInherited(a, id, onObject)$
   Κληρονομούμενη δήλωση
3. $FindExplicit(gc(a), id, onInsts)$
   Δήλωση $onInsts$ στην άμεσες κλάσεις
4. $FindInherited(gc(a), id, onInsts)$
   Δήλωση $onInsts$ στην έμμεσες κλάσεις
5. $FindExplicit(fromClass(a), id, onAttrs)$
   Δήλωση onAttrs στην $fromClass(a)$
6. $FindInherited(fromClass(a), id, onAttrs)$
   Κληρονομούμενη δήλωση onAttrs στην $fromClass(a)$
7. $FindExplicit(gSc(fromClass(a)), id, onAttrs)$
   Δήλωση onAttrs στην άμεση κλάση συστήματος της $fromClass(a)$
8. $FindInherited(gSc(fromClass(a)), id, onAttrs)$
   Κληρονομούμενη δήλωση onAttrs στην κλάση συστήματος της $fromClass(a)$
9. $FindExplicit(gSc(a), id, onObject)$
   Δήλωση στην άμεση κλάση συστήματος
10. $FindInherited(gSc(a), id, onObject)$
    Δήλωση στις έμμεσες κλάσεις συστήματος

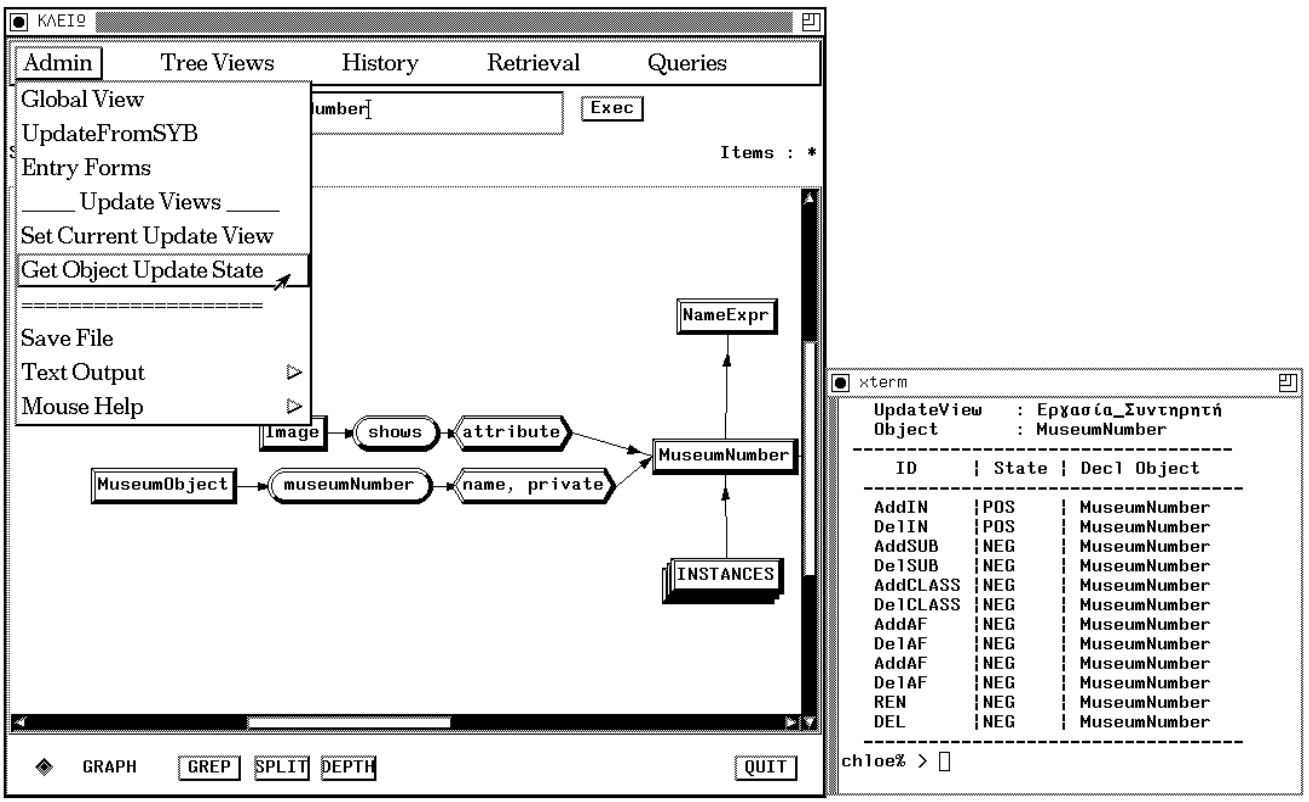
Σχήμα 6.65: Αλγόριθμος υπολογισμού της *GetIdState* για γνώρισμα.

6.9.2 Αλγόριθμος Εύρεσης Κατηγορημάτων Αντικειμένου

Η εύρεση των κατηγορημάτων που αφορούν ένα αντικείμενο υπολογίζεται από τον προηγούμενο αλγόριθμο εκτός από τα κατηγορήματα με αναγνωριστικό $AddIn, DelIn$. Στο σχήμα 6.67 παρουσιάζεται ο σχετικός αλγόριθμος ενώ σχετικό παράδειγμα παρουσιάστηκε στην ενότητα 6.7.6 στο σχήμα 6.47.

6.9.3 Πολυπλοκότητα Αλγορίθμων

Η επαγωγή των δηλώσεων που αφορούν μια οντότητα γίνεται διασχίζοντας ένα σύνολο από συνδέσμους της SIS-Telos . Η διαδικασία αυτή μπορεί να γίνει αρκετά γρήγορα αφού οι διασχίσεις συνδέσμων στην SIS-Telos είναι αρκετά γρήγορες. Οι απαιτήσεις σε ταχύτητα θα ήταν πολύ μεγαλύτερες αν το ΜΜΟΕ ήταν υπεύθυνο και για τον



Σχήμα 6.66: Πρωτότυπη εφαρμογή των αλγορίθμων του ΜΜΟΕ

Το μενού *Admin* της επαφής χρήσεως του SIS επεκτάθηκε με δύο λειτουργίες:

- **Set Current Update View** η οποία θέτει την τρέχουσα ΟΕ.
- **Get Update State Of** η οποία υπολογίζει και τυπώνει σε ένα ξεχωριστό παράθυρο τις καταστάσεις για όλα τα ID της εστιακής οντότητας όσο αφορά την τρέχουσα ΟΕ.

Διερεύνηση $< obj, class, id > \in Pred_t$ όπου $id \in \{AddIn, DelIn\}$

Ισχύει $id \in \{AddIn, DelIn\}$

1. Εύρεση της κλάσης με την κυρίαρχη δήλωση με αναγνωριστικό id που αφορά τη $class$. Έστω A αυτή η κλάση.
2. Εύρεση της κλάσης στην οποια καταλήγει το ειδικότερο γνώρισμα τύπου *relatedClasses* ,έστω l, το οποίο έχει δήλωση με αναγνωριστικό *id.* Έστω B η κλάση αυτή.
3. Αν $\nexists\ l$ ή $obj \notin gai(gfv(l))$ τότε
   $< obj, class, id > \in Pred_t$ αν $(class, id, onObject, \text{POS}, t) \in FD_t^{all}$
4. Αν $\exists\ l\ \wedge\ obj \in gai(gfv(l))$ τότε
   (a) Αν $A\ isA\ B$
   $< obj, class, id > \in Pred_t$ ανν $(A, id, onObject, \text{POS}, t) \in FD_t^{all}$
   (b) αλλιώς :
   $< obj, class, id > \in Pred_t$ ανν $(l, id, onObject, \text{POS}, t) \in FD_t^{all}$

Σχήμα 6.67: Αλγόριθμος υπολογισμού των κατηγορημάτων με $AddIn, DelIn$.

έλεγχο των προσπελάσεων ανάγνωσης στην βάση (τότε θα επρόκειτο για ένα πλήρη μηχανισμό δικαιοδοσίας προσπελάσεων), διαφορετικά η ταχύτητα απάντησης ερωτήσεων θα μειωνόταν δραματικά.

Ακολουθεί ο υπολογισμός της χρονικής πολυπλοκότητας της διαδικασίας επαγωγής των δηλώσεων που αφορούν μια οντότητα ενώ στην ενότητα 6.9.4 προτείνεται ένας μηχανισμός για βελτίωση των επιδόσεων.

Πολυπλοκότητα της $FindInherited$

Θα συμβολίζουμε με w_{FE} το κόστος εύρεσης μιας άμεσης δήλωσης (που αναφέρεται σε ένα ID και σε μια OE) ενός αντικειμένου. Πρόκειται δηλαδή για το κόστος της κλήσης της $FindExplicit$ με παράμετρο ένα μόνο αντικείμενο. Το κόστος αυτό είναι σταθερό.

Με w_{FI} συμβολίζουμε το κόστος κλήσης της $FindInherited$ με παράμετρο ένα μόνο αντικείμενο. Στον υπολογισμό του w_{FI} που παρουσιάζεται στον πίνακα που ακολουθεί, χρησιμοποιούνται οι ακόλουθοι συμβολισμοί :

- w_q : κόστος μιας απλής μονοβάθμιας ερώτησης (σταθερό).
- d_{av} : μέσος αριθμός υπερκλάσεων μιας κλάσης.

- $decl_{av}$: μέσος αριθμός κληρονομούμενων δηλώσεων (της ίδιας ΟΕ και με το ίδιο ID) από υπερκλάση μιας κλάσης.
- K : μια σταθερά
- $gilf_{decl}$: η ερώτηση η οποία επιστρέφει τα κληρονομούμενα γνωρίσματα που είναι δηλώσεις του ΜΜΟΕ (ανήκουν στην κατηγορία *updateDecl from Telos_Object*) και ανήκουν στην τρέχουσα ΟΕ.

FindInherited		
Βήμα		Κόστος
1	$set_1 = gilf_{decl}$	$w_{FE} * d_{av}$
2	$set_2 = gfv(set_1)$	$w_q * decl_{av}$
3	$set_3 = gasc(set_2)$	$w_q * decl_{av} * d_{av}$
4	$set_4 = set_2 - set_3$	K

$$\text{Άρα } w_{FI} = w_{FE} * d_{av} + w_q * decl_{av}(1 + d_{av}) + K \tag{6.1}$$

Θα κάνουμε τώρα κάποιες προσεγγιστικές εκτιμήσεις. Συμβολίζουμε w_{FIi}^{est} το μέσο κόστος της $FindInherited$ με παράμετρο οντότητα. Για την περίπτωση αυτή αν κάνουμε την εκτίμηση ότι $d_{av} = 5$ και $decl_{av} = 1$, προκύπτει :

$$w_{FIi}^{est} = 5 * w_{FE} + 6 * w_q \tag{6.2}$$

Συμβολίζοντας w_{FIa}^{est} το μέσο κόστος της $FindInherited$ με παράμετρο γνώρισμα και κάνοντας την προσέγγιση ότι $d_{av} = 1$ και $decl_{av} = 1$ προκύπτει :

$$w_{FIa}^{est} = w_{FE} + 2 * w_q \tag{6.3}$$

Πολυπλοκότητα της $GetIDState(i \in I)$

Συμβολίζουμε με w_{Gi} το κόστος υπολογισμού της $GetIDState$ για μια οντότητα. Με c_{av} θα συμβολίζουμε τον μέσο αριθμό κλάσεων μιας οντότητας.

$GetIDState(i \in I)$			
βήμα	κόστος	βήμα	κόστος
1	w_{FE}	4	$w_{FI} * c_{av}$
2	w_{FI}	5	w_{FE}
3	$w_{FE} * c_{av}$	6	w_{FI}

$$\text{Άρα}: \; w_{Gi} \;=\; (2 + c_{av}) * (w_{FE} + w_{FI}) \tag{6.4}$$

Κάνοντας την προσέγγιση ότι $c_{av} = 1$ προκύπτει :

$$w_{Gi}^{est} \;=\; 3 * (w_{FE} + w_{FI}) \overset{6.2}{\Rightarrow} \tag{6.5}$$

$$w_{Gi}^{est} \;=\; 18 * w_{FE} + 18 w_q \tag{6.6}$$

Πολυπλοκότητα της $GetIDState(a \in A)$

Συμβολίζουμε με w_{Ga} το κόστος υπολογισμού της $GetIDState$ για ένα γνώρισμα. Με c_{av}^a θα συμβολίζουμε τον μέσο αριθμό κλάσεων ενός γνωρίσματος.

$GetIDState(a \in A)$			
βήμα	κόστος	βήμα	κόστος
1	w_{FE}	6	w_{FIi}
2	w_{FIa}	7	w_{FE}
3	$w_{FE} * c_{av}^a$	8	w_{FIi}
4	$w_{FIa} * c_{av}^a$	9	w_{FE}
5	w_{FE}	10	w_{FIi}

$$w_{Ga} = (4 + c_{av}^a) * w_{FE} + (1 + c_{av}^a) * w_{FIa}^a + 3 * w_{FIi}$$

Κάνοντας την προσέγγιση $c_{av}^a = 2$ προκύπτει [11]:

$$w_{Ga}^{est} \;=\; 3 * (2 * w_{FE} + w_{FIa} + w_{FIi}) \overset{6.3}{\Rightarrow} \tag{6.7}$$

$$w_{Ga}^{est} \;=\; 24 * w_{FE} + 36 w_q \tag{6.8}$$

6.9.4 Βελτιστοποίηση επιδόσεων

Στην ενότητα αυτή παρουσιάζονται κάποιες προτάσεις για την βελτίωση του χρόνου υπολογισμού της $GetIdState$, παρόλου που ο χρόνος αυτός στην Telos είναι ελάχιστος.

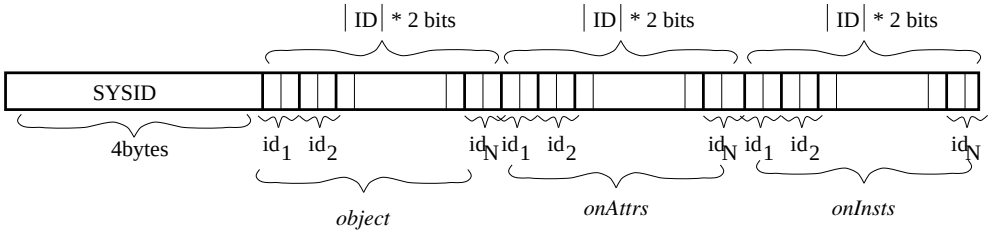
Μια λύση είναι η χρήση μιας **Κρυφής Μνήμης Επαγόμενων Δηλώσεων (ΚΜΕΔ)**. Σκοπός της είναι η ελαχιστοποίηση του χρόνου που απαιτείται για την εύρεση των

[11] Προσεγγίζουμε το c_{av}^a με 2, ενώ το c_{av} με 1, γιατί στα γνωρίσματα η πολλαπλή ταξινόμηση εμφανίζεται συχνότερα από ότι στις οντότητες.

δηλώσεων που αφορούν ένα αντικείμενο. Για το σκοπό αυτό σε αυτήν αποθηκεύονται επαγόμενες δηλώσεις αντικειμένων με σκοπό τη μελλοντική χρησιμοποίηση τους και όχι τον επανυπολογισμό τους. Η λύση της κρυφής μνήμης επιλέχθηκε γιατί μπορεί να προσφέρει ικανοποιητικά αποτελέσματα και δεν αυξάνει υπερβολικά το μέγεθος της βάσης .

Τα στοιχεία μνήμης της προτεινόμενης ΚΜΕΔ περιγράφονται στο σχήμα 6.68. Κάθε στοιχείο της ΚΜΕΔ περιέχει το SYSID του αντικειμένου που αφορά, και δύο bit για κάθε ID και για κάθε πεδίο εφαρμογής ($onObject,\ onAttrs,\ onInsts$). Στα bits που αντιστοιχούν σε ένα ID και ένα πεδίο εφαρμογής αποθηκεύεται η κατάσταση της αντίστοιχης κληρονομούμενης δήλωσης ($\xrightarrow{isA}$), αν υπάρχει. Άρα καταλαβαίνουμε ότι η ΚΜΕΔ είναι μια κρυφή μνήμη αποθήκευσης των στοιχείων του $D_t^{inh} - D_t$. Η σημασία των αποθηκευόμενων bits παρουσιάζεται στον ακόλουθο πίνακα :

bits	Σημασία
00	Θετική
01	Αρνητική
10	Ανύπαρκτη
11	Άκυρη



Σχήμα 6.68: Τα στοιχεία της Κρυφής Μνήμης Επαγόμενων Δηλώσεων

Το γέμισμα και η χρήση των τιμών της ΚΜΕΔ γίνεται από τη συνάρτηση $FindInherited$. Η διαδικασία αυτή ψάχνει αρχικά τη ζητούμενη δήλωση στην ΚΜΕΔ. Αν δεν την βρεί εκεί, τότε την υπολογίζει κανονικά και στο τέλος εμπλουτίζει την ΚΜΕΔ με την τιμή της. Βέβαια τα δεδομένα που αποθηκεύονται στην ΚΜΕΔ έχουν δυναμική φύση αφού μεταβάλλονται είτε γιατί οι δηλώσεις του ΜΜΟΕ μεταβάλλονται, είτε γιατί η βάση ενημερώνεται. Επειδή όμως η επικαιρότητα των δεδομένων της ΚΜΕΔ είναι απαραίτητη, πρέπει να ακυρώνονται τα δεδομένα που την χάνουν. Αλλοίωση της επικαιρότητας των τιμών της ΚΜΕΔ προκύπτει όταν πραγματοποιούνται :

1. **Ενημέρωση των δηλώσεων του Μεταμοντέλου** (D_t)

- *Κρίσιμα γεγονότα ενημέρωσης*
  Οποιαδήποτε αλλαγή του D_t (προσθήκη/διαγραφή/αλλαγή δήλωσης, τροποποίηση ενός σύνθετου τύπου).

- *Στοιχεία προς ακύρωση*
  Εξαρτάται από το είδος της αλλαγής. Για παράδειγμα η ενημέρωση ενός σύνθετου τύπου, μπορεί να έχει ως συνέπεια την ακύρωση πολλών στοιχείων.

- *Ειδοποίηση κρίσιμων γεγονότων ενημέρωσης*
  Είτε μέσω του πυρήνα της SIS-Telos , είτε μέσω του εργαλείου με το οποίο ενημερώνονται οι δηλώσεις.

- *Αντίδραση μηχανισμού*
  Ακύρωση των περιεχομένων ολόκληρης της ΚΜΕΔ. Πρόκειται για την καλύτερη και ευκολότερη λύση αφού οι δηλώσεις δεν ενημερώνονται συχνά. Εξάλλου η επιλεκτική ακύρωση μεμονομένων στοιχείων της ΚΜΕΔ, ενώ δεν θα προσέφερε σημαντικό πλεονέκτημα, θα προσέθετε πολυπλοκότητα στο μηχανισμού.

2. **Ενημέρωση της βάσης**

- *Κρίσιμα γεγονότα ενημέρωσης*
  Ενημέρωση των ιεραρχιών εξειδίκευσης (προσθήκη/διαγραφή/αλλαγή συνδέσμων εξειδίκευσης).

- *Στοιχεία προς ακύρωση*
  Τα στοιχεία που αφορούν τις υποκλάσεις (άμεσες και έμμεσες) των αντικειμένων που αποτελούν άκρα του συνδέσμου εξειδίκευσης που προστέθηκε ή διαγράφηκε από τη βάση.

- *Ειδοποίηση κρίσιμων γεγονότων ενημέρωσης*
  Τροποποίηση του κώδικα των **AddSubClass** , **DeleteSubClass** ώστε να στέλνουν ένα μήνυμα στην ΚΜΕΔ.

- *Αντίδραση μηχανισμού*
  Ακύρωση των τιμών της ΚΜΕΔ που αφορούν τα ‘‘στοιχεία προς ακύρωση’’.

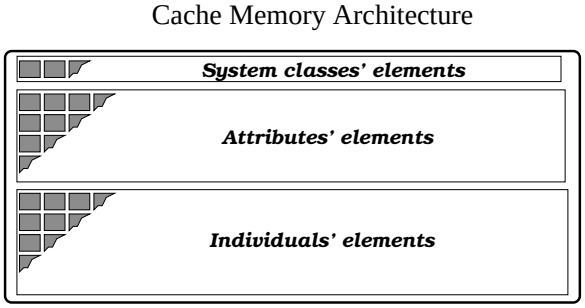
Η ΚΜΕΔ ελαττώνει το κόστος υπολογισμού του w_{FI} άρα και της $GetIDState$. Το ότι αποθηκεύει μόνο τις κληρονομούμενες δηλώσεις από υπερκλάσεις, και όχι τα μέλη του FD_t^{all}, έγινε ώστε να διατηρείται εύκολα και αποδοτικά η επικαιρότητα της ΚΜΕΔ.

Προτεινόμενη οργάνωση της ΚΜΕΔ

Η ΚΜΕΔ αποτελείται από τμήματα. Κάθε τμήμα περιέχει στοιχεία αντικειμένων κάποιου συγκεκριμένου τύπου :

- **ΚΜΕΔ Κλάσεων Συστήματος**
  Περιέχει τις δηλώσεις που αφορούν τις **κλάσεις συστήματος**. Ενημέρωση των δεδομένων της βάσης δεν επηρεάζουν την επικαιρότητα τους.
- **ΚΜΕΔ κλάσεων Individuals**
  Περιέχει δεδομένα για αντικείμενα του $C \cap I$.
- **ΚΜΕΔ κατηγοριών Attributes**
  Περιέχει δεδομένα για αντικείμενα του $C \cap A$.

Σχετικό σχήμα παρουσιάζεται στο σχήμα 6.69.



Σχήμα 6.69: Η οργάνωση της ΚΜΕΔ

Ενεργοποίηση της ΚΜΕΔ

Υπάρχουν διάφορες εναλλακτικες προτάσεις :

- Μια ΚΜΕΔ ανά ΟΕ.
  Για κάθε μια ενεργή ΟΕ δημιουργείται, αν το επιθυμεί ο χρήστης, μια ΚΜΕΔ.
- Μια ολική ΚΜΕΔ
  Κάθε στοιχείο της ΚΜΕΔ επεκτείνεται με ένα πεδίο για το SYSID της ΟΕ. Για την βελτιστοποίηση των επιδόσεων μπορεί να χρησιμοποιηθεί μια tagged cache (tag θα είναι το SYSID της ΟΕ).

Κεφάλαιο 7

Πρόταση Εφαρμογής του MMOE

Οι όψεις ενημέρωσης που ορίζονται με το MMOE μπορούν να συνδιαστούν με ένα σύνολο **γενικών λειτουργιών ενημέρωσης** και να στηρίξουν την ανάπτυξη ενός **διαλογικού δελτίου εξέλιξης γνώσης**. Το δελτίο αυτό θα προσφέρει στον τελικό χρήστη ένα συγκεκριμένο σύνολο **διαδικασιών ενημέρωσης**. Η όλη ιδέα παρουσιάζεται σχηματικά στο σχήμα 7.1.

Σχήμα 7.1: Αξιοποίηση των Όψεων Ενημέρωσης

Στο κεφάλαιο αυτό αρχικά περιγράφονται οι επιθυμητές λειτουργίες ενημέρωσης και ο τρόπος με τον οποίο συνδιάζονται με τις ΟΕ. Εν συνεχεία προτείνεται ένα σύνολο από συγκεκριμένες διαδικασίες ενημέρωσης και μια ενδεικτική **επαφή χρήσης** ενός διαλογικού δελτίου, το οποίο τις υποστηρίζει.

7.1 Λειτουργίες Ενημέρωσης

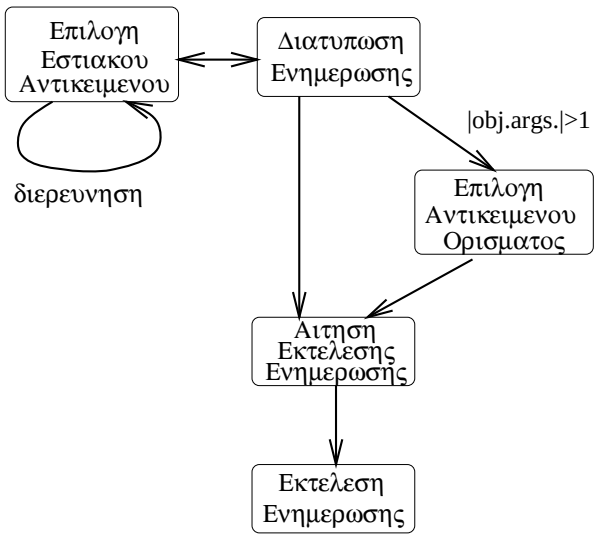
Για την διαλογική εξέλιξη γνώσης στο SIS προτείνονται οι ακόλουθες γενικές λειτουργίες :

- Πραγματοποίηση ενημερώσεων κατά την διάρκεια της **διερεύνησης** .
- Γνωστοποίηση των ΟΕ που αφορούν έναν χρήστη.
- Κατανοητή παρουσίαση των περιεχομένων (ΣΕΕ) μιας ΟΕ.
- Υπόδειξη **σημείων εκκίνησης** μιας ΟΕ.
- Υπόδειξη **ενημερώσεων ανά αντικείμενο ανάλογα με την τρέχουσα ΟΕ**.
- Βοήθεια κατά την εισαγωγή **σύνθετων αντικειμένων**.

Το γενικό διάγραμμα ροής ελέγχου του προτεινόμενου διαλογικού μηχανισμού ενημερώσεων που παρέχει τη δυνατότητα διερεύνησης και ενημερώσεων παρουσιάζεται στο σχήμα 7.2. Οι λειτουργίες που εμφανίζονται επεξηγούνται παρακάτω :

- **Επιλογή Εστιακού Αντικειμένου**
  Ένα αντικείμενο της βάσης τίθεται ως εστιακό αντικείμενο (ΕΑ).
- **Διατύπωση Ενημέρωσης**
  Σκοπό έχει τη διατύπωση μιας ή περισσοτέρων ΣΕΕ που αφορούν άμεσα το ΕΑ.
- **Επιλογή Αντικειμένου Ορίσματος**
  Το βήμα αυτό χρειάζεται γιατί ορισμένες ΣΕΕ δέχονται σαν ορίσματα περισσότερα από ένα αντικείμενα της βάσης.
- **Αίτηση Εκτέλεσης Ενημέρωσης**
  Πρόκειται για την αίτηση εκτέλεσης της ενημέρωσης που έχει διατυπωθεί, η οποία απευθύνεται στον πυρήνα της SIS-Telos (Semantic Checker)
- **Εκτέλεση Ενημέρωσης**
  Η λειτουργία αυτή πραγματοποιείται αν ο Semantic Checker δεν βρει κανένα σημασιολογικό σφάλμα.

Το διάγραμμα ροής ελέγχου ενός μηχανισμού ο οποίος επιπλέον υποστηρίζει και *Όψεις Ενημέρωσης* παρουσιάζεται σχηματικά στο σχήμα 7.3 και αναλύεται στις επόμενες



Σχήμα 7.2: Ενημερώσεις και Διερεύνηση : Ροή Ελέγχου

ενότητες. Όπως φαίνεται ένα βασικό χαρακτηριστικό του προτεινόμενου μηχανισμού (και συνάμα μια κύρια διαφορά του από τον προηγούμενο) είναι ότι ο χρήστης δεν είναι αναγκασμένος να επιλέξει την ΟΕ αρχικά, αλλά όποτε αυτός το θελήσει κατά τη διάρκεια της διερεύνησης.

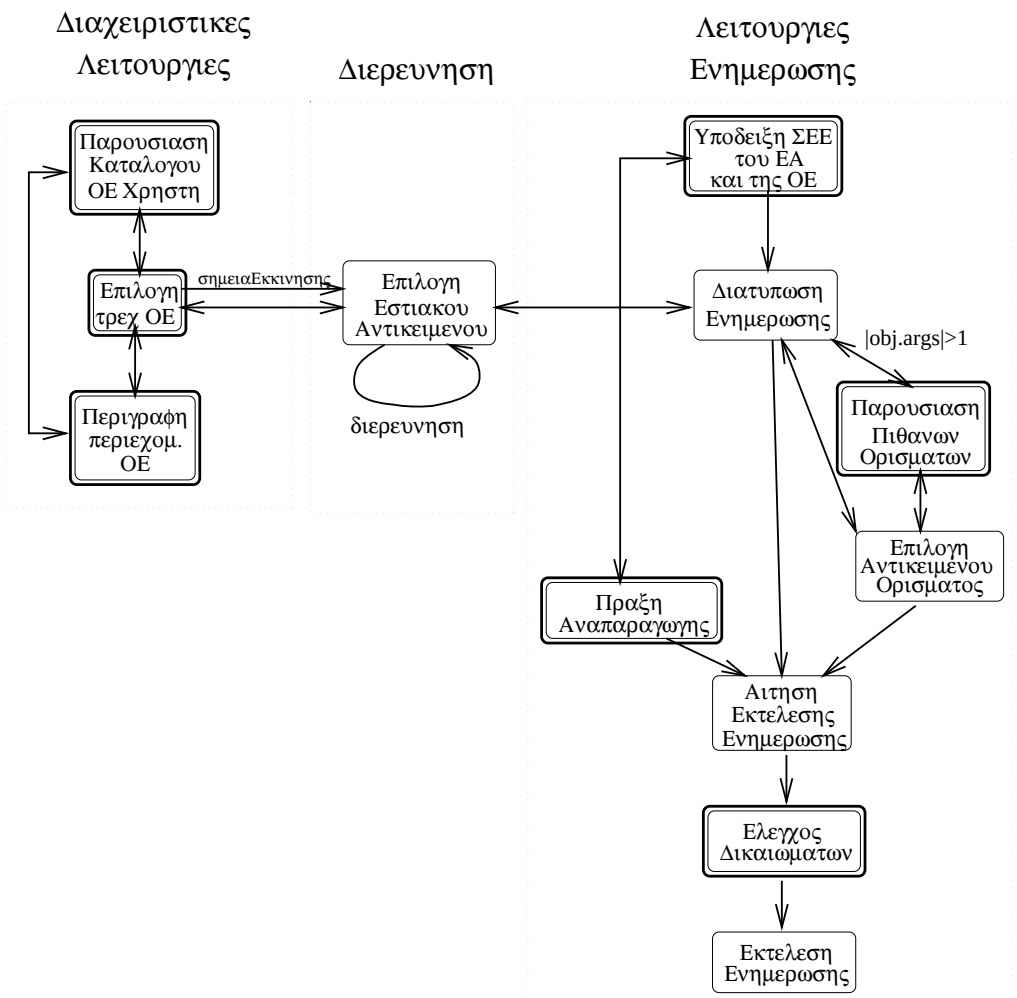
7.2 Διαχειριστικές Λειτουργίες

Το τμήμα του ολικού διαγράμματος ροής που αφορά τις διαχειριστικές λειτουργίες παρουσιάζεται στο σχήμα 7.4. Οι εικονιζόμενες λειτουργίες περιγράφονται παρακάτω:

1. **Παρουσίαση καταλόγου ΟΕ του χρήστη.**
   Παρουσίαση στο χρήστη ενός καταλόγου με τις ΟΕ που τον αφορούν

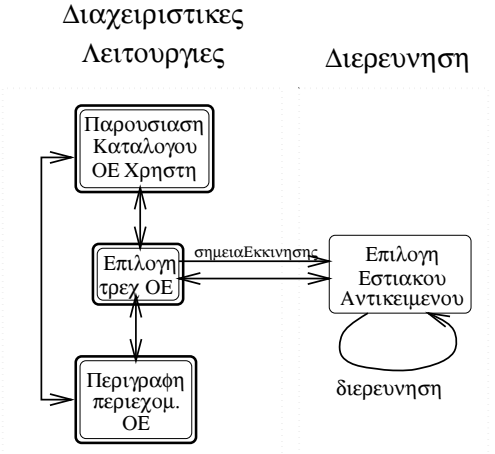
2. **Επιλογή τρέχουσας ΟΕ**
   Επιλογή μιας ΟΕ από τον προηγούμενο κατάλογο η οποία τίθεται τρέχουσα.

3. **Περιγραφή περιεχομένων ΟΕ.**
   Παρουσίαση της λειτουργικότητας μιας ΟΕ στη μορφή *απλού* ή *μη-γραμμικού* κειμένου. Η παρουσίαση αυτή συνοδεύεται από μια λίστα με τα *σημεία εκκίνησης* καθώς επίσης και από μια *ημιτυπική περιγραφή* των ενημερώσεων που η ΟΕ ορίζει. Η περιγραφή αυτή μπορεί να παράγεται αυτόματα από έναν *μεταφραστή δηλώσεων*.



Σχήμα 7.3: Διαλογική Εξέλιξη Γνώσης και Ενημέρωση Όψεων : Ροή Ελέγχου

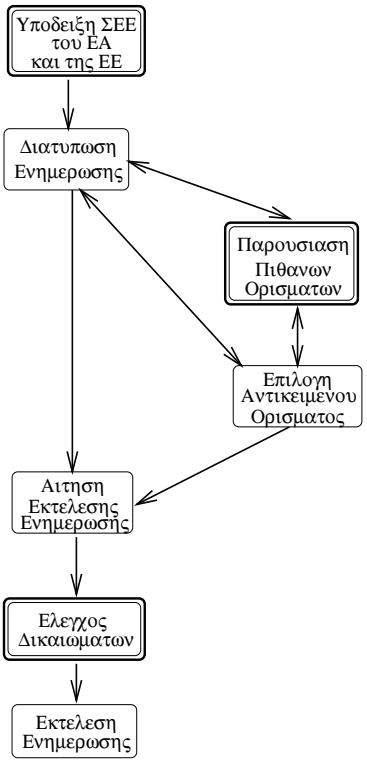
Τα έντονα πλαίσια σηματοδοτούν τις λειτουργίες που αφορούν τις Όψεις Ενημέρωσης.



Σχήμα 7.4: Διάγραμμα ροής του ελέγχου των διαχειριστικών λειτουργιών

7.3 Λειτουργίες Ενημέρωσης

Το τμήμα του ολικού διαγράμματος ροής που αφορά τις λειτουργίες ενημέρωσης παρουσιάζεται στο σχήμα 7.5.



Σχήμα 7.5: Διάγραμμα ροής ελέγχου των λειτουργιών ενημέρωσης.

Η λειτουργία **Διατύπωση Ενημέρωσης** δίνει στο χρήστη τη δυνατότητα διατύπωσης μιας ή περισσοτέρων ΣΕΕ. Η διατύπωση τους επιτυγχάνεται με την εκτέλεση ενός συνόλου **διαδικασιών ενημέρωσης**. Ο κατάλογος με τις συγκεκριμένες διαδικασίες που προτείνονται ακολουθεί παρακάτω. Υπενθυμίζουμε ότι σε κάθε χρονική στιγμή ένα αντικείμενο της SIS-Telos έχει τεθεί ‘‘τρέχον’’, το οποίο και ονομάζουμε **εστιακό αντικείμενο (ΕΑ)**. Επίσης στον παρακάτω πίνακα η λέξη *γνώρισμα* αντιπροσωπεύει οποιοδήποτε γνώρισμα του ΕΑ, ευθύ ή ανάστροφο, άμεσο ή κληρονομημένο.

Κατάλογος Διαδικασιών Ενημέρωσης

1. **Μετονομασία** του ΕΑ.
2. **Μετονομασία** γνωρισμάτων του ΕΑ.
3. **Διαγραφή** του ΕΑ.
4. **Διαγραφή** γνωρισμάτων του ΕΑ.
5. **Διαγραφή** συνδέσμων ταξινόμησης του ΕΑ .
6. **Διαγραφή** συνδέσμων ταξινόμησης των γνωρισμάτων του ΕΑ.
7. **Διαγραφή** συνδέσμων γενίκευσης του ΕΑ.
8. **Διαγραφή** συνδέσμων γενίκευσης των γνωρισμάτων του ΕΑ.
9. **Ταξινόμηση** του ΕΑ.
10. **Ταξινόμηση** των γνωρισμάτων του ΕΑ.
11. **Γενίκευση** του ΕΑ.
12. **Γενίκευση** των γνωρισμάτων του ΕΑ.
13. **Δημιουργία** νέου αντικειμένου σαν **υποκλάση** του ΕΑ.
14. **Δημιουργία** νέου γνωρίσματος στο ΕΑ σαν **υποκλάση** ενός κληρονομημένου.
15. **Δημιουργία** νέου αντικειμένου σαν **περίπτωση** του ΕΑ.
16. **Δημιουργία** νέου γνωρίσματος στο ΕΑ σαν **περίπτωση** μιας ή περισσοτέρων κατηγοριών.
17. **Δημιουργία αντιγράφου** του ΕΑ (**MakeCopy**) .

Όπως θα παρουσιαστεί και παρακάτω οι παραπάνω διαδικασίες ενεργοποιούνται από ένα δελτίο όμοιο με το *Δελτίο Παρουσίασης Οντότητας (ΔΠΟ)*, το οποίο παρουσιάζει όλες τις άμεσα συνδεδεμένες πληροφορίες μιας οντότητας. Λαμβάνοντας υπόψη και το παραπάνω σχόλιο οι *διαδικασίες ενημέρωσης* χωρίζονται σε δύο κατηγορίες :

α) Σε αυτές που η εκτέλεσή τους δεν απαιτεί την εύρεση ενός ή περισσοτέρων αντικειμένων σαν ορίσματα.

β) Σε αυτές που ένα βήμα της εκτέλεσης τους είναι η εύρεση ενός ή περισσοτέρων αντικειμένων σαν ορίσματα.

Οι διαδικασίες της κατηγορίας (β) απαιτούν διευκόλυνση γιατί ο εντοπισμός και η επιλογή των επιθυμητών αντικειμένων (ορίσματα) συχνά δεν είναι εύκολος. Οι διαδικασίες

αυτές είναι οι εξής :

1. **Ταξινόμηση** του ΕΑ.
2. **Ταξινόμηση** των γνωρισμάτων του ΕΑ.
3. **Γενίκευση** του ΕΑ.
4. **Γενίκευση** των γνωρισμάτων του ΕΑ.
5. **Δημιουργία** νέου γνωρίσματος στο ΕΑ σαν **υποκλάση** ενός κληρονομημένου.
6. **Δημιουργία** νέου γνωρίσματος σαν **περίπτωση** μιας ή περισσοτέρων κατηγοριών.

Οι υπόλοιπες διαδικασίες ανήκουν στην κατηγορία διαδικασιών (α) και η εκτέλεση τους δεν απαιτεί ιδιαίτερη διευκόλυνση.

Για διευκόλυνση των διαδικασιών της κατηγορίας (β) προτείνεται ένα σύνολο βοηθητικών διαδικασιών οι οποίες συνθέτουν τη γενική λειτουργία υπό το όνομα **Λειτουργία Παρουσίασης Πιθανών Ορισμάτων**. Σκοπός της λειτουργίας αυτής είναι ο υπολογισμός και εν συνεχεία η παρουσίαση στο χρήστη ενός συνόλου από πιθανά ορίσματα. Για τον υπολογισμό αυτό λαμβάνεται υπόψη το περιβάλλον (context) στο οποίο ένα αντικείμενο είναι ενταγμένο. Τα ορίσματα που προτείνονται πρέπει :

- Να συγκροτούν **σημασιολογικά ορθές** ενημερώσεις (τουλάχιστο όχι σίγουρα λανθασμένες), ώστε να ελαχιστοποιούν τις λανθασμένες επιλογές του χρήστη (φιλικότητα χρήσης).
- Σε περίπτωση που τα πιθανώς σημασιολογικά ορθά ορίσματα είναι πολλά, θα ήταν καλό να υπολογίζονται και να προτείνονται **τα πιθανότερα** με κριτήριο τον τρόπο με τον οποίο μια βάση της SIS-Telos εξελίσσεται. Εναλλακτικά θα ήταν καλό να παρέχεται η δυνατότητα σταδιακής προσέγγισης των επιθυμητών αντικειμένων.

Παρακάτω κάθε μία από τις διαδικασίες ενημέρωσης της κατηγορίας (β), αναλύτεται σε ένα σύνολο *βοηθητικών διαδικασιών* [1].

1. **Ταξινόμηση του εστιακού αντικειμένου.**

   Αν το ΕΑ δεν είναι ήδη ταξινομημένο σε μια κλάση, πλην της απαραίτητης κλάσης συστήματος, τότε προτείνονται οι λειτουργίες :

   α) Ταξινόμηση σε *Κλάση Δημιουργίας Οντοτήτων*
   Επιλογή κλάσης από το σύνολο των ΚΔΟ της τρέχουσας ΟΕ.

[1] Οι ερωτηματικές εντολές (του **qi**) που εμφανίζονται στους παρακάτω επεξηγούνται στο παράρτημα C.

β) Ταξινόμηση σε κλάση που ορίζει ο χρήστης.
Ο χρήστης συμπληρώνει σε ένα πεδίο το λογικό όνομα της επιθυμητής κλάσης.

Αν το ΕΑ είναι ήδη ταξινομημένο σε μια τουλάχιστο κλάση τότε συχνή είναι η ταξινόμηση σε ειδικότερη κλάση ως συνέπεια της αύξησης της γνώσης που υπάρχει για ένα αντικείμενο, είτε συνέπεια της εξέλιξης (εξειδίκευσης) του σχήματος, το οποίο μπορεί να παραστήσει περισσότερη γνώση. Επίσης συχνή είναι η ταξινόμηση σε γενικότερη ή ειδικότερη κλάση ως συνέπεια της αλλαγής του πραγματικού κόσμου (Πχ *Πρόσωπο* → *Φοιτητής* → *Πρόσωπο*).

Εκ τούτου προτείνονται οι επόμενες διαδικασίες οι οποίες παίρνουν σαν όρισμα μια κλάση του ΕΑ:

γ) Αλλαγή μιας κλάσης με μια γενικότερη της
Το προτεινόμενο σύνολο προκύπτει από την ερώτηση :

$$\{c \mid c \in gasc(d) \wedge d \in gc(EA)\}$$

δ) Αλλαγή μιας κλάσης με μια ειδικότερη της
Το προτεινόμενο σύνολο προκύπτει από την ερώτηση :

$$\{c \mid c \in gasb(d) \wedge d \in gc(EA)\}$$

ε) Αλλαγή μιας κλάσης με μια ‘‘συνάλληλή’’ της
Προκύπτει από την ανάγκη ενημέρωσης χρονικά μεταβαλλόμενων ιδιοτήτων (Πχ: *Άνθρωπος* → *Φοιτητής* → *Φαντάρος*). Το προτεινόμενο σύνολο προκύπτει από την ερώτηση :

$$\{c \mid c \in gasb(gasc(d)) - gasb(d) \wedge d \in gc(EA)\}$$

στ) Ταξινόμηση σε ‘‘συγγενή’’ ιεραρχία
Λόγω του ότι η ταξινόμηση χρησιμοποιείται συχνά αντί της απόδοσης γνωρίσματος. Το προτεινόμενο σύνολο προκύπτει από την ερώτηση :

$$\{c \mid c \in gasb(gtv(categ)) \wedge relatedClasses \in gc(categ) \wedge categ \in gilf(gc(EA))\}$$

2. **Ταξινόμηση των γνωρισμάτων του ΕΑ.**

Το σύνολο των πιθανών κατηγοριών για την ταξινόμηση ενός γνωρίσματος (l) του ΕΑ είναι το υποσύνολο των κληρονομημένων κατηγοριών που περιέχει τις κατηγορίες

με την κατάλληλη στάθμη αφαίρεσης :

$$\{c \mid c \in (gilf(gc(gfv(l))) \cap gilt(gc(gtv(l)))) \wedge R^{in}_{sys}(c) = R^{in}_{sys}(R^{in}_{sys}(l))\}$$

3. **Γενίκευση του εστιακού αντικειμένου.**

Αν το ΕΑ δεν έχει καμία υπερκλάση προτείνονται οι λειτουργίες :

α) Γενίκευση σε *Κλάση Εξειδίκευσης*
Επιλογή υπερκλάσης από το σύνολο κλάσεων εξειδίκευσης της τρέχουσας ΟΕ.

β) Γενίκευση σε κλάση που ορίζει ο χρήστης.
Ο χρήστης συμπληρώνει σε ένα πεδίο το λογικό όνομα της επιθυμητής υπερκλάσης.

Αν το ΕΑ έχει ήδη μία ή περισσότερες υπερκλάσεις συχνά απαιτείται η προσθήκη μιας νέας υπερκλάσης (πολλαπλή γενίκευση) ή η αλλαγή μιας υπάρχουσας υπερκλάσης με μια άλλη λόγω της εξέλιξης της ιεραρχίας (πχ οι ιεραρχίες όρων των *Facets*). Προτείνονται οι ακόλουθες διαδικασίες οι οποίες παίρνουν σαν όρισμα μια υπάρχουσα υπερκλάση :

γ) Αλλαγή μιας υπερκλάσης με μια γενικότερη της
Το προτεινόμενο σύνολο προκύπτει από την ερώτηση :

$$\{c \mid c \in gasc(\text{ΕΑ})\}$$

δ) Αλλαγή μιας υπερκλάσης με μια ειδικότερη της
Το προτεινόμενο σύνολο προκύπτει από την ερώτηση :

$$\{c \mid c \in gasb(s) \wedge s \in gsc(EA)\}$$

ε) Αλλαγή μιας υπερκλάσης με μια ‘‘συνάλληλή’’ της
Το προτεινόμενο σύνολο προκύπτει από την ερώτηση :

$$\{c \mid c \in gasb(gasc(s)) - gasb(s) \wedge s \in gsc(EA)\}$$

Επίσης προτείνεται η αξιοποίηση των **συζυγών ζευγών κλάσεων-μετακλάσεων** [2] κατά την προσθήκη/διαγραφή υπερκλάσεων. Ειδικότερα προτείνεται η αυτόματη

[2] Η έννοια των συζυγών ζευγών κλάσεων-μετακλάσεων για την Telos προτάθηκε στην εργασία [26]. Παρόμοια πρόταση υπάρχει στο [34].

προσθήκη/διαγραφή συνδέσμων ταξινόμησης στη συζυγή μετακλάση ως αποτέλεσμα της προσθήκης/διαγραφής υπερκλάσεων στο ΕΑ αντίστοιχα.

Η λειτουργικότητα αυτή θα συνέβαλλε στην ευκολία του χρήστη και στη συνέπεια της εξέλιξης γνώσης που πραγματοποιείται διαλογικά.

4. **Γενίκευση των γνωρισμάτων του ΕΑ.**

Οι πιθανές υπερκλάσεις ενός γνωρίσματος (l) είναι το σύνολο των κληρονομημένων γνωρισμάτων που αποτελείται από γνωρίσματα που ανήκουν στην ίδια στάθμη αφαίρεσης.

Το σύνολο αυτό προκύπτει από την ερώτηση :

$$\{c \mid c \in ((gilf(gfv(l)) \cap gilt(gtv(l))) - gasb(l)) \wedge R^{in}_{sys}(l) = R^{in}_{sys}(c)\}$$

5. **Δημιουργία νέου γνωρίσματος σαν εξειδίκευση ενός κληρονομημένου**

Η λειτουργία αυτή αναλύεται στις διαδικασίες :

α) Επιλογή του επιθυμητού γνωρίσματος (ή γνωρισμάτων) για εξειδίκευση
Ο χρήστης μαρκάρει ένα ή περισσότερα κληρονομημένα γνωρίσματα (το σύνολο προκύπτει από την ερώτηση : $gilf(0) - glf(0)$.

β) Επιλογή τιμής του γνωρίσματος
Η τιμή του γνωρίσματος είναι μια κλάση που υπάρχει στη βάση, ή μια που θα δημιουργηθεί για το σκοπό αυτό. Το σύνολο των δυνατών υπερκλάσεων της είναι :

$$\{s \mid s \in \bigcap_{c_i}(gasb(gtv(c_i)) \cup (gtv(c_i)) \text{ όπου } c_i \text{ οι υπερκλάσεις } \}$$

Ο χρήστης τότε μπορεί να προσθέσει μια νέα υποκλάση σε μια από αυτές ή να επιλέξει μια από το σύνολο :

$$\{c \mid c \in \bigcap_{c_i} gasb(gtv(c_i)) \text{ όπου } c_i \text{ οι υπερκλάσεις } \}$$

6. **Δημιουργία νέου γνωρίσματος σαν περίπτωση μιας ή περισσοτέρων κατηγοριών.**

Η διαδασία αυτή αναλύεται στις ακόλουθες διαδικασίες :

α) Επιλογή της επιθυμητής κατηγορίας (ή κατηγοριών)
Το σύνολο των δυνατών κατηγοριών είναι το σύνολο των κληρονομημένων

κατηγοριών, τα οποία δεν είναι ατομικά γνωρίσματα. Προκύπτουν από την ερώτηση :

$$\{c \mid c \in (gilf(gc(EA)) \cup gilf(gSc(EA))) \wedge c \in C\}$$

β) Επιλογή τιμής του γνωρίσματος

Η τιμή του γνωρίσματος μπορεί να είναι μια οντότητα που υπάρχει στη βάση, ή μια που θα δημιουργηθεί για το σκοπό αυτό. Το σύνολο των δυνατών κλάσεων της είναι:

$$\{c \mid c \in \bigcap_{c_i} gasb(gtv(c_i)) \text{ όπου } c_i \text{ οι κατηγορίες } \}$$

ενώ το σύνολο των πιθανών τιμών που υπάρχουν είναι το :

$$\{c \mid c \in gi(\bigcap_{c_i} gasb(gtv(c_i))) \text{ όπου } c_i \text{ οι κατηγορίες } \}$$

7.4 Λειτουργίες Ενημέρωσης και Όψεις Ενημέρωσης

7.4.1 Έλεγχος Δικαιωμάτων

Πρόκειται για μια λειτουργία η οποία δέχεται σαν όρισμα μια *αίτηση εκτέλεσης ενημέρωσης* και το αναγνωριστικό της *τρέχουσας ΟΕ*, και επιστρέφει ένα *ΝΑΙ* αν η ενημέρωση ορίζεται από την ΟΕ, ή ένα *ΌΧΙ* διαφορετικά.

Μια αίτηση εκτέλεσης ενημέρωσης μπορεί να αποτελείται από πολλές ΣΕΕ όπως συμβαίνει για παράδειγμα στην **MakeCopy** και στις ακόλουθες διαδικασίες ενημέρωσης :

Λειτουργία Ενημέρωσης	**ΣΕΕ**
Δημιουργία νέου αντικ σαν **υποκλάση** του ΕΑ.	$o :=$ **CreateIndividual** n, S **AddSubClass** o, EA
Δημιουργία νέου γνωρ σαν **υποκλάση** ενός κληρ.	$o :=$ **CreateAttribute** *from* , *to* , n, S **AddSubClass** o, l l : ένα κληρονομημένο γνώρισμα του ΕΑ
Δημιουργία νέου αντικ σαν **περίπτωση** του ΕΑ	$o :=$ **CreateIndividual** n, S **AddInstance** o, EA

Ο αλγόριθμος που εξετάζει το αν μια ΣΕΕ οριζεται από τις δηλώσεις μιας ΟΕ παρουσιάστηκε στο προηγούμενο κεφάλαιο και μπορεί να υλοποιηθεί είτε στο διαλογικό δελτίο ενημερώσεων, ή σε πιο χαμηλό επίπεδο, στο run-time σύστημα της SIS-Telos . Στη δεύτερη περίπτωση θα είχαμε έλεγχο και των μαζικών ενημερώσεων που γίνονται μέσω του μεταφραστή της SIS-Telos (Telos parser).

7.4.2 Υπόδειξη Ενημερώσιμων Πληροφοριών

Οι λειτουργίες ενημέρωσης που περιγράφηκαν παραπάνω ενεργοποιούνται από ένα δελτίο όμοιο με το ΔΠΟ. Αν κατά τη διάρκεια του ''ανοίγματος'' του δελτίου υπολογίζονται οι ενημερώσιμες πληροφορίες (αυτές που μπορούν να διαγραφούν ή που μπορούν να εμπλουτιστούν) τότε μπορούμε **διαφοροποιώντας την παρουσίαση τους** (διαφορετικό χρώμα ή γραμματοσειρά) να γνωστοποιούμε στο χρήστη εξ αρχής το τι μπορεί να κάνει. Η λειτουργία αυτή ονομάζεται **Υπόδειξη ΣΕΕ του ΕΑ και ΟΕ** (δες σχήμα 7.2). Όπως αναφέρεται και στην ενότητα 7.5.3 η διάκριση αυτή (ενημερώσιμων και μη ενημερώσιμων πληροφοριών) μπορεί να αποτελέσει κριτήριο φιλτραρίσματος της εμφανιζόμενης πληροφορίας (παρουσίαση μόνο των ενημερώσιμων πληροφοριών).

Η λειτουργία αυτή πρέπει να εκτελείται αποδοτικά (γρήγορα). Αν ξανασκεφτούμε τον αλγόριθμο της διαδικασίας $GetIdState$ θα δούμε ότι η εύρεση της τιμής ενός ID στο ΕΑ, απαιτεί την εξέταση μιας ''γειτονιάς'' του ΕΑ (υπερκλάσεις, κλάσεις , κλάση συστήματος) . Αν κατά τη διάσχιση αυτή συλλέγουμε όλες τις δηλώσεις (για όλα τα ID) που υπάρχουν και τις αποθηκεύουμε σε ένα σύνολο πινάκων, τότε μπορούμε στον ίδιο χρόνο να υπολογίσουμε τις τιμές όλων των ID που αφορούν το ΕΑ. Η πρωτότυπη υλοποίηση των αλγορίθμων που έγινε το υποστηρίζει.

7.4.3 Εύρεση Πιθανών Ορισμάτων

Η λειτουργία *εύρεση πιθανών ορισμάτων* πρέπει να ξαναορισθεί ώστε να λαμβάνεται υπόψη η τρέχουσα ΟΕ. Άρα το προτεινόμενο σύνολο ορισμάτων, set, πρέπει να συγκροτεί:

- **σημασιολογικά ορθές** ενημερώσεις (τουλάχιστο όχι σίγουρα λανθασμένες).
  Συμβολίζουμε με set_{sem} το σύνολο αυτό.

- **ενημερώσεις που ανήκουν στην τρέχουσα ΟΕ**.
  Συμβολίζουμε με set_{task} το σύνολο αυτό.

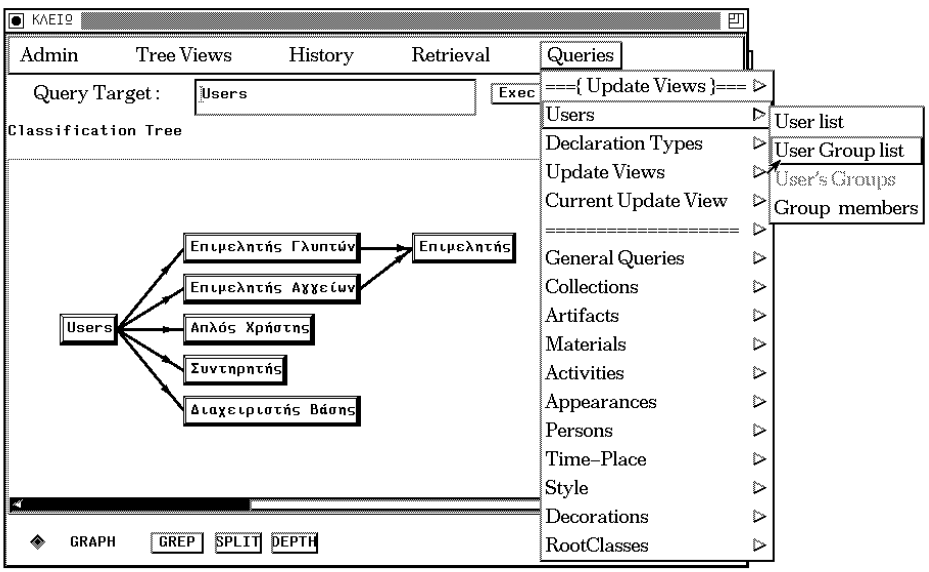
Δηλαδή $set = set_{sem} \cap set_{task}$. Ο χρόνος υπολογισμού του εξαρτάται από την σειρά με την οποία θα υπολογιστούν τα set_{sem} και set_{task}. Αν το set_{sem} είναι μικρό και μπορεί να υπολογισθεί γρήγορα, είναι καλό να το υπολογίσουμε πρώτο (πχ το σύνολο πιθανών κατηγοριών ενός γνωρίσματος). Αν όμως αυτό είναι μεγάλο (πχ το σύνολο πιθανών μετακλάσεων ταξινόμησης) τότε είναι καλύτερο να υπολογίσουμε αρχικά το set_{task}. Σημειώνουμε επίσης ότι η προτεινόμενη ΚΜΕΔ μπορεί να βελτιώσει αισθητά τον απαιτούμενο χρόνο, ο οποίος είναι ήδη μικρός.

7.5 Προτεινόμενη Επαφή Χρήσης

Στην ενότητα αυτή περιγράφεται ενδεικτικά η επαφή χρήσης του προτεινόμενου μηχανισμού διαλογικής εξέλιξης γνώσης στο SIS.

Μια πρωτότυπη υλοποίηση των διαχειριστικών λειτουργιών έγινε χρησιμοποιώντας τις υπάρχουσες δυνατότητες του SIS. Στα σχήματα 7.6 έως 7.10 παρουσιάζονται εικόνες από τη χρήση τους στο σύστημα *ΚΛΕΙΩ*.

Στο σχήμα 7.6 βλέπουμε ότι στο σύνολο των προκαθορισμένων ερωτήσεων του *ΚΛΕΙΩ* έχει προστεθεί ένα σύνολο ερωτήσεων με τίτλο *Update Views*. Υπό τον τίτλο **Users** ομαδοποιούνται οι ερωτήσεις που αφορούν τους χρήστες και τα σύνολα χρηστών του συστήματος, ενώ χρησιμοποιώντας το γραφικό παράθυρο μπορούμε να δούμε γραφικά την ιεραρχική οργάνωση των συνόλων.

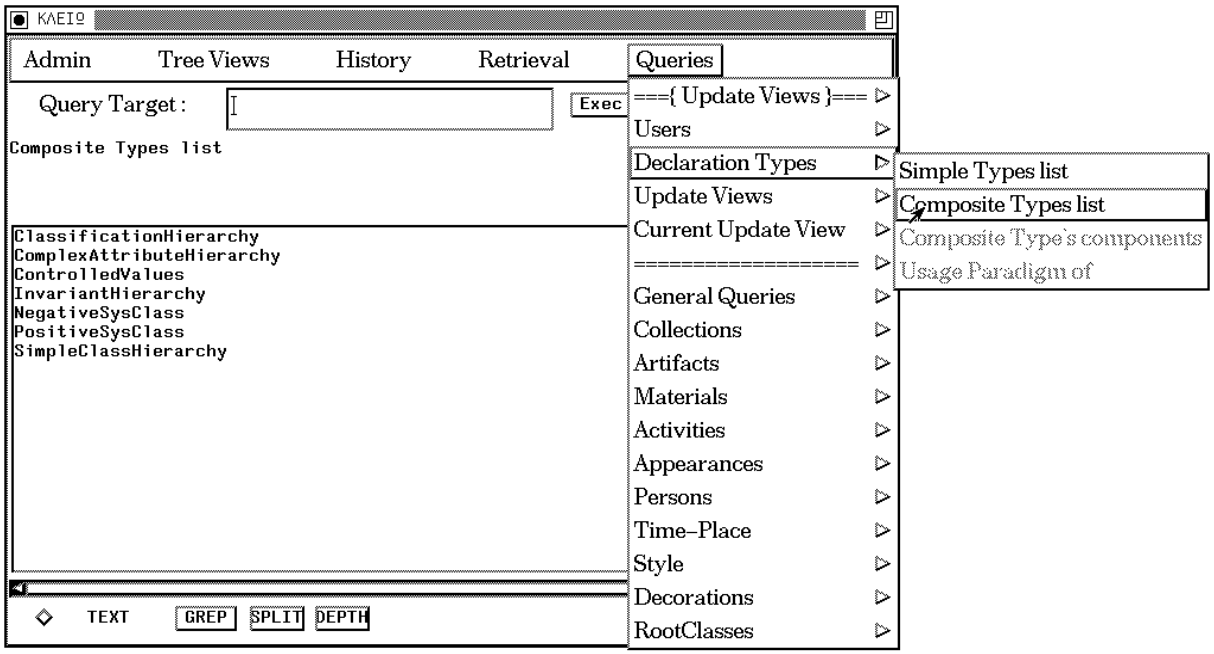


Σχήμα 7.6: Λειτουργίες που αφορούν τους χρήστες του συστήματος

Στο σχήμα αυτό παρατηρούμε ότι το σύνολο χρηστών Επιμελητής, ως υποκλάση των *Επιμελητή Αγγείων* και *Επιμελητή Γλυπτών*, κληρονομεί τις ΟΕ που έχουν εκχωρηθεί σε αυτούς

Υπό τον τίτλο **Declaration Types** ομαδοποιούνται οι ερωτήσεις που πληροφορούν τον χρήστη για τους διαθέσιμους *τύπους δήλωσης*, απλούς ή σύνθετους (δες σχήμα 7.7) και αναλύουν τους σύνθετους τύπους στους συστατικούς τους (δες σχήμα 7.8). Επίσης η

ερώτηση *Usage Paradigm of* παρουσιάσει παραδείγματα χρήσης των σύνθετων δηλώσεων για να κατανοηθεί καλύτερα η χρήση τους.

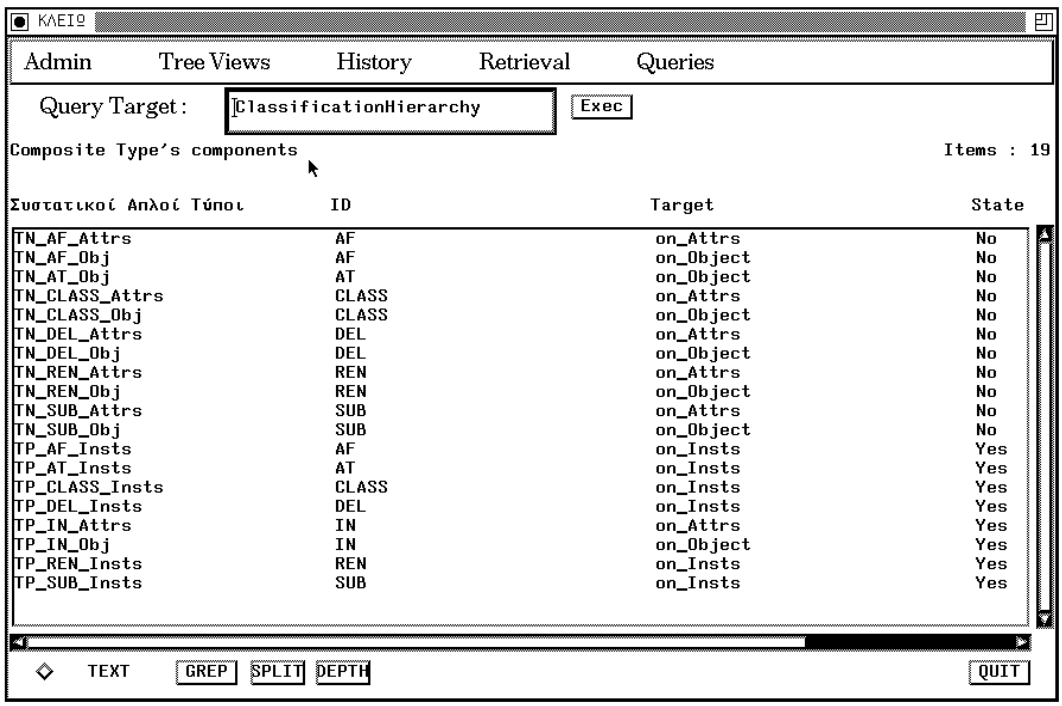


Σχήμα 7.7: Παρουσίαση καταλόγου με τους σύνθετους τύπους που έχουν ορισθεί

Υπό τον τίτλο **Update Views** υπάρχουν ερωτήσεις που εμφανίζουν το σύνολο των ΟΕ που έχουν ορισθεί, ή επιστρέφουν το σύνολο των ΟΕ που αφορούν άμεσα έναν χρήστη ή ένα σύνολο χρηστών.

Υπό τον τίτλο **Current Update Task** υπάρχει ένα σύνολο ερωτήσεων που αφορούν την ‘‘τρέχουσα ΟΕ’’: σχετικά σύνολα χρηστών, χρήστες, δηλώσεις ενημέρωσης, κλάσεις δημιουργίας οντοτήτων και κλάσεις εξειδίκευσης. Στο σχήμα 7.9 βλέπουμε τις δηλώσεις ενημέρωσης της ΟΕ *Εργασία Επιμελητή*, το σχετικό σχόλιο και το αντίστοιχο δελτίο παρουσίασης (Object Card).

Το ΜΜΟΕ επιτρέπει την σύνθεση ΟΕ. Οι κλάσεις δηλώσεων των ΟΕ (μέλη της μετακλάσεις *DeclarationsType*) μπορούν να συνδέονται με σχέσεις *isA*. Σχετικό παράδειγμα παρουσιάζεται στο σχήμα 7.10 όπου βλέπουμε ότι η κλάση *Δηλώσεις_ΕργΕπιμελητή* είναι υπερκλάση της κλάσης *Δηλώσεις_Ασφάλειας*. Με αυτό τον τρόπο τα μέλη της τελευταίας (που είναι ένα σύνολο αρνητικών δηλώσεων που αφορούν την ασφάλεια του συστήματος) είναι και έμμεσα μέλη της πρώτης.



Σχήμα 7.8: Ανάλυση του σύνθετου τύπου *ClassificationHierarchy* στους συστατικούς του.

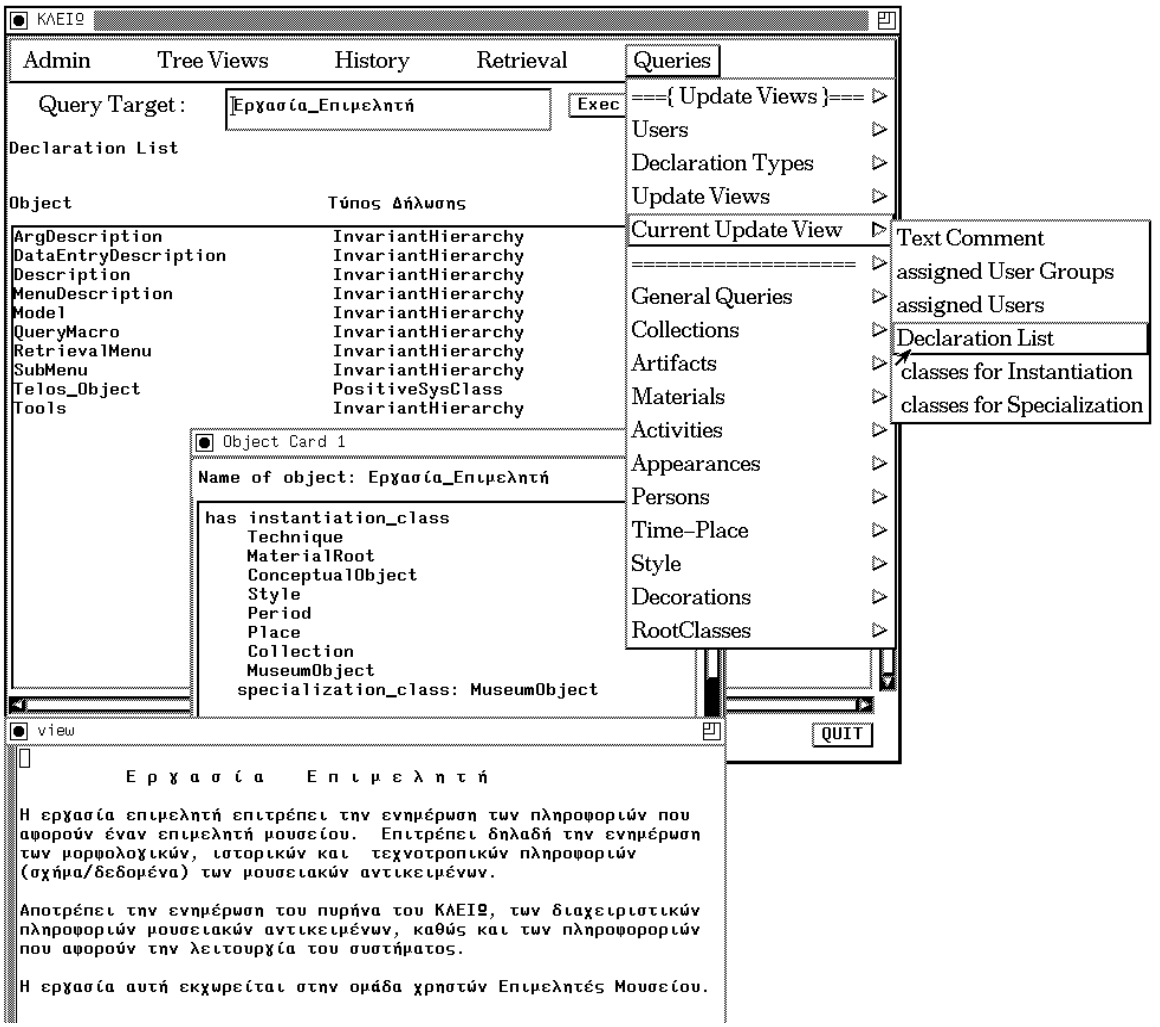
Όσο αφορά τώρα τις διαδικασίες ενημέρωσης, η προτεινόμενη επαφή χρήσης για την πραγματοποίηση των *λειτουργιών ενημέρωσης* βασίζεται στην επιθυμητή και κοινώς αποδεκτή, ιδιότητα της **ομοιομορφίας μεταξύ της παρουσίασης και της ενημέρωσης** της πληροφορίας μιας ΒΔ.

Για την παρουσίαση της πληροφορίας το SIS προσφέρει το Δελτίο Παρουσίασης Οντότητας (ΔΠΟ) το οποίο μπορεί να παρουσιάσει όλες τις πληροφορίες που αφορούν άμεσα μια οντότητα. Προτείνεται η επέκταση του και η ανάπτυξη ενός Δελτίου Παρουσίασης και Ενημέρωσης Αντικειμένου (ΔΠΕΑ) για την παρουσίαση αλλά και την εισαγωγή ή ενημέρωση των πληροφοριών ενός αντικειμένου.

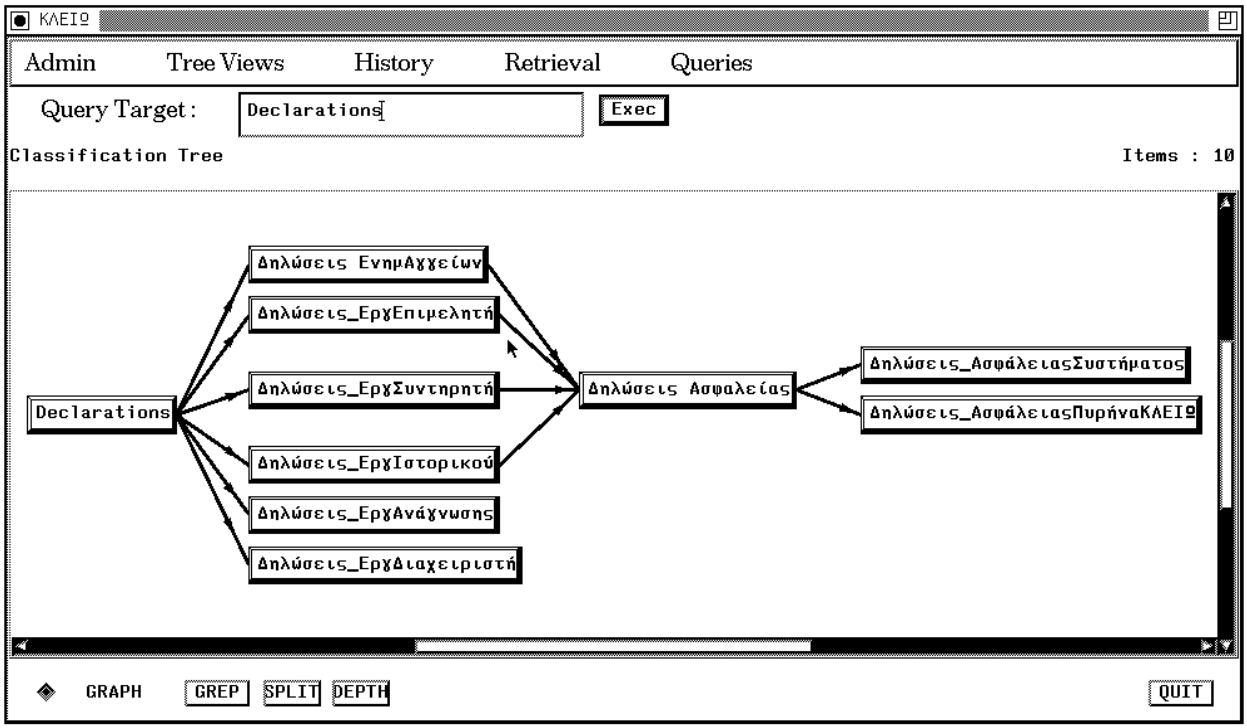
Οι πληροφορίες που παρουσιάζει ένα ΔΠΟ για ένα αντικείμενο μπορούν επίσης να παρουσιάζονται στη μορφή αστεροειδούς γράφου από το γραφικό παράθυρο του SIS. Άρα οι προτεινόμενες επεκτάσεις του ΔΠΟ θα μπορούσαν να ενεργοποιούνται και από το γραφικό παράθυρο.

7.5.1 Δελτίο Παρουσίασης Οντότητας

Στο σχήμα 7.12 φαίνεται το ΔΠΟ της οντότητας *Φοιτητής*, η οποία ανήκει στο μοντέλο



Σχήμα 7.9: Παρουσίαση πληροφοριών για την όψη ενημέρωσης *Εργασία_Επιμελητή*

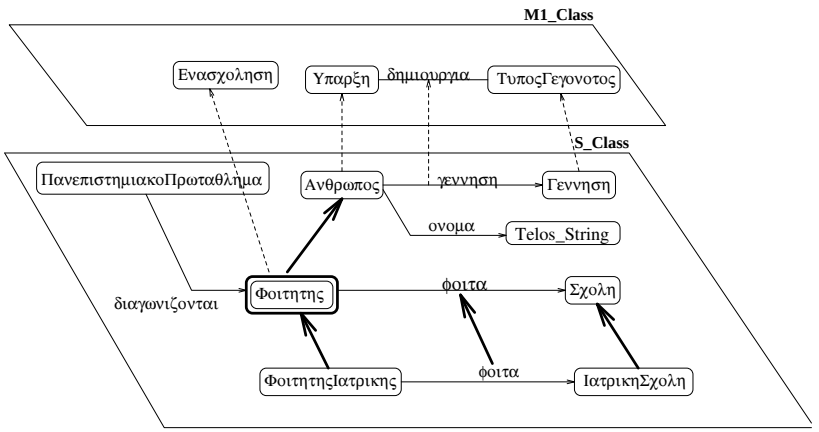


Σχήμα 7.10: Ιεραρχική οργάνωση των κλάσεων δήλωσης

πληροφοριών που παρουσιάζεται σχηματικά στο σχήμα 7.11. Γράφοντας το λογικό όνομα μιας οντότητας στο πεδίο *QueryTarget* εμφανίζεται το αντίστοιχο ΔΠΟ. Επίσης επιλέγοντας με το ποντίκι μια οντότητα που αναγράφεται σε ένα ΔΠΟ μπορούμε να εμφανίζουμε το αντίστοιχη ΔΠΟ . Η δυνατότητα αυτή επιτρέπει την πλοήγηση στα δεδομένα της βάσης. Μάλιστα υπάρχει δυνατότητα παράλληλης πλοήγησης αφού μπορούμε να έχουμε ανοικτά ταυτόχρονα πολλά ΔΠΟ.

7.5.2 Ένα Δελτίο Παρουσίασης και Ενημέρωσης Αντικειμένων

Προτείνεται η εξέλιξη του ΔΠΟ σε ένα Δελτίο Παρουσίασης και Ενημέρωσης Αντικειμένου (ΔΠΕΑ), το οποίο θα δίνει τη δυνατότητα προσθήκης, διαγραφής ή τροποποίησης των πληροφοριών της βάσης. Οι προτεινόμενες λειτουργίες που πρέπει να υποστηρίζει περιγράφηκαν σε προηγούμενη ενότητα. Μια ενδεικτική επαφή χρήσης του ΔΠΕΑ παρουσιάζεται στα σχήματα 7.13, 7.14, 7.15 και 7.16.



Σχήμα 7.11: Ένα απλό μοντέλο της SIS-Telos

7.5.3 Συμπληρωματικές λειτουργίες

Για λόγους ευχρηστίας προτείνεται ένα σύνολο από συμπληρωματικές λειτουργίες (δες σχήμα 7.15), οι οποίες φιλτράρουν ή χωροθετούν τις πληροφορίες που εμφανίζονται στο ΔΠΕΑ. Φιλτράροντας την εμφανιζόμενη πληροφορία αποτρέπεται η γνωστική επιβάρυνση, ενώ γίνεται δυνατή η παρατήρηση της πληροφορίας από διάφορες όψεις. Συνάμα ο χρόνος εμφάνισης ενός δελτίου μειώνεται (απαιτούνται λιγότερες ερωτήσεις) καθώς επίσης και το μέγεθος των δελτίων κάνοντας έτσι εφικτή την ταυτόχρονη θέαση (και σύγκριση) πολλών δελτίων στην ίδια οθόνη.

Ειδικότερα :

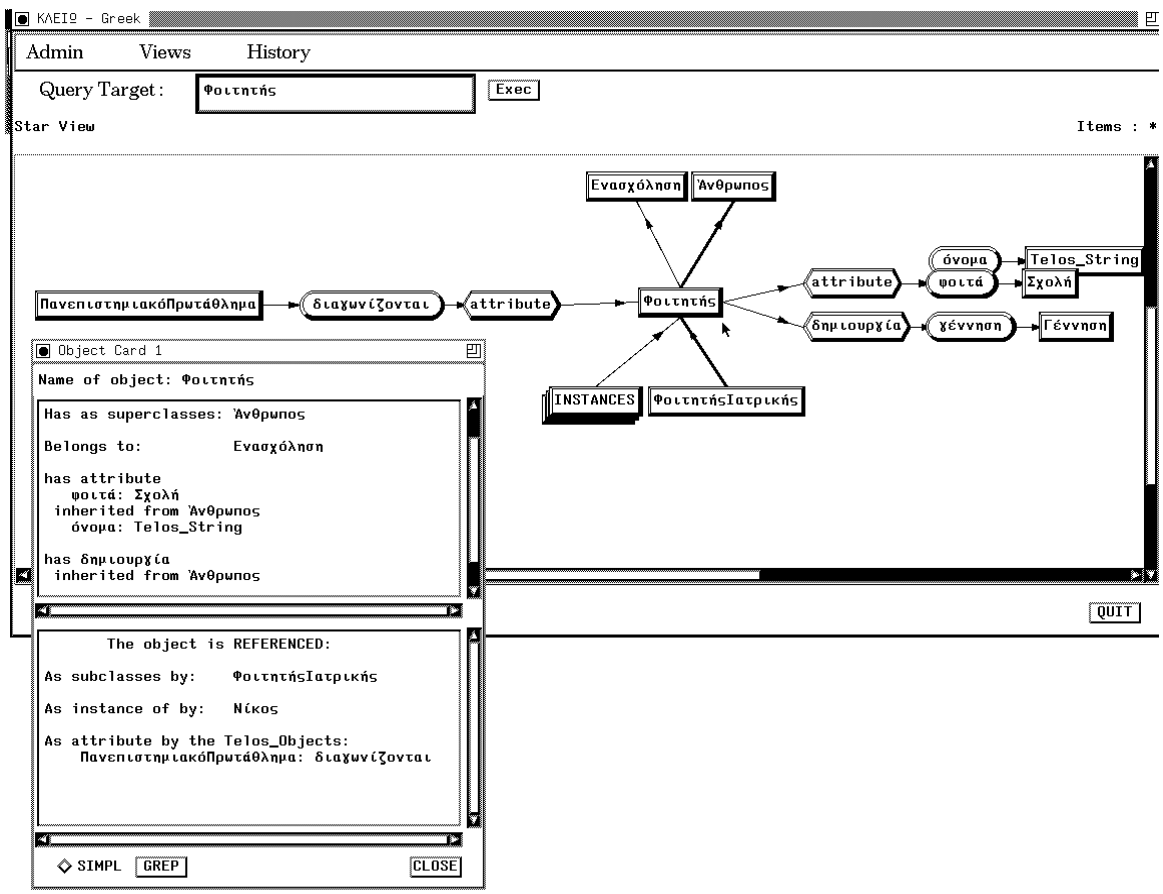
- Φιλτράρισμα με κριτήριο το είδος της πληροφορίας.
  (κλάσεις, υπερκλάσεις, κληρονομούμενα γνωρίσματα,κλπ)

- Φιλτράρισμα με κριτήριο την ενημερωσιμότητα των πληροφοριών.
  Παρουσιάζονται οι σύνδεσμοι που μπορούν να διαγραφούν και γνωστοποιείται το είδος πληροφορίας που μπορεί να εμπλουτιστεί.

- Χωροθέτηση γνωρισμάτων
  Με τη λειτουργία αυτή είναι εφικτή η ομαδοποίηση των γνωρισμάτων και των κατηγοριών που κληρονομεί ένα αντικείμενου με τα κριτήρια :

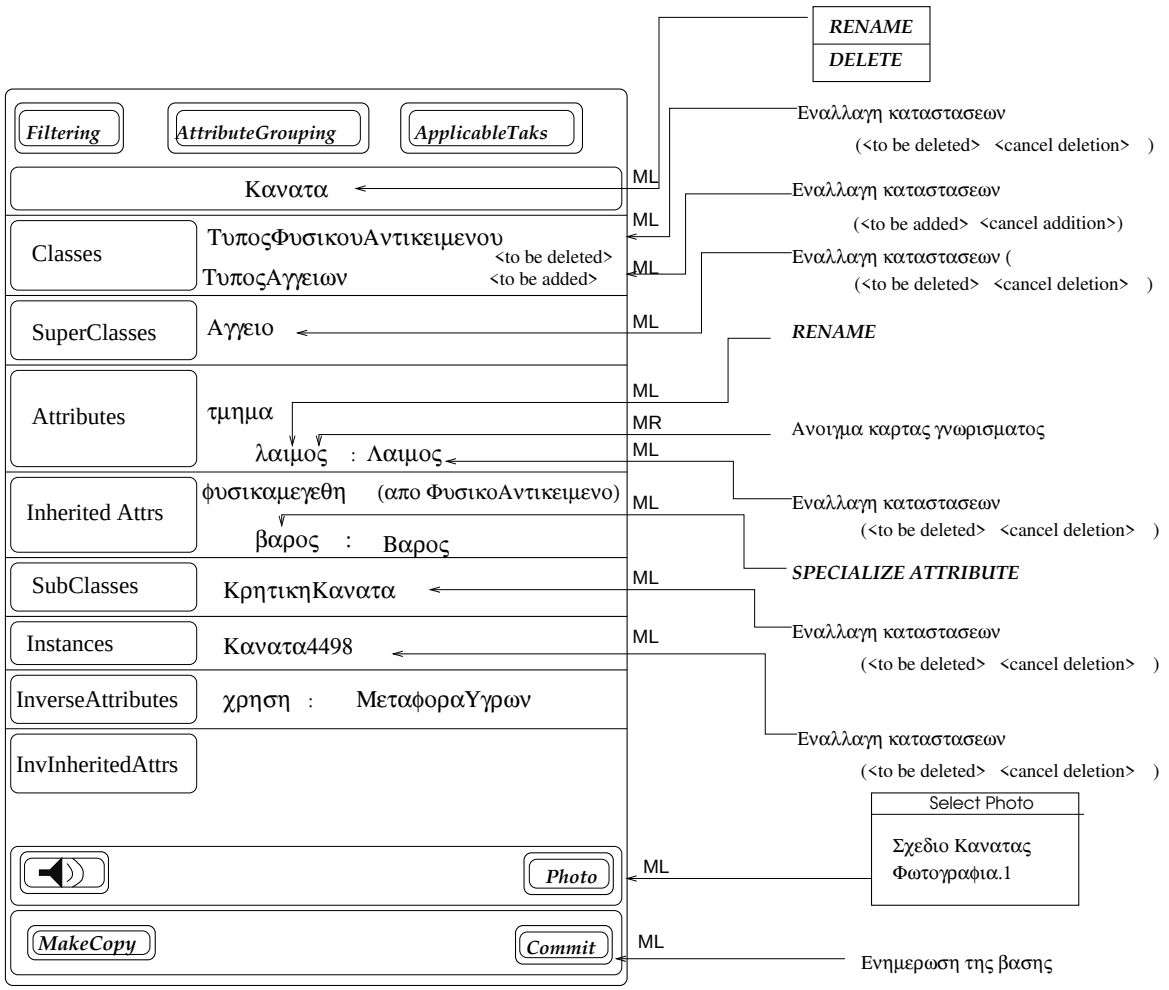
  -- Ομαδοποίηση γνωρισμάτων ανά μετακατηγορία

  -- Ομαδοποίηση γνωρισμάτων ανά κλάση απόδοσης (για τα κληρονομημένα)



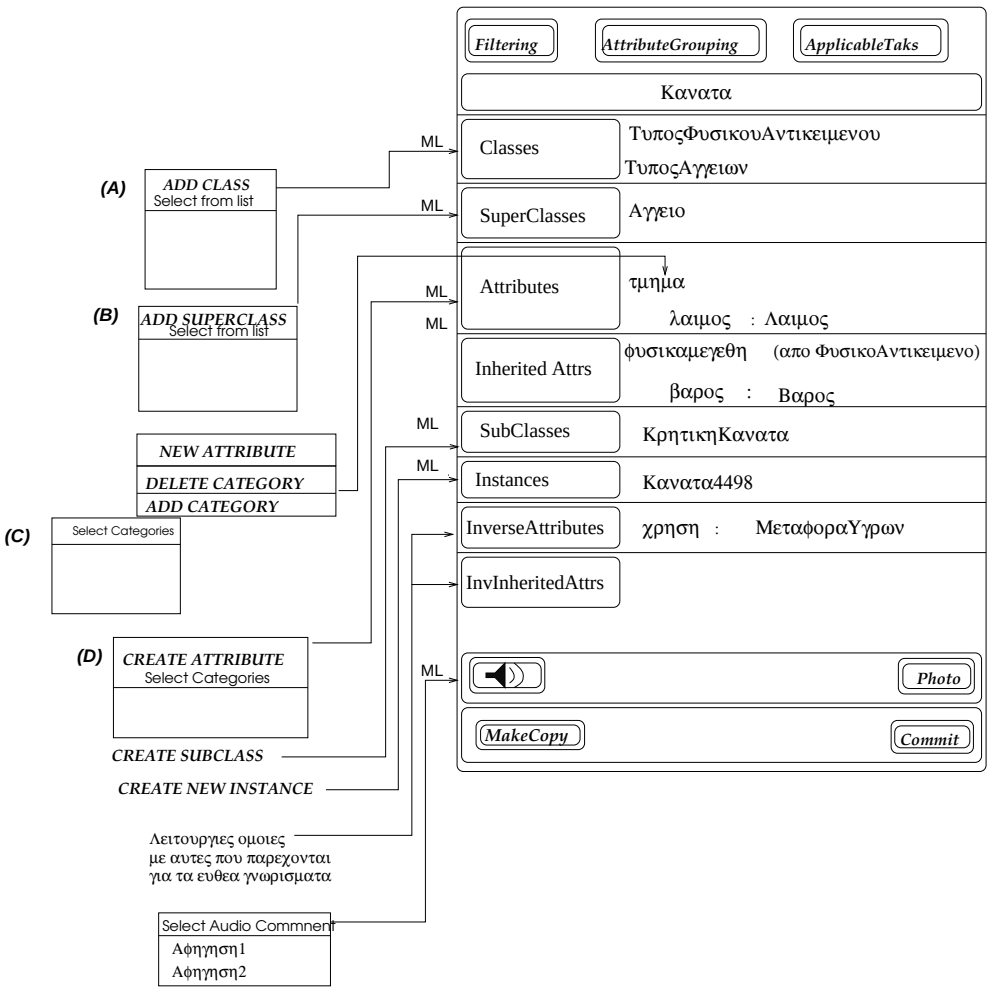
Σχήμα 7.12: Δελτίο Παρουσίασης Οντότητας

Φωτογραφία της επαφής χρήσης του SIS στην οποία φαίνονται το ΔΠΟ της οντότητας *φοιτητής* και ο αντίστοιχος αστεροειδής γράφος. Στην πρώτη γραμμή του δελτίου αναγράφεται το όνομα της εστιακής οντότητας, ενώ στην τελευταία υπάρχουν μερικά κουμπιά ελέγχου. Το κεντρικό τμήμα του δελτίου είναι χωρισμένο σε δύο μέρη. Στο πάνω μέρος αναγράφονται οι σύνδεσμοι που έχουν αρχή το *Φοιτητή*, ενώ στο κάτω μέρος αυτοί που καταλήγουν σε αυτόν. Δίπλα και κάτω από τη φράση *Has as superclasses* φαίνονται όλες οι άμεσες υπερκλάσεις του (*Άνθρωπος*). Δίπλα και κάτω από τη φράση *Belongs to* φαίνονται όλες οι άμεσες κλάσεις του (*Ενασχόληση*). Ακολουθούν όλα τα γνωρίσματα (συμπεριλαμβανομένων αυτών που κληρονομούνται). Για κάθε γνώρισμα παρουσιάζεται το λογικό του όνομα, η τιμή του, οι κλάσεις του (κατηγορίες) και αν πρόκειται για κληρονομημένο γνώρισμα τότε αναγράφεται επίσης η οντότητα από την οποία κληρονομείται. Δίπλα και κάτω από τη φράση *As subclasses by* φαίνονται όλες οι άμεσες υποκλάσεις του *Φοιτητή* (*ΦοιτητήςΙατρικής* ενώ δίπλα και κάτω από τη φράση *As instance of by* φαίνονται όλες οι άμεσες περιπτώσεις του (*Νίκος*). Εν συνεχεία αναγράφονται όλα τα γνωρίσματα που έχουν τιμή (καταλήγουν) στο *Φοιτητή*. Για κάθε γνώρισμα παρουσιάζεται το λογικό του όνομα, η οντότητα στην οποία έχει αποδοθεί και οι κλάσεις του (κατηγορίες).



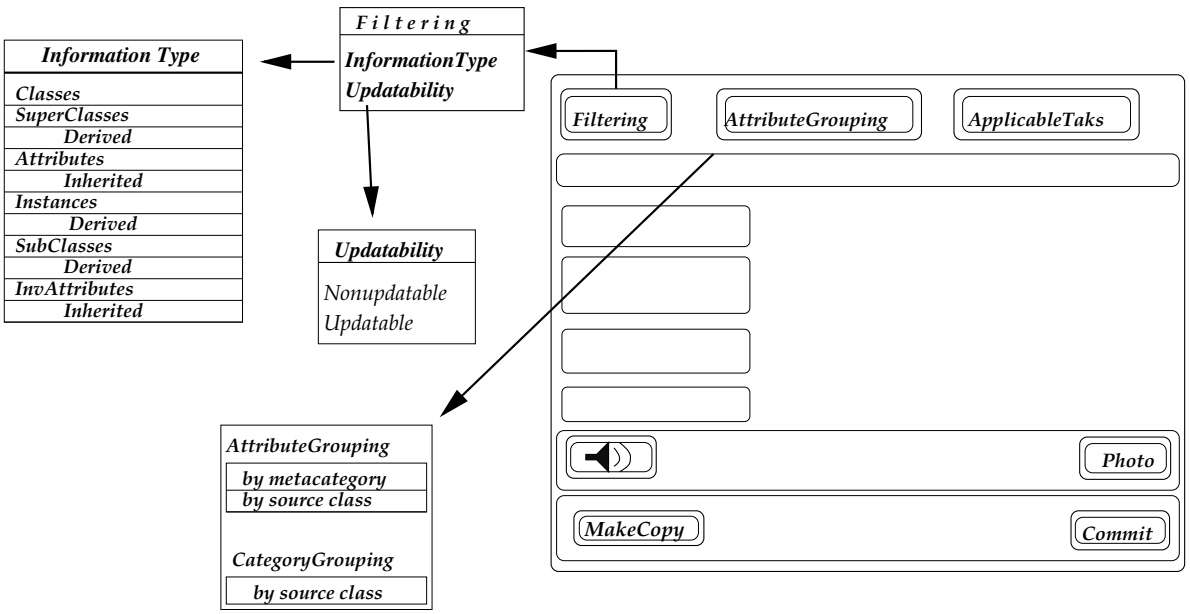
Σχήμα 7.13: Η επαφή χρήσης του προτεινόμενου ΔΠΕΑ (Α)

ML: mouse left button click, MR : mouse right button click.



Σχήμα 7.14: Η επαφή χρήσης του προτεινόμενου ΔΠΕΑ (Β)

ML: mouse left button click, MR : mouse right button click.



Σχήμα 7.15: Η επαφή χρήσης του προτεινόμενου ΔΠΕΑ (Γ)

Ενεργεια χρηστη	Αποτελεσμα
ML στο εστιακο αντικειμενο	Δυνατοτητα μετονομασιας η διαγραφης εστιακου αντικειμενου μεσω ενος μενου 2 επιλογων. Η επιλογη της μετονομασιας κανει την επικεφαλιδα του δελτιου ενημερωσιμη.
ML^2 σε λογικο ονομα οντοτητας	Ανοιγμα ΔΠΕΔ οντοτητας
ML^2 σε λογικο ονομα γνωρισματος	Ανοιγμα ΔΠΕΔ γνωρισματος
ML σε οντοτητα που αναφερεται ως ακρο υπαρχοντος αμεσου συνδεσμου.	Αιτηση διαγραφης συνδεσμου <to be deleted>
ML σε οντοτητα που αναφερεται ως ακρο αμεσου συνδεσμου προς διαγραφη .	Ακυρωνει την αιτηση διαγραφης <cancel deletion>
ML σε οντοτητα που αναφερεται ως ακρο αμεσου προς εισαγωγη .	Ακυρωνει την αιτηση εισαγωγης <cancel addition>
ML σε κληρονομημενο γνωρισμα (ευθυ η αναστροφο)	Αιτηση εξειδικευσης κληρονομημενου γνωρισματος Δημιουργουνται πεδια για τη συμπληρωση του λογικου ονοματος και της τιμης του νεου γνωρισματος.
MR μετα απο ML	Ακυρωνει το αποτελεσμα του ML (Undo like xfig)
ML σε computed superclasses,subclasses	Κανενα αποτελεσμα.

Υπομνημα

MR	mouse right button click
ML	mouse left button click
MM	mouse middle button click
Mx^2	double click of mouse button Mx

Σχήμα 7.16: Λειτουργίες της επαφής χρήσης του προτεινόμενου ΔΠΕΑ

-- Ομαδοποίηση κατηγοριών ανά κλάση απόδοσης
Η ομαδοποίηση αυτή, συν τοις άλλοις, ομαδοποιεί τις ω-κατηγορίες (αυτές έχουν κλάση εκκίνησης μια κλάση συστήματος) και τις ειδικές κατηγορίες γνωρισμάτων (για παράσταση περιορισμών ακεραιότητας) που πρόκειται να υλοποιηθούν στην SIS-Telos .

Οι παράμετροι των λειτουργιών αυτών είναι καλό να μπορούν να εφαρμοσθούν σε όλες τις ανοικτές κάρτες ώστε να είναι εφικτή η ταυτόχρονη παρατήρηση πολλών αντικειμένων με εστίαση σε ένα είδος πληροφορίας.

Κεφάλαιο 8

Συμπεράσματα και Μελλοντικές Κατευθύνσεις

8.1 Αποτελέσματα της εργασίας

Η παρούσα εργασία ασχολήθηκε με την ενημέρωση όψεων σε βάσεις γνώσεις και πιο συγκεκριμένα σε βάσεις γνώσεις της Telos . Μια όψη ενημέρωσης ορίζεται ως ένα μη διατεταγμένο σύνολο στοιχειωδών ενημερώσεων οι οποίες αφορούν τα δεδομένα ή το σχήμα της βάσης [1]. Με την παράσταση και τη διαχείριση ΟΕ επιτυγχάνεται ο έλεγχος των ενημερώσεων που πραγματοποιούνται στη βάση ώστε να μπορεί να διασφαλισθεί η ποιότητα των περιεχομένων της, καθώς και η καλή λειτουργία ενός συστήματος που βασίζεται σε αυτήν. Επιπλέον η ζεύξη τους με ένα σύνολο *γενικών διαδικασιών διαλογικής ενημέρωσης* κάνει εφικτή την ανάπτυξη ενός γενικού δελτίου διαλογικής ενημέρωσης της βάσης, το οποίο να δρά σε όψεις ενημέρωσης. Το δελτίο αυτό μπορεί να βοηθήσει σημαντικά το χρήστη αφού μπορεί να εστιάζει και να υποδεικνύει τις ενημερώσιμες πληροφορίες που αφορούν μια όψη ενημέρωσης.

Για τον ορισμό των ΟΕ αρχικά χρησιμοποιήθηκε ένα *Μοντέλο Διεργασιών Ενημέρωσης* στο οποίο βασίστηκε ανάπτυξη ενός διαλογικού **Δελτίου Εισαγωγής Δεδομένων (ΔΕΔ)** [2] το οποίο με τη σειρά του χρησιμοποιήθηκε στην εφαρμογή **SIB Class Management System**.

[1] Πεδίο εφαρμογής των ΟΕ αυτών είναι κυρίως οι *σχεδιαστικές εφαρμογές* οι οποίες δε χαρακτηρίζονται από προκαθορισμένες και αυστηρές ακολουθίες ενημερώσεων (διεργασίες). Αυτό όμως δε σημαίνει ότι οι προτεινόμενες ΟΕ δεν μπορούν να συνδιασθούν ένα μοντέλο διεργασιών ενημέρωσης.

[2] Η υλοποίηση του ΔΕΔ έγινε από το Δημήτρη Δασκαλάκη

Η δήλωση μιας ΟΕ με το ΜΔΕ γίνεται με χρήση των ερωτηματικών συναρτήσεων της Telos . Η προσέγγιση αυτή δεν επιτρέπει την αποδοτική αξιοποίηση των ΟΕ, αφού δεν είναι εφικτή η αποδοτική εξαγωγή συμπερασμάτων που αφορούν τις ΟΕ. Επίσης η συνέπεια των δηλώσεων με τα περιεχόμενα της βάσης δεν μπορεί να διασφαλισθεί, με αποτέλεσμα την ανάγκη συντήρησης των δηλώσεων καθώς η βάση εξελίσσεται.

Για την άρση των αδυναμιών αυτών αναπτύχθηκε ένα *μεταμοντέλο εργασιών ενημέρωσης* (ΜΜΟΕ) [3], με το οποίο οι αδυναμίες που είχαμε με το ΜΔΕ ξεπεράστηκαν. Οι δηλώσεις ΟΕ γίνονται με τη βοήθεια ενός συνόλου τύπων δήλωσης, ενώ οι δηλώσεις ενσωματώνονται στο μοντέλο εφαρμογής της εκάστοτε βάσης σαν ειδικές σχέσεις της SIS-Telos . Το χαρακτηριστικό αυτό αναβαθμίζει το ρόλο των ΟΕ, αφού έτσι οι δηλώσεις διατηρούν τη συνέπεια τους με τα περιεχόμενα της βάσης, δεν έχουν ανάγκη συντήρησης, η ορθότητα τους εξασφαλίζεται αυτόματα, και το κυριότερο, οι δυνατότητες και οι προοπτικές αξιοποίησης τους αυξάνονται.

Επίσης με το ΜΜΟΕ ο ορισμός ΟΕ είναι εύκολος αφού χρησιμοποιείται η ίδια η SIS-Telos , εκφραστικός και ευέλικτος, αφού υποστηρίζεται η δυνατότητα άμεσων/έμμεσων και θετικών/αρνητικών δηλώσεων, καθώς επίσης και σύνθετων δηλώσεων με τη βοήθεια των σύνθετων τύπων, οι οποίοι μπορούν να δημιουργούνται και να εξελίσσονται από τον ίδιο το χρήστη. Ένα επιπλέον πλεονέκτημα (υλοποίησης) είναι το ότι χρησιμοποιούνται οι υπάρχουσες δυνατότητες του SIS για την αποθήκευση, ανάκτηση και παρουσίαση των δηλωμένων ΟΕ. Για τη βελτιστοποίηση του χρόνου εξαγωγής συμπερασμάτων σχεδιάστηκε και προτείνεται η χρήση μιας κρυφής μνήμης επαγόμενων δηλώσεων.

Τέλος για τον έλεγχο και τη διευκόλυνση των ενημερώσεων που γίνονται διαλογικά προτείνεται ένα σύνολο γενικών διαδικασιών διαλογικής ενημέρωσης οι οποίες αξιοποιούν το περιβάλλον (context) στο οποίο είναι ενταγμένο ένα αντικείμενο και τις δηλωμένες όψεις ενημέρωσης. Μεταξύ των διαδικασιών αυτών περιλαμβάνεται μια σύνθετη εντολή ενημέρωσης, η εντολή αναπαραγωγής, η οποία διευκολύνει σημαντικά την εισαγωγή σύνθετων αντικειμένων. Η εντολή αυτή μπορεί να χρησιμοποιηθεί για την ημιαυτόματη δημιουργία σύνθετων αντικειμένων που ικανοποιούν εξαρχής μια συνθήκη. Η συνθήκη αυτή παριστάνεται με αντικείμενα της βάσης που την ικανοποιούν, τα οποία ονομάζονται υποδείγματα.

[3] Ο σχεδιασμός και η υλοποίηση του ΜΜΟΕ έγιναν από το συγγραφέα.

8.2 Συμπεράσματα και Σύγκριση με Σχετικές Εργασίες

Η έννοια της Όψης Ενημέρωσης συμβάλλει στην εξέλιξη των Συστημάτων Διαχείρισης Βάσεων Δεδομένων και των υπηρεσιών που παρέχουν. Κάνει εφικτό τον έλεγχο των ενημερώσεων (στα δεδομένα και στο σχήμα) και την ανάπτυξη εύκολα προσαρμόσιμων γενικών εργαλείων διαλογικής ενημέρωσης.

Ο πίνακας του σχήματος 8.1 συγκρίνει τα αποτελέσματα της παρούσας εργασίας, και πιο συγκεκριμένα του SIS' (το SIS με το MMOE), με άλλες σχετικές εργασίες ή συστήματα : τις οντοκεντρικές βάσεις δεδομένων O2([1]) και CoOMS([18]), το μοντέλο δικαιοδοσιών GSM([43]) της βάσης γνώσης ConceptBase και το μηχανισμό δικαιοδοσιών για την ΟΒΔ ORION([36]). Τα κριτήρια ως προς τα οποία γίνεται η σύγκριση είναι **η δυνατότητα** :

- *ενημέρωσης του σχήματος μέσω όψεων*
- *διατύπωσης αρνητικών δηλώσεων*
- *ορισμού όψεων ενημέρωσης με λεπτομέρεια στοιχειώδους ενημέρωσης*
- *χρήσης σύνθετων τύπων*
- *ανάλυσης των δηλώσεων και εξαγωγής συμπερασμάτων*
- *έμμεσων δηλώσεων που αφορούν σύνθετα αντικείμενα*
  Πιο συγκεκριμένα αναφερόμαστε στη δυνατότητα ''κληρονομόμησης'' δηλώσεων από ένα σύνθετο αντικείμενο στα τμήματα του (parts). Ενδιαφέρον παρουσιάζει ο τρόπος υλοποίησης των δηλώσεων αυτών, ώστε να επιτρέπει τον αποδοτικό υπολογισμό των κληρονομούμενων δηλώσεων. Η δυνατότητα αυτή υποστηρίζεται, με αρκετές όμως αδυναμίες, από την ORION. Η εισαγωγή της στο MMOE , προϋποθέτει τον ακριβή ορισμό της έννοιας του ''σύνθετου αντικειμένου'' στην Telos , ο οποίος με τη σειρά του απαιτεί την υλοποίηση των ειδικών κατηγοριών γνωρισμάτων (βλέπε [25]).
- *δηλώσεων σε υποσύνολα κλάσεων*
  Αναφερόμαστε σε υποσύνολα τα οποία ορίζονται με τη βοήθεια μιας ερωτηματικής γλώσσας. Η χρήση του ΜΔΕ είχε αυτή τη δυνατότητα αλλά είχε τις αδυναμίες που αναλύθηκαν στο κεφάλαιο 5.

Από τον πίνακα παρατηρούμε ότι η χρήση του MMOE συνεπάγεται πολλά πλεονεκτήματα. Συμπερασματικά μπορούμε να πούμε ότι το MMOE μπορεί με ομοιόμορφο τρόπο

ΣΔΒΔ / Χαρακτηρ.	**O2**	**CoOMS NO^2**	**ConceptBase** GSM	**ORION** (Kim)	**SIS′**
Ενημερωση Σχηματος	−	−	+	+	+
Αρνητικες Δηλωσεις	+	−	−	+	+
Λεπτομερεια ΣΕ	−	−	+	ιεραρχ. οργανωση	+
Συνθετοι Τυποι			−		+
Αναλλοιωτο Μοντελο Εφαρμ.	−	−	+	+	+
Συνθεση ΟΕ	+	−	+	ιεραρχ. οργανωση	+
Αναλυση Δηλωσεων	−	−	+	−	+
Εμμεσες δηλ. συνθετων αντικ.	−	−	−	+	−
Δηλωσεις σε υποσυνολα κλασεων	+	+	−	−	(−)

Σχήμα 8.1: Σύγκριση με σχετικές εργασίες

να ορίζει όψεις ενημέρωσης των δεδομένων ή του σχήματος, ενώ η δυνατότητα χρήσης άμεσων/έμμεσων, θετικών/αρνητικών δηλώσεων, παρέχει την ευελιξία του μηχανισμού που προτείνεται στο [36]. Οι ΟΕ που ορίζει αποτελούνται από ΣΕΕ και όχι από αντικείμενα, άρα προσφέρει μεγαλύτερη εκφραστική ικανότητα από συστήματα όπως τα [18], [1] και συνάμα ευκολία, αφού υποστηρίζει σύνθετους τύπους δήλωσης. Η χρήση ενός μεταμοντέλου προσφέρει άμεση σύνδεση μεταξύ των ΟΕ και του μοντέλου εφαρμογής προσφέροντας έτσι το προαναφερθέν (στην ενότητα 8.1) σύνολο πλεονεκτημάτων, ενώ συνάμα δεν αλλοιώνει το μοντέλο εφαρμογής, όπως συμβαίνει σε συστήματα όπως το [40].

Γενικά μπορούμε να πούμε ότι η παρούσα εργασία αποτελεί ένα βήμα της πορείας προς την ανάπτυξη Συστημάτων Διαχείρισης Βάσεων Δεδομένων στα οποία, η έννοια της όψης [4] αποτελεί έναν ισχυρό εγγενή και βασικό μηχανισμό τους.

8.3 Μελλοντική Εργασία και Κατευθύνσεις

8.3.1 Αξιοποίηση του MMOE

Υλοποίηση του διαλογικού δελτίου ενημέρωσης για την Telos το οποίο να υλοποιεί τις διαδικασίες ενημέρωσης που περιγράφηκαν στο κεφάλαιο 7, ώστε το MMOE να αξιολογηθεί στην πράξη.

Ενσωμάτωση του ελέγχου ενημερώσεων στον πυρήνα της Telos ώστε να ελέγχονται οι ενημερώσεις που γίνονται μαζικά (μέσω του parser).

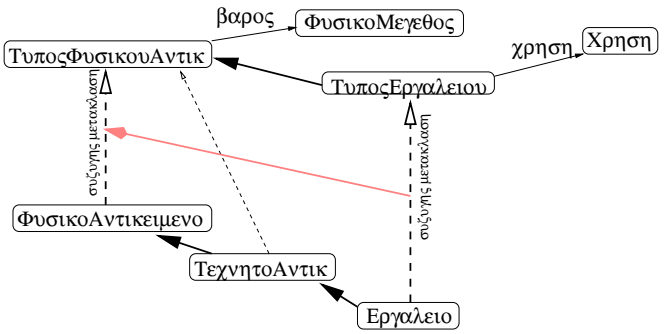
8.3.2 Επεκτάσεις της SIS-Telos

Πιθανές εξελίξεις στο μοντέλο δεδομένων της SIS-Telos θα είχαν άμεση επίδραση στις εκφραστικές ικανότητες του MMOE .

Για παράδειγμα μια τέτοια εξέλιξη είναι η αναβάθμιση της έννοιας των **συζυγών ζευγών κλάσεων, μετακλάσεων** η οποία προτάθηκε στην εργασία [26] και εμφανίζεται επίσης στην εργασία [34]. Αν επεκτείνουμε τους δομικούς περιορισμούς της SIS-Telos , ώστε κάθε περίπτωση της μετακλάσης να πρέπει να είναι συνάμα και υποκλάση της συζυγούς υπερκλάσης, και επιπλέον επιτρέψουμε την εξειδίκευση των συζυγών ζευγαριών (δες σχήμα 8.2), τότε μια δήλωση αναγνωριστικό ενημέρωσης $AddIn$ σε μια μετακλάση ορίζει κάτι πιο εξειδικευμένο : τις ΣΕΕ ταξινόμησης των υποκλάσεων της συζυγούς υπερκλάσης

[4] Ή *context* όπως αναφέρεται στο [7].

στη συζυγή μετακλάση. Άρα καταλαβαίνουμε τη άμεση σχέση μοντέλου δεδομένων και ΟΕ.



Σχήμα 8.2: Εξειδίκευση συζυγών ζευγαριών

Επίσης η εισαγωγή στην SIS-Telos των *ειδικών κατηγοριών γνωρισμάτων* θα κάνει εφικτή την υλοποίηση και χρήση της εντολής αναπαραγωγής **MakeCopy** .

8.3.3 Επεκτάσεις του SIS

Προτείνεται η εξέλιξη της επαφής χρήσεως του SIS ώστε να μπορεί να αλλάζει η διαμόρφωσή της (setup) κατά το χρόνο εκτέλεσης (και όχι μόνο κατά την εκκίνηση του συστήματος). Έτσι για κάθε ΟΕ θα μπορεί να χρησιμοποιείται ξεχωριστή διαμόρφωση, τέτοια ώστε να βοηθά το χρήστη στην διεκπεραίωση της εργασίας που επιθυμεί.

8.3.4 Επεκτάσεις του MMOE

Οι επεκτάσεις ή εξελίξεις του MMOE αφορούν κυρίως την **εκφραστική του ικανότητα** και το **χρόνο υπολογισμού των επαγόμενων δηλώσεων**. Η εκφραστική του ικανότητα καθορίζεται από τους υποστηριζόμενους τύπους δήλωσης και τη σημασιολογία τους.

Ιδιαίτερο ενδιαφέρον παρουσιάζει η μελέτη έμμεσων δηλώσεων που αφορούν σύνθετα αντικείμενα, ή πιο συγκεκριμένα, η ‘‘κληρονομόμηση’’ δηλώσεων από ένα αντικείμενο στα τμήματα του.

Ενδιαφέρον επίσης παρουσιάζουν τα τεχνικά θέματα που αφορούν την υλοποίηση του MMOE . Η υλοποίηση των απλών τύπων δήλωσης στον πυρήνα της Telos θα βελτίωνε πολύ τις επιδόσεις, ενώ η προτεινόμενη ΚΜΕΔ επιδέχεται μελέτη και δοκιμές.

Παράρτημα Α

Συντομογραφίες

Στο κείμενο χρησιμοποιούνται κάποιες συντομογραφίες οι οποίες αναγράφονται παρακάτω συνοδευόμενες από τη σημασία τους και την ενότητα στην οποία πρωτοεμφανίζονται.

Συντομογραφία	**Επεξήγηση**	**Ενότητα**
ΑΕΤ	Αρχείο Εντολών Telos	5.1.4
API	Application Program Interface	4
ΔΕ	Δηλώσεις Ενημέρωσης	6.2
ΔΕΔ	Δελτίο Εισαγωγής Δεδομένων	1.2
ΔΠΕΑ	Δελτίο Παρουσίασης και Ενημέρωσης Αντικειμένου	7.5
ΔΠΟ	Δελτίο Παρουσίασης Οντότητας	7.3
DC	Describe Class	5.3
DO	Describe Object	5.3
ΕΑ	Εστιακό Αντικείμενο	7.1
ΕΑΔ	Εστιακό Αντικείμενο Δήλωσης	6.3
ΕΓ	Εξέλιξη Γνώσης	2.1
FD	Focus Declarations	6.3
ΚΔΟ	Κλάση Δημιουργίας Οντοτήτων	6.5

Συντομογραφία	**Επεξήγηση**	**Ενότητα**
ΚΜΕΔ	Κρυφή Μνήμη Επαγόμενων Δηλώσεων	6.9.4
ΜΔΕ	Μοντέλο Διεργασιών Ενημέρωσης	1.2
ΜΜΟΕ	ΜεταΜοντέλο Όψεων Ενημέρωσης	1.2
ΟΒΔ	Οντοκεντρικές Βάσεις Δεδομένων	2
ΟΕ	Όψη Ενημέρωσης	1.1
ΣΔΒΔ	Σύστημα Διαχείρισης Βάσεων Δεδομένων	1.1
ΣΓΕ	Στοιχειώδης Γλώσσα Ενημερώσεων	4.1
ΣΕΕ	Στοιχειώδεις Εντολές Ενημέρωσης	1.1
SIS	Semantic Index System	3.4
qi	Query Interface	5.1.1

Παράρτημα Β

Το Μοντέλο Διεργασιών Ενημέρωσης

Ακολουθεί η περιγραφή του Μοντέλου Διεργασιών Ενημέρωσης σε μορφή Telos .

TELL Individual Users **in S_Class** with
attribute
task : UpdateTask
end

TELL Individual UpdateTask **in S_Class** with
attribute
object : QueryMacro;
export : QueryMacro ;
update : Update
end

TELL Individual Update **in S_Class** with
attribute
update_type: UpdateType;
argument : QueryMacro;
sequence : Update
end

TELL Individual UpdateType **in S_Class**
end

TELL Individual CHANGE_CLASSES **in Token** ,
UpdateType
end

TELL Individual CHANGE_SUPERCLASSES **in Token** ,
UpdateType
end

TELL Individual CREATE_INDIVIDUAL **in Token** ,
UpdateType
end

TELL Individual RENAME_INDIVIDUAL **in Token** ,
UpdateType
end

TELL Individual DELETE_INDIVIDUAL **in Token** ,
UpdateType
end

TELL Individual CHANGE_ATTRIBUTES **in Token** ,
UpdateType
end

Παράρτημα C

Οι Ερωτηματικές Εντολές του qi

Ο κατάλογος των βασικότερων εντολών του **qi** ακολουθεί παρακάτω (η πλήρης περιγραφή υπάρχει στο [23]).

Κάθε εντολή (εκτός της **scn**) δέχεται σαν όρισμα ένα φυσικό αριθμό *s* , ο οποίος καθορίζει το αντικείμενο ή τα αντικείμενα στα οποία απευθύνεται η ερώτηση και επιστρέφει ένα σύνολο αντικειμένων. Αν το *s* ισούται με 0, η ερώτηση απευθύνεται στο εστιακό αντικείμενο, διαφορετικά απευθύνεται σε κάθε στοιχείο του συνόλου αντικειμένων με αναγνωριστικό *s* .

scn *l* (set current node)
Το αντικείμενο με λογικό όνομα *l* τίθεται ως εστιακό.

gc *s* (**g**et **c**lasses)
Το σύνολο άμεσων κλάσεων του *s*

gac *s* (get all classes)
Το σύνολο άμεσων και έμμεσων κλάσεων του *s*

gSc *s* (get System classes)
Η κλάση συστήματος του *s*

gaSc *s* (get all System classes)
Το σύνολο με την άμεση και τις έμμεσες κλάσεις συστήματος του *s*

gi *s* (get instances)
Το σύνολο άμεσων περιπτώσεων του *s*

gai *s* (get all instances)
Το σύνολο άμεσων και έμμεσων περιπτώσεων του *s*

gsc *s* (get superclasses)

Το σύνολο άμεσων υπερκλάσεων του *s*

gasc *s* (get all superclasses)

Το σύνολο άμεσων και έμμεσων υπερκλάσεων του *s*

gsb *s* (get subclasses)

Το σύνολο άμεσων υποκλάσεων του *s*

gasb *s* (get all subclasses)

Το σύνολο άμεσων και έμμεσων υποκλάσεων του *s*

glf *s* (get links from)

Το σύνολο άμεσων γνωρισμάτων του *s*

gilf *s* (get inherited links from)

Το σύνολο άμεσων και κληρονομημένων γνωρισμάτων του *s*

glt *s* (get links to)

Το σύνολο άμεσων γνωρισμάτων που καταλήγουν στο *s*

gilt *s* (get inherited links to)

Το σύνολο άμεσων και κληρονομημένων γνωρισμάτων που καταλήγουν στο *s*

gfv *s* (get from value)

Το αντικείμενο εκκίνησης του *s* .

glt *s* (get link to)

Η οντότητα στην οποία καταλήγει το *s* .

glfc *s* , ctg (get link from by category)

Το σύνολο των άμεσων γνωρισμάτων του *s* που έχουν κατηγορία ctg.

gfnc *s* , ctg (get from node by category)

Το σύνολο των αντικειμένων στα οποία έχουν αποδοθεί γνωρίσματα που έχουν τιμή το *s* και έχουν κατηγορία ctg.

gtnc *s* , ctg (get to node by category)

Το σύνολο των οντοτήτων που αποτελούν τιμή γνωρισμάτων του *s* και έχουν κατηγορία ctg.

Παράρτημα D

Βιβλιογραφία ανά θεματική ενότητα

- **Οντοκεντρικά Συστήματα Διαχείρισης Βάσεων Δεδομένων (Object Oriented DBMS)**
  [22], [28], [10], [8], .
- **Repositories** [7], [39] .
- **Σημασιολογικά Δίκτυα (Semantic Network)**
  [11], [25], [21], [33], [41], [34], [45].
- **Εξέλιξη Γνώσης (Knowledge Evolution)**
  [4], [37], [35], [5].
- **Όψεις (Views)**
  [19], [3], [40], [12], [1], [6], [35], [42], [20], [18], [2], [27], [31], [15].
- **Μηχανισμοί Ελέγχου Προσβάσεων (Access Contol and Authorization Mechanims)**
  [36], [43], [20], [44], [30].
- **Μεταμοντέλα (Metamodels)**
  [29], [5], [16].
- **Επαφές Χρήσεως (User Interfaces)**
  [9], [17] .
- **Telos και Σύστημα Σημασιολογικού Ευρετηριασμού (SIS)**
  [32], [14], [13], [24], [25], [26].

Βιβλιογραφία

[1] Serge Abiteboul and Anthony Bonner. ‘‘Objects and Views”. In *Proceedings of ACM SIGMOD Conference of Management of Data*, pages 238--247, February 1991.

[2] Rakesh Agrawal and Linda G. DeMichiel. ‘‘Type Derivation Using the Projection Operation”. *Information Systems*, 19(1):55--68, 1994.

[3] F. Bancilhon and N. Spyratos. ‘‘Update Semantics of Relational Views”. *ACM Transactions on Database Systems*, December 1981.

[4] Jay Banerjee, Won Kim, Hyoung-Joo Kim, and Henry F. Korth. ‘‘Semantics and Implementation of Schema Evolution in Object-Oriented Databases”. *SIGMOD RECORD*, 16(3):311--322, 1987. ACM SIGMOD Conference of Management of Data.

[5] Z. Bellahsene. ‘‘An Active Meta-Model for Knowledge Evolution in an Object-oriented Database”. Technical report. LIRMM UMR CNRS/Montpellier II, 1992.

[6] Z. Bellahsene. ‘‘The Point of View Notion for Defining and Updating Views in an Object-oriented Database”. Technical report. LIRMM UMR CNRS/Montpellier II, 1992.

[7] Philip A. Bernstein and Umeshwar Dayal. ‘‘An Overview of Repository Technology”. In *Proceedings of the 20th VLDB Conference*, pages 705--713, Santiago, Chile, 1994.

[8] Ralph B. Bisland. *‘‘Database Management. Developing Application System with Oracle”.* Prentice-Hall, 1989.

[9] P. Borras, J.C. Mamou, D. Plateau, B. Poyet, and D. Tallot. ‘‘Building User Interfaces for Database Applications : The O2 experience”. *SIGMOD RECORD*, 21(1):32--38, March 1992.

[10] A. Brown. *‘‘Object Oriented Databases. Applications in Sofware Engineering”.* McGraw-Hill, 1988.

[11] Dionysios C. Tsichritzis and Frederic H. Lochovsky. ‘‘*Data Models*”, chapter 10, pages 210--224. Prentice-Hall, 1982.

[12] Daniel K.C. Chan and David A. Kerr. ‘‘Improving One’s Views of Object-Oriented Databases”. In *Proceedings of the Colloquium on Object Orientation in Databases and Software Engineering, the 62nd Congress of l’ Association Canadienne Francaise pour l’ Avancement de Sciences*, Montreal, Canada, May 1994.

[13] Panos Constantopoulos and Martin Doerr. ‘‘Component Classification in the Software Information Base”. in O.Nierstrasz and D.Tsichritzis, eds.,Object-Oriented Software Composition, Prentice-Hall,1995.

[14] Panos Constantopoulos, Martin Doerr, and Yannis Vassiliou. ‘‘Repositories for Software Reuse : The Software Information Base”. In *Proceedings IFIP WG 8.1 Conference on Information System Development Process*, pages 285--307, Como,Italy, September 1993.

[15] Bogdan Czejdo and David W. Embley. ‘‘View Specification and Manipulation for a Semantic Data Model”. *Information Systems*, 16(6):585--612, 1991.

[16] Oscar Diaz and Norman W. Paton. ‘‘Extending ODBMS Using Metaclasses”. *IEEE Software*, pages 40--47, May 1994.

[17] Belinda B. Flynn and David Maier. ‘‘Supporting Display Generation for Complex Database Objects”. *SIGMOD RECORD*, 21(1):18--24, March 1992.

[18] Andreas Geppert, Stefan Scherrer, and Klaus R. Dittrich. ‘‘Derived Types and Subschemas : Towards Better Support for Logical Data Independence in Object-Oriented Data Models”. Technical Report 93.27, Institut für Informatik, Universität Zürich, June 1993.

[19] Georg Gottlob, Paolo Paolini, and Roberto Zicari. ‘‘Properties & Update Semantics of Consistent Views”. *ACM Transactions on Database Systems*, 13(4):486--524, December 1988.

[20] Brent Hailpern and Harold Ossher. ‘‘Extending Objects to Support Multiple Interfaces and Access Control”. *IEEE Transactions on Software Engineering*, 16(11):1247--1257, November 1990.

[21] Richard Hull and Roger King. ‘‘Semantic Database Modeling”. *ACM Computing Surveys*, 19(3):202--260, September 1987.

[22] A.R. Hurson, Simin H. Pakzad, and Jia bing Cheng. ''Object-Oriented Database Management Systems : Evolution and Performance Issues". *IEEE Computer*, pages 48--60, February 1993.

[23] Κώστας Νταντουρής . ''*Βιβλιοθήκη Στοιχειωδών Ερωτηματικών Συναρτήσεων και Επεξεργασία Ερωτήσεων για την γλώσσα TELOS*", Μεταπτυχιακή Εργασία, Τμήμα Επιστήμης Υπολογιστών, Πανεπιστήμιο Κρήτης, Ιούλιος, 1993 . (ftp.csd.uch.gr/pub/thesis/MS/dadouris/thesis.ps).

[24] Γιώργος Γεωργιαννάκης . ''*Ο Μηχανισμός Αποθήκευσης και Διαχείρισης Οντοτήτων για τη Γλώσσα Παράστασης Γνώσης Telos*", Μεταπτυχιακή Εργασία, Τμήμα Επιστήμης Υπολογιστών, Πανεπιστήμιο Κρήτης, Φεβρουάριος, 1994 . (ftp.csd.uch.gr/pub/thesis/MS/georgianakis/thesis.ps).

[25] Μάνος Θεοδωράκης . ''*Εμβέλεια Ονόματος σε Σημασιολογικά Μοντέλα Δεδομένων*", Μεταπτυχιακή Εργασία, Τμήμα Επιστήμης Υπολογιστών, Πανεπιστήμιο Κρήτης, Σεπτέβρης, 1995 . (ftp.csd.uch.gr/pub/thesis/MS/theodorakis/thesis.ps).

[26] Μαρία Χριστοφοράκη . ''*Τεκμηρίωση Πολιτιστικών Αγαθών με το Σύστημα ΚΛΕΙΩ*", Μεταπτυχιακή Εργασία, Τμήμα Επιστήμης Υπολογιστών, Πανεπιστήμιο Κρήτης, Φεβρουάριος,1994 . (ftp.csd.uch.gr/pub/thesis/MS/christoforaki/thesis.ps).

[27] Arthur M. Keller. ''The Role of Semantics in Translating View Updates". *IEEE Computer*, pages 63--73, January 1986.

[28] Won Kim. ''Object-Oriented Databases : Definition and Research Dierctions". *IEEE Transactions on Knowledge and Data Engineering*, 2(3):327--341, September 1990.

[29] Leo Mark and Nick Roussopoulos. ''Metadata Management". *IEEE Computer*, pages 26--36, December 1986.

[30] Dave Menninger. ''A Room with a View : Give Users Access to their Data". *Data Based Advisor*, 11(10):107--110, October 1993.

[31] M. Missikoff and M. Scholl. ''An Algorithm for Insertion into a Lattice: Application to Type Classification". In *Proceedings 3nd International Conference on Foundations of Data Organisation and Algorithms - FODO*, pages 64--82, Paris, June 1989. Springer-Verlag.

[32] John Mylopoulos, Alex Borgida, Matthias Jarke, and Manolis Koubarakis. ''Telos : Representing Knowledge about Information Systems". *ACM Transactions on Information Systems*, 8(4), October 1990.

[33] Joan Peckham and Fred Maryanski. ‘‘Semantic Data Models”. *ACM Computing Surveys*, 20(3):153--189, September 1988.

[34] Alain Pirotte, Esteban Zimanyi, David Massart, and Tatiana Yakusheva. ‘‘Materialization: a powerful and ubiquitous abstraction pattern”. In *Proceedings of the 20th VLDB Conference*, pages 630--641, Santiago, Chile, 1994.

[35] YoungGook Ra and Elke A. Rundensteiner. ‘‘A Transparent Object-Oriented Schema Change Approach Using View Evolution”. Technical Report MI 48109-2122, Software Systems Research Laboratory -Department of Electrical Engineering and Computer Science -University of Michigan, April 1994.

[36] Fausto Rabiti, Elisa Bertino, Won Kim, and Darrell Woelk. ‘‘A Model of Authorization for Next-Generation Database Systems”. *ACM Transactions on Object Oriented Database Systems-TODS*, 16(1), March 1991.

[37] John F. Roddick. ‘‘Schema Evolution in Database Systems - An Annotated Bibliography”. *SIGMOD RECORD*, 21(4):35--40, December 1992.

[38] W.D. Ross. ‘‘Αριστοτέλης” . Μορφωτικό Ίδρυμα Εθνικής Τραπέζης , 1993.

[39] Prieto-Diaz Ruben. ‘‘Implementing Faceted Classification for Software Reuse”. *Communications of the ACM*, 34(5), 1991.

[40] Marc H. Scholl, Cristian Laasch, and Markus Tresch. ‘‘Updatable Views in Object-Oriented Databases”. In *Proceedings of the 2nd International Conference on Deductive and Object-Oriented Databases*, pages 189--207, 1991.

[41] Lokendra Shastri. ‘‘Default Reasoning in Semantic Networks: A Formalization of Recognition and Inheritance ”. *Artificial Intelligence*, 39:283--355, 1989.

[42] John J. Shilling and Peter F. Sweeney. ‘‘Three Steps to Views : Extending the Object-Oriented Paradigm”. In *Proceedings of Object-Oriented Programming, Systems, Languages and Applications - OOPSLA*, pages 353--361, October 1989.

[43] Gerhard Steinke. ‘‘*Task-Based Security for Knowledge Base Systems*”. PhD thesis, University of Passau, July 1992.

[44] M.B. Thuraisingham. ‘‘Mandatory Security in Object-Oriented Database Systems”. In *Proceedings of Object-Oriented Programming, Systems, Languages and Applications - OOPSLA*, pages 203--210, October 1989.

[45] David Touretzky, John Horty, and Richmond Thomason. ‘‘A Clash of Intuitions : The Current State of Nonmonotonic Multiple Inheritance Systems”. In *Proceedings of the 10th IJCAI, Milan, Italy*, pages 476--482, 1989.