Πανεπιστήμιο Κρήτης
Σχολή Θετικών Επιστημών
Τμήμα Επιστήμης Υπολογιστών

Πύλη Διασύνδεσης του Παγκόσμιου Ιστού με το Σύστημα Σημασιολογικού Ευρετηριασμού

Αικατερίνη Χ. Τόλιου

Μεταπτυχιακή Εργασία

Ηράκλειο, Οκτώβριος 1998

Αφιερώνεται στους γονείς μου
και στον αδερφό μου Κωστή

Πύλη Διασύνδεσης του Παγκόσμιου Ιστού με το Σύστημα Σημασιολογικού Ευρετηριασμού

Αικατερίνη Χ. Τόλιου

Μεταπτυχιακή Εργασία

Τμήμα Επιστήμης Υπολογιστών
Πανεπιστήμιο Κρήτης

Περίληψη

Μια Πύλη Διασύνδεσης Βάσεων Δεδομένων με τον Παγκόσμιο Ιστό (ΠΔΒΔΠΙ), είναι ένα ενδιάμεσο λογισμικό που επιτρέπει την σύνδεση συστημάτων βάσεων δεδομένων με τον Παγκόσμιο Ιστό. Μια ΠΔΒΔΠΙ μπορεί να είναι γενική, επιτρέποντας την πρόσβαση σε πολλές και ετερογενείς βάσεις δεδομένων, ή να είναι ειδική επιτρέποντας επικοινωνία με συγκεκριμένη βάση δεδομένων.

Σκοπός της παρούσας εργασίας είναι η κατασκευή μιας πύλης διασύνδεσης του Συστήματος Σημασιολογικού Ευρετηριασμού με τον Παγκόσμιο Ιστό. Το εργαλείο επιτρέπει την πλοήγηση και παρουσίαση σε HTML των οντοτήτων οποιασδήποτε βάσης που έχει κατασκευασθεί σύμφωνα με το μοντέλο δεδομένων της Telos.

Για τον καθορισμό της παρουσίασης των οντοτήτων σε HTML έχει κατασκευασθεί ένα μοντέλο παρουσίασης ιστοσελίδων. Με τη χρήση του μοντέλου παρουσίασης ιστοσελίδων μπορεί να προσδιορισθεί παρουσίαση για κάθε οντότητα της Telos, ανεξαρτήτως του επιπέδου ταξινόμησης στο οποίο ανήκει. Επιπλέον, το μοντέλο διαχωρίζει τη λογική, σημασιολογική και φυσική δομή της παρουσίασης και επιτρέπει τον καθορισμό όψεων παρουσίασης.

Η χρήση του μοντέλου επιτρέπει στον προγραμματιστή εφαρμογής να προσδιορίζει με δηλωτικό τρόπο την παρουσίαση ενός ή περισσότερων αντικειμένων.

Η πύλη διασύνδεσης που υλοποιήθηκε, βασιζόμενη στην πληροφορία που περιέχει το μοντέλο παρουσίασης, παράγει την τελική παρουσίαση HTML.

Το σύστημα χρησιμοποιήθηκε σε ένα σύνολο εφαρμογών με ικανοποιητικά αποτελέσματα.

Επόπτης: Πάνος Κωνσταντόπουλος
Καθηγητής Επιστήμης Υπολογιστών
Πανεπιστήμιο Κρήτης

World Wide Web Gateway to the Semantic Index System

Katerina C. Toliou

Master of Science Thesis

Computer Science Department
University of Crete

Abstract

A Web Database Gateway (WDG) is a middleware allowing the cooperation between the World Wide Web and database systems. A WDG may be generic, allowing access from the World Wide Web to different and heterogeneous database systems or it may be specific allowing access to a given database system.

This thesis deals with the development of a gateway between the Semantic Index System and the World Wide Web. This tool allows the navigation and presentation in HTML of any database which is based on the data model of the knowledge representation language Telos.

We have defined a presentation model in order to specify the HTML presentation of objects. This model is a part of the database and is applicable to every database object independently from its classification level. The model separates the semantic, physical and logical structure of the presentation. Furthermore, different presentation views can be defined over the same data.

The presentation model provides the application programmer with the means to prescribe the presentation of an object or a set of objects in a declarative way.

The gateway uses the information from the presentation model and produces the final HTML page.

The tool has been tested in several SIS applications producing satisfactory results.

Supervisor: Panos Constantopoulos
Professor of Computer Science
University of Crete

Ευχαριστίες

Στο σημείο αυτό θα ήθελα να ευχαριστήσω θερμά τον επόπτη καθηγητή μου κ. Πάνο Κωνσταντόπουλο που μου εμπιστεύθηκε την εργασία αυτή καθώς και για τη μέχρι τώρα καθοδήγηση και συμβουλές του. Θεωρώ ότι η συνεργασία μας υπήρξε καθοριστική για τον ερευνητικό μου προσανατολισμό.

Επίσης, ευχαριστώ τα μέλη της εισηγητικής επιτροπής κ. Ευάγγελο Μαρκάτο και κ. Γεώργιο Δ. Σταμούλη για τις εποικοδομητικές τους παρατηρήσεις και διορθώσεις στην εργασία αυτή.

Δεν θα μπορούσα να παραλείψω να πω ένα μεγάλο ευχαριστώ στον επιβλέποντα της εργασίας μου κ. Βασίλη Χριστοφίδη για την άψογη πραγματικά συνεργασία μας, την σωστή του καθοδήγηση και τις γνώσεις, τις οποίες απέκτησα κοντά του τα τελευταία δύο χρόνια.

Θα πρέπει να ευχαριστήσω το Ινστιτούτο Πληροφορικής του Ι.Τ.Ε. και κυρίως τον Τομέα Πληροφοριακών Συστημάτων για την υλικοτεχνική υποστήριξη που μου παρείχε τα τελευταία χρόνια. Επίσης, θέλω να ευχαριστήσω όλα τα μέλη του Τομέα Πληροφοριακών Συστημάτων και Τεχνολογίας Λογισμικού για τη βοήθειά τους κατά τη διάρκεια της παρούσας εργασίας. Ευχαριστώ επίσης το Τμήμα Επιστήμης Υπολογιστών του Π.Κ. για το σύγχρονο περιβάλλον εργασίας που μου παρείχε και τις γνώσεις που μου έδωσε την ευκαιρία να αποκτήσω.

Ακόμη, θα ήθελα να ευχαριστήσω τον Βασίλη Ραξενίδη για τη βοήθεια και συμπαράστασή της στις δύσκολες στιγμές της συγγραφής της εργασίας αυτής.

Το μεγαλύτερο όμως ευχαριστώ απ' όλους το αξίζουν οι γονείς μου, Χρήστος και Δέσποινα. Αν έφτασα σε αυτό το σημείο το χρωστάω σε αυτούς και μόνο. Το λιγότερο που μπορώ να κάνω ως ένδειξη ευχαριστίας, είναι να τους αφιερώσω την παρούσα εργασία, ελπίζοντας ότι κατάφερα να φανώ αντάξια των προσδιοκιών τους.

Αικατερίνη Χ. Τόλιου

Περιεχόμενα

Περίληψη **i**

Abstract **iii**

Ευχαριστίες **v**

Περιεχόμενα **vii**

Κατάλογος Σχημάτων **xi**

1 Εισαγωγή **1**
1.0.1 Ενδιάμεσο λογισμικό . . . 2
1.0.2 Πύλη Διασύνδεσης Βάσεων Δεδομένων με τον Παγόσμιο Ιστό . . . 5
1.1 Αντικείμενο της εργασίας . . . 5
1.2 Υλοποίηση . . . 6
1.3 Εφαρμογές . . . 7
1.4 Δομή της εργασίας . . . 7

2 Πύλες Διασύνδεσης Βάσεων Δεδομένων με τον Παγκόσμιο Ιστό **9**
2.1 Αρχιτεκτονική εξυπηρετούμενου/εξυπηρέτη τριών επιπέδων . . . 9
2.1.1 Σύγκριση αρχιτεκτονικών δύο και τριών επιπέδων . . . 11
2.2 Απαιτήσεις ΠΔΒΔ με τον ΠΙ . . . 12
2.2.1 Απαιτήσεις Αρχιτεκτονικής Υλοποίησης . . . 12
2.2.2 Απαιτήσεις Λειτουργικότητας . . . 13
2.3 Αρχιτεκτονικές Υλοποίησης . . . 13
2.4 Αρχιτεκτονική CGI Executable . . . 15
2.4.1 Περιγραφή της αρχιτεκτονικής . . . 15
2.4.2 DB2WWW . . . 17
2.4.2.1 Περιγραφή της αρχιτεκτονικής . . . 17
2.4.2.2 Δομή ενός αρχείου μακρο-εντολών . . . 18
2.4.3 Illustra's Web DataBlade Module . . . 20
2.4.3.1 Περιγραφή της αρχιτεκτονικής . . . 20
2.5 Αρχιτεκτονική CGI Application Server . . . 21
2.5.1 Περιγραφή της αρχιτεκτονικής . . . 21
2.5.2 O_2Web . . . 23
2.5.2.1 Περιγραφή της αρχιτεκτονικής . . . 23
2.5.2.2 Παράδειγμα . . . 25
2.5.3 Oracle Web Application Server . . . 26

2.5.3.1 Περιγραφή της αρχιτεκτονικής . . . 26
2.5.4 LINCKS . . . 27
2.5.4.1 Περιγραφή της αρχιτεκτονικής . . . 27
2.5.4.2 General Presentation Descriptor . . . 28
2.6 Αρχιτεκτονική Server API . . . 29
2.6.1 Περιγραφή της αρχιτεκτονικής . . . 30
2.6.2 Sybase web.sql . . . 31
2.6.2.1 Περιγραφή της αρχιτεκτονικής . . . 31
2.6.2.2 Παράδειγμα ενός αρχείου HyperText Sybase . . . 32
2.7 Αρχιτεκτονική Proprietary Server . . . 33
2.7.1 Περιγραφή της αρχιτεκτονικής . . . 33
2.8 Αρχιτεκτονική External Viewer . . . 34
2.8.1 Περιγραφή της αρχιτεκτονικής . . . 34
2.9 Αρχιτεκτονική Browser Extension . . . 35
2.9.1 Περιγραφή τυπικής αρχιτεκτονικής . . . 36
2.10 Σύστημα Διαχείρισης Κόμβων του Παγκόσμιου Ιστού . . . 37
2.10.1 Strudel . . . 37
2.10.1.1 Παράδειγμα Γράφου Δεδομένων . . . 39
2.10.1.2 Παράδειγμα ορισμού ενός κόμβου του Παγκόσμιου Ιστού 39
2.10.1.3 Παραγωγή παρουσίασης σε κείμενο HTML . . . 41
2.11 Συμπεράσματα . . . 42

3 Μοντέλο Παρουσίασης Ιστοσελίδων 45
3.1 Παράδειγμα . . . 45
3.2 Το Σύστημα Σημασιολογικού Ευρετηριασμού . . . 46
3.2.1 Η γλώσσα παράστασης γνώσης Telos . . . 46
3.2.1.1 Η δομή της γλώσσας . . . 47
3.2.1.2 Δομικοί περιορισμοί . . . 50
3.2.1.3 Μηχανισμοί Διαχείρισης του ΣΣΕ . . . 50
3.3 Μοντέλο Παρουσίασης . . . 51
3.3.1 Προσδιορισμός της παρουσίασης ενός αντικειμένου της βάσης . . 52
3.3.2 Σημασιολογική δομή . . . 54
3.3.3 Φυσική δομή . . . 55
3.3.4 Λογική δομή . . . 55
3.3.4.1 Σύνδεση σημασιολογικής, φυσικής και λογικής δομής . . 58
3.3.4.2 Η δομή FrameContainer . . . 58
3.3.4.3 Η δομή BasicContainer . . . 61
3.3.4.4 Η δομή BodyElement . . . 63
3.3.4.5 Η δομή τύπος παρουσίασης . . . 64
3.3.4.6 Επέκταση . . . 68
3.3.5 Δεδομένα παρουσιάσεων . . . 72
3.3.5.1 Δεδομένα πολυμέσων . . . 72
3.3.5.2 Δεδομένα αποθηκευμένα στη βάση . . . 73
3.3.5.3 Η κλάση This . . . 77
3.3.6 Προδηλωμένη Παρουσίαση . . . 78
3.4 Παράδειγμα κατασκευής ιστοσελίδας . . . 79
3.5 Παρατηρήσεις - Συμπεράσματα . . . 80

4 Πύλη Διασύνδεσης ΣΣΕ - Παγκόσμιου Ιστού **83**
4.1 Πύλη Διασύνδεσης ΣΣΕ-ΠΙ 83
4.1.1 Εξυπηρέτης εφαρμογής 84
4.1.1.1 Τμήμα επικοινωνίας 85
4.1.1.2 Τμήμα παραγωγής σελίδας HTML 86
4.1.1.3 Τμήμα επερωτήσεων 88
4.2 Σύγκριση με άλλες υλοποιήσεις 89

5 Εμπειρία χρήσης του συστήματος **93**
5.1 Ιστοσελίδες Τομέα Πληροφοριακών Συστημάτων και Τεχνολογίας Λογισμικού 93
5.1.1 Σενάριο παρουσίασης 93
5.1.2 Αποτελέσματα 95
5.2 Σύστημα Διαχείρισης Θησαυρών Όρων 98
5.3 Παρατηρήσεις 100

6 Επίλογος / Συμπεράσματα **103**
6.1 Αποτελέσματα της εργασίας 103
6.2 Μελλοντικές κατευθύνσεις 104
6.2.1 Διαλογική ενημέρωση του μοντέλου παρουσίασης 104
6.2.2 Παρουσίαση σε XML - SGML 105
6.2.3 Υποστήριξη πολύπλοκων επερωτήσεων 105
6.2.4 Βέλτιστος αριθμός ενεργών διεργασιών και καταλαμβανόμενης μνήμης 105
6.2.5 Διανεμητής Αιτήσεων 106
6.2.6 Αύξηση του αριθμού βημάτων ιστορίας 106
6.2.7 Γραφική παρουσίαση των δεδομένων 106
6.2.8 Δημιουργία ενός ΣΔΚΠΙ χρησιμοποιώντας το ΣΣΕ 107

Α Συντομογραφίες **109**

Β Αγγλο-ελληνικό γλωσσάριο **111**

Γ Μοντέλο παρουσίασης ιστοσελίδων σε Telos **113**

Βιβλιογραφία **123**

Κατάλογος Σχημάτων

1.1 Ενδιάμεσο Λογισμικό είναι ένα ενδιάμεσο επίπεδο με εφαρμογές 2

2.1 Η αρχιτεκτονική τριών επιπέδων 10
2.2 Ταξινόμηση των διαφόρων αρχιτεκτονικών 14
2.3 Η αρχιτεκτονική CGI executable 15
2.4 Η αρχιτεκτονική που χρησιμοποιεί το DB2WWW 17
2.5 Η αρχιτεκτονική που ακολουθεί το Illustra 21
2.6 Η αρχιτεκτονική CGI Application Server 22
2.7 Η αρχιτεκτονική που ακολουθεί η O_2. 24
2.8 Η αρχιτεκτονική που ακολουθεί η Oracle 27
2.9 Η αρχιτεκτονική που ακολουθεί το LINCKS 27
2.10 Ένα παράδειγμα του τμήματος STRUCTURE ενός GPD 29
2.11 Η αρχιτεκτονική Server API 30
2.12 Η αρχιτεκτονική που χρησιμοποιεί η Sybase 31
2.13 Η αρχιτεκτονική Proprietary Server 34
2.14 Η αρχιτεκτονική External Viewer 35
2.15 Η αρχιτεκτονική Browser Extension 36
2.16 Η αρχιτεκτονική του συστήματος Strudel 38
2.17 Παράδειγμα γράφου δεδομένων 38
2.18 Παράδειγμα γράφου ιστοκόμβων 40

3.1 Παράδειγμα κόμβων, γνωρίσματος και σχέσεων ταξινόμησης και γενίκευσης 47
3.2 Ένα Ερευνητικό Ίδρυμα αποτελείται από Ινστιτούτα και αυτά από Δραστηριότητες 48
3.3 Έργα, Δημοσιεύσεις και Προσωπικό 49
3.4 Παράδειγμα προσδιορισμού παρουσίασης 53
3.5 Οι κλάσεις αντικειμένων *View* και *Presentation* 54
3.6 Η κλάση *Document* και τα γνωρίσματα που της έχουν αποδοθεί 56
3.7 Μια λογική παράσταση ταξινομείται στην όψη και το είδος διαμόρφωσης κειμένου σύμφωνα με τα οποία θα παραχθεί η παρουσίαση 58
3.8 Οι κλάσεις *FrameContainer*, *FrameSet* και *FrameElement* 59
3.9 Παράδειγμα χρήσης πλαισίων 61
3.10 Η κλάση *BasicContainer* και τα γνωρίσματά της. 62
3.11 Η κλάση *BodyElement* και οι κλάσεις με τις οποίες σχετίζεται 63
3.12 Παράδειγμα χρήσης του *BasicContainer* 64
3.13 Παράδειγμα *Header* και *Footer* 65
3.14 Η κλάση *MediaType* και οι περιπτώσεις της. 66

3.15 Η κλάση *Format* και οι υποκλάσεις της. 67
3.16 Η κλάση *Expansion* και οι υποκλάσεις της. 69
3.17 Η κλάση *MMediaData* και οι υποκλάσιες της 73
3.18 Η κλάση *FactualData* και τα γνωρίσματα που της έχουν αποδοθεί 74
3.19 Η κλάση *FactualData* και οι υποκλάσεις της 75
3.20 Παράδειγμα κυρίου τμήματος μιας σελίδας HTML 76
3.21 Η πρώτη ιστοσελίδα με τη χρήση πλαισίων 79
3.22 Η πρώτη ιστοσελίδα χωρίς πλαίσια 80

4.1 Γενική άποψη της αρχιτεκτονικής που υλοποιήθηκε 84
4.2 Ο εξυπηρέτης εφαρμογής 85
4.3 Ο εξυπηρέτης εφαρμογής υποστηρίζει παρουσίαση σε HTML και XML . . 89

5.1 Παρουσίαση των μελών με χρήση πλαισίων 94
5.2 Παρουσίαση της δραστηριότητας ‘‘Συστήματα Πολιτισμικής Πληροφορικής και Συστημάτων Τεκμηρίωσης’’ με χρήση πλαισίων 95
5.3 Παρουσίαση της δραστηριότητας ‘‘Συστήματα Πολιτισμικής Πληροφορικής και Συστημάτων Τεκμηρίωσης’’ χωρίς πλαίσια 96
5.4 Παράδειγμα αρχείου αναφοράς 99

.

.

Κεφάλαιο 1

Εισαγωγή

Κατά τη διάρκεια των τελευταίων δεκαετιών έχει αυξηθεί εντυπωσιακά ο αριθμός των εφαρμογών που έχουν αναπτυχθεί για την κάλυψη διαφόρων αναγκών. Το βασικό μειονέκτημα των υπαρχουσών εφαρμογών είναι ότι οι περισσότερες έχουν υλοποιηθεί σύμφωνα με την αρχιτεκτονική εξυπηρετούμενου-εξυπηρέτη δύο επιπέδων (two-tier client/server architecture)[1]. Σύμφωνα με αυτήν την αρχιτεκτονική μια εφαρμογή αποτελείται από τρία τμήματα που είναι κατανεμημένα σε δύο επίπεδα. Τα τρία τμήματα είναι η διεπιφάνεια χρήσης της εφαρμογής, η διαχείριση διαδικασιών επεξεργασίας (Process Management) και η διαχείριση των δεδομένων. Η διεπιφάνεια χρήσης του συστήματος βρίσκεται στον εξυπηρετούμενο. Η διαχείριση των δεδομένων βρίσκεται στον εξυπηρέτη ενώ η διαχείριση των διαδικασιών επεξεργασίας διαιρείται σε δύο τμήματα όπου το ένα βρίσκεται στον εξυπηρέτη και το άλλο στον εξυπηρετούμενο. Η τροποποίηση μιας εφαρμογής με την αρχιτεκτονική εξυπηρετούμενου-εξυπηρέτη δύο επιπέδων και η συνεργασία της με άλλες εφαρμογές δεν είναι εύκολη επειδή τα τμήματα που συνιστούν την εφαρμογή δεν είναι ανεξάρτητα μεταξύ τους.

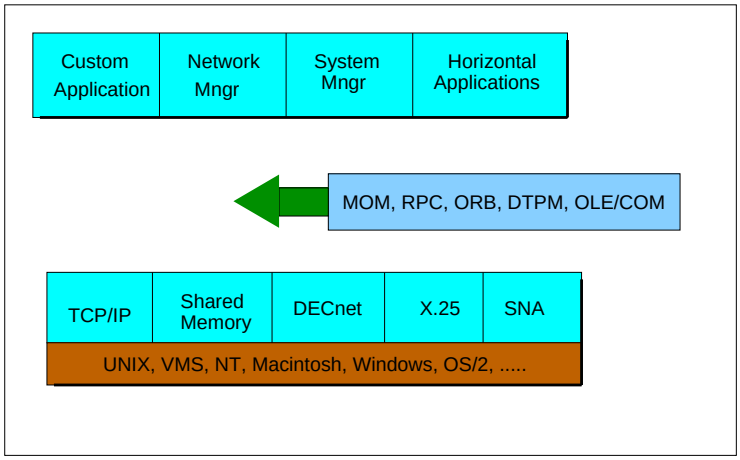
Τα προβλήματα και οι περιορισμοί που επέβαλε η αρχιτεκτονική δύο επιπέδων, οδήγησαν πολύ σύντομα στην αναζήτηση μιας άλλης αρχιτεκτονικής. Έτσι προέκυψε η αρχιτεκτονική τριών ή περισσότερων επιπέδων[2, 3]. Σύμφωνα με αυτήν την αρχιτεκτονική εισάγεται ένα ενδιάμεσο επίπεδο (ή περισσότερα) μεταξύ του εξυπηρέτη και του εξυπηρετούμενου. Με αυτόν τον τρόπο το τμήμα που έχει σχέση με την παράσταση των δεδομένων στο χρήστη παραμένει στην πλευρά του εξυπηρετούμενου αλλά η διαχείριση των διαδικασιών επεξεργασίας μεταφέρεται σε ένα άλλο επίπεδο, στο επίπεδο εφαρμογής. Στο επίπεδο του εξυπηρέτη παραμένει η διαχείριση των δεδομένων. Έτσι η εφαρμογή είναι περισσότερο ευέλικτη σε αλλαγές. Επιπλέον, δεν είναι απαραίτητο τα διάφορα τμήματα να λειτουργούν στον ίδιο υπολογιστή. Χρησιμοποιώντας διαφορετικούς υπολογιστές και διαφορετικά περιβάλλοντα εργασίας, είναι δυνατή η καλύτερη εκμετάλλευση των δυνατοτήτων που αυτά μπορούν να προσφέρουν.

Ουσιαστικά, περνώντας από την αρχιτεκτονική δύο επιπέδων στην αρχιτεκτονική

τριών (ή και περισσότερων) επιπέδων, αυξάνεται ο βαθμός κατανομής μιας εφαρμογής σε τμήματα. Ο βαθμός αυτός εξαρτάται από την πολυπλοκότητα της εφαρμογής. Η σύνδεση των κατανεμημένων τμημάτων δημιουργεί το κατάλληλο υπόβαθρο για την ανάπτυξη συνεργαζόμενων εφαρμογών.

1.0.1 Ενδιάμεσο λογισμικό

Το ενδιάμεσο επίπεδο με τα κατανεμημένα τμήματα, μερικοί το ονομάζουν *Ενδιάμεσο Λογισμικό - ΕΛ* (Middleware). Άλλοι αναφέρονται στην έννοια του ΕΛ ως μια ‘‘ομπρέλα’’ που καλύπτει όλα τα τμήματα μιας κατανεμημένης εφαρμογής. Ακριβής ορισμός για το τί είναι ΕΛ δεν υπάρχει. Στην διαθέσιμη βιβλιογραφία δίνονται διάφοροι ορισμοί για το τί είναι το ΕΛ, ορισμοί που προσπαθούν να περιγράψουν την ίδια έννοια σύμφωνα με διαφορετικές θεωρήσεις [4, 5, 6, 7]. Στο μόνο σημείο που όλοι συμφωνούν είναι η λειτουργία του. Ουσιαστικά το ΕΛ είναι το ενδιάμεσο επίπεδο με τις στοιχιεώδεις λειτουργίες οι οποίες στη συνέχεια θα χρησιμοποιηθούν για τη δημιουργία μιας εφαρμογής, σχήμα 1.1. Αν η αρχιτεκτονική είναι περισσότερων των τριών επιπέδων τότε υπάρχουν και πολλά επίπεδα από ΕΛ, τα οποία συνεργάζονται για την δημιουργία εφαρμογών.



Σχήμα 1.1: Ενδιάμεσο Λογισμικό είναι ένα ενδιάμεσο επίπεδο με εφαρμογές

Ένας απλός ορισμός για το τί είναι ΕΛ[8] είναι ο εξής:

”*Ενδιάμεσο Λογισμικό είναι ένα επίπεδο λογισμικού που βρίσκεται μεταξύ του επιπέδου της εφαρμογής και του επιπέδου δικτύωσης των ετερογενών πρωτοκόλλων και περιβάλλοντων εργασίας. Διαχωρίζει τις εφαρμογές από οποιεσδήποτε εξαρτήσεις από ετερογενή λειτουργικά συστήματα, πλατφόρμες υλικού και πρωτόκολλα επικοινωνίας.*”

Όπως είναι εμφανές και από τον απλό ορισμό που δόθηκε, η έννοια ΕΛ είναι πολύ γενική και μπορεί να εφαρμοστεί σε πολλά διαφορετικά πεδία. Διακρίνονται πέντε

κατηγορίες από ΕΛ ανάλογα με τους μηχανισμούς που χρησιμοποιούν και τις λειτουργίες που επιτελούν. Έτσι ένα ΕΛ μπορεί να χρησιμοποιεί :

- Απομακρυσμένη κλήση διαδικασιών (Remote Procedure Calls - RPC)
- Ανταλλαγή μηνυμάτων (Message Oriented Middlewares)
- Κατανεμημένα αντικείμενα (Distributed Objects)

Επιπλέον, μπορεί να λειτουργεί ως

- Επιτηρητής κατανεμημένων δοσοληψιών (Distributed Transaction Processing (DTP) Monitor)
- Κατανεμητής αντικειμένων (Object Broker)
- Πύλη διασύνδεσης (Gateway)
- ΕΛ για Βάσεις Δεδομένων (Database Middleware)

Οι κατηγορίες αυτές δεν είναι απόλυτες και στην πορεία είναι πολύ πιθανό ότι θα αλλάξουν. Κυρίως θα συγχωνευτούν μερικές καθώς θα αυξάνεται ο αριθμός των προϊόντων που θα ανήκουν σε μια κατηγορία αλλά που θα προσφέρουν υπηρεσίες σύμφωνες με περισσότερες από μια κατηγορίες.

Τα προϊόντα που χρησιμοποιούν *Remote Procedure Calls (RPC)* χρησιμοποιούν ως πολύ βασική μονάδα την συνάρτηση ή τη διαδικασία (function ή procedure). Οι προγραμματιστές εφαρμογής ορίζουν κάποιες συναρτήσεις χρησιμοποιώντας μια γλώσσα ορισμού διεπιφάνειας χρήσης (Interface Definition Language). Η χρησιμοποίηση RPC υπονοεί μια στατική ανάθεση ρόλων στα διάφορα τμήματα από τα οποία αποτελείται μια εφαρμογή. Δεν προσφέρει αρκετή ευελιξία για αλλαγές. Η επικοινωνία μπορεί να είναι σύγχρονη ή ασύγχρονη, δηλαδή όσο ο εξυπηρέτης επεξεργάζεται μια αίτηση, ο εξυπηρετούμενος μπορεί να περιμένει την απάντηση και στο ενδιάμεσο να μην εκτελεί καμία άλλη λειτουργία ή να συνεχίζεται η λειτουργία του εξυπηρετούμενου και όταν ο εξυπηρέτης τελειώσει με την επεξεργασία της αίτησης, συλλέγει τα αποτελέσματα.

Τα προϊόντα που χρησιμοποιούν μηνύματα (*Message Oriented Middlewares (MOM)*) προσφέρουν ένα σύνολο βασικών εντολών που χρησιμοποιούνται για την επικοινωνία μεταξύ των τμημάτων μιας εφαρμογής μέσω του δικτύου. Για τη δομή των μηνυμάτων δεν υπάρχει κάποιο πρότυπο. Τα μηνύματα είναι αλφαριθμητικά που έχουν κάποια σημασία για τις εφαρμογές που τα ανταλλάσουν. Τα προϊόντα αυτά είναι αρκετά ευέλικτα σε αλλαγές και δεν υπονοούν στατική ανάθεση ρόλων. Ο εξυπηρετούμενος μπορεί οποιαδήποτε στιγμή να απευθυνθεί σε κάποιον άλλο εξυπηρέτη για την αποστολή αίτησης. Επιπλέον, υποστηρίζεται σύγχρονη και ασύγχρονη λειτουργία.

Ένας πολύ σημαντικός παράγοντας στην ανάπτυξη ολοκληρωμένων συστημάτων είναι η επαναχρησιμοποίηση έτοιμων εφαρμογών. Οι μηχανισμοί που παρέχονται

από την οντοκεντρική τεχνολογία μπορούν να συμβάλουν σημαντικά προς μια τέτοια προσπάθεια. Ένα ΕΛ που έχει ως κεντρική έννοια το αντικείμενο και όχι τη διαδικασία ονομάζεται *Object Oriented Middleware (OOM)*. Τα τελευταία χρόνια γίνονται πολλές προσπάθειες ορισμού ενός προτύπου και έχουν προταθεί κάποιες αρχιτεκτονικές. Παραδείγματα τέτοιων αρχιτεκτονικών είναι η αρχιτεκτονική CORBA[9], τα προϊόντα της Microsoft OLE/COM/DCOM[10] και το OpenDOC[11] που είναι προϊόν των εταιρειών Apple και IBM.

Ένας επιτηρητής κατανεμημένων δοσοληψιών (*Distributed Transaction Monitor*) είναι ένα ενδιάμεσο περιβάλλον που προσφέρει σημασιολογικές δοσοληψίες σε ένα κατανεμημένο σύστημα. Με τον όρο "σημασιολογική δοσοληψία" εννοούμε ένα σύνολο λειτουργιών που πρέπει να αντιμετωπισθούν από το σύστημα ως μια ατομική λειτουργία. Εκτός από εντολές για την αποστολή και λήψη μηνυμάτων, προσφέρει εντολές για την εκκίνηση και τον τερματισμό δοσοληψίας. Ένα τέτοιο σύστημα είναι απαραίτητο σε κατανεμημένα περιβάλλοντα όπου η σειρά των λειτουργιών και η ακεραιότητα των δεδομένων είναι πάρα πολύ σημαντική.

Μια *Πύλη Διασύνδεσης* (Gateway) είναι ένα πρόγραμμα για την επίτευξη συνεργασίας μεταξύ εφαρμογών. Πρόκειται για ενδιάμεσες εφαρμογές που δέχονται αιτήσεις σύμφωνα με ένα σύνολο εντολών, Διεπιφάνεια Προγραμματισμού Εφαρμογών (Application Programmatic Interface - API), και πρωτόκολλα επικοινωνίας και στέλνουν τις αιτήσεις σε ένα σύνολο εξυπηρετών χρησιμοποιώντας ένα συνήθως διαφορετικό σύνολο εντολών και πρωτοκόλλων επικοινωνίας. Μια πύλη διασύνδεσης αποτελεί ουσιαστικά έναν ενδιάμεσο μεταφραστή μεταξύ του εξυπηρετούμενου και του εξυπηρέτη.

Τα προϊόντα που προσφέρουν πρόσβαση σε βάσεις δεδομένων, *Ενδιάμεσο Λογισμικό για Βάσεις Δεδομένων (ΕΛΒΔ)*, (*Database Middlewares*), χειρίζονται την σύνδεση και εκτελούν τα λογικά βήματα που χρειάζονται για την ανάκτηση και αποθήκευση δεδομένων σε συστήματα βάσεων δεδομένων. Ένα ΕΛΒΔ μπορεί να είναι γενικό, παρέχοντας μια προγραμματιστική διεπιφάνεια χρήσης (API) για επικοινωνία με πολλές βάσεις δεδομένων, ή να είναι ειδικό επιτρέποντας επικοινωνία με συγκεκριμένη βάση δεδομένων. Στην περίπτωση που το ΕΛΒΔ είναι γενικό, μπορεί να χρησιμοποιεί τα πρότυπα ODBC (Open DataBase Connectivity)[12] και JDBC (Java DataBase Connectivity)[13] που έχουν ορισθεί για την επικοινωνία με συστήματα σχεσιακών βάσεων δεδομένων. Μια σημαντική κατηγορία ΕΛΒΔ αποτελούν προϊόντα που υπάγονται και στην προηγούμενη κατηγορία, στις πύλες διασύνδεσης. Πρόκειται δηλαδή, για *Πύλες Διασύνδεσης Βάσεων Δεδομένων (Database Gateways)*. Με τη χρησιμοποίηση πυλών διασύνδεσης βάσεων δεδομένων είναι εύκολα υλοποιήσιμο ένα σύστημα που παρέχει πρόσβαση σε πολλά και ετερογενή συστήματα βάσεων δεδομένων και διαχείρησή τους.

1.0.2 Πύλη Διασύνδεσης Βάσεων Δεδομένων με τον Παγόσμιο Ιστό

Μια *Πύλη Διασύνδεσης Βάσεων Δεδομένων με τον Παγκόσμιο Ιστό* (Web Database Gateway) είναι μια ειδική κατηγορία πυλών διασύνδεσης βάσεων δεδομένων και χρησιμοποιείται για την σύνδεση του Παγκόσμιου Ιστού με συστημάτα βάσεων δεδομένων.

Η σύνδεση μιας βάσης δεδομένων με τον Παγκόσμιο Ιστό έχει ως αποτέλεσμα την κατασκευή μιας εφαρμογής που συνδυάζει όλα τα πλεονεκτήματα του Παγκόσμιου Ιστού και αρκετά από τα πλεονεκτήματα των συστημάτων διαχείρισης βάσεων δεδομένων. Δεν έχει όλα τα πλεονεκτήματα των συστημάτων βάσεων δεδομένων λόγω περιορισμών που επιβάλλουν τα πρότυπα που χρησιμοποιεί ο Παγκόσμιος Ιστός και κυρίως το ότι το πρωτόκολλο HTTP[14] δεν υποστηρίζει καταστάσεις (state-less).

Ένα κύριο ερώτημα είναι το τί περισσότερο μπορεί να προσφέρει ο συνδυασμός αυτών των δύο ευρέως χρησιμοποιούμενων συστημάτων. Έχοντας την πληροφορία αποθηκευμένη σε μία βάση δεδομένων, είναι επιθυμητό να υπάρχει ομοιόμορφη προσπέλαση, δηλαδή ανεξάρτητα του υπολογιστικού συστήματος που χρησιμοποιείται και ανεξάρτητα του αν η προσπέλαση γίνεται από εσωτερικό χρήστη ή όχι, κάτι που μπορεί να προσφέρει η χρησιμοποίηση του Παγκόσμιου Ιστού ως διεπιφάνεια χρήσης. Επιπλέον, η χρησιμοποίηση του Παγκόσμιου Ιστού ως διεπιφάνεια χρήσης για την προσπέλαση πληροφορίας που βρίσκεται αποθηκευμένη σε βάσεις δεδομένων, παρέχει τη δυνατότητα ευκολότερης δημοσιοποίησης αυτής της πληροφορίας στο Διαδίκτυο (Internet).

Από την πλευρά του Παγκόσμιου Ιστού, η χρησιμοποίηση ενός συστήματος διαχείρισης βάσεων δεδομένων για την αποθήκευση των δεδομένων προσφέρει ευελιξία και ευκολία στην ενημέρωση των δεδομένων, τη διατήρηση της πληροφορίας λόγω της λογικής και φυσικής ανεξαρτησίας των δεδομένων και τη διασφάλιση της εγκυρότητας των παρεχόμενων υπηρεσιών. Το τελευταίο είναι εφικτό λόγω της δυνατότητας δυναμικής παραγωγής των σελίδων HTML[15] (HyperText Markup Language) με τη χρήση δηλωτικών γλωσσών επερωτήσεων (query languages).

1.1 Αντικείμενο της εργασίας

Σκοπός της παρούσας εργασίας είναι η κατασκευή μιας πύλης διασύνδεσης μεταξύ του *Συστήματος Σημασιολογικού Ευρετηριασμού - ΣΣΕ* (Semantic Index System - SIS)([16]) και του Παγκόσμιου Ιστού. Το ΣΣΕ είναι ένα οντοκεντρικό σύστημα για τη διαχείριση μεγάλου όγκου αλληλένδετων δεδομένων, εννοιών και πολύπλοκων σχέσεων. Πυρήνας του ΣΣΕ είναι η Telos([17]), μία γλώσσα παράστασης γνώσης που παρέχει τους εξής βασικούς μηχανισμούς δόμησης : ταξινόμηση, γενίκευση και απόδοση γνωρίσματος.

Ποιό συγκεκριμένα, το πρόβλημα που θα μας απασχολήσει είναι ο σχεδιασμός και η ανάπτυξη ενός εργαλείου για την πλοήγηση σε οποιοδήποτε σημασιολογικό δίκτυο

που έχει κατασκευαστεί με τη χρήση της Telos. Ως διεπιφάνεια χρήσης του εργαλείου θα μπορεί να χρησιμοποιηθεί οποιοσδήποτε γραφικός αναδιφητής του Παγκόσμιου Ιστού (Web Browser). Το σύστημα πρέπει να είναι :

- Ευέλικτο
  Με τον ευελιξία ενοούμε οτι πρέπει να υποστηρίζεται :
  - ▷ η δυνατότητα προσδιορισμού παρουσίασης σε κάθε αντικείμενο της βάσης ανεξάρτητα από τον τύπο του και από το επίπεδο ταξινόμησης στο οποίο βρίσκεται
  - ▷ η δυνατότητα παρουσίασης σε HTML διαφορετικών όψεων του ίδιου αντικειμένου
- Επεκτάσιμο
  Πρέπει να υποστηρίζεται η δυνατότητα παρουσίασης του ίδιου αντικειμένου χρησιμοποιώντας διαφορετικά είδη διαμόρφωσης κειμένου
- Δηλωτικό
  Ο προγραμματιστής εφαρμογής πρέπει να μπορεί να προσδιορίζει με δηλωτικό τρόπο την παρουσίαση των αντικειμένων και όχι χρησιμοποιώντας κάποια γλώσσα προγραμματισμού ή την HTML.

Με βάση τις παραπάνω απαιτήσεις κατασκευάστηκε ένα μοντέλο για την παράσταση των αντικειμένων της Telos σε κείμενο σύμφωνο με το πρότυπο HTML. Όποια πληροφορία είναι απαραίτητη για τη δυναμική παραγωγή του κειμένου, αποθηκεύεται χρησιμοποιώντας αυτό το μοντέλο. Το μοντέλο αποτελεί τμήμα του σημασιολογικού δικτύου και η σχέση του με τα αντικείμενα που συνιστούν το σημασιολογικό δίκτυο είναι άμεση, όπως θα φανεί και στη περιγραφή του μοντέλου. Ο προγραμματιστής εφαρμογής συμπληρώνει την αντίστοιχη πληροφορία.

Η χρήση του μοντέλου προσφέρει αρκετή ευελιξία όσον αφορά τον δυναμικό ορισμό της παράστασης των αντικειμένων της Telos σε κείμενο σύμφωνο με το πρότυπο HTML. Για την υποστήριξη μελλοντικής επαύξησης της λειτουργικότητας, έχει σχεδιαστεί έτσι το μοντέλο, ώστε η πληροφορία που υπάρχει να μην συσχετίζει άμεσα την παράσταση των αντικειμένων σε κείμενο σύμφωνο με το πρότυπο HTML αλλά έμμεσα. Δηλαδή, η ίδια ακριβώς πληροφορία μπορεί να χρησιμοποιηθεί για την παράσταση των αντικειμένων χρησιμοποιώντας κάποιο άλλο μορφότυπο διαμόρφωσης κειμένου, όπως XML[18], SGML[19] ή ASCII.

1.2 Υλοποίηση

Για την υλοποίηση του συστήματος χρησιμοποιήθηκε η γλώσσα προγραμματισμού C και η ανάπτυξη έγινε σε λειτουργικό σύστημα Solaris 5.6. Για την επικοινωνία με

το ΣΣΕ χρησιμοποιήθηκε η διεπιφάνεια προγραμματισμού εφαρμογών (API) που αυτό προσφέρει. Για την εξυπηρέτηση μιας αίτησης του τελικού χρήστη χρησιμοποιείται το πρωτόκολλο HTTP μέχρι η αίτηση να φτάσει σε μια ενδιάμεση διεργασία, η οποία έχει υλοποιηθεί σύμφωνα με το πρότυπο CGI, ενώ η επικοινωνία με τον εξυπηρέτη εφαρμογής και τον εξυπηρέτη του ΣΣΕ γίνεται με ανταλλαγή μηνυμάτων μέσω sockets και το πρωτόκολλο που χρησιμοποιείται είναι το TCP/IP. Για τον εξυπηρέτη του Παγκόσμιου Ιστού χρησιμοποιήθηκε το προϊόν Apache[20].

Το εργαλείο υποστηρίζει την HTML 3.2[21]. Για παρουσιάσεις που δεν καλύπτονται από την HTML 3.2 παρέχεται η δυνατότητα χρησιμοποίησης Java applets και scripts.

1.3 Εφαρμογές

Το σύστημα χρησιμοποιήθηκε για την παραγωγή σελίδων HTML του Τομέα Πληροφοριακών Συστημάτων και Τεχνολογίας Λογισμικού του Ινστιτούτου Πληροφορικής (ΙΠ) του Ιδρύματος Έρευνας και Τεχνολογίας (ΙΤΕ). Από αυτήν τη χρήση προέκυψε ότι το σύστημα είναι αρκετά ευέλικτο και το παραγόμενο αποτέλεσμα κρίνεται ικανοποιητικό.

Το σύστημα επίσης χρησιμοποιήθηκε στην παραγωγή απλού κειμένου για το Σύστημα Διαχείρισης Θησαυρών Όρων - ΣΔΘΟ (Thesaurus Management System - TMS), το οποίο έχει αναπτυχθεί από τον Τομέα Πληροφοριακών Συστημάτων και Τεχνολογίας Λογισμικού του ΙΠ-ΙΤΕ.

Από τη χρήση του συστήματος υπάρχουν ενδείξεις ότι η συντήρηση διαφορετικών όψεων παρουσίασης της ίδιας πληροφορίας είναι εύκολη διαδικασία.

1.4 Δομή της εργασίας

Στο δεύτερο κεφάλαιο γίνεται μια αναφορά στις υπάρχουσες αρχιτεκτονικές υλοποίησης μιας πύλης διασύνδεσης βάσεων δεδομένων και του Παγκόσμιου Ιστού. Για την κάθε αρχιτεκτονική περιγράφεται ο τρόπος λειτουργίας, τα πλεονεκτήματα και τα μειονεκτήματά της καθώς και μερικές αντιπροσωπευτικές υλοποιήσεις.

Στο τρίτο κεφάλαιο περιγράφεται το μοντέλο παράστασης που χρησιμοποιήθηκε στην παρούσα εργασία. Χρησιμοποιούνται παραδείγματα για την καλύτερη κατανόηση του μοντέλου.

Στο τέταρτο κεφάλαιο περιγράφεται η αρχιτεκτονική υλοποίησης που υιοθετήθηκε, η υλοποίηση, καθώς και μια σύγκριση με συστήματα που περιγράφηκαν στο δεύτερο κεφάλαιο.

Στο πέμπτο κεφάλαιο, γίνονται κάποιες παρατηρήσεις που προέκυψαν από τη χρήση του συστήματος ενώ στο έκτο και τελευταίο κεφάλαιο γίνεται μια αναφορά σε

μελλοντικές επεκτάσεις του συστήματος.

Κεφάλαιο 2

Πύλες Διασύνδεσης Βάσεων Δεδομένων με τον Παγκόσμιο Ιστό

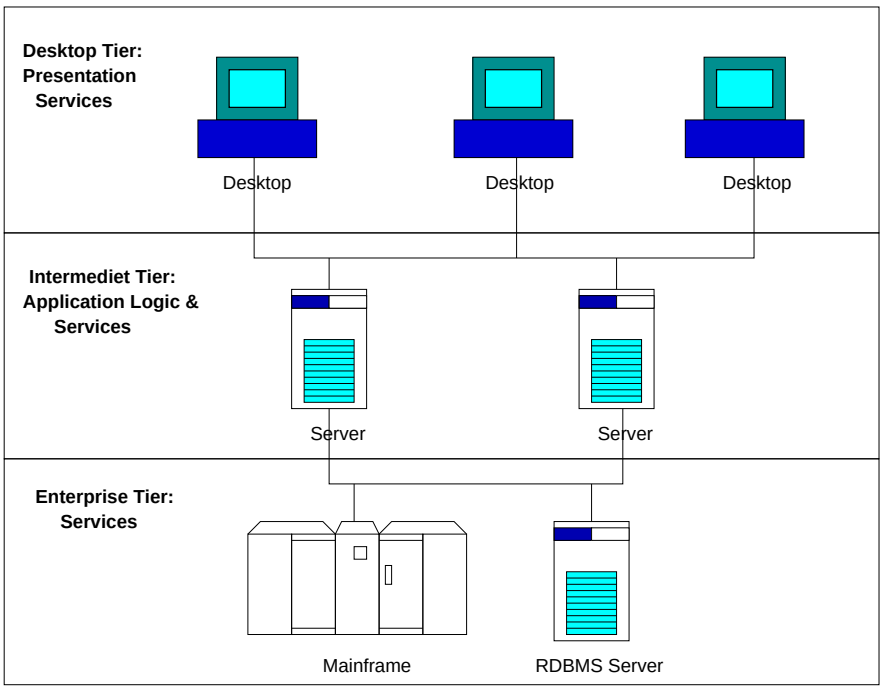
Στο κεφάλαιο αυτό εξετάζουμε αρχικά την αρχιτεκτονική εξυπηρετούμενου/εξυπηρέτη τριών επιπέδων. Μια σύγκριση των αρχιτεκτονικών δύο και τριών επιπέδων κρίνεται απαραίτητη. Επιπλέον, εξετάζουμε τις απαιτήσεις που υπάρχουν από μια πύλη διασύνδεσης βάσεων δεδομένων (ΠΔΒΔ) με τον Παγκόσμιο Ιστό (ΠΙ) και στη συνέχεια τις αρχιτεκτονικές σύμφωνα με τις οποίες μπορεί να υλοποιηθεί. Για κάθε αρχιτεκτονική εξετάζουμε τον τρόπο λειτουργίας, τα πλεονεκτήματα και μειονεκτήματά της και στη συνέχεια περιγράφονται μερικά προϊόντα που αποτελούν αντιπροσωπευτικές υλοποιήσεις της αρχιτεκτονικής. Για το κάθε προϊόν εξετάζεται ο τρόπος υλοποίησης της αρχιτεκτονικής που ακολουθεί και ο τρόπος παρουσίασης των δεδομένων της βάσης σε κείμενο σύμφωνο με το πρότυπο HTML. Τέλος, γίνεται μια σύντομη αναφορά στα συστήματα διαχείρισης κόμβων του Παγκόσμιου Ιστού και περιγράφουμε ένα τέτοιο σύστημα.

2.1 Αρχιτεκτονική εξυπηρετούμενου/εξυπηρέτη τριών επιπέδων

Η αρχιτεκτονική τριών επιπέδων επεκτείνει την αρχιτεκτονική δύο επιπέδων προσθέτοντας ένα ενδιάμεσο επίπεδο. Αυτό το ενδιάμεσο επίπεδο περιλαμβάνει ενδιάμεσους εξυπηρέτες για την υποστήριξη της λογικής λειτουργίας των εφαρμογών και κατανεμημένων λειτουργιών. Το ενδιάμεσο επίπεδο είναι πολύ σημαντικό να προσφέρει διαφάνεια όσον αφορά την κατανομή των πόρων του συστήματος. Επιπλέον, πρέπει να προσφέρει ένα περιβάλλον αξιόπιστο, ασφαλές και αποδοτικό για την υποστήριξη κατανεμημένων εφαρμογών.

Η αρχιτεκτονική τριών επιπέδων έχει δύο διαστάσεις: αρχιτεκτονική εφαρμογής και αρχιτεκτονική ανάπτυξης [22]. Η αρχιτεκτονική εφαρμογής αναφέρεται στα λογικά

τμήματα από τα οποία αποτελείται μια εφαρμογή. Αυτά είναι το τμήμα παρουσίασης, το τμήμα διαχείρισης των διαδικασιών επεξεργασίας και το τμήμα διαχείρισης των δεδομένων. Τα τμήματα αυτά συνιστούν τη λογική παράσταση της εφαρμογής. Η αρχιτεκτονική ανάπτυξης αναφέρεται στη φυσική σχέση μεταξύ των υπολογιστών που χρησιμοποιεί η εφαρμογή. Τα τρία επίπεδα είναι ο υπολογιστής που χρησιμοποιεί ο χρήστης, ο υπολογιστής στον οποίο εκτελείται ο ενδιάμεσος εξυπηρέτης, και ο υπολογιστής στον οποίο εκτελείται ο εξυπηρέτης που χειρίζεται τα δεδομένα (σχήμα 2.1). Ο υπολογιστής τον οποίο χρησιμοποιεί ο χρήστης αντιστοιχεί στο τμήμα που διαχειρίζεται την παρουσίαση των δεδομένων. Ο υπολογιστής στον οποίο εκτελείται ο ενδιάμεσος εξυπηρέτης αντιστοιχεί στο τμήμα διαχείρισης των διαδικασιών επεξεργασίας. Τέλος, ο υπολογιστής στον οποίο εκτελείται ο εξυπηρέτης αντιστοιχεί στο τμήμα που διαχειρίζεται τα δεδομένα. Επομένως, το φυσικό περιβάλλον καθρεπτίζει την λογική λειτουργία της εφαρμογής.



Σχήμα 2.1: Η αρχιτεκτονική τριών επιπέδων

Το βασικό στοιχείο για την ανάπτυξη συστημάτων σύμφωνα με την αρχιτεκτονική τριών επιπέδων είναι η εξασφάλιση ότι το κάθε τμήμα της εφαρμογής είναι ένα ξεχωριστό, ανεξάρτητο τμήμα μέσα στα πλαίσια της εφαρμογής. Ένας άλλος τρόπος έκφρασης του παραπάνω είναι ότι οι εφαρμογές που είναι σύμφωνες με την αρχιτεκτονική τριών επιπέδων διακρίνονται στα τμήματα παράστασης, λογικής λειτουργίας και δεδομένων. Ο διαχωρισμός των εφαρμογών σε ανεξάρτητα τμήματα δεν σημαίνει ότι πρόκειται για διαφορετικές διεργασίες. Μπορεί μέσα στα πλαίσια μιας διεργασίας να συνυπάρχουν

περισσότερα του ενός τμήματα. Η ανεξαρτησία των τμημάτων σημαίνει ότι το κάθε τμήμα μπορεί να αλλάξει ή να αντικατασταθεί χωρίς να επηρεάζει τα υπόλοιπα, όχι ότι πρόκειται για ανεξάρτητες διεργασίες.

Ένα άλλο βασικό στοιχείο της αρχιτεκτονικής τριών επιπέδων είναι ότι μπορεί να γίνει εκμετάλευση των δυνατοτήτων που προσφέρουν τα διάφορα περιβάλλοντα εργασίας ώστε το κάθε τμήμα από τα οποία αποτελείται μια εφαρμογή να εκτελείται στο βέλτιστο περιβάλλον εργασίας.

2.1.1 Σύγκριση αρχιτεκτονικών δύο και τριών επιπέδων

Η ανάπτυξη εφαρμογών σύμφωνα με την αρχιτεκτονική τριών επιπέδων προσφέρει κάποια πλεονεκτήματα σε σχέση με τη χρήση αρχιτεκτονικής δύο επιπέδων. Τα πλεονεκτήματα προέρχονται κυρίως από το γεγονός ότι υπάρχει ένας σαφής διαχωρισμός ρόλων και αρμοδιοτήτων καθώς και από το ότι στο ενδιάμεσο επίπεδο μπορούν να συνδυαστούν πολλές εφαρμογές για την δημιουργία της ζητούμενης.

Με βάση τα δύο αυτά βασικά στοιχεία είναι κατανοητό ότι η αρχιτεκτονική τριών (ή και περισσότερων) επιπέδων προσφέρει τη δυνατότητα επαναχρησιμοποίησης εφαρμογών. Η επαναχρησιμοποίηση γίνεται ως προς δύο κατευθύνσεις :

- Επαναχρησιμοποίηση υπαρχουσών εφαρμογών.
- Χρησιμοποίηση των ενδιάμεσων εφαρμογών σε περισσότερα από ένα συστήματα.

Υπάρχουν πάρα πολλές εφαρμογές οι οποίες χρησιμοποιούνται για την εξυπηρέτηση συγκεκριμένων λειτουργιών στα πλαίσια μιας επιχείρησης. Με τη χρησιμοποίηση της αρχιτεκτονικής τριών επιπέδων οι υπάρχουσες εφαρμογές μπορούν να χρησιμοποιηθούν για την κατασκευή μιας νέας ολοκληρωμένης εφαρμογής. Με αυτόν τον τρόπο δεν απορρίπτονται παλιά συστήματα στα οποία μια επιχείρηση έχει επενδύσει, αλλά χρησιμοποιούνται με τον καλύτερο δυνατό τρόπο.

Επιπλέον, επειδή τα διάφορα τμήματα που θα χρησιμοποιηθούν για την τελική εφαρμογή είναι ανεξάρτητα και επικοινωνούν με κάποια διεπιφάνεια ζεύξης, τα ίδια τμήματα που βρίσκονται στο ενδιάμεσο επίπεδο μπορούν να χρησιμοποιηθούν από πολλές εφαρμογές χωρίς να χρειάζεται η επανάληψή τους. Αυτό προσφέρει μια πολύ μεγάλη ευελιξία στην ανάπτυξη εφαρμογών και τη συντήρησή τους.

Η ανεξαρτησία των διαφόρων τμημάτων που συνιστούν μια εφαρμογή, προσφέρει επιπλέον τη δυνατότητα ανάπτυξης των διαφόρων τμημάτων στο βέλτιστο περιβάλλον εργασίας. Με τον όρο ‘‘βέλτιστο περιβάλλον εργασίας’’ εννοούμε ότι μπορεί να χρησιμοποιηθεί με βέλτιστο τρόπο το φυσικό περιβάλλον για την ανάπτυξη και τη λειτουργία των εφαρμογών.

Η ύπαρξη των ενδιάμεσων εξυπηρετών και η ανεξαρτησία των τμημάτων που συνιστούν μια εφαρμογή, διευκολύνει στην υποστήριξη πολύπλοκων λειτουργιών που

απαιτούν την συνεργασία πολλών εξειδικευμένων εφαρμογών. Με αυτόν τον τρόπο μπορεί να υποστηριχθεί η ύπαρξη ενός διαχειριστή σημασιολογικών δοσοληψιών. Ένας παρατηρητής διαχειρίζεται όλες τις αιτήσεις των εξυπηρετούμενων και τις κατευθύνει στον καταλληλότερο εξυπηρέτη που μπορεί να εξυπηρετήσει την κάθε αίτηση βάσει διαφόρων κριτηρίων, όπως η επίτευξη εξισορρόπησης φόρτου εργασίας στους εξυπηρέτες. Επιπλέον, με την ύπαρξη ενός παρατηρητή σημασιολογικών δοσοληψιών μπορεί να υποστηριχθεί η απόδοση προτεραιοτήτων στις αιτήσεις ώστε αιτήσεις με μεγάλη προτεραιότητα να προηγούνται στην εκτέλεσή τους. Τέλος, ένας παρατηρητής σημασιολογικών δοσοληψιών μπορεί να συνδυαστεί με τη χρήση πυλών διασύνδεσης επικοινωνίας (communication gateways) και βάσεων δεδομένων (database gateways). Όλες οι παραπάνω λειτουργίες μπορούν και πρέπει να επιτυγχάνονται με τρόπο διαφανή προς το χρήστη του συστήματος.

Η αρχιτεκτονική τριών (ή περισσότερων) επιπέδων είναι ιδανική για την υποστήριξη κατανεμημένων εφαρμογών με τρόπο διάφανο προς το χρήστη του συστήματος. Αν ληφθεί υπόψη ότι ένας βασικός στόχος στην ανάπτυξη κατανεμημένων εφαρμογών είναι η διαφάνεια προς το χρήστη, είναι κατανοητό πόσο σημαντική προσφορά είναι η αρχιτεκτονική περισσοτέρων των δύο επιπέδων.

Όπως έγινε κατανοητό και από την περιγραφή των πλεονεκτημάτων που προσφέρει η υιοθέτηση της αρχιτεκτονικής τριών (ή περισσότερων) επιπέδων, το βασικό στοιχείο είναι η ύπαρξη των ενδιάμεσων εξυπηρετών και η τμηματοποίηση της εφαρμογής σε ανεξάρτητα τμήματα. Αυτή είναι και η βασική διαφορά από την αρχιτεκτονική δύο επιπέδων όπου η τμηματοποίηση της εφαρμογής δεν είναι σε τελείως ανεξάρτητα τμήματα.

2.2 Απαιτήσεις ΠΔΒΔ με τον ΠΙ

Από μια πύλη διασύνδεσης βάσεων δεδομένων με τον Παγκόσμιο Ιστό υπάρχουν συγκεκριμένες απαιτήσεις όσον αφορά την αρχιτεκτονική υλοποίησης και την υποστηριζόμενη λειτουργικότητα [23].

2.2.1 Απαιτήσεις Αρχιτεκτονικής Υλοποίησης

Οι απαιτήσεις της αρχιτεκτονικής υλοποίησης αφορούν σε κάποια γενικά κριτήρια που πρέπει να ικανοποιούνται από κάθε εργαλείο. Τα κριτήρια που αναφέρονται δεν αποτελούν μια αναλυτική καταγραφή όλων των κριτηρίων αλλά είναι ενδεικτικά.

Έτσι, απαιτείται :

- Η υψηλή απόδοση,
  δηλαδή να είναι αρκετά γρήγορη,

- Το κόστος ανάπτυξης να είναι το ελάχιστο δυνατό,
  δηλαδή όσο το δυνατόν λιγότερη οικονομική και χρονική επιβάρυνση χρησιμοποιώντας κοινόχρηστες βιβλιοθήκες και κοινά μοντέλα που έχουν υιοθετηθεί από διεθνείς οργανισμούς.

- Η αρχιτεκτονική να είναι κλιμακούμενη (scaleable) και ανοιχτή,
  δηλαδή να είναι απλή, επεκτάσιμη, αποτελούμενη από τμήματα και βασισμένη σε διεθνή πρότυπα.

- Πρέπει να χρησιμοποιούνται πρότυπες διεπιφάνειες χρήσης (standard interfaces) μεταξύ των τμημάτων απο τα οποία αποτελείται και η οποιαδήποτε ανταλλαγή δεδομένων να γίνεται χρησιμοποιώντας πρότυπους μορφότυπους δεδομένων (standard data formats).

2.2.2 Απαιτήσεις Λειτουργικότητας

Οι απαιτήσεις λειτουργικότητας αφορούν στην λειτουργικότητα που πρέπει να υποστηρίζουν τα διάφορα τμήματα από τα οποία αποτελείται μια πύλη διασύνδεσης βάσεων δεδομένων με τον παγκόσμιο ιστό.

Ενδεικτικά αναφέρονται :

- Γρήγορη απόκριση του συστήματος
  Σε ένα διαλογικό σύστημα είναι πολύ σημαντικό η αντίδραση (reaction) του συστήματος να είναι όσο το δυνατόν ταχύτερη.

- Υποστήριξη πολυμέσων
  Το παραγόμενο κείμενο, που είναι σύμφωνο με το πρότυπο HTML, να περιέχει δεδομένα πολυμέσων τα οποία υποστηρίζει ο Παγκόσμιος Ιστός, όπως για παράδειγμα εικόνες, ήχος και βίντεο.

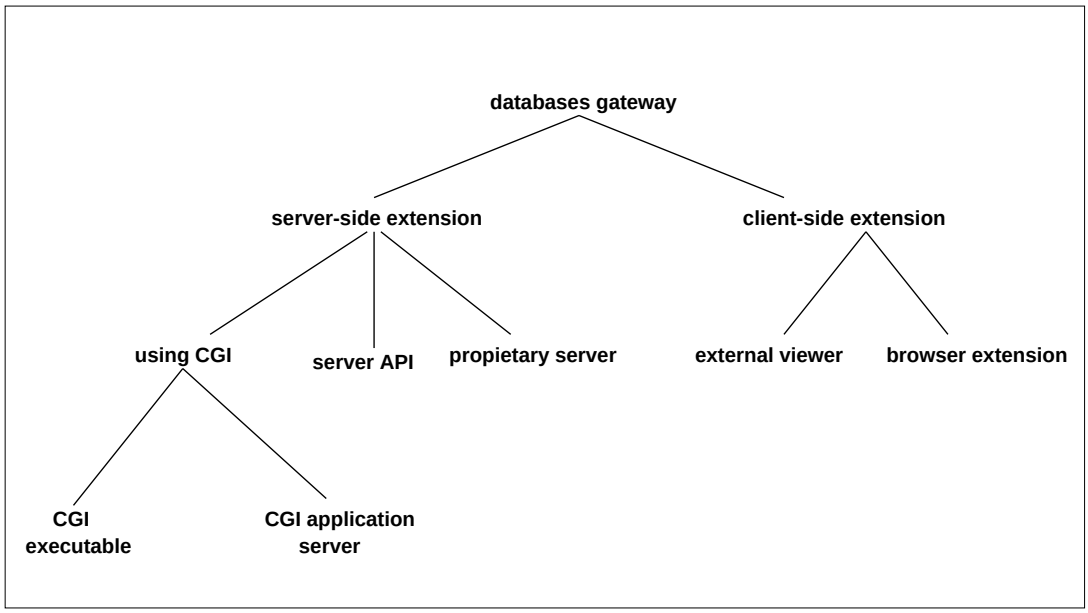
- Υποστήριξη δοσοληψιών
  Η έννοια της δοσοληψίας είναι βασική στα συστήματα βάσεων δεδομένων. Ο Παγκόσμιος Ιστός δεν έχει τη δυνατότητα υποστήριξης δοσοληψιών. Επομένως, πρέπει η πύλη διασύνδεσης να υποστηρίζει με κάποιο τρόπο αυτή τη βασική έννοια.

- Υποστήριξη μόνιμων συνδέσεων με τη βάση,
  δηλαδή, αποφυγή της επαναρχικοποίησης των συνδέσεων σε κάθε επερώτηση.

2.3 Αρχιτεκτονικές Υλοποίησης

Οι πύλες διασύνδεσης βάσεων δεδομένων με τον Παγκόσμιο Ιστό μπορούν να ταξινομηθούν σε διάφορες κατηγορίες ανάλογα με την αρχιτεκτονική ολοκλήρωσης που

χρησιμοποιούν. Για την ταξινόμηση των αρχιτεκτονικών υλοποίησης[24] το κριτήριο που χρησιμοποιείται είναι το ποιό τμήμα της εφαρμογής επικοινωνεί με το σύστημα της βάσης δεδομένων. Η ταξινόμηση φαίνεται στο σχήμα 2.2.



Σχήμα 2.2: Ταξινόμηση των διαφόρων αρχιτεκτονικών

Όπως διακρίνεται και στο σχήμα, οι πύλες διασύνδεσης διακρίνονται σε δύο μεγάλες κατηγορίες αυτές που ‘‘προστίθενται’’ στην πλευρά του εξυπηρέτη του Παγκόσμιου Ιστού (Web Server) και αυτές που ‘‘προστίθενται’’ στην πλευρά του εξυπηρετούμενου του Παγκόσμιου Ιστού (Web Client). Από αυτές που προστίθενται στην πλευρά του εξυπηρέτη, η πύλη διασύνδεσης υλοποιείται χρησιμοποιώντας το πρωτόκολλο CGI (Common Gateway Interface)[25], τη διεπιφάνεια προγραμματισμού εφαρμογών (API) του εξυπηρέτη του Παγκόσμιου Ιστού ή αποτελεί τμήμα του εξυπηρέτη. Από αυτές που προστίθενται στην πλευρά του εξυπηρετούμενου του Παγκόσμιου Ιστού, η πύλη διασύνδεσης μπορεί να χρησιμοποιηθεί ως εξωτερική εφαρμογή ή να συμπεριληφθεί στον αναδιφητή του Παγκόσμιου Ιστού (Web Browser). Η χρησιμοποίηση του πρωτοκόλλου CGI μπορεί να γίνει είτε υλοποιώντας τη πύλη διασύνδεσης ως ένα εκτελέσιμο cgi είτε σε συνδυασμό με κάποιον εξυπηρέτη εφαρμογής (application server) στον οποίο το εκτελέσιμο cgi μεταφέρει τις αιτήσεις.

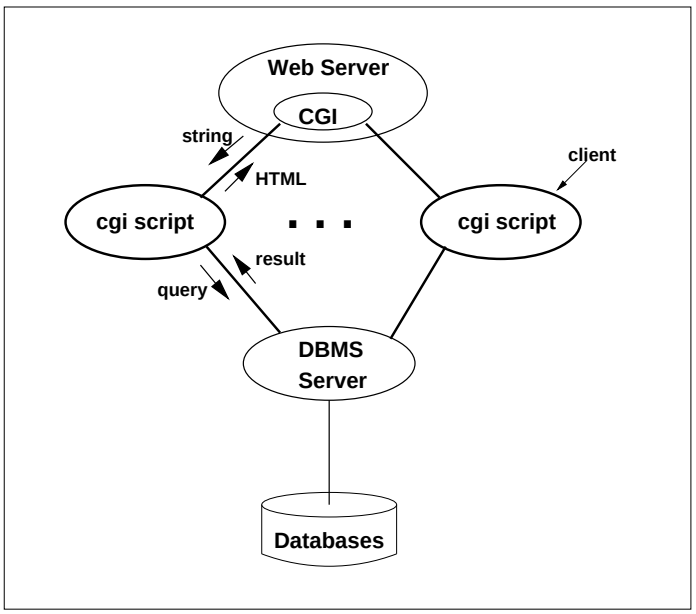
Στη συνέχεια αναλύεται κάθε αρχιτεκτονική και περιγράφονται μερικές αντιπροσωπευτικές υλοποιήσεις για καθεμιά.

2.4 Αρχιτεκτονική CGI Executable

Το CGI είναι ένα πρωτόκολλο επικοινωνίας προγραμμάτων με έναν εξυπηρέτη του Παγκόσμιου Ιστού. Όλοι οι γνωστοί εξυπηρέτες του Παγκόσμιου Ιστού εμπορικοί ή μη υποστηρίζουν αυτό το πρωτόκολλο. Ένα εκτελέσιμο πρόγραμμα σύμφωνο με το πρωτόκολλο CGI ονομάζεται *εκτελέσιμο cgi.* Ο προσδιορισμός του γίνεται μέσω του μηχανισμού URL/URI (Uniform Resource Locator/Identifier)[26].

2.4.1 Περιγραφή της αρχιτεκτονικής

Η αρχιτεκτονική CGI Executable παρουσιάζεται στο σχήμα 2.3. Σύμφωνα με



Σχήμα 2.3: Η αρχιτεκτονική CGI executable

αυτή την αρχιτεκτονική, ο εξυπηρέτης του Παγκόσμιου Ιστού (Web Server) μεταφέρει μία αίτηση του χρήστη προς τη πύλη διασύνδεσης μέσω του πρωτοκόλλου CGI. Η διευθυνσιοδότηση γίνεται χρησιμοποιώντας το πρότυπο URL/URI. Η πύλη διασύνδεσης επεξεργάζεται την αίτηση και επικοινωνεί με τον εξυπηρέτη της βάσης δεδομένων (Database Server), δηλαδή είναι ένας εξυπηρετούμενος του εξυπηρέτη της βάσης όπως φαίνεται και στο σχήμα 2.3, για την αποστολή της επερώτησης και τη συλλογή των αποτελεσμάτων. Στη συνέχεια μετατρέπει τα αποτελέσματα της επερώτησης σε κείμενο σύμφωνο με το πρότυπο HTML, το οποίο στέλνει στον χρήστη μέσω του εξυπηρέτη του Παγκόσμιου Ιστού.

Η πύλη διασύνδεσης μπορεί να υλοποιηθεί χρησιμοποιώντας μία πληθώρα γλωσσών προγραμματισμού σε συνδυασμό με ένα περιβάλλον ανάπτυξης εφαρμογών βάσεων δεδομένων, όπως ESQL/C ή C/C++ API.

Όταν ένα αντικείμενο που ζητείται από έναν εξυπηρέτη του Παγκόσμιου Ιστού είναι ένα εκτελέσιμο cgi, ο εξυπηρέτης του Παγκόσμιου Ιστού δημιουργεί ένα αντίγραφο του εκτελέσιμου κώδικα στη μνήμη στο οποίο περνάει ως όρισμα ένα τμήμα του URL και τον έλεγχο της εκτέλεσης. Τα ορίσματα κλήσης του εκτελέσιμου cgi συνήθως περιέχουν πληροφορίες σχετικά με τις επερωτήσεις που πρέπει να αποσταλούν στον εξυπηρέτη της βάσης. Στη συνέχεια, το εκτελέσιμο cgi συνδέεται με τον εξυπηρέτη της βάσης δεδομένων και στέλνει την επερώτηση. Μόλις, συλλέξει τα αποτελέσματα της επερώτησης αποσυνδέεται από τον εξυπηρέτη της βάσης. Με αυτόν τον τρόπο δεν γίνεται εκμετάλλευση των τεχνικών βελτιστοποίησης που μπορεί να προσφέρει ένα σύστημα διαχείρισης βάσεων δεδομένων (αρχικοποίηση ενταμιευτών (buffers)) που αποφέρουν σημαντική χρονική επιβάρυνση αρχικά, η οποία εξισορροπείται στη συνέχεια με πολύ καλύτερο συνολικό χρόνο απόκρισης του συστήματος. Μόλις τελειώσει την επεξεργασία των αποτελεσμάτων και την αποστολή της σελίδας HTML στον εξυπηρέτη του Παγκόσμιου Ιστού, το εκτελέσιμο cgi παύει να υφίσταται. Στην επόμενη αίτηση του χρήστη θα δημιουργηθεί ένα άλλο αντίγραφο του εκτελέσιμου cgi από τον εξυπηρέτη του Παγκόσμιου Ιστού.

Το πλεονέκτημα αυτής της αρχιτεκτονικής είναι η απλότητά της. Η πύλη διασύνδεσης είναι μια ξεχωριστή διεργασία που εκτελείται στον ίδιο υπολογιστή με τον εξυπηρέτη του Παγκόσμιου Ιστού. Δεν επηρεάζεται η λειτουργία του εξυπηρέτη της βάσης ή του εξυπηρέτη του Παγκόσμιου Ιστού και δεν απαιτείται καμία αλλαγή των υπαρχουσών τεχνολογιών όπως URL/URI, HTML, HTTP και CGI. Επιπλέον, επειδή η πύλη διασύνδεσης είναι ανεξάρτητη διεργασία και δεν μοιράζεται πόρους με τον εξυπηρέτη του Παγκόσμιου Ιστού, όπως χώρο διευθύνσεων και προτιμήσεις χρηστών, είναι πολύ εύκολη η επέκταση της λειτουργικότητάς της χωρίς να απαιτούνται αλλαγές στον εξυπηρέτη του Παγκόσμιου Ιστού.

Τα μειονεκτήματά της είναι :

- Επειδή το εκτελέσιμο cgi συνδέεται με τον εξυπηρέτη της βάσης και κάνει όλη την επεξεργασία, είναι μεγάλο σε μέγεθος. Αυτό σημαίνει ότι το σύστημα δεν μπορεί να εξυπηρετήσει μεγάλο αριθμό ταυτόχρονων αιτήσεων, κατάσταση επιθυμητή όταν πολλοί χρήστες χρησιμοποιούν το σύστημα, επειδή δεν μπορούν να δημιουργηθούν πολλά αντίγραφα του εκτελέσιμου cgi λόγω μεγέθους.

- Δεσμεύει έναν μεγάλο αριθμό πόρων του συστήματος, όπως μνήμη, τους οποίους δεν μοιράζεται με καμία άλλη διεργασία.

- Επειδή κάθε φορά γίνεται σύνδεση με τον εξυπηρέτη της βάσης τίθεται η επερώτηση και στη συνέχεια κλείνει η σύνδεση με τη βάση, η πύλη διασύνδεσης επιβαρύνεται με όλη τη χρονική καθυστέρηση που προκαλούν οι τεχνικές βελτιστοποίησης που εφαρμόζει το σύστημα διαχείρισης της βάσης, χωρίς να εκμεταλλεύεται τα πλεονεκτήματα που προσφέρουν.

Υπάρχουν πολλά συστήματα που έχουν υλοποιηθεί σύμφωνα με αυτή την αρχιτε-

κτονική. Ενδεικτικά αναφέρονται τα DB2WWW[27], Illustra's Web DataBlade Module[28], PHP/FI[29], Webin Tool[30], GSQL[31], Web/Genera[32], WBD[33] και SQLGate[34].

Αναλυτικότερα θα περιγραφούν τα DB2WWW και Illustra's Web DataBlade Module ως αντιπροσωπευτικότερες υλοποιήσεις.

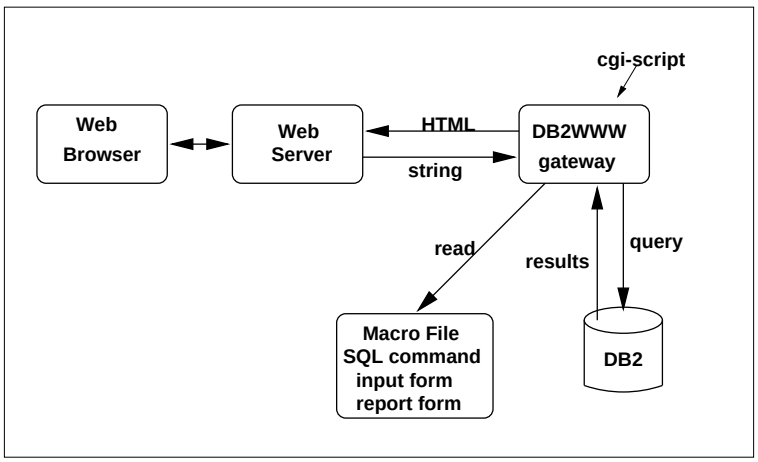
2.4.2 DB2WWW

Το DB2WWW[27] είναι μια πύλη διασύνδεσης που δημιουργήθηκε από την IBM και προσφέρει διασύνδεση μιας βάσης δεδομένων DB2 με τον Παγκόσμιο Ιστό. Η DB2 είναι σχεσιακή βάση δεδομένων και προϊόν της IBM.

Το προϊόν αυτό χρησιμοποιεί αρχεία με μακρο-εντολές (macro-files) για τον καθορισμό της αρχικής σελίδας HTML η οποία θα περιέχει κάποιες φόρμες HTML που θα συμπληρωθούν από τους χρήστες του συστήματος για τον προσδιορισμό των παραμέτρων κλήσης της πύλης διασύνδεσης. Επιπλέον, τα αρχεία περιέχουν τις επερωτήσεις SQL και τον τρόπο μετατροπής των αποτελεσμάτων σε κείμενο HTML.

2.4.2.1 Περιγραφή της αρχιτεκτονικής

Η αρχιτεκτονική που χρησιμοποιεί το DB2WWW φαίνεται στο σχήμα 2.4.



Σχήμα 2.4: Η αρχιτεκτονική που χρησιμοποιεί το DB2WWW

Η διευθυνσιοδότηση γίνεται μέσω URL/URI και είναι της μορφής : *../db2www[.exe]/*{*macro-file*}*/*{*cmd*}*[?*$name$=val&*...]*, όπου {macro-file} είναι το όνομα ενός αρχείου στο οποίο περιέχονται μακρο-εντολές. Οι μακρο-εντολές αφορούν την αρχική σελίδα HTML από την οποία θα κληθεί η πύλη διασύνδεσης, τις επερωτήσεις προς τη βάση και τον τρόπο μετατροπής των αποτελεσμάτων των επερωτήσεων σε κείμενο HTML.

Η μεταβλητή {cmd} προσδιορίζει το τμήμα του αρχείου μακρο-εντολών που θα επεξεργαστεί και μπορεί να πάρει δύο τιμές input και report.

Το σύστημα DB2WWW χρησιμοποιεί έναν γενικό μηχανισμό αντικατάστασης μεταβλητών και χρησιμοποιείται για την επεξεργασία του αρχείου μακρο-εντολών.

2.4.2.2 Δομή ενός αρχείου μακρο-εντολών

Στο τμήμα αυτό θα περιγραφεί η δομή ενός αρχείου μακρο-εντολών του προϊόντος DB2WWW ακολουθώντας το παράδειγμα που ακολουθεί.

```
1 %define{
2  dbtbl="urldb"
3  L_INFO = USE_URL ? "$(dbtbl).url LIKE '%$(SEARCH)%'" : ""
4  L_INFO = USE_TITLE ? "$(dbtbl).title LIKE '%$(SEARCH)%'" : ""
5  WHERELIST = ? "WHERE $(LINFO)"
6  %}
7
8  %SQL[(sql-section-name)]{
9  SELECT url FROM $(dbtbl) $(WHERELIST) order by title
10 %SQL_REPORT{
11 HTML Text..
12 <UL>
13 %ROW{ <LI> <A HREF="something"> text </A>%}
14 </UL>
15 HTML Text
16 %}
17 %}
18
19 %HTML_INPUT{
20 HTML Text
21 %}
22
23 %define{
24 more define statements
25 %}
26
27 %HTML_REPORT{
28 HTML Text
29 %EXEC_SQL[(sql-section-name)]
30 HTML Text
31 %}
```

Στο παραπάνω παράδειγμα όπου υπάρχει το ‘‘HTML Text’’ (γραμμές 11, 15, 20, 28 και 30) σημαίνει ότι μπορεί εκεί να συμπεριληφθεί κείμενο HTML στατικό ή/και

δυναμικό. Ως δυναμικό κείμενο HTML ορίζεται στατικό κείμενο HTML που περιέχει και μεταβλητές οι οποίες θα αντικατασταθούν από τις τιμές τους.

Όλες οι μακρο-εντολές ξεκινούν με το όνομά τους και τελειώνουν με το σύμβολο % (γραμμές 6, 16, 17, 21, 25 και 31). Η μακρο-εντολή που ξεκινάει με το όνομα *%define* (στις γραμμές 1 και 23) προσδιορίζει κάποιες μεταβλητές στις οποίες αντιστοιχίζονται τιμές. Μπορούν να χρησιμοποιηθούν δηλώσεις ελέγχου ροής (conditional statements) για την απόδοση τιμών. Στο παραπάνω παράδειγμα για την απόδοση τιμής στη μεταβλητή L_INFO χρησιμοποιείται ένας έλεγχος ροής (γραμμές 3 και 4). Μεταβλητές οι οποίες δεν έχουν δηλωθεί μέσα στα όρια ενός τμήματος define, όπως οι μεταβλητές USE_TITLE (γραμμή 4), USE_URL (γραμμή 3) και SEARCH (γραμμές 3 και 4) στο παραπάνω παράδειγμα, θεωρείται ότι αποτελούν ονόματα πεδίων της φόρμας HTML από την οποία θα κληθεί το εκτελέσιμο cgi. Ο μηχανισμός αυτός επιτρέπει μια μεταβλητή να θεωρείται κενή όταν έχει ως τιμή κενή συμβολοσειρά ή δεν έχει ορισθεί.

Η μακρο-εντολή με το όνομα *%SQL[(sql-section-name)]* (γραμμή 8) προσδιορίζει μια επερώτηση στη βάση δεδομένων (γραμμή 9). Μπορεί να υπάρχουν περισσότερες από μία μακρο-εντολές %SQL. Για να χρησιμοποιήσει κανείς περισσότερες από μία μακρο-εντολές %SQL πρέπει να τις ονοματίσει. Ο λόγος ονοματοδοσίας των μακρο-εντολών %SQL έχει σχέση με την εκτέλεση των επερωτήσεων και θα εξηγηθεί παρακάτω. Για την ονομασία των μακρο-εντολών %SQL χρησιμοποιείται το *[sql-section-name]*. Μέσα σε κάθε μακρο-εντολή %SQL υπάρχει η μακρο-εντολή *%SQL_REPORT* (γραμμή 10) στην οποία προσδιορίζεται ο τρόπος μετατροπής των αποτελεσμάτων της επερώτησης που προσδιορίστηκε σε κείμενο HTML. Για όλα τα αποτελέσματα της επερώτησης χρησιμοποιείται κοινός τρόπος μετατροπής τους σε κείμενο HTML. Αυτό γίνεται χρησιμοποιώντας την μακρο-εντολή %ROW (γραμμή 13). Επιπλέον, μπορεί να ενσωματωθεί και κείμενο HTML, στατικό και δυναμικό, πριν και μετά από την μακρο-εντολή %SQL_REPORT (γραμμές 11 και 15).

Η μακρο-εντολή με όνομα *%HTML_INPUT* (γραμμή 19) περιέχει κείμενο HTML, στατικό και δυναμικό. Αυτή η μακρο-εντολή εκτελείται όταν το cmd στο URL/URI έχει την τιμή input. Συνήθως περιέχει την φόρμα HTML που θα χρησιμοποιήσει ο χρήστης για την σύνθεση της επερώτησης.

Μετά από αυτήν την μακρο-εντολή μπορεί να ακολουθεί άλλη μία μακρο-εντολή με ορισμούς μεταβλητών, *%define* (γραμμή 23) και τέλος υπάρχει η μακρο-εντολή *%SQL_REPORT* (γραμμή 27) που εκετλείται όταν το cmd είναι report. Μπορεί να περιέχει κείμενο HTML (γραμμή 28) και την μακρο-εντολή *%EXEC_SQL[(sql-section-name)]* (γραμμή 29). Μέσω αυτής της μακρο-εντολής γίνεται η εκτέλεση των μακρο-εντολών %SQL και αποστέλλονται οι επερωτήσεις στον εξυπηρέτη της βάσης προς εκτέλεση. Η μακρο-εντολή %EXEC_SQL μπορεί να χρησιμοποιηθεί μία μόνο φορά ανά [sql-section-name], δηλαδή δεν μπορεί να υπάρχει περισσότερες από μια φορές η μακρο-εντολή %EXEC_SQL[(sql-section-name)] με το ίδιο [sql-section-name]. Χρησιμοποιώντας αυτήν την μακρο-εντολή με το όνομα μιας μακρο-εντολής %SQL (η οποία έχει ήδη

προσδιορισθεί) έχει ως αποτέλεσμα την εκτέλεση της συγκεκριμένης μακρο-εντολής %SQL. Με τον τρόπο αυτό μπορεί να προσδιορισθεί η σειρά με την οποία θα εκτελεστούν οι επερωτήσεις και επομένως θα παρουσιασθούν τα αποτελέσματα στον χρήστη. Αν υπάρχουν περισσότερα του ενός τμήματα %SQL και χρησιμοποιηθεί μόνο η μακρο-εντολή %EXEC_SQL αυτό θα έχει ως αποτέλεσμα την εκτέλεση όλων των μακρο-εντολών %SQL με τυχαία σειρά.

Το DB2WWW παρέχει αρκετή ευελιξία αλλά σε χαμηλό επίπεδο. Ο προγραμματιστής εφαρμογής έχει τη δυνατότητα να προσδιορίσει τις επερωτήσεις και τον τρόπο παρουσίασής τους ως κείμενο HTML, όμως ένα αρχείο μακρο-εντολών αντιστοιχεί σε μια φόρμα HTML. Για ένα σύνολο λειτουργιών, όπου η κάθε λειτουργία αντιστοιχεί σε διαφορετική φόρμα HTML, απαιτείται ένα αντίστοιχο σύνολο αρχείων μακρο-εντολών. Επιπλέον, το DB2WWW δεν δημιουργεί αυτόματα υπερ-συνδέσμους παρά μόνο απλό κείμενο. Για την παραγωγή υπερ-συνδέσμων πρέπει ο προγραμματιστής εφαρμογής να τους δηλώσει.

Λόγω αρχιτεκτονικής, εμφανίζει όλα τα μειονεκτήματα που έχουν προαναφερθεί.

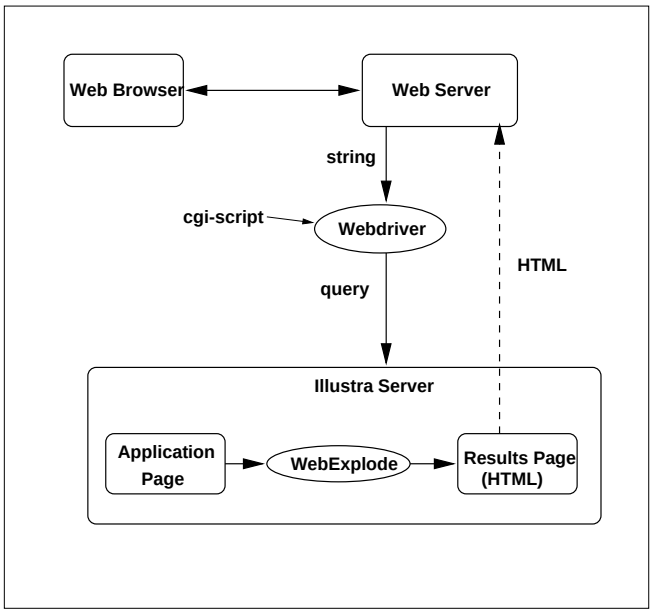
2.4.3 Illustra's Web DataBlade Module

Το Illustra's Web DataBlade Module[28] είναι προϊόν της εταιρείας Illustra για τη διασύνδεση της βάσης δεδομένων Illustra με τον Παγκόσμιο Ιστό. Η Illustra είναι μια οντοκεντρική, σχεσιακή βάση δεδομένων.

Τα βασικά στοιχεία που χρησιμοποιούνται στην πύλη διασύνδεσης είναι οι σελίδες εφαρμογής (Application Pages). Πρόκειται για σελίδες HTML στις οποίες περιέχονται ετικέτες διαμόρφωσης που δεν είναι ορισμένες σύμφωνα με το πρότυπο HTML. Πρόκειται για ετικέτες που έχει εισαγάγει το σύστημα και χρησιμοποιούνται για τον προσδιορισμό των επερωτήσεων στη βάση, τη χρησιμοποίηση μεταβλητών, την παραγωγή μηνυμάτων λάθους, την κλήση εξωτερικών προγραμμάτων και την χρήση εντολών ελέγχου ροής για την μετατροπή των αποτελεσμάτων των επερωτήσεων σε κείμενο HTML. Οι σελίδες εφαρμογής είναι αντικείμενα της βάσης.

2.4.3.1 Περιγραφή της αρχιτεκτονικής

Η διευθυνσιοδότηση των σελίδων εφαρμογής γίνεται μέσω URL/URI. Μέσω του εξυπηρέτη του Παγκόσμιου Ιστού το εκτελέσιμο cgi (που ονομάζεται webdriver) λαμβάνει την πληροφορία για το ποιά σελίδα εφαρμογής πρέπει να ανακτήσει από τη βάση. Ο εξυπηρέτης της βάσης αναλαμβάνει τον εντοπισμό της σελίδας. Στη συνέχεια, την επεξεργασία της σελίδας αναλαμβάνει ένα τμήμα, που ονομάζεται WebExplode. Αυτό το τμήμα εξετάζει αν υπάρχουν ειδικές ετικέτες διαμόρφωσης (εκτός από ετικέτες της HTML) και, αν υπάρχουν, εκτελεί τις επερωτήσεις που προσδιορίζονται και αντικαθιστά στη θέση των επερωτήσεων τα αποτελέσματά τους. Στη συνέχεια, η παραγόμενη σελίδα



Σχήμα 2.5: Η αρχιτεκτονική που ακολουθεί το Illustra

HTML επιστρέφεται στον εξυπηρέτη του Παγκόσμιου Ιστού. Η αρχιτεκτονική φαίνεται στο σχήμα 2.5.

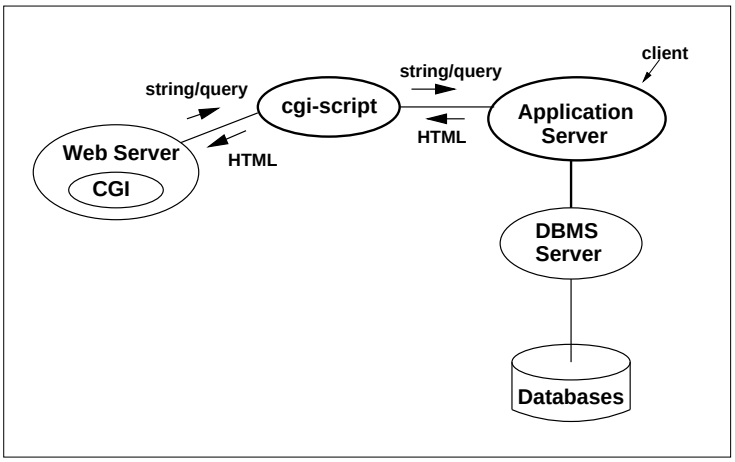
Δεν υπάρχουν αρκετές διαθέσιμες πληροφορίες για το ποιές είναι οι επιπλέον ετικέτες διαμόρφωσης που χρησιμοποιούνται, ούτε και παραδείγματα. Το πλεονέκτημα που παρουσιάζει είναι ότι επειδή η βάση υποστηρίζει εκδόσεις, μπορεί να χρησιμοποιηθεί για την καταγραφή εκδόσεων των σελίδων εφαρμογής.

2.5 Αρχιτεκτονική CGI Application Server

Για να είναι δυνατή η εκμετάλλευση των τεχνικών βελτιστοποίησης που προσφέρουν τα συστήματα διαχείρισης βάσεων δεδομένων, η πύλη διασύνδεσης χωρίζεται σε δύο τμήματα, στο τμήμα που επικοινωνεί με τον εξυπηρέτη της βάσης και στο τμήμα που επικοινωνεί με τον εξυπηρέτη του Παγκόσμιου Ιστού.

2.5.1 Περιγραφή της αρχιτεκτονικής

Σύμφωνα με την αρχιτεκτονική CGI Application Server, που φαίνεται στο σχήμα 2.6, το εκτελέσιμο cgi μεταφέρει τα δεδομένα που δέχεται από τον εξυπηρέτη του Παγκόσμιου Ιστού σε έναν άλλον εξυπηρέτη, στον εξυπηρέτη εφαρμογής (Application server). Ο δεύτερος, επεξεργάζεται τα δεδομένα, παράγει τις κατάλληλες επερωτήσεις, επικοινωνεί με τον εξυπηρέτη της βάσης για την εκτέλεση των επερωτήσεων, συλλέγει τα αποτελέσματα και κάνει τη μετατροπή τους σε κείμενο HTML. Το κείμενο HTML



Σχήμα 2.6: Η αρχιτεκτονική CGI Application Server

επιστρέφεται στο εκτελέσιμο cgi που το επιστρέφει στον εξυπηρέτη του Παγκόσμιου Ιστού.

Επομένως, το εκτελέσιμο cgi χρειάζεται μόνο ως σύνδεσμος μεταξύ του εξυπηρέτη του Παγκόσμιου Ιστού και του εξυπηρέτη της εφαρμογής. Αυτό έχει ως αποτέλεσμα να είναι μικρός ο κώδικας του εκτελέσιμου cgi που σημαίνει λιγότερη επιβάρυνση του εξυπηρέτη του Παγκόσμιου Ιστού.

Όπως διακρίνεται και στο σχήμα 2.6, ο εξυπηρέτης εφαρμογής είναι και εξυπηρετούμενος του εξυπηρέτη της βάσης.

Τα πλεονεκτήματα αυτής της αρχιτεκτονικής είναι :

- Το εκτελέσιμο cgi είναι ανεξάρτητο του συστήματος βάσης δεδομένων.
- Ο εξυπηρέτης εφαρμογής δεν χρειάζεται να γνωρίζει την ύπαρξη του Παγκόσμιου Ιστού. Μπορεί να είναι ένας εξυπηρέτης ο οποίος είναι ανεξάρτητος του Παγκόσμιου Ιστού. Η παραγωγή κειμένου HTML δεν καθορίζει ότι τα δεδομένα θα παρουσιασθούν χρησιμοποιώντας έναν αναδιφητή του Παγκόσμιου Ιστού. Μπορεί να χρησιμοποιηθεί οποιαδήποτε εφαρμογή που κατανοεί τη γλώσσα παράστασης κειμένου HTML.
- Το εκτελέσιμο cgi είναι μικρό και δεν καταλαμβάνει πολύ μνήμη. Έτσι μπορούν να εξυπηρετηθούν πολλές ταυτόχρονες αιτήσεις, που είναι συχνό φαινόμενο δεδομένου ότι πολλοί πελάτες χρησιμοποιούν τις ίδιες υπηρεσίες. Είναι θέμα του εξυπηρέτη εφαρμογής το πώς διαχειρίζεται πολλές ταυτόχρονες αιτήσεις.
- Δεδομένου ότι ο εξυπηρέτης εφαρμογής είναι συγχρόνως εξυπηρετούμενος της βάσης, δεν χρειάζεται να ανοίγει σύνδεση για κάθε επερώτηση και με το τέλος να την κλείνει. Επομένως, είναι δυνατή η εκμετάλλευση των τεχνικών βελτιστοποίησης

του συστήματος διαχείρισης της βάσης. Επιπλέον, με τον τρόπο αυτό μπορεί να εισάγει την έννοια της κατάστασης, που όπως έχει ήδη λεχθεί δεν υποστηρίζεται από το πρωτόκολλο HTTP.

Το μεγάλο μειονέκτημα της μεθόδου αυτής είναι ότι για να εξυπηρετηθεί μία αίτηση πρέπει να συνεργαστούν πολλές διεργασίες. Η επικοινωνία μεταξύ των διεργασιών είναι γενικά χρονοβόρα και μπορεί να προκαλέσει σημαντική χρονική καθυστέρηση.

Μερικά προϊόντα που ακολουθούν αυτή την αρχιτεκτονική είναι τα O_2Web[35], Oracle Web Application Server[36], LINCKS[37], Sibylla[38], ObjectStore[39] και NetDynamics[40]. Ενδεικτικά περιγράφονται τα O_2Web και LINCKS, ενώ γίνεται μια μικρή αναφορά και στον Oracle Web Application Server.

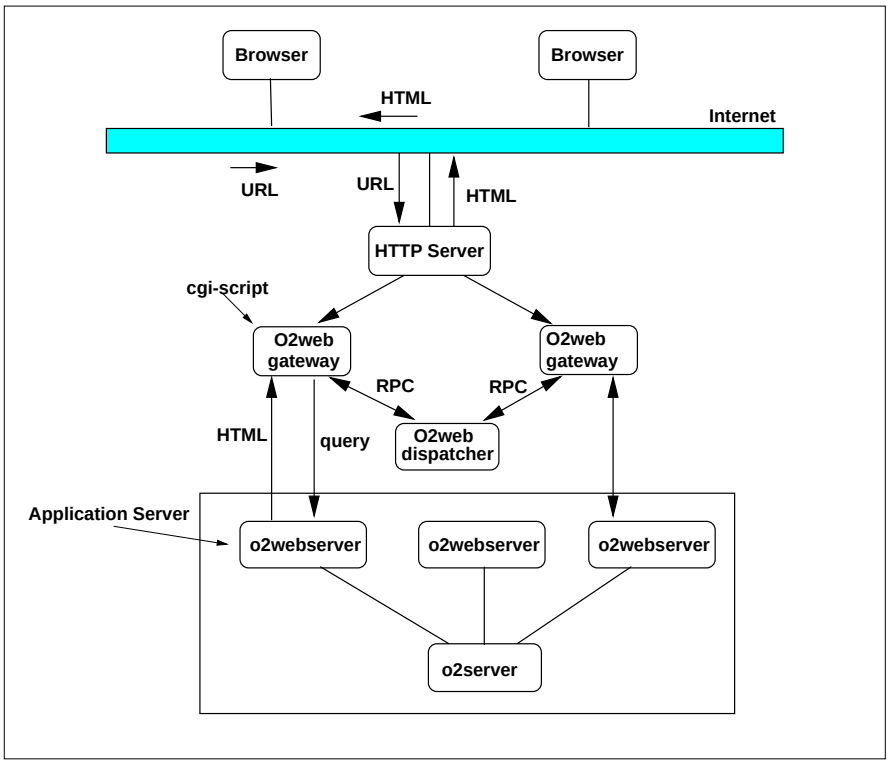
2.5.2 O_2Web

Η πύλη διασύνδεσης O_2Web[35], είναι προϊόν της εταιρείας O_2[41], και χρησιμοποιείται για τη διασύνδεση της οντοκεντρικής βάσης O_2 με τον Παγκόσμιο Ιστό.

Η διευθυνσιοδότηση των επερωτήσεων OQL (Object Query Language) και των αντικειμένων της βάσης γίνεται μέσω URL/URI. Η μορφή είναι ‘‘http://server-name/o2webgateway-path/o2webgateway/[extrapath][?search]’’, όπου ‘‘o2webgateway’’ είναι το όνομα του εκτελέσιμου cgi και ‘‘extrapath’’, ‘‘search’’ είναι η επιπλέον πληροφορία που προσδιορίζει την επερώτηση OQL (ένα αντικείμενο της βάσης ή ένα σύνολο αντικειμένων προσδιορίζεται χρησιμοποιώντας μια επερώτηση OQL).

2.5.2.1 Περιγραφή της αρχιτεκτονικής

Ο χρήστης στέλνει την αίτησή του μέσω ενός εξυπηρετούμενου του Παγκόσμιου Ιστού στον εξυπηρέτη του Παγκόσμιου Ιστού. Η αίτηση του χρήστη είναι ένα URL/URI που προσδιορίζει μια επερώτηση OQL. Ο εξυπηρέτης του Παγκόσμιου Ιστού στέλνει την επερώτηση στην πύλη διασύνδεσης, που ονομάζεται o2webgateway. Η πύλη διασύνδεσης επικοινωνεί με έναν διανεμητή αιτήσεων (dispatcher), ο οποίος προσδιορίζει σε ποιόν εξυπηρέτη εφαρμογής πρέπει να απευθυνθεί. Για αυτήν την επικοινωνία χρησιμοποιείται ο μηχανισμός RPC (Remote Procedure Call). Ο διανεμητής αιτήσεων, που ονομάζεται $O_2Webdispatcher$, λαμβάνει υπόψη του το ποιός εξυπηρέτης εφαρμογής εξυπηρετεί ποιά βάση και τον φόρτο του εξυπηρέτη εφαρμογής έτσι ώστε να υπάρχει εξισορρόπηση φόρτου εργασίας (load balancing), όσο αυτό είναι δυνατό. Στη συνέχεια η πύλη διασύνδεσης συνδέεται στον κατάλληλο εξυπηρέτη εφαρμογής, που ονομάζεται $O_2WebServer$, και στέλνει την επερώτηση του χρήστη με μορφή αίτησης κατανοητής στον $O_2WebServer$. Ο $O_2WebServer$ συνδέεται με τον αντίστοιχο εξυπηρέτη της βάσης, $O_2Server$, στέλνει την επερώτηση OQL, συλλέγει και μετατρέπει τα αποτελέσματα σε κείμενο HTML, το οποίο



Σχήμα 2.7: Η αρχιτεκτονική που ακολουθεί η O_2.

μέσω της πύλης διασύνδεσης επιστρέφεται στον εξυπηρέτη του Παγκόσμιου Ιστού και από εκεί στον χρήστη που έθεσε την επερώτηση. Τα παραπάνω φαίνονται στο σχήμα 2.7.

Το γενικό μοντέλο παράστασης διακρίνει πέντε διαφορετικά τμήματα σε μία σελίδα HTML. Τα τμήματα είναι ο πρόλογος (prolog), το άνω μέρος της σελίδας (header), το κύριο τμήμα (body), το κάτω μέρος της σελίδας (footer) και ο επίλογος (epilog). Ο πρόλογος αντιστοιχεί στους τύπους MIME που πρέπει να σταλούν, σύμφωνα με το πρωτόκολλο HTTP, πριν από τη σελίδα HTML. Το άνω μέρος της σελίδας αντιστοιχεί στο τμήμα μεταξύ των ετικετών διαμόρφωσης της HTML <HEAD> και </HEAD>. Το κύριο τμήμα αντιστοιχεί στο τμήμα μεταξύ των ετικετών διαμόρφωσης της HTML <BODY> και λίγο πριν το </BODY>. Το κάτω μέρος της σελίδας χρησιμοποιείται για το τελευταίο τμήμα μέχρι και την ετικέτα διαμόρφωσης </HTML>. Ο επίλογος δεν χρησιμοποιείται, έχει δηλωθεί για μελλοντική χρήση.

Για την μετατροπή των αποτελεσμάτων σε κείμενο HTML χρησιμοποιούνται κατάλληλες μέθοδοι των αντικειμένων. Υπάρχει μία μέθοδος για κάθε ένα από τα παραπάνω τμήματα (html_prolog, html_header, html_body, html_footer και html_epilog). Κάθε μία από τις παραπάνω μεθόδους ορίζεται για κάθε κλάση αντικειμένων της οντοκεντρικής βάσης O_2. Με αυτόν τον τρόπο μπορεί άμεσα να εξαρτηθεί η παρουσίαση αντικειμένων σε κείμενο HTML από τον τύπο και την σημασιολογία τους που ο προγραμματιστής

εφαρμογής είναι σε θέση να γνωρίζει. Επιπλέον, λόγω της κληρονόμησης των μεθόδων, αν δεν έχει ορισθεί μία μέθοδος από τις παραπάνω, χρησιμοποιείται αυτή που κληρονομείται σύμφωνα με την ιεραρχία των κλάσεων. Με τον τρόπο αυτό, είναι δυνατό να ορισθεί κοινός πρόλογος, άνω και κάτω μέρος σελίδας για ένα σύνολο κλάσεων οι οποίες κληρονομούν τη συμπεριφορά τους από μια κοινή υπερ-κλάση. Όπως επίσης και μια τυπική παρουσίαση για όλα τα αντικείμενα που είναι μέλη υποκλάσεων μιας κλάσης. Στην ιεραρχία των κλάσεων της O_2 είναι ορισμένες αυτές οι μέθοδοι, προσδιορίζοντας με τον τρόπο αυτό μια τυπική παρουσίαση για όλα τα αντικείμενα της βάσης.

Οι μέθοδοι ορίζονται χρησιμοποιώντας μία οντοκεντρική γλώσσα προγραμματισμού που μοιάζει πολύ με τη C++ και ονομάζεται O_2C.

2.5.2.2 Παράδειγμα

Ακολουθεί ένα παράδειγμα ορισμού των μεθόδων για την μετατροπή των αποτελεσμάτων μιας επερώτησης σε κείμενο HTML στο O_2Web.

- Παράδειγμα της μεθόδου html_header,

```
1  method body html_header: bits in class O2WebInteractor {
2    O2 bits result;
3    result += "<html><header>\n";
4    result += "<title>" + self->title + "</title></header>";
5    result += "<body><H1>" + self->title + "</H1>";
6    return result;
7  }
```

- Παράδειγμα της μεθόδου html_body,

```
1  method body html_report: bits in class O2WebInteractor {
2    O2 bits result;
3    O2 bits anchor;
4    result += "<dl>";
5    anchor = toolbox->make_anchor(
6      toolbox->make_url(self->get_query + "->entries", "",
7             self->title, 0), "Browse the Directory");
8    result += "<dt> <h3>" + anchor + "</h3></dt></dl>\n";
9    return result;
10 }
```

Η μεταβλητή ‘‘self’’ (γραμμές 4, 5 στο πρώτο παράδειγμα και 6,7 στο δεύτερο) είναι το αντίστοιχο ‘‘this’’ των οντοκεντρικών γλωσσών προγραμματισμού και προσδιορίζει το τρέχον κάθε φορά αντικείμενο. Η μέθοδος ‘‘get_query’’ (γραμμή 6 στο δεύτερο παράδειγμα) έχει ορισθεί στην συγκεκριμένη κλάση αντικειμένων και επιστρέφει μια επερώτηση OQL για τον προσδιορισμό του τρέχοντος αντικειμένου. Την μέθοδο αυτή την ορίζει ο προγραμματιστής εφαρμογής.

Οι συναρτήσεις ‘‘make_anchor’’ και ‘‘make_url’’ (γραμμές 5,6 στο δεύτερο παράδειγμα) παρέχονται από το περιβάλλον προγραμματισμού του προϊόντος O_2Web με σκοπό την διευκόλυνση του προγραμματιστή εφαρμογής και ανήκουν σε ένα γενικότερο σύνολο συναρτήσεων.

Μέσω του O_2Web μπορούν να γίνουν και άλλες λειτουργίες στη βάση, εκτός από επερωτήσεις, όπως εισαγωγή νέας πληροφορίας, διαγραφή και ενημέρωση.

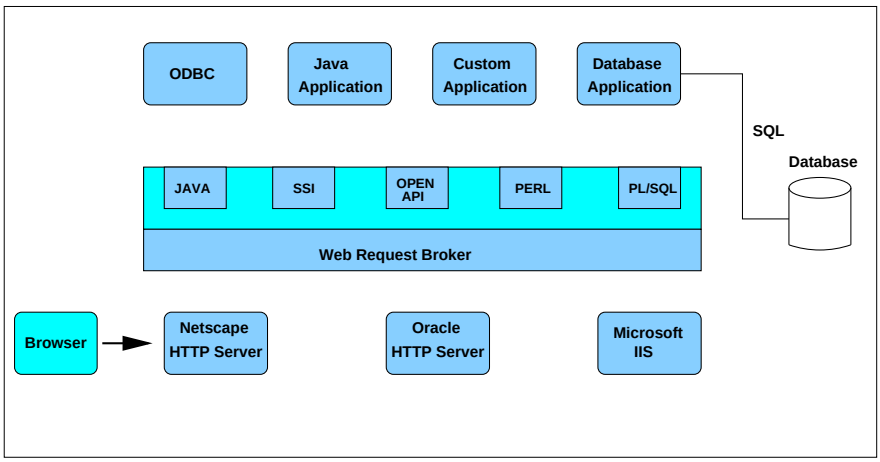
Η παραγωγή των σελίδων HTML γίνεται μέσω μεθόδων που μπορούν να επαναπροσδιορισθούν (overloading). Η χρησιμοποίηση του διανεμητή αιτήσεων διευκολύνει την αντιμετώπιση πολλών κλήσεων σε μία βάση δεδομένων. Δεδομένου ότι την περισσότερη επεξεργασία την κάνει ο εξυπηρέτης εφαρμογής, αυτός αποτελεί και το σημείο συμφόρησης. Μέσω του διανεμητή αιτήσεων μπορούν να υπάρξουν περισσότεροι του ενός εξυπηρέτες εφαρμογής για την εξυπηρέτηση μιας βάσης δεδομένων.

2.5.3 Oracle Web Application Server

Είναι ένα προϊόν της Oracle[36], και προσφέρει γενικότερες λειτουργίες από την διασύνδεση της σχεσιακής βάσης Oracle με τον Παγκόσμιο Ιστό, όπως κλήση εφαρμογών χρήστη και επικοινωνία με άλλες βάσεις δεδομένων. Η ιδιαιτερότητα που παρουσιάζει είναι η χρησιμοποίηση αρχιτεκτονικής CORBA. Όλο και περισσότερα προϊόντα ακολουθούν τελευταία αυτή την προσέγγιση λόγω της ευελιξίας που προσφέρει η υιοθέτηση της αρχιτεκτονικής CORBA. Ο εξυπηρέτης εφαρμογής είναι στην ουσία ένας Web Request Broker. Με τον τρόπο αυτό προσφέρεται ομοιόμορφη πρόσβαση σε δεδομένα που είναι αποθηκευμένα σε βάσεις δεδομένων, το σύστημα αρχείων κ.α. Επιπλέον, ο Web Request Broker είναι και ο επιτηρητής των δοσοληψιών (Transaction Monitor).

2.5.3.1 Περιγραφή της αρχιτεκτονικής

Σύμφωνα με αυτήν την αρχιτεκτονική, που φαίνεται στο σχήμα 2.8, η αίτηση του χρήστη μεταφέρεται μέσω οποιουδήποτε εξυπηρέτη του Παγκόσμιου Ιστού στον Web Request Broker (WRB). Ο WRB αναλαμβάνει να ανακατευθύνει την αίτηση στην κατάλληλη εφαρμογή που θα αναλάβει την εξυπηρέτησή της. Η χρησιμοποίηση ενός WRB προσφέρει έναν μηχανισμό για ομοιόμορφη επικοινωνία με ένα σύνολο εφαρμογών, όπως εφαρμογές βάσεων δεδομένων και Java.



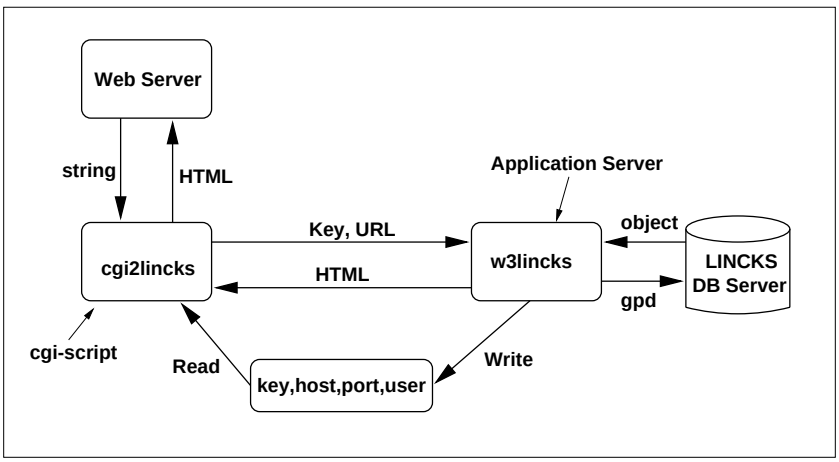
Σχήμα 2.8: Η αρχιτεκτονική που ακολουθεί η Oracle

2.5.4 LINCKS

Το προϊόν LINCKS[37] είναι μια οντοκεντρική βάση δεδομένων. Η ομάδα εργασίας που υλοποίησε τη βάση, έχει υλοποιήσει και μια πύλη διασύνδεσής της με τον Παγκόσμιο Ιστό.

2.5.4.1 Περιγραφή της αρχιτεκτονικής

Η αίτηση ενός χρήστη, που γίνεται μέσω ενός εξυπηρετούμενου του Παγκόσμιου



Σχήμα 2.9: Η αρχιτεκτονική που ακολουθεί το LINCKS

Ιστού, στέλνεται στον εξυπηρέτη του Παγκόσμιου Ιστού με τη μορφή URL/URI. Το URL/URI είναι της μορφής ‘‘../cgi2lincks.exe/<object>-<gpd>’’, όπου ‘‘object’’ είναι το όνομα ή η συντομογραφία ενός αντικειμένου της βάσης. Η χρησιμοποίηση συντομογρα-

φιών είναι χαρακτηριστικό της οντοκεντρικής βάσης, η οποία και τις διαχειρίζεται. Το ‘‘gpd’’ είναι το όνομα ενός αντικειμένου της βάσης που έχει ειδική δομή και ονομάζεται ‘‘General Presentation Descriptor’’. Η δομή αυτή χρησιμοποιείται για την παράσταση ενός αντικειμένου και της συσχετιζόμενης πληροφορίας σε κείμενο HTML. Τέλος, στο URL/URI προσδιορίζεται και μία άλλη μεταβλητή που ονομάζεται ‘‘Key’’ και προσδιορίζει ένα αρχείο στο οποίο περιέχονται πληροφορίες για το πού βρίσκεται ο εξυπηρέτης εφαρμογής. Το URL/URI δίνεται στο εκτελέσιμο cgi το οποίο αφού διαβάσει το προσδιοριζόμενο από το ‘‘Key’’ αρχείο, συνδέεται με τον εξυπηρέτη εφαρμογής και στέλνει το URL/URI. Ο εξυπηρέτης εφαρμογής, που ονομάζεται w3lincks, επεξεργάζεται το URL/URI για την παραγωγή της κατάλληλης επερώτησης και την στέλνει στον εξυπηρέτη της βάσης. Ο εξυπηρέτης της βάσης επιστρέφει το αντικείμενο, που προσδιοριζόταν από την επερώτηση, και το αντικείμενο gpd. Το w3lincks με βάση το gpd παράγει την σελίδα HTML και την στέλνει στο εκτελέσιμο cgi, το οποίο με τη σειρά του την επιστρέφει στον εξυπηρέτη του Παγκόσμιου Ιστού και από εκεί στον τελικό χρήστη. Τα παραπάνω φαίνονται στο σχήμα 2.9.

Το αρχείο που προσδιορίζεται από το ‘‘Key’’ δημιουργείται από το w3licks κάθε φορά που αρχικοποιείται και περιέχει το μηχάνημα και την πόρτα στην οποία μπορεί να δεχθεί αιτήσεις καθώς και το ποιός χρήστης έχει αρχικοποιηθεί.

2.5.4.2 General Presentation Descriptor

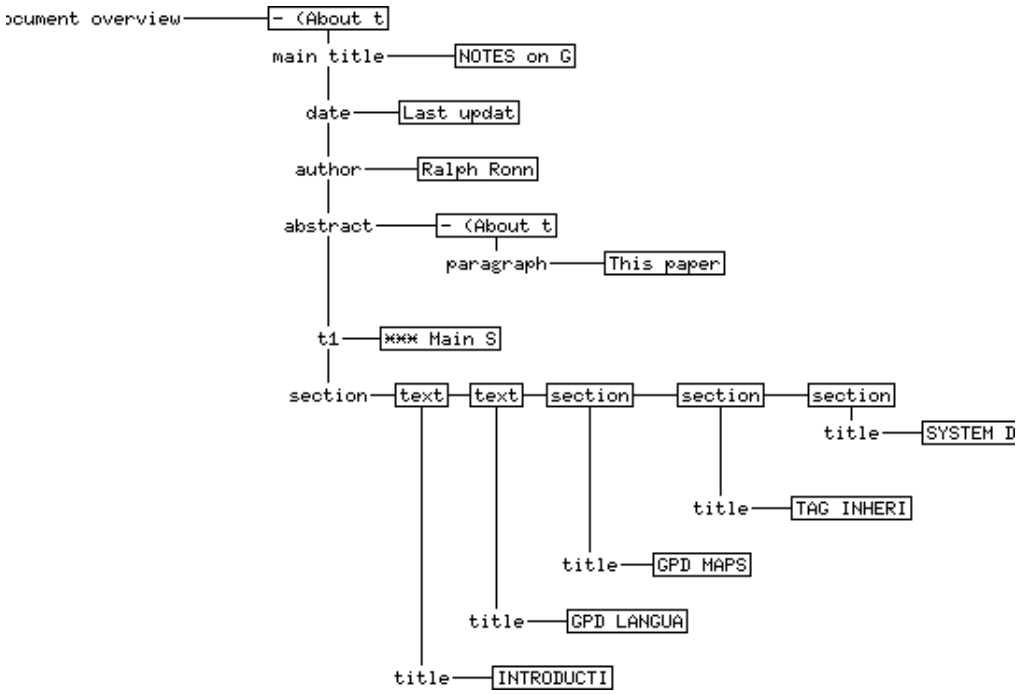
Η δομή General Presentation Descriptor (GPD), αποτελείται από τα παρακάτω τμήματα :

- STRUCTURE
  Η λογική δομή της όψης του αντικειμένου που θα παρουσιασθεί. Ένα παράδειγμα φαίνεται στο σχήμα 2.10. Μοιάζει πολύ με απλουστευμένο DTD της SGML. Προσδιορίζει με ποιά σειρά πρέπει να παρουσιασθούν ποιά στοιχεία του αντικειμένου.

- ACCESS
  Πού βρίσκεται η απαραίτητη πληροφορία στη βάση για την κατασκευή της παράστασης.

- EXPAND
  Ποιό GPD πρέπει να χρησιμοποιηθεί όταν επιλέγεται ένας υπερ-σύνδεσμος από την παραγόμενη σελίδα HTML.

- EDIT MODE
  Περιορισμοί στο είδος του τερματικού στο οποίο θα γίνει η παρουσίαση, δηλαδή τί δυνατότητες έχει το τερματικό. Χρησιμοποιείται κυρίως στην έκδοση του LINCKS σε X-Windows.



Σχήμα 2.10: Ένα παράδειγμα του τμήματος STRUCTURE ενός GPD

- FORMAT
  X11 format attributes, για την αντίστοιχη έκδοση του LINCKS.

- FORMATX
  Προσδιορίζονται οι ενέργειες που πρέπει να πραγματοποιηθούν για την παραγωγή του κειμένου HTML. Παραδείγματα εντολών που μπορούν να χρησιμοποιηθούν είναι :

  ▷ IF(integer-expr, formatx, formatx)

  ▷ EQ/GE/GT/LE/LT/NE(int-exp, int-exp)

Το ενδιαφέρον στοιχείο στην προσέγγιση που περιγράφηκε, είναι η εισαγωγή του μοντέλου παράστασης των αντικειμένων που σε όλα τα προηγούμενα προϊόντα βασιζόταν σε προγραμματιστικές μετατροπές και χρήση φίλτρων. Το μοντέλο αυτό μπορεί να χρησιμοποιηθεί και για την μετατροπή σε άλλους μορφότυπους παράστασης δεδομένων, όπως απλό κείμενο, SGML, XML κ.α.

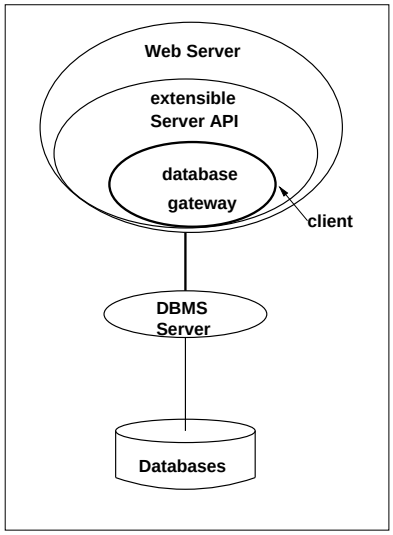
2.6 Αρχιτεκτονική Server API

Αρκετοί εξυπηρέτες του Παγκόσμιου Ιστού προσφέρουν μια διεπιφάνεια προγραμματισμού εφαρμογών (API) μέσω της οποίας μπορούν οι προγραμματιστές εφαρμογών να

επαυξήσουν τις λειτουργίες του εκάστοτε εξυπηρέτη του Παγκόσμιου Ιστού. Παραδείγματα είναι το NSAPI του Netscape[42] και το ISAPI του Microsoft IIS (Internet Information Server)[43].

2.6.1 Περιγραφή της αρχιτεκτονικής

Όπως διακρίνεται και στο σχήμα 2.11, η πύλη διασύνδεσης αποτελεί τμήμα



Σχήμα 2.11: Η αρχιτεκτονική Server API

του εξυπηρέτη του Παγκόσμιου Ιστού. Όταν χρειαστεί να γίνει μία κλήση προς την πύλη διασύνδεσης δεν χρειάζεται να δημιουργηθεί κανένα αντίγραφο διεργασίας, όπως εκτελέσιμου cgi, από τον εξυπηρέτη του Παγκόσμιου Ιστού. Ουσιαστικά δίνει τον έλεγχο στην πύλη διασύνδεσης που εξυπηρετεί την αίτηση, μειώνοντας το κόστος διαχείρισης διεργασιών σύμφωνα με το πρωτόκολλο CGI.

Συνήθως αυτή η αρχιτεκτονική συνδυάζεται με την εισαγωγή στη σελίδα HTML ειδικών ετικετών διαμόρφωσης για τον προσδιορισμό των επερωτήσεων και την κλήση script σε κάποια γλώσσα προγραμματισμού script, όπως perl scripts, java scripts κ.α. Η χρήση ειδικών ετικετών διαμόρφωσης δεν αποτελεί χαρακτηριστικό της αρχιτεκτονικής, είναι ορθογώνια. Σ'αυτή την περίπτωση η πύλη διασύνδεσης προσδιορίζει, μέσω των ειδικών ετικετών διαμόρφωσης, τις επερωτήσεις προς τη βάση, τις εκτελεί και αντικαθιστά τα αποτελέσματα στη θέση των επερωτήσεων στη σελίδα HTML. Εξαρτάται από την υλοποίηση το αν τα αποτελέσματα επεξεργάζονται πριν εισαχθούν στη σελίδα HTML ή εισάγονται όπως επιστρέφονται από τη βάση.

Το μεγάλο πλεονέκτημα αυτής της μεθόδου είναι ότι είναι ταχύτερη από τις δύο που περιγράφηκαν ήδη. Αυτό ισχύει επειδή ο εξυπηρέτης εφαρμογής δεν δημιουργεί κανένα αντίγραφο και ο έλεγχος περνάει αμέσως στην πύλη διασύνδεσης. Δεν χρειάζεται να

μεσολαβήσουν άλλες διεργασίες.

Το μεγάλο μειονέκτημα αυτής της αρχιτεκτονικής είναι ότι δεν υπάρχει κανένα πρότυπο για διεπιφάνεια προγραμματισμού εφαρμογών εξυπηρέτη του Παγκόσμιου Ιστού και επομένως η υλοποίηση της πύλης διασύνδεσης εξαρτάται από την υλοποίηση της διεπιφάνειας προγραμματισμού εφαρμογών που προσφέρει ο κάθε εξυπηρέτης του Παγκόσμιου Ιστού. Η πύλη διασύνδεσης δεν είναι εύκολα μεταφέρσιμη για χρησιμοποίηση σε άλλον εξυπηρέτη του Παγκόσμιου Ιστού και είναι πολύ ευαίσθητη σε αλλαγές που γίνονται στην διεπιφάνεια προγραμματισμού εφαρμογών του εξυπηρέτη για τον οποίο έχει γραφεί.

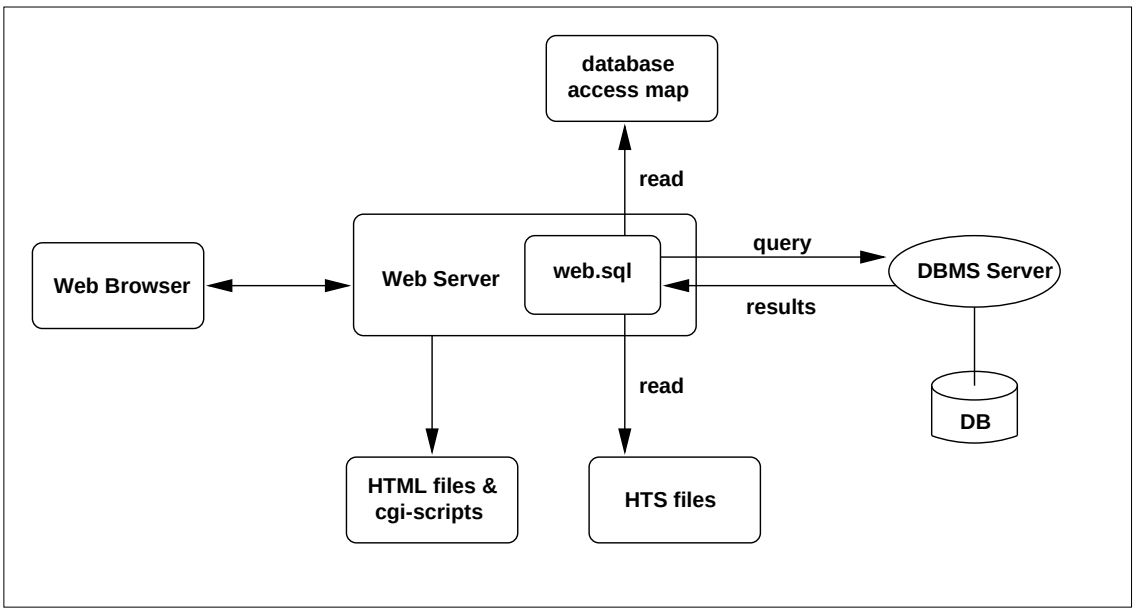
Το ποιό χαρακτηριστικό προϊόν που υλοποιεί αυτή την αρχιτεκτονική είναι η πύλη διασύνδεσης που προσφέρει η Sybase.

2.6.2 Sybase web.sql

Το web.sql[44] είναι ένα προϊόν της εταιρείας Sybase[45] για την διασύνδεση της σχεσιακής βάσης Sybase με τον Παγκόσμιο Ιστό. Χρησιμοποιεί αρχεία HTS (HyperText Sybase) που περιέχουν ειδικές ετικέτες διαμόρφωσης για τον προσδιορισμό των επερωτήσεων SQL στη βάση και τη χρήση perl scirpts.

2.6.2.1 Περιγραφή της αρχιτεκτονικής

Η αρχιτεκτονική του προϊόντος web.sql φαίνεται στο σχήμα 2.12. Η κλήση γίνεται



Σχήμα 2.12: Η αρχιτεκτονική που χρησιμοποιεί η Sybase

μέσω URL/URI και είναι της μορφής ‘‘http://<servername>/<web.sql>/<file.hts>’’. Όπου

<servername> είναι η διεύθυνση του εξυπηρέτη του Παγκόσμιου Ιστού, <web.sql> είναι το τμήμα web.sql και <file.hts> είναι το όνομα ενός αρχείου HTS. Ο εξυπηρέτης του Παγκόσμιου Ιστού εξυπηρετεί όλες τις αιτήσεις για σελίδες HTML και εκτελέσιμα cgi. Όταν καλείται ο εξυπηρέτης του Παγκόσμιου Ιστού να εξυπηρετήσει μια αίτηση για κάποιο αρχείο HTS, δίνει τον έλεγχο στο web.sql. Το web.sql διαβάζει το αρχείο που προσδιορίζεται από το URL/URI και κάποια αρχεία που περιγράφουν το που βρίσκονται οι εξυπηρέτες των βάσεων της Sybase. Στέλνει στον αντίστοιχο εξυπηρέτη Sybase τις επερωτήσεις SQL ενώ καλεί ως εξωτερική εφαρμογή τον interpreter της perl για την εκτέλεση των perl scripts που μπορεί να περιέχονται στο αρχείο.

Η ετικέτα διαμόρφωσης που χρησιμοποιείται είναι η ‘‘<SYB>’’ με όρισμα το αν πρόκειται για επερώτηση SQL ή για perl script.

Το web.sql δεν κάνει καμία επεξεργασία των αποτελεσμάτων της επερώτησης ή των αποτελεσμάτων του perl script. Τα εισάγει στην σελίδα HTML στη θέση της ειδικής ετικέτας διαμόρφωσης όπως είναι.

Το χαρακτηριστικό αυτής της υλοποίησης είναι ότι ο εξυπηρέτης του Παγκόσμιου Ιστού χειρίζεται τα αρχεία HTS όπως και τα αρχεία HTML. Δεν υπάρχει καμία διαφορά.

2.6.2.2 Παράδειγμα ενός αρχείου HyperText Sybase

Στην ενότητα αυτή θα περιγραφεί ένα παράδειγμα αρχείου HTS.

```
1  <HTML>
2  <HEAD><TITLE>Παράδειγμα ενός αρχείου HTS</TITLE></HEAD>
3  <BODY><H1>Παράδειγμα ενός αρχείου HTS</H1>
4  <P><B>Τιμοκατάλογος Βιβλίων</B>
5  <SYB TYPE=SQL>
6   select title, price
7   from books
8  </SYB>
9  .. κείμενο HTML
10 <SYB TYPE=PERL>
11  $sql_stmt = qq!select title, price
12        from books!;
13  $format = "%s κοστίζει %s.<P>\n";
14  ws_sql($ws_db, $sql_stmt, $format);
15 </SYB>
16 .. Επιπλέον κείμενο HTML
17 </BODY></HTML>
```

Όπως φαίνεται και από το παραπάνω παράδειγμα, ένα αρχείο HTS είναι ένα αρχείο HTML που περιέχει επιπλέον τις ετικέτες διαμόρφωσης *<SYB TYPE=SQL>*

(γραμμές 5-8) και <*SYB TYPE=PERL*> (γραμμές 10-15). Στις γραμμές 6 και 7 φαίνεται μία απλή επερώτηση για τα αποτελέσματα της οποίας δεν χρησιμοποιείται κάποιος ειδικός τρόπος παράστασης σε κείμενο HTML εν αντιθέσει με το παράδειγμα χρήσης perl όπου χρησιμοποιείται η μεταβλητή/συνάρτηση *$format* για την παράσταση των αποτελεσμάτων σε κείμενο HTML (γραμμές 13,14). Η εντολή *qq* περικλύει σε εισαγωγικά (") το κείμενο μεταξύ των χαρακτήρων ! (γραμμές 11 και 12). Η εντολή *ws_sql* είναι μια συνάρτηση που προσφέρει το web.sql και χρησιμοποιείται για την αποστολή μιας επερώτησης στον εξυπηρέτη της βάσης δεδομένων. Το τελευταίο όρισμα αυτής της συνάρτησης χρησιμοποιείται για την μετατροπή των αποτελεσμάτων της επερώτησης σε HTML (στην παραπάνω περίπτωση είναι το *$format*). Πρέπει να παρατηρηθεί ότι στο παράδειγμα χρήσης perl script η μετατροπή των αποτελεσμάτων σε κείμενο HTML γίνεται μέσω της perl και όχι από το web.sql.

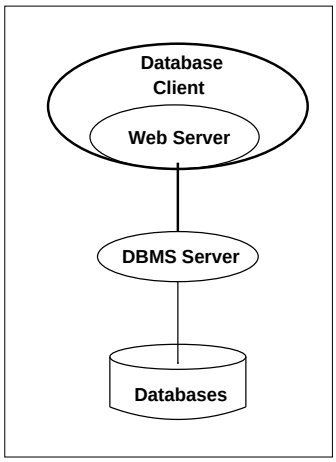
2.7 Αρχιτεκτονική Proprietary Server

Από την πλευρά ενός συστήματος διαχείρισης βάσης δεδομένων, μια πύλη διασύνδεσης με τον Παγκόσμιο Ιστό είναι ένας εξυπηρετούμενος που καταλαβαίνει την τεχνολογία του Παγκόσμιου Ιστού. Σύμφωνα με αυτή τη λογική, μπορεί κάποιος να υλοποιήσει μία εφαρμογή που είναι εξυπηρετούμενος μιας βάσης δεδομένων και υποστηρίζει το πρωτόκολλο HTTP άμεσα, δηλαδή δέχεται αιτήσεις και απαντά χρησιμοποιώντας το πρωτόκολλο HTTP. Μία τέτοιου είδους εφαρμογή μπορεί να θεωρηθεί ως ένας εξυπηρέτης του Παγκόσμιου Ιστού που είναι εξειδικευμένος για ένα συγκεκριμένο σύστημα διαχείρισης βάσεων δεδομένων.

2.7.1 Περιγραφή της αρχιτεκτονικής

Όπως φαίνεται στο σχήμα 2.13, η πύλη διασύνδεσης είναι μία εφαρμογή που είναι εξυπηρετούμενος της βάσης και εξυπηρέτης του Παγκόσμιου Ιστού. Δηλαδή, συνδέεται απευθείας με τον εξυπηρέτη της βάσης δεδομένων χωρίς να χρειάζεται η μεσολάβηση κάποιας άλλης διεργασίας ή η ενσωμάτωση κάποιας άλλης εφαρμογής.

Το μεγάλο πλεονέκτημα αυτής της αρχιτεκτονικής, όπως και της προηγούμενης, είναι η ταχύτητα, επειδή δεν μεσολαβούν άλλες διεργασίες μέχρι να φτάσει η αίτηση στον εξυπηρέτη της βάσης και από αυτόν μέχρι να επιστραφεί η σελίδα HTML στον εξυπηρέτη του Παγκόσμιου Ιστού. Επιπλέον, το τμήμα που συμπεριφέρεται ως εξυπηρέτης του Παγκόσμιου Ιστού δεν χρειάζεται να προσφέρει τις λειτουργίες ενός τυπικού εξυπηρέτη του Παγκόσμιου Ιστού. Μπορεί να είναι εξειδικευμένος για την υποστήριξη μόνο της βάσης. Με τον τρόπο αυτό η ταχύτητα απόκρισης είναι καλύτερη. Σε σχέση με την αρχιτεκτονική Server API, η ταχύτητα απόκρισης δεν έχει σημαντική διαφορά. Διαφορά μπορεί να υπάρξει όταν στην αρχιτεκτονική Server API το τμήμα του εξυπηρέτη του Παγκόσμιου Ιστού που είναι η πύλη διασύνδεσης φορτώνεται δυναμικά στη μνήμη μόλις



Σχήμα 2.13: Η αρχιτεκτονική Proprietary Server

γίνει μια αίτηση γι' αυτό και διαγράφεται από τη μνήμη όταν τελειώσει η εξυπηρέτηση της αίτησης. Σ' αυτήν την περίπτωση ταχύτερη είναι η αρχιτεκτονική Proprietary Server.

Το μειονέκτημα αυτής της αρχιτεκτονικής είναι ότι σε μια αλλαγή ενός των πρωτοκόλλων HTML και HTTP, αν οι υπόλοιποι εξυπηρέτες του παγκόσμιου ιστού δεν υποστηρίζουν παλιές εκδόσεις, πρέπει να ακολουθηθούν οι αλλαγές.

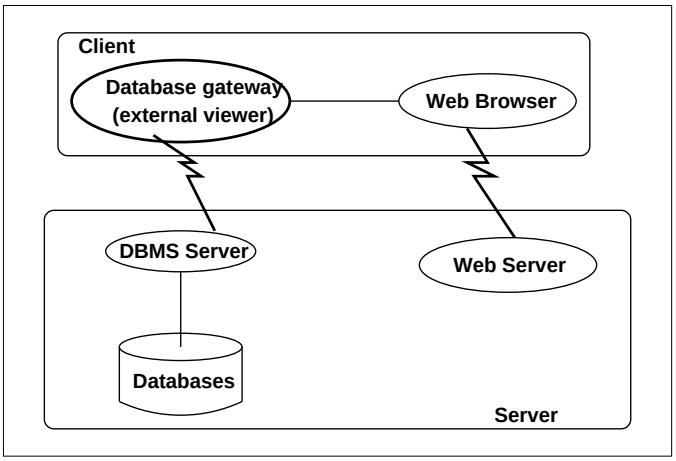
Παραδείγματα προϊόντων που ακολουθούν αυτή την αρχιτεκτονική είναι ο AOL Server[46] που χρησιμοποιεί την βάση δεδομένων Illustra και InterServ που χρησιμοποιεί τη βάση δεδομένων UniSQL[47].

2.8 Αρχιτεκτονική External Viewer

Οι αναδιφητές του Παγκόσμιου Ιστού (Web Browsers) έχουν την δυνατότητα, αν ορισθεί κατάλληλα από τον χρήστη, να καλέσουν εξωτερικές εφαρμογές για την παρουσίαση δεδομένων τα οποία δεν είναι σε θέση να παρουσιάσουν. Η κλήση εξωτερικών εφαρμογών γίνεται με τη χρήση των κατάλληλων τύπων MIME. Για παράδειγμα μπορεί να καλεσθεί η εφαρμογή ''mpeg_player'' για την παρουσίαση βίντεο σύμφωνου με το πρότυπο MPEG. Αυτή η δυνατότητα μπορεί να χρησιμοποιηθεί για την σύνδεση των βάσεων δεδομένων με τον Παγκόσμιο Ιστό.

2.8.1 Περιγραφή της αρχιτεκτονικής

Σύμφωνα με την αρχιτεκτονική External Viewer, σχήμα 2.14, η πύλη διασύνδεσης καλείται από τον αναδιφητή του Παγκόσμιου Ιστού ως εξωτερική εφαρμογή η οποία συνδέεται με τον εξυπηρέτη της βάσης. Σε αυτή την αρχιτεκτονική, όπως και στην προηγούμενη, δεν χρησιμοποιείται το πρωτόκολλο CGI. Επιπλέον, δεν χρειάζεται καμία αλλαγή στον εξυπηρέτη ή στον αναδιφητή του Παγκόσμιου Ιστού. Χαρακτηριστικό της



Σχήμα 2.14: Η αρχιτεκτονική External Viewer

αρχιτεκτονικής είναι ότι ο εξυπηρέτης του Παγκόσμιου Ιστού δεν επικοινωνεί άμεσα ή έμμεσα με τον εξυπηρέτη της βάσης.

Το μεγάλο πλεονέκτημα αυτής της αρχιτεκτονικής είναι ότι μπορούν να επαναχρησιμοποιηθούν έτοιμες εφαρμογές της βάσης χωρίς να χρειάζεται οποιαδήποτε προσθήκη για την υποστήριξη του Παγκόσμιου Ιστού. Επιπλέον, η παρουσίαση δεν χρειάζεται να περιορισθεί στα μέσα που παρέχει ο Παγκόσμιος Ιστός, δηλαδή διαμόρφωση σε HTML.

Ένα μεγάλο μειονέκτημα αυτής της αρχιτεκτονικής είναι ότι δεν μπορούν να χρησιμοποιηθούν εύκολα τεχνικές εξισορρόπησης φορτίου. Όλοι οι εξυπηρετούμενοι θα γνωρίζουν έναν εξυπηρέτη στον οποίο πρέπει να απευθυνθούν και ο εξυπηρέτης αυτός θα αποτελέσει το σημείο συμφόρησης του όλου συστήματος. Επιπλέον, δεδομένου ότι από την πλευρά του εξυπηρετούμενου του Παγκόσμιου Ιστού επιτυγχάνεται η σύνδεση με τη βάση, δεν είναι δυνατό να γίνει μια καλή διαχείριση των συνδέσεων με τον εξυπηρέτη της βάσης. Στα συστήματα διαχείρισης βάσεων δεδομένων συνήθως υπάρχει ένας περιορισμός ως προς τον αριθμό των ταυτόχρονων ανοιχτών συνδέσεων ενώ η κάθε σύνδεση αρχικοποιείται με συγκεκριμένες άδειες. Αντιθέτως, όταν τον έλεγχο της σύνδεσης με τη βάση τον έχει μια κεντρική διεργασία μπορεί να εφαρμόσει κάποιους αλγορίθμους για την καλύτερη δρομολόγηση των αιτήσεων και χρησιμοποίηση των συνδέσεων με τη βάση.

2.9 Αρχιτεκτονική Browser Extension

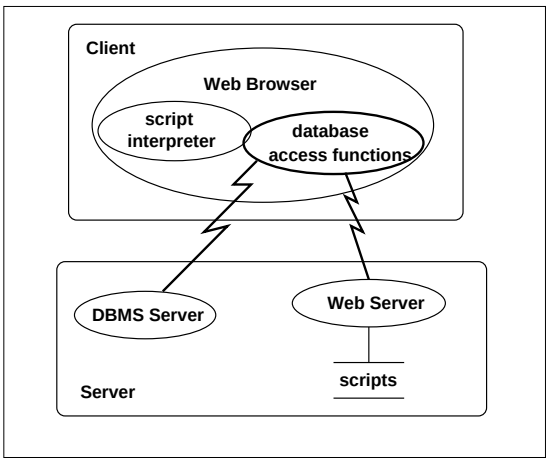
Οι αναδιφητές του Παγκόσμιου Ιστού διαρκώς επαυξάνουν τις λειτουργίες τους. Αυτές οι επεκτάσεις έχουν ως κύριο άξονα την παράκαμψη περιορισμών που επιβάλλει το πρότυπο HTML. Έτσι για παράδειγμα, οι αναδιφητές του Παγκόσμιου Ιστού υποστηρίζουν τη χρήση εφαρμογών Java, Java script κ.α.

Με την ίδια λογική μπορούν να επεκταθούν οι αναδιφητές του Παγκόσμιου Ιστού ώστε να υποστηρίζουν την άμεση σύνδεση με βάσεις δεδομένων.

Στην περίπτωση επέκτασης του αναδιφητή του Παγκόσμιου Ιστού δεν υπάρχει συγκεκριμένη αρχιτεκτονική ή κάποιο πρότυπο. Μια τυπική αρχιτεκτονική που εμπίπτει σε αυτήν την κατηγορία και θα παρουσιάσουμε στη συνέχεια, φαίνεται στο σχήμα 2.15.

2.9.1 Περιγραφή τυπικής αρχιτεκτονικής

Στην περίπτωση αυτή, επεκτείνεται ο αναδιφητής του Παγκόσμιου Ιστού να



Σχήμα 2.15: Η αρχιτεκτονική Browser Extension

υποστηρίζει μία επέκταση της HTML με ειδικές ετικέτες διαμόρφωσης που επιτρέπουν την απευθείας σύνδεση του εξυπηρετούμενου του Παγκόσμιου Ιστού με τον εξυπηρέτη της βάσης. Επιπλέον, η χρησιμοποίηση scripts μέσα στην σελίδα HTML είναι δυνατή. Η αρχιτεκτονική φαίνεται στο σχήμα 2.15.

Όπως έχει ήδη αναφερθεί, η χρησιμοποίηση ειδικών ετικετών διαμόρφωσης είναι πρόβλημα ορθογώνιο προς την αρχιτεκτονική υλοποίησης της πύλης διασύνδεσης. Η διαφορά στην αρχιτεκτονική αυτή είναι ότι ο αναδιφητής του Παγκόσμιου Ιστού επεκτείνεται και όχι ο εξυπηρέτης του Παγκόσμιου Ιστού.

Το μεγάλο μειονέκτημα αυτής της αρχιτεκτονικής είναι ότι δεν υπάρχει κάποιο πρότυπο σύμφωνα με το οποίο να γίνονται οι επεκτάσεις.

Ένα παράδειγμα προϊόντων που ακολουθούν αυτή την υλοποίηση είναι οι εφαρμογές Java που καλούνται ως applets και χρησιμοποιούν JDBC (Java DataBase Conectivity) για την επίτευξη σύνδεσης με σχεσιακές βάσεις δεδομένων.

2.10 Σύστημα Διαχείρισης Κόμβων του Παγκόσμιου Ιστού

Η τεράστια αύξηση της χρήσης του Παγκόσμιου Ιστού δημιούργησε ένα νέο πρόβλημα διαχείρησης δεδομένων, την κατασκευή και διατήριση κόμβων του Παγκόσμιου Ιστού (Web Sites). Η δημιουργία ενός κόμβου του Παγκόσμιου Ιστού περιλαμβάνει πολλές ενέργειες όπως η επιλογή της πληροφορίας που πρέπει να παρουσιασθεί, η οργάνωση της πληροφορίας σε ανεξάρτητες σελίδες ή σε γράφους συνδεδεμένων σελίδων και ο καθορισμός της παρουσίασης της κάθε σελίδας. Η δημιουργία και η διατήριση μεγάλου όγκου σελίδων HTML είναι δύσκολη και επίπονη διαδικασία επειδή πρέπει όλα τα παραπάνω να γίνονται ταυτόχρονα με τη δημιουργία των σελίδων HTML.

Η δημιουργία και η διατήρηση κόμβων του Παγκόσμιου Ιστού αποτελεί μια μοναδική ευκαιρία για την εφαρμογή εννοιών και τεχνικών από το πεδίο των συστημάτων βάσεων δεδομένων. Τέτοιες έννοιες είναι ο διαχωρισμός της λογικής παράστασης από τα δεδομένα, το πού βρίσκονται αποθηκευμένα τα δεδομένα και η δυνατότητα αναδημιουργίας των σελίδων HTML χρησιμοποιώντας επερωτήσεις.

Συστήματα που έχουν αναπτυχθεί και εφαρμόζουν τεχνικές των συστημάτων διαχείρισης βάσεων δεδομένων στον Παγκόσμιο Ιστό ονομάζονται *Συστήματα Διαχείρισης Κόμβων του Παγκόσμιου Ιστού - ΣΔΚΠΙ* (‘‘Web Site Management Systems’’ ή ‘‘Web-Base Management Systems’’). Μερικά αντιπροσωπευτικά συστήματα αυτής της κατηγορίας είναι τα Strudel([48],[49],[50]), Araneus([51],[52],[53],[54],[55]) και YATT([56]).

Στη συνέχεια θα περιγράψουμε το σύστημα Strudel.

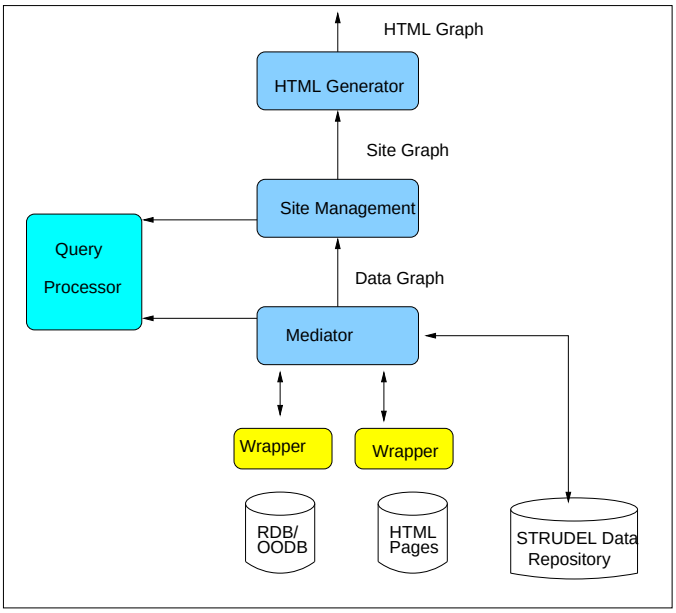
2.10.1 Strudel

Η βασική ιδέα στο σύστημα Strudel είναι ο σαφής διαχωρισμός της λογικής όψης της πληροφορίας, που είναι διαθέσιμη σε έναν κόμβο του Παγκόσμιου Ιστού, από τη δομή της πληροφορίας σε συνδεδεμένες σελίδες και τη γραφική παράσταση των σελίδων σε HTML.

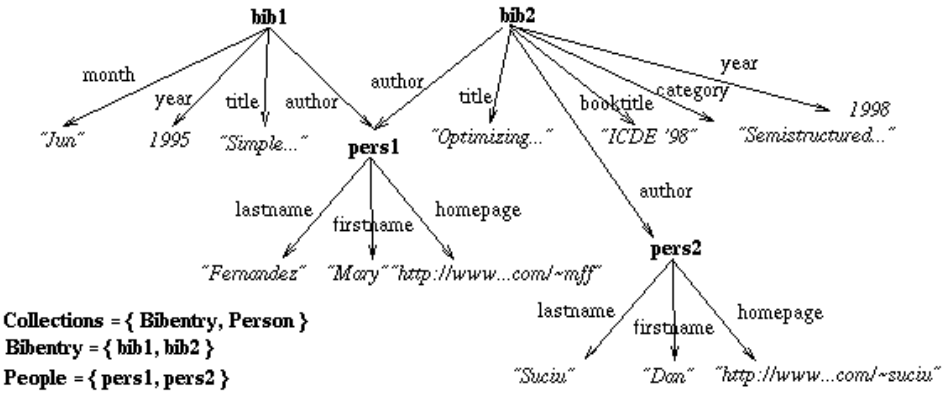
Η δημιουργία ενός κόμβου του Παγκόσμιου Ιστού με τη χρήση του Strudel απαιτεί δύο βήματα. Αρχικά, ο υπεύθυνος του κόμβου του Παγκόσμιου Ιστού καθορίζει τα δεδομένα που πρέπει να περιέχονται στις σελίδες. Στη συνέχεια καθορίζει πώς θα οργανωθούν και θα παρουσιασθούν τα δεδομένα που καθόρισε στο πρώτο βήμα. Δηλαδή, η δομή του κόμβου του Παγκόσμιου Ιστού που καθόρισε στο πρώτο βήμα. Η δομή του κόμβου του Παγκόσμιου Ιστού καθορίζεται ως μια όψη των δεδομένων. Επομένως, ορίζοντας διαφορετικές όψεις πάνω στα ίδια δεδομένα, μπορεί να παρουσιασθεί με διαφορετικούς τρόπους η ίδια πληροφορία.

Για την αποθήκευση των δεδομένων μπορούν να χρησιμοποιηθούν πολλά και διαφορετικά συστήματα, όπως βάσεις δεδομένων, κόμβοι του Παγκόσμιου Ιστού και

αρχεία συστήματος. Το Strudel έχει υλοποιήσει ένα μηχανισμό για την κατασκευή ενός γραφικού μοντέλου δεδομένων που περιέχει όλα τα δεδομένα από όλες τις πηγές. Αυτό επιτυγχάνεται με τη χρήση μεταφραστών (wrappers) και διαμεσολαβητών (mediators). Υπάρχει ένας μεταφραστής για κάθε είδος πηγής που χρησιμοποιείται. Ο κάθε μεταφραστής μεταφράζει το σχήμα των δεδομένων της πηγής για την οποία είναι υπεύθυνος, στο γραφικό μοντέλο δεδομένων του Strudel. Επιπλέον, μεταφράζει τις επερωτήσεις που είναι εκφρασμένες στη γλώσσα επερώτησης του Strudel σε επερωτήσεις ή λειτουργίες που είναι κατανοητές από τη πηγή.



Σχήμα 2.16: Η αρχιτεκτονική του συστήματος Strudel



Σχήμα 2.17: Παράδειγμα γράφου δεδομένων

Σε κάθε επίπεδο του συστήματος Strudel η όψη των δεδομένων είναι ομοιόμορφη ως ένας γράφος, σχήμα 2.16. Στο χαμηλότερο επίπεδο, τα δεδομένα είναι αποθηκευμένα στο σύστημα αποθήκευσης του Strudel ή σε εξωτερικές πηγές και το Strudel τα βλέπει ως γράφους δεδομένων. Δηλαδή ένας γράφος δεδομένων (Data Graph) ορίζεται ως μια όψη των δεδομένων που περιέχονται στις εξωτερικές πηγές. Το σύστημα Strudel υποστηρίζει πολλούς ατομικούς τύπους που εμφανίζονται πολύ συχνά σε ιστοσελίδες, όπως για παράδειγμα URL, κείμενο, εικόνα και αρχείο HTML. Τα αντικείμενα ομαδοποιούνται σε συλλογές. Ένα αντικείμενο μπορεί να ανήκει σε περισσότερες από μια συλλογές και αντικείμενα της ίδιας συλλογής μπορεί να έχουν διαφορετική παρουσίαση σε κείμενο HTML.

Η δομή του κόμβου του Παγκόσμιου Ιστού ορίζεται ως υπογράφος του γράφου δεδομένων και ονομάζεται γράφος ιστοκόμβου ή γράφος κόμβου του Παγκόσμιου Ιστού (Site Graph). Οι κόμβοι του γράφου ιστοκόμβου αντιστοιχούν σε δεδομένα ενώ οι ακμές σε συσχετίσεις δεδομένων. Ο κάθε κόμβος του γράφου ιστοκόμβου έχει ένα γνώρισμα που προσδιορίζει την παράστασή του σε κείμενο HTML.

Η διαχείριση των γράφων, δηλαδή ο ορισμός των γράφων ως όψεων σε εξωτερικές πηγές, ο ορισμός των υπογράφων ως όψεων ενός γράφου και η διάσχιση των γράφων δεδομένων, γίνεται μέσω μιας γλώσσας που ονομάζεται StruQL, που μοιάζει με το OEM[57]. Επιπλέον, η γλώσσα StruQL επιτρέπει τη χρήση κανονικών εκφράσεων μονοπατιών (regular path expresion) που καθορίζουν πώς θα ταιριάξουν (match) μονοπάτια οποιουδήποτε μήκους σε ένα γράφο δεδομένων.

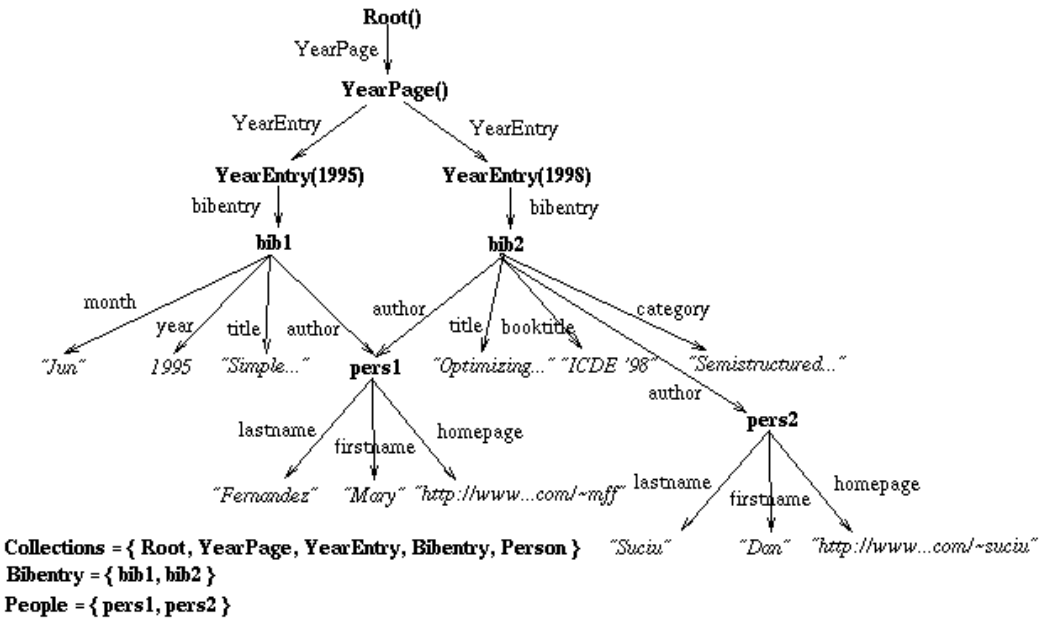
Η χρησιμοποίηση ενός γράφου δεδομένων για την παράσταση των δεδομένων σε συνδυασμό με τη χρήση κανονικών εκφράσεων μονοπατιών, επιτρέπει την επιβολή περιορισμών από το χρήστη του συστήματος, όπως για παράδειγμα να μην υπάρχουν υπερ-σύνδεσμοι που δεν δείχνουν πουθενά, η δομή του κόμβου του Παγκόσμιου Ιστού να ακολουθεί κάποιο συγκεκριμένο μορφότυπο κ.α.

2.10.1.1 Παράδειγμα Γράφου Δεδομένων

Ένα παράδειγμα ενός γράφου δεδομένων που αφορά βιβλιογραφικά δεδομένα διακρίνεται στο σχήμα 2.17. Στο παράδειγμα διακρίνονται δύο συλλογές όπου η μία (Bibentry) αφορά εγγραφές βιβλίων ενώ η δεύτερη (People) αφορά άτομα.

2.10.1.2 Παράδειγμα ορισμού ενός κόμβου του Παγκόσμιου Ιστού

Για τον ορισμό ενός κόμβου του Παγκόσμιου Ιστού, χρησιμοποιείται η γλώσσα επερώτησης StruQL. Δεδομένου ενός ή περισσότερων γράφων δεδομένων, μια επερώτηση εκφρασμένη στη γλώσσα επερώτησης StruQL εξάγει αντικείμενα, γνωρίσματα και τιμές



Σχήμα 2.18: Παράδειγμα γράφου ιστοκόμβων

από τους γράφους εισόδου και κατασκευάζει έναν νέο γράφο. Ο νέος γράφος είναι ένας γράφος δεδομένων.

Ένας γράφος δεδομένων ονομάζεται γράφος ιστοκόμβου ή γράφος κόμβου του Παγκόσμιου Ιστού (Site Graph), όταν το τμήμα του Strudel που παράγει την παρουσίαση σε κείμενο σύμφωνο με το πρότυπο HTML, έχοντας αυτόν το γράφο ως είσοδο, παράγει έναν κόμβο του Παγκόσμιου Ιστού.

Η ακόλουθη επερώτηση δημιουργεί ένα γράφο ιστοκόμβου που ομαδοποιεί τις βιβλιογραφικές εγγραφές ανάλογα με την τιμή του γνωρίσματος *year*. Επιλέγει μόνο τα αντικείμενα που το γνώρισμα *''publicationtype''* έχει τιμή *''inproceedings''*, *''article''* ή *''techreport''*.

```
1. collect WebPage{Root()}, YearPage(), Root()
2. { where Bibentry{x},
3.      x -> "publicationtype" -> v,
4.     v in { "inproceedings", "article", "techreport" },
5.     x -> "year" -> y
6.  { where l = "year"
7.   collect WebPage{YearPage()}, YearEntry(y)
8.   link  Root() -> "YearPage" -> YearPage(),
9.       YearPage() -> "YearEntry" -> YearEntry(y),
10.      YearEntry(y) -> "bibentry" -> x,
11.      YearEntry(y) -> "year" -> v
```

```
12. }
13. }
```

Έτσι, αν στον γράφο δεδομένων του σχήματος 2.17 εφαρμοστεί η παραπάνω επερώτηση θα προκύψει ο γράφος δεδομένων του σχήματος 2.18 που είναι και γράφος κόμβου.

Η χρήση κανονικών εκφράσεων μονοπατιών επιτρέπει τη διάσχιση μονοπατιών οποιουδήποτε μήκους του γράφου εισόδου. Στο παραπάνω παράδειγμα, στις γραμμές 3, 4 και 5 φαίνεται η χρήση κανονικών εκφράσεων. Στη γραμμή 3 προσδιορίζεται ότι το *x* που είναι τύπου *Bibentry* έχει ένα γνώρισμα με όνομα *''publicationtype''* το οποίο έχει ως προορισμό το *v*. Στη γραμμή 4 προσδιορίζεται ότι το *v* πρέπει να είναι ένα από τα αλφαριθμιτικά *''inproceedings''*, *''article''* και *''techreport''* ενώ στη γραμμή 5 δηλώνεται ότι το *x* πρέπει να έχει και ένα γνώρισμα με όνομα *''year''* που να έχει ως προορισμό το αντικείμενο *y*.

2.10.1.3 Παραγωγή παρουσίασης σε κείμενο HTML

Για τον καθορισμό της παρουσίασης σε κείμενο HTML ενός κόμβου του γράφου ιστοκόμβου, χρησιμοποιούνται σκελετοί HTML (templates). Ένας σκελετός HTML αναφέρεται σε έναν κόμβο του γράφου ιστοκόμβων. Αν σε ένα αντικείμενο δεν έχει προσδιορισθεί σκελετός HTML, αυτό κληρονομεί τον σκελετό HTML της συλλογής στην οποία ανήκει.

Για το γράφο ιστοκόμβου του σχήματος 2.18, ένας σκελετός HTML που θα μπορούσε να χρησιμοποιηθεί είναι :

```
1. <a name="<SFMT bibtexkey EMBED>">
2. <SFOR a IN author ORDER=ascend KEY=ordernumber>
3.   <SFMT @a EMBED>,
4. </SFOR>
5. <i><SIF postscript> <SFMT postscript LINK=title> </i>
6.   <SELSE><SFMT title>
7. </SIF></i>
9. <BR>
10. <SIF publicationtype = "inproceedings">
11.   In <SFMT booktitle EMBED>
12.   <SIF year>, <SFMT year EMBED> </SIF>
13. <SELSE>
14.   <SIF publicationtype = "article">
15.     Published in <SFMT journal EMBED>
16.     <SIF volume>, vol. <SFMT volume> </SIF>
```

```
17.     <SIF number>, no. <SFMT number> </SIF>
18.     <SIF pages>, pp. <SFMT pages> </SIF>
19.     <SIF month>, <SFMT month EMBED> </SIF>
20.     <SIF year>, <SFMT year EMBED> </SIF>
21.   <SELSE>
22.     <SIF publicationtype = "techreport">
23.      <SFMT institution EMBED> Technical Report, Number <SFMT number>
24.       <SIF year>, <SFMT year EMBED> </SIF>
25.     </SIF>
26.   </SIF>
27. </SIF>
28. <BR>
29. <SIF abstract>
30.   Click <SFMT abstract LINK="here"> for an abstract. <BR>
31. </SIF>
32. <BR>
```

Δεδομένου ενός αντικειμένου A και του παραπάνω σκελετού HTML, το τμήμα παραγωγής κειμένου HTML του Strudel, παράγει μια σελίδα HTML εξετάζοντας τις εκφράσεις του σκελετού και αντικαθιστώντας κάθε έκφραση γνωρίσματος με την αντίστοιχη τιμή. Μια έκφραση γνωρίσματος είναι είτε ένα γνώρισμα του αντικειμένου A είτε ένα γνώρισμα ενός αντικειμένου το οποίο συνδέεται με κάποιο μονοπάτι με το A.

Σε έναν σκελετό HTML μπορεί να υπάρχουν τμήματα που περικλύονται σε δηλώσεις ελέγχου, όπως στις γραμμές 14-26 του παραδείγματος όπου ανάλογα με τον τύπο του αντικειμένου χρησιμοποιείται το αντίστοιχο τμήμα του σκελετού.

2.11 Συμπεράσματα

Από μια πύλη διασύνδεσης με τον Παγκόσμιο Ιστό έχουμε τις εξής απαιτήσεις :

- Ευελιξία
  Οποιοδήποτε αντικείμενο της βάσης να μπορεί να παρουσιασθεί σε HTML και η παρουσίαση ενός αντικειμένου να μπορεί να διαφοροποιηθεί από την παρουσίαση της κλάσης στην οποία ανήκει, εν αντιθέσει, για παράδειγμα, με την O_2 όπου παρουσίαση δεν μπορεί να ορισθεί στα δεδομένα παρά μόνο στις κλάσεις.

  Επιπλέον, πρέπει να υποστηρίζεται η δήλωση διαφορετικών όψεων παρουσίασης σε HTML του ίδιου αντικειμένου.

- Επεκτασιμότητα
  Πρέπει να υποστηρίζεται η δήλωση διαφορετικών όψεων παρουσίασης του ίδιου

αντικειμένου χρησιμοποιώντας διαφορετικά είδη διαμόρφωσης κειμένου, όπως XML.

- Δηλωτικότητα
  Ο προγραμματιστής εφαρμογής να μην είναι υποχρεωμένος να χρησιμοποιήσει κάποια γλώσσα προγραμματισμού ή/και να γράψει κείμενο HTML για την παραγωγή του ζητούμενου αποτελέσματος. Η δηλωμένη από το χρήστη παρουσίαση πρέπει να είναι υψηλότερου επιπέδου.

Από τα προϊόντα που περιγράφηκαν κανένα δεν είναι δηλωτικό και επεκτάσιμο, ενώ όσον αφορά την ευελιξία, εκτός από το O_2Web, όλα τα προϊόντα υποστηρίζουν την παρουσίαση σε HTML οποιουδήποτε αντικειμένου της βάσης. Η ύπαρξη διαφορετικών όψεων παρουσίασης σε κείμενο HTML του ίδιου αντικειμένου, υποστηρίζεται μόνο από το προϊόν LINCKS.

Ιδιαίτερο ενδιαφέρον παρουσιάζουν τα συστήματα LINCKS και O_2Web λόγω της προσπάθειας μοντελοποίησης της παράστασης HTML και της υποστήριξης μηχανισμού εξισορόπησης φορτίου. Επιπλέον, το LINCKS είναι το μοναδικό από τα συστήματα που περιγράφηκαν που το μοντέλο αυτό βρίσκεται στη βάση και δεν χρησιμοποιούνται προγράμματα/φίλτρα μετατροπής. Όμως, το LINCKS είναι πολύ πρωτόγονο επειδή όλες οι ενέργειες που πρέπει να εφαρμοστούν για την μετάφραση του μοντέλου παράστασης σε HTML πρέπει να περιγραφούν πάρα πολύ αναλυτικά.

Επίσης, πρέπει να σημειωθεί η προσπάθεια συνδυασμού και άλλων ευρέως γνωστών και χρησιμοποιούμενων αρχιτεκτονικών όπως CORBA, για την κατασκευή της πύλης διασύνδεσης.

Με βάση την περιγραφή των προϊόντων που έχει προηγηθεί, καταλήγουμε στο συμπέρασμα οτι δεν υπάρχει προϊόν που να είναι πύλη διασύνδεσης και να ικανοποιεί όλες τις απαιτήσεις που θέσαμε. Γι'αυτόν τον λόγο σχεδιάσαμε και υλοποιήσαμε μια πύλη διασύνδεσης βάσεων δεδομένων με τον Παγκόσμιο Ιστό που ικανοποιεί τις απαιτήσεις που θέσαμε.

Με βάση τα όσα περιγράφηκαν είναι κατανοητό ότι η ταχύτητα μιας πύλης διασύνδεσης και οι λειτουργίες που μπορεί να προσφέρει, εξαρτώνται άμεσα από την αρχιτεκτονική σύμφωνα με την οποία έχει υλοποιηθεί. Από τις αρχιτεκτονικές που περιγράφηκαν και είναι οι CGI Executable, CGI Application Server, Server API, Proprietary Server, External Viewer και Browser Extension, ταχύτερες είναι οι αρχιτεκτονικές Server API και Proprietary Server με προτιμότερη την πρώτη όπου και γίνονται προσπάθειες ορισμού ενός προτύπου. Στην υλοποίηση του συστήματος που περιγράφουμε δεν προτιμήσαμε καμία από αυτές τις δύο. Η αρχιτεκτονική Server API απορρίφθηκε επειδή αυτή τη στιγμή δεν υπάρχει κάποιο πρότυπο για Server API και ο κάθε εξυπηρέτης του Παγκόσμιου Ιστού προσφέρει τη δική του υλοποίηση με αποτέλεσμα να μην υπάρχει συμβατότητα μεταξύ των διαφόρων υλοποιήσεων. Η αρχιτεκτονική Proprietary Server απορρίφθηκε επειδή δεν είναι πολύ ευέλικτη σε αλλαγές ενός εκ των πρωτοκόλλων του

Παγκόσμιου Ιστού ή προσθήκη κάποιου νέου, όπως για παράδειγμα υποστήριξη XML. Προτιμήθηκε η αρχιτεκτονική CGI Application Server επειδή σε μια μελλοντική χρησιμοποίηση της διεπιφάνειας προγραμματισμού εφαρμογών (API) ενός εξυπηρέτη του Παγκόσμιου Ιστού, χρειάζεται ένα τμήμα μόνο να αντικατασταθεί.

Κεφάλαιο 3

Μοντέλο Παρουσίασης Ιστοσελίδων

Σ' αυτό το κεφάλαιο αναπτύσσεται το μοντέλο παρουσίασης σε κείμενο HTML των περιεχομένων μιας Βάσης Γνώσης η οποία έχει κατασκευαστεί με το *Σύστημα Σημασιολογικού Ευρετηριασμού*. Το μοντέλο παρουσίασης έχει κατασκευαστεί χρησιμοποιώντας τη γλώσσα παράστασης γνώσης Telos και αποτελεί μέρος της Βάσης Γνώσης. Το μοντέλο χρησιμοποιείται από τον προγραμματιστή εφαρμογής και περιέχει όλη την απαραίτητη πληροφορία για την παρουσίαση των περιεχομένων μιας Βάσης Γνώσης σε HTML κείμενο. Η πύλη διασύνδεσης χρησιμοποιεί την πληροφορία του μοντέλου παρουσίασης για την παραγωγή της τελικής HTML σελίδας. Για την καλύτερη κατανόηση του μοντέλου παρουσίασης, χρησιμοποιείται ένα παράδειγμα, το οποίο εκτίθεται στην αρχή του κεφαλαίου. Πριν την περιγραφή του μοντέλου παρουσίασης είναι απαραίτητη μια σύντομη περιγραφή του Συστήματος Σημασιολογικού Ευρετηριασμού.

Στη συνέχεια του κεφαλαίου όπου γίνεται αναφορά σε Βάση Γνώσης, εννοούμε Βάση Γνώσης κατασκευασμένη με το Σύστημα Σημασιολογικού Ευρετηριασμού.

3.1 Παράδειγμα

Για την καλύτερη κατανόηση της δομής της γλώσσας Telos και του μοντέλου παρουσίασης, θα χρησιμοποιήσουμε το ακόλουθο παράδειγμα. Έστω ένα Ερευνητικό Ίδρυμα το οποίο διαιρείται σε Ινστιτούτα, όπως Πληροφορικής, Εφαρμοσμένων Μαθηματικών κ.α. Ένα Ινστιτούτο υποδιαιρείται σε τομείς. Έτσι για παράδειγμα το Ινστιτούτο Πληροφορικής διαιρείται στους τομείς Πληροφοριακών Συστημάτων και Τεχνολογίας Λογισμικού, Αρχιτεκτονικής, Παράλληλων και Κατανεμημένων Συστημάτων κ.α. Ένας τομέας περιλαμβάνει δραστηριότητες, όπως για παράδειγμα ο Τομέας Πληροφοριακών Συστημάτων και Τεχνολογίας Λογισμικού περιλαμβάνει τις δραστηριότητες Πολιτισμικής Τεκμηρίωσης, Συστημάτων Βάσεων Δεδομένων και Γνώσης και Συστήματος Σημασιολογικού Ευρετηριασμού. Σε κάθε τομέα υπάρχει ένας επικεφαλής του τομέα,

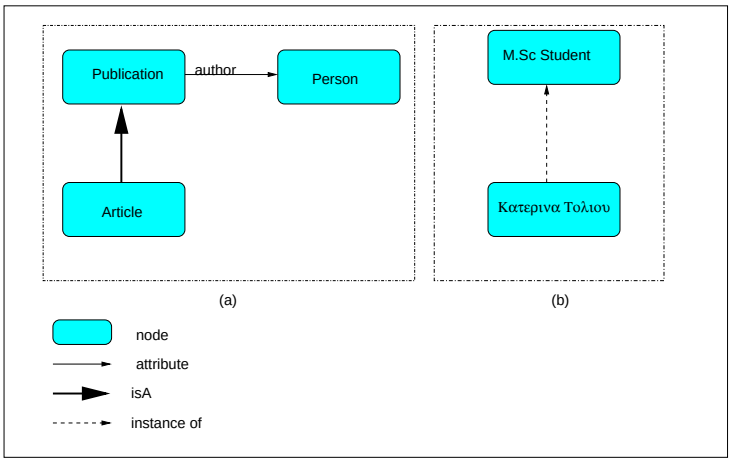
ερευνητές, εργαζόμενοι και φοιτητές. Στα πλαίσια ενός τομέα έχει αναπτυχθεί κάποιο λογισμικό, ο τομέας συμμετέχει σε κάποια έργα, έχει οργανώσει ή συμμετάσχει σε κάποιες εκδηλώσεις και έχει μια διεύθυνση. Τέλος, το προσωπικό ενός τομέα έχει κάνει κάποιες δημοσιεύσεις. Οι δημοσιεύσεις διακρίνονται σε δημοσιεύσεις σε επιστημονικά περιοδικά και συνέδρια, κεφάλαια από βιβλία, μεταπτυχιακές και διδακτορικές εργασίες, τεχνικές αναφορές και κείμενα τα οποία δεν έχουν δημοσιευθεί. Το κάθε έργο χρηματοδοτείται από κάποιον οργανισμό, συμμετέχουν οργανισμοί, αποτελεί τη συνέχεια κάποιου παλαιότερου έργου και μπορεί να χρησιμοποιεί συστήματα που αναπτύχθηκαν σε κάποιο προηγούμενο έργο.

3.2 Το Σύστημα Σημασιολογικού Ευρετηριασμού

Το Σύστημα Σημασιολογικού Ευρετηριασμού (Semantic Index System)[16] (ΣΣΕ - SIS) είναι ένα εργαλείο περιγραφής και τεκμηρίωσης στενά συνδεδεμένων μεταξύ τους δεδομένων, εννοιών και πολύπλοκων σχέσεων. Η περιγραφή αυτή γίνεται χρησιμοποιώντας τη γλώσσα παράστασης γνώσης Telos[17]. Το ΣΣΕ αποτελείται από έναν μηχανισμό μόνιμης αποθήκευσης δεδομένων που βασίζεται σε ένα μοντέλο δεδομένων που είναι ένα οντοκεντρικό σημασιολογικό δίκτυο. Η εισαγωγή των δεδομένων γίνεται με τη βοήθεια του μεταφραστή της γλώσσας Telos ή του Δελτίου Εισαγωγής Δεδομένων. Και τα δύο αυτά εργαλεία επικοινωνούν με το σύστημα σημασιολογικού ελέγχου, το οποίο δίνει τη δυνατότητα ενημερώσεων. Για την άντληση της πληροφορίας το ΣΣΕ παρέχει μια διεπιφάνεια προγραμματισμού εφαρμογών (API) [58]. Τόσο το API όσο και το σύστημα σημασιολογικού ελέγχου επικοινωνούν με το σύστημα διαχείρισης μνήμης του ΣΣΕ και αυτό με τη βάση. Το περιβάλλον συμπληρώνεται με ένα σύστημα διεπαφής χρήσης που ονομάζεται Γραφικός Αναλυτής. Το εργαλείο αυτό χρησιμεύει στην παρουσίαση της πληροφορίας τόσο σε γραφική μορφή όσο και σε μορφή κειμένου.

3.2.1 Η γλώσσα παράστασης γνώσης Telos

Η Telos είναι μια γλώσσα παράστασης γνώσης που υποστηρίζει ένα οντοκεντρικό μοντέλο δεδομένων. Οι μηχανισμοί που προσφέρει η γλώσσα είναι ανάλογοι των μηχανισμών που υποστηρίζουν τα σημασιολογικά δίκτυα. Οι στοιχειώδεις δομές είναι οντότητες και σχέσεις μεταξύ οντοτήτων που μπορούν να συνδυαστούν για την παράσταση πολύπλοκων δομών. Οι οντότητες μπορούν να είναι είτε ατομικές είτε κλάσεις οντοτήτων. Οι βασικοί μηχανισμοί δόμησης είναι οι σχέσεις ταξινόμησης, γενίκευσης και απόδοσης γνωρίσματος. Οι ιεραρχίες ταξινόμησης και εξειδίκευσης μπορεί να είναι απεριόριστες και πολλαπλές.

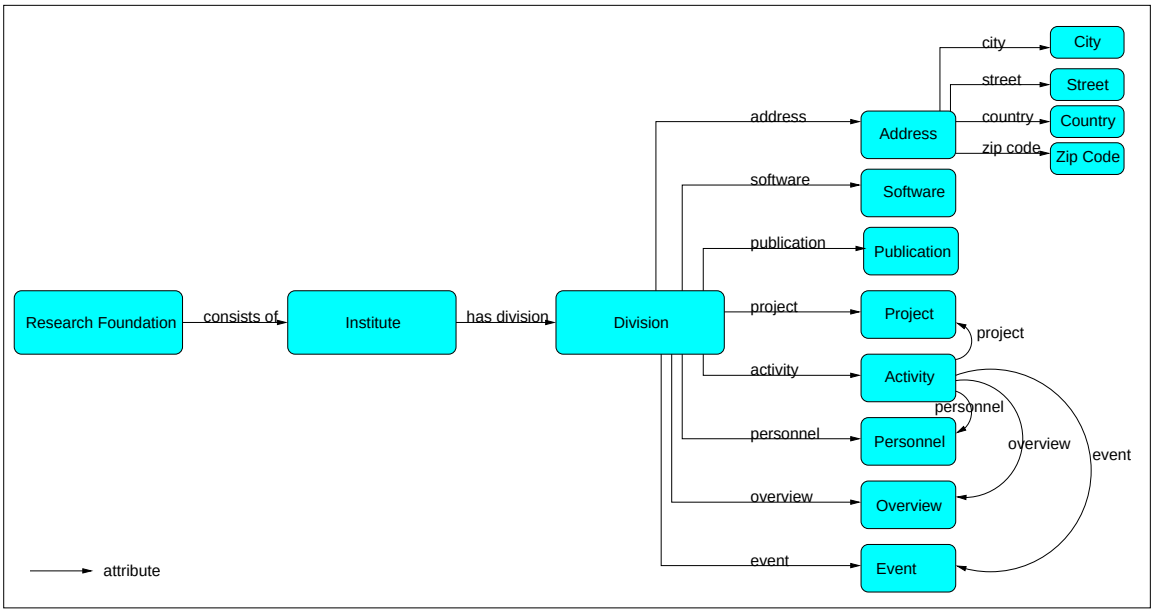


Σχήμα 3.1: Παράδειγμα κόμβων, γνωρίσματος και σχέσεων ταξινόμησης και γενίκευσης

3.2.1.1 Η δομή της γλώσσας

Το σημασιολογικό δίκτυο που κατασκευάζεται χρησιμοποιώντας τη γλώσσα Telos αποτελείται από οντότητες και σχέσεις μεταξύ οντοτήτων. Οι οντότητες διακρίνονται σε *ανεξάρτητες* και *εξαρτημένες*. Μια οντότητα ονομάζεται *ανεξάρτητη* όταν η ύπαρξή της δεν εξαρτάται από την ύπαρξη άλλων οντοτήτων. Μια οντότητα λέγεται *εξαρτημένη* όταν η ύπαρξή της εξαρτάται από την ύπαρξη άλλων οντοτήτων. Στην Telos μια οντότητα είναι εξαρτημένη όταν παριστάνει σχέση μεταξύ οντοτήτων ή ιδιότητα οντότητας. Οι ανεξάρτητες οντότητες ονομάζονται *κόμβοι* ενώ οι εξαρτημένες ονομάζονται *γνωρίσματα*. Ένα γνώρισμα έχει μια αφετηρία που μπορεί να είναι κόμβος ή γνώρισμα και ονομάζεται *οντότητα αφετηρίας* και έναν προορισμό που είναι κόμβος και ονομάζεται *κόμβος προορισμού*. Υπάρχουν σχέσεις μεταξύ οντοτήτων που δεν αποτελούν γνωρίσματα και είναι οι σχέσεις ταξινόμησης και γενίκευσης. Έτσι, χρησιμοποιώντας τις έννοιες του παραδείγματος που περιγράφηκε στην αρχή του κεφαλαίου, οι έννοιες *Δημοσίευση*, *ρθρο* και *Πρόσωπο* θα παρασταθούν ως κόμβοι, αφού είναι ανεξάρτητες οντότητες. Αντίθετα, η σχέση ‘‘η δημοσίευση έχει συγγραφέα’’, παριστάνεται ως γνώρισμα με οντότητα αφετηρίας τον κόμβο *Δημοσίευση* και οντότητα προορισμού τον κόμβο *Πρόσωπο* (σχήμα 3.2).

Στα οντοκεντρικά μοντέλα παράστασης δεδομένων οι κλάσεις οντοτήτων χρησιμοποιούνται για την ομαδοποίηση όλων των οντοτήτων με κοινά χαρακτηριστικά. Έτσι οι κλάσεις αποτελούν αφηρημένους τύπους δεδομένων και χρησιμοποιούνται για τη δημιουργία της *σχέσης ταξινόμησης*. Μία οντότητα που ταξινομείται σε μια κλάση οντοτήτων ονομάζεται *περίπτωση* ή *μέλος* της κλάσης. Μια κλάση οντοτήτων είναι μια οντότητα η οποία μπορεί να ταξινομηθεί σε μια άλλη γενικότερη κλάση οντοτήτων. Έτσι, δημιουργείται μια μη φραγμένη ιεραρχία από κλάσεις. Το πρώτο επίπεδο αυτής της ιεραρχίας αποτελούν οι *ατομικές οντότητες* (Tokens), το επόμενο επίπεδο οι *κλάσεις*

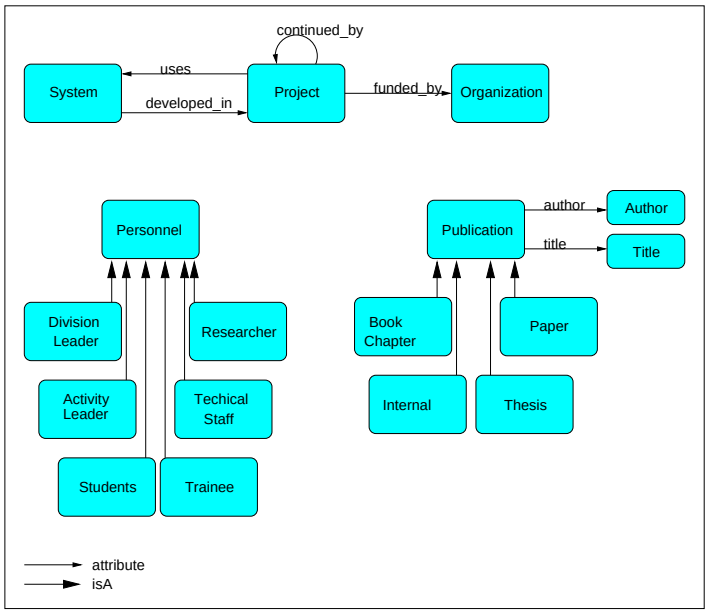


Σχήμα 3.2: Ένα Ερευνητικό Ίδρυμα αποτελείται από Ινστιτούτα και αυτά από Δραστηριότητες

οντοτήτων, που περιλαμβάνουν ατομικές οντότητες, το επόμενο επίπεδο οι *μετα-κλάσεις*, που περιλαμβάνουν κλάσεις οντοτήτων κ.ο.κ. Μια οντότητα μπορεί να ανήκει σε περισσότερες κλάσεις (περίπτωση πολλαπλής ταξινόμησης) και χρησιμοποιεί τα γνωρίσματα των κλάσεων στις οποίες ανήκει ως κλάσεις των δικών της γνωρισμάτων. Χρησιμοποιώντας το παράδειγμα, η Κατερίνα Τόλιου είναι μεταπτυχιακή φοιτήτρια. Επομένως, ο κόμβος *Κατερίνα Τόλιου* ταξινομείται στον κόμβο *M.Sc Student* ή διαφορετικά ο κόμβος *Κατερίνα Τόλιου* είναι περίπτωση του κόμβου *M.Sc Student*, (σχήμα 3.1 (b)).

Η σχέση *γενίκευσης* (αντίστροφα σχέση εξειδίκευσης) ορίζει μια σχέση υπερσυνόλου (υποσυνόλου) μεταξύ κλάσεων οντοτήτων. Έτσι, κλάσεις οντοτήτων μπορούν να ομαδοποιηθούν σε μια νέα κλάση, η οποία αποτελεί τη *γενίκευση* των κλάσεων αυτών, ενώ κάθε μια από αυτές τις κλάσεις αποτελεί *εξειδίκευση* της νέας αυτής κλάσης. Μια οντότητα μπορεί να ανήκει σε περισσότερες από μια υπερ-κλάσεις (πολλαπλή εξειδίκευση). Μια οντότητα κληρονομεί όλα τα γνωρίσματα των υπερ-κλάσεών της (αυστηρή κληρονόμηση). Εξετάζοντας το παράδειγμα, παρατηρούμε ότι ο κόμβος *Δημοσίευση* που αντιστοιχεί στην έννοια δημοσίευση του παραδείγματος, έχει ως εξειδίκευση τον κόμβο *Τεχνικές Αναφορές* και αντίστροφα ο κόμβος *Τεχνικές Αναφορές* έχει ως γενίκευση τον κόμβο *Δημοσίευση* (σχήμα 3.1 (a)). Στο κόμβο *Δημοσίευση* έχουν αποδοθεί τα γνωρίσματα *συγγραφέας* και *τίτλος*. Ο κόμβος *Τεχνικές Αναφορές* που είναι εξειδίκευση του *Δημοσίευση* κληρονομεί και τα δύο αυτά γνωρίσματα.

Ένα σύνολο από κλάσεις αποτελούν τον αρχικό πληθυσμό κάθε βάσης. Αυτές είναι κλάσεις συστήματος που αποτελούνται από τα επίπεδα ταξινόμησης (Token,



Σχήμα 3.3: Έργα, Δημοσιεύσεις και Προσωπικό

S_Class, M1_Class κ.ο.κ), τους πρωτογενείς τύπους (Telos_Integer, Telos_Real, Telos_String και Telos_Time) και τις γενικές κλάσεις οντοτήτων (Individual και Attribute). Στη συνέχεια όταν γίνεται αναφορά στον όρο κλάση αναφερόμαστε στις οριζόμενες από το χρήστη κλάσεις και όχι στις κλάσεις συστήματος.

Για τα γνωρίσματα το επίπεδο ταξινόμησης στο οποίο ανήκουν προσδιορίζεται να είναι χαμηλότερο ή ίσο με το ελάχιστο επίπεδο ταξινόμησης μεταξύ των επιπέδων ταξινόμησης της οντότητας αφετηρίας και της οντότητας προορισμού. Αν σε ένα γνώρισμα δεν προσδιορισθεί ρητά το επίπεδο ταξινόμησης στο οποίο ανήκει, θεωρείται ίσο με το μικρότερο επίπεδο ταξινόμησης των κόμβων αφετηρίας και προορισμού. Στο παραπάνω παράδειγμα όλοι οι κόμβοι βρίσκονται στο S_Class επίπεδο ταξινόμησης και επομένως αν δεν προσδιορισθεί το επίπεδο ταξινόμησης κάποιου γνωρίσματος, τότε το γνώρισμα βρίσκεται σε S_Class επίπεδο ταξινόμησης.

Τα γνωρίσματα μπορούν να έχουν ως αφετηρία μια οντότητα, κόμβο ή γνώρισμα, και ως προορισμό μόνο έναν κόμβο ή έναν από τους πρωτογενείς τύπους της Telos. Οι περιπτώσεις των πρωτογενών τύπων δεν μπορούν να δημιουργηθούν αλλά θεωρούμε ότι προϋπάρχουν.

Κάθε οντότητα χαρακτηρίζεται κατά τη δημιουργία της ως κόμβος (Individual) ή γνώρισμα (Attribute) και ο χαρακτηρισμός αυτός δεν μπορεί να αλλάξει καθόλη τη διάρκεια ζωής αυτής της οντότητας στο σύστημα. Μία οντότητα συνδέεται υποχρεωτικά και μοναδικά με ένα εσωτερικό αναγνωριστικό (Object Identifier - oid). Για τους κόμβους είναι υποχρεωτική η σύνδεσή τους και με ένα εξωτερικό λογικό αναγνωριστικό που δίνεται από το χρήστη του συστήματος ενώ για τα γνωρίσματα δεν είναι υποχρεωτικό.

Ένα άλλο αναλλοίωτο χαρακτηριστικό μιας οντότητας είναι το επίπεδο ταξινόμησης. Θεωρητικά υπάρχουν άπειρα επίπεδα ταξινόμησης με τις ατομικές οντότητες να βρίσκονται στο επίπεδο Token, τις κλάσεις στο επίπεδο S_Class, τις μετα-κλάσεις στο επίπεδο M1_Class κ.ο.κ.

Με βάση τις έννοιες που περιγράφησαν σε αυτήν την ενότητα, οι οντότητες που αντιστοιχούν στο παράδειγμα και οι μεταξύ τους σχέσεις, διακρίνονται στα σχήματα 3.2 και 3.3.

3.2.1.2 Δομικοί περιορισμοί

Στις βάσεις γνώσεων που υλοποιούνται με το μοντέλο δεδομένων της Telos έχει εφαρμοστεί ένα σύνολο δομικών περιορισμών. Οι περιορισμοί αυτοί είναι :

1. Το λογικό όνομα των κόμβων είναι υποχρεωτικό και μοναδικό στη βάση.

2. Το λογικό όνομα των γνωρισμάτων είναι προαιρετικό αλλά μοναδικό στην εμβέλεια της οντότητας αφετηρίας του. Δηλαδή, δυο γνωρίσματα που έχουν κοινή οντότητα αφετηρίας δεν μπορούν να έχουν το ίδιο λογικό όνομα. Αν ένα γνώρισμα δεν έχει όνομα τότε πρέπει η οντότητα προορισμού του να είναι διαφορετική από την οντότητα προορισμού οποιουδήποτε άλλου γνωρίσματος χωρίς λογικό όνομα που έχει κοινή με αυτό οντότητα αφετηρίας.

3. Μια οντότητα μπορεί να ταξινομηθεί σε μία άλλη, μόνο αν είναι του ίδιου τύπου (Individual ή Attribute) και η πρώτη βρίσκεται ένα επίπεδο ταξινόμησης χαμηλότερο από τη δεύτερη.

4. Μια οντότητα μπορεί να γίνει γενίκευση μιας άλλης, μόνο αν είναι του ίδιου τύπου και βρίσκονται στο ίδιο επίπεδο ταξινόμησης. Σχέσεις ταξινόμησης δεν μπορούν να υπάρξουν μεταξύ ατομικών οντοτήτων. Τέλος, δεν επιτρέπεται να δημιουργηθεί κύκλος από σχέσεις γενίκευσης.

5. Μια οντότητα μπορεί να διαγραφεί μόνο αν δεν είναι αφετηρία ή προορισμός γνωρισμάτων και δεν έχει περιπτώσεις.

6. Για την ταξινόμηση ενός γνωρίσματος σε μια κλάση γνωρισμάτων C πρέπει η οντότητα αφετηρίας του γνωρίσματος να είναι περίπτωση της οντότητας αφετηρίας της C και η οντότητα προορισμού του γνωρίσματος να είναι περίπτωση της οντότητας προορισμού της C. Ανάλογος κανόνας ισχύει στην περίπτωση γενίκευσης μιας κλάσης γνωρισμάτων.

3.2.1.3 Μηχανισμοί Διαχείρισης του ΣΣΕ

Το Σύστημα Σημασιολογικού Ευρετηριασμού παρέχει ένα σύνολο μηχανισμών για την καλύτερη διαχείριση του. Οι μηχανισμοί είναι :

- **Μηχανισμός επερωτήσεων**
  Για την άντληση της πληροφορίας από μια βάση το ΣΣΕ παρέχει δύο βασικούς μηχανισμούς :

  ▷ Την *Ερωτηματική Διεπιφάνεια Χρήσης* (Query Interface)[59] για τη διαλογική άντληση της πληροφορίας

  ▷ Την *Διεπιφάνεια Προγραμματισμού Εφαρμογών* (API)[58] για την άντληση πληροφορίας μέσω εξωτερικών εφαρμογών.

- **Μηχανισμός Εισαγωγής Δεδομένων**
  Η εισαγωγή δεδομένων στο *ΣΣΕ* γίνεται με δύο τρόπους:

  ▷ Μαζική εισαγωγή δεδομένων με τη βοήθεια του μεταφραστή της Telos.

  ▷ Διαλογική εισαγωγή δεδομένων με χρήση των Δελτίων Εισαγωγής Δεδομένων.

Οι ερωτηματικές συναρτήσεις που προσφέρει το API κατατάσσονται ανάλογα με τη λειτουργικότητά τους σε πέντε κατηγορίες:

- Συναρτήσεις για την έναρξη και λήξη της επικοινωνίας μιας εξωτερικής εφαρμογής με το ΣΣΕ.
- Συναρτήσεις για την αποστολή επερώτησης στο ΣΣΕ.
- Συναρτήσεις για την τροποποίηση γενικών παραμέτρων που επηρεάζουν τις συναρτήσεις αποστολής επερώτησης.
- Συναρτήσεις χειρισμού συνόλων αποθήκευσης
- Συναρτήσεις ανάκτησης πληροφορίας από τα σύνολα αποθήκευσης.

3.3 Μοντέλο Παρουσίασης

Στην υλοποίηση που περιγράφουμε, όπως έχει ήδη αναφερθεί, όλη η απαραίτητη πληροφορία για την παρουσίαση των αντικειμένων της βάσης βρίσκεται στο μοντέλο παρουσίασης. Η πύλη διασύνδεσης βασιζόμενη σε αυτήν την πληροφορία παράγει το τελικό HTML κείμενο.

Στο τέλος του δεύτερου κεφαλαίου τέθηκαν τρείς συγκεκριμένες απαιτήσεις που είναι πολύ σημαντικό να ικανοποιούνται από μια πύλη διασύνδεσης βάσεων δεδομένων με τον παγκόσμιο ιστό. Σύμφωνα με την πρώτη για να είναι το σύστημα ευέλικτο, πρέπει να μπορεί να προσδιορισθεί παρουσίαση σε οποιοδήποτε αντικείμενο της βάσης, ανεξάρτητα από το επίπεδο ταξινόμησης στο οποίο βρίσκεται. Επιπλέον, πρέπει να υποστηρίζονται διαφορετικές παρουσιάσεις ενός αντικειμένου σε HTML. Σύμφωνα με

τη δεύτερη απαίτηση, για να είναι το σύστημα επεκτάσιμο πρέπει να υποστηρίζεται η παρουσίαση ενός αντικειμένου χρησιμοποιώντας διαφορετικά είδη διαμόρφωσης κειμένου, όπως XML, Latex και απλό κείμενο (ASCII). Για να είναι το σύστημα δηλωτικό, που είναι και η τρίτη απαίτηση, πρέπει η παρουσίαση των δεδομένων να προσδιορίζεται από τον προγραμματιστή εφαρμογής με δηλωτικό τρόπο.

Για να είναι δυνατή η παρουσίαση οποιουδήποτε αντικειμένου της βάσης, είτε πρόκειται για γνώρισμα είτε για κόμβο και ανεξάρτητα από το επίπεδο ταξινόμησης στο οποίο βρίσκεται, πρέπει αυτή να είναι ανεξάρτητη του κόσμου που παριστούν τα αντικείμενα της βάσης.

Για την υποστήριξη πολλαπλών παρουσιάσεων ενός αντικειμένου σε HTML χρησιμοποιούνται όψεις πάνω σε μια βάση SIS-Telos. Μια παρουσίαση ορίζεται σε σχέση με μια όψη.

Για να είναι δυνατή η υποστήριξη πολλών ειδών διαμόρφωσης κειμένου διαχωρίζεται ο προσδιορισμός του είδους διαμόρφωσης κειμένου από τον προσδιορισμό της δομής και του περιεχομένου της παρουσίασης, που στο υπόλοιπο κεφάλαιο θα ονομάζουμε λογική παράσταση. Μια λογική παράσταση αντιστοιχίζεται σε κάποιο είδος διαμόρφωσης κειμένου, προσδιορίζοντας με αυτόν τον τρόπο την τελική παρουσίαση.

Σχετικά με την δηλωτικότητα του μοντέλου, ο προγραμματιστής εφαρμογής περιγράφει τη δομή και το περιεχόμενο της παρουσίασης με δηλωτικό τρόπο, αποφεύγοντας τη χρησιμοποίηση κάποιας γλώσσας προγραμματισμού ή κάποιου συγκεκριμένου είδους διαμόρφωσης κειμένου.

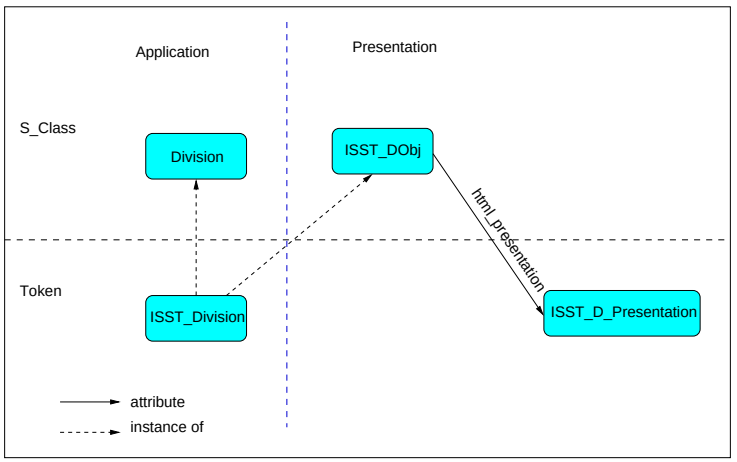
Επομένως, το μοντέλο παρουσίασης πρέπει να περιλαμβάνει τρείς βασικές δομές. Οι δομές αυτές είναι η λογική, που περιγράφει τη λογική δομή και το περιεχόμενο της παρουσίασης, η σημασιολογική, που περιγράφει την όψη, ή αλλιώς την εφαρμογή, σύμφωνα με την οποία θα γίνει η παρουσίαση, και η φυσική, που αντιστοιχίζεται στο είδος διαμόρφωσης κειμένου που θα χρησιμοποιηθεί.

3.3.1 Προσδιορισμός της παρουσίασης ενός αντικειμένου της βάσης

Ένα αντικείμενο τη βάσης συνδέεται με τις υπόλοιπες οντότητες της βάσης μέσω σχέσεων ταξινόμησης, γενίκευσης και απόδοσης γνωρίσματος. Οι οντότητες μιας βάσης συνιστούν μια ή περισσότερες εφαρμογές. Η παρουσίαση ενός αντικειμένου είναι ανεξάρτητη της εφαρμογής στην οποία βρίσκεται. Συνεπώς, η παρουσίαση δεν πρέπει να επηρεάζει την εφαρμογή της βάσης.

Για τη συσχέτιση του μοντέλου παρουσίασης με τα αντικείμενα της βάσης πρέπει να λάβουμε υπόψη ότι η παρουσίαση ενός αντικειμένου σύμφωνα με κάποιο είδος διαμόρφωσης κειμένου, αποτελεί μία διαφορετική ερμηνεία/θεώρηση του αντικειμένου. Δεν είναι κάποιο χαρακτηριστικό, οπότε η συσχέτιση να γίνει χρησιμοποιώντας σχέση απόδοσης γνωρίσματος, ούτε προσδιορίζει σχέση συνόλου - υποσυνόλου ώστε η συσχέτιση

να γίνει χρησιμοποιώντας σχέση γενίκευσης. Αντίθετα, η σχέση ταξινόμησης είναι κατάλληλη για να αποδώσει ερμηνείες ή θεωρήσεις σύμφωνα με τις οποίες περιγράφονται τα αντικείμενα. Για να γίνει περισσότερο κατανοητός ο ρόλος της σχέσης ταξινόμησης, ας εξετάσουμε ένα παράδειγμα. Έστω το φυσικό πρόσωπο με όνομα Κατερίνα Τόλιου, που στη βάση παριστάνεται από το αντικείμενο *Κατερίνα Τόλιου*. Το αντικείμενο *Κατερίνα Τόλιου* ταξινομείται στην κλάση *Φυσικό Πρόσωπο* επειδή αυτός είναι ο τύπος του. Το φυσικό πρόσωπο Κατερίνα Τόλιου είναι και μεταπτυχιακή φοιτήτρια, αν το εξετάσουμε σε σχέση με την επαγγελματική ιδιότητα που έχει, που είναι μια διαφορετική θεώρηση. Επομένως το αντικείμενο *Κατερινα Τόλιου* ταξινομείται και στην κλάση *Μεταπτυχιακός Φοιτητής*.



Σχήμα 3.4: Παράδειγμα προσδιορισμού παρουσίασης

Επιπλέον, η παρουσίαση ενός αντικειμένου είναι χαρακτηριστικό της κλάσης στην οποία είναι μέλος. Όπως έχει ειπωθεί, μια κλάση αντικειμένων χρησιμοποιείται για την ομαδοποίηση αντικειμένων με κοινά χαρακτηριστικά. Έτσι, αντικείμενα που έχουν την ίδια παρουσίαση πρέπει να είναι μέλη της ίδιας κλάσης.

Κατ' αναλογία κρίνουμε ότι πρέπει να χρησιμοποιηθεί η σχέση ταξινόμησης για τον προσδιορισμό της παρουσίασης ενός αντικειμένου.

Σύμφωνα με αυτήν την προσέγγιση, η παρουσίαση ενός αντικειμένου, έστω X, δεν αποδίδεται στο ίδιο το αντικείμενο, αλλά σε μία κλάση στην οποία το αντικείμενο X ταξινομείται. Άρα, αν θέλουμε να συσχετίσουμε μια παρουσίαση με ένα αντικείμενο X τότε το X θα ταξινομηθεί σε μια κλάση, έστω Y, και στην Y θα αποδοθεί η παρουσίαση μέσω γνωρίσματος. Η παρουσίαση θα βρίσκεται στο χαμηλότερο επίπεδο ταξινόμησης, Token επίπεδο ταξινόμησης, ανεξάρτητα από το επίπεδο ταξινόμησης του αντικειμένου X. Η κλάση Y δεν αποτελεί μέρος της εφαρμογής στην οποία βρίσκεται το X αλλά δημιουργείται για τον προσδιορισμό της παρουσίασης του αντικειμένου X.

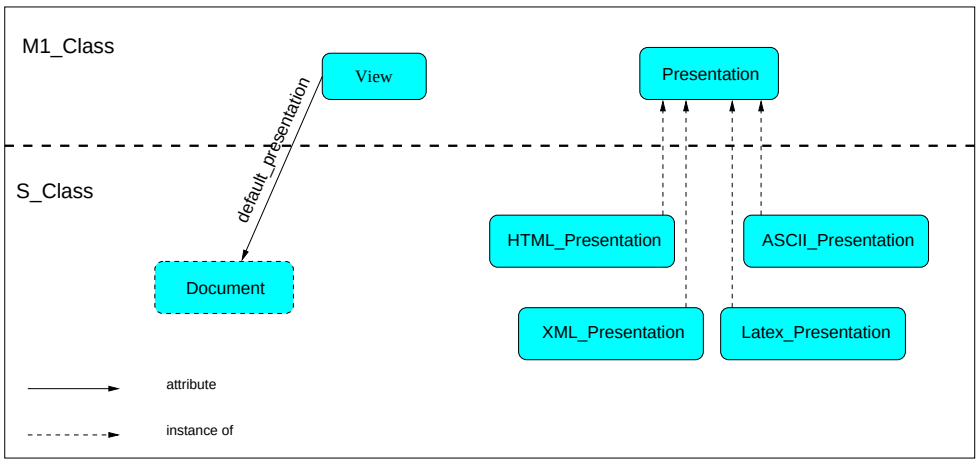
Θα προσπαθήσουμε να προσδιορίσουμε την παρουσίαση ενός αντικειμένου, χρη-

σιμοποιώντας το παράδειγμα του σχήματος 3.4. Έστω το αντικείμενο *ISST_Division* το οποίο είναι περίπτωση της κλάσης *Division* και παριστάνει τον Τομέα Πληροφοριακών Συστημάτων και Τεχνολογίας Λογισμικού του ΙΠ-ΙΤΕ. Για την συσχέτιση του μοντέλου παράστασης με το αντικείμενο *ISST_Division* δημιουργούμε ένα ενδιάμεσο αντικείμενο, που το ονομάζουμε *ISST_DObj* σε ένα επίπεδο ταξινόμησης υψηλότερο από αυτό που βρίσκεται το *ISST_Division*. Επομένως, η κλάση *ISST_DObj* βρίσκεται σε S_Class επίπεδο ταξινόμησης. Στη κλάση *ISST_DObj* αποδίδουμε ένα γνώρισμα της κλάσης γνωρισμάτων *html_presentation* που έχει ως προορισμό το αντικείμενο *ISST_D_Presentation* που είναι και το μοντέλο παρουσίασης του *ISST_Division*. Η δομή του *ISST_D_Presentation* θα περιγραφεί στη συνέχεια.

Ο μηχανισμός προσδιορισμού παρουσίασης είναι αρκετά γενικός ώστε να επιτρέπει τον προσδιορισμό παρουσίασης σε οποιαδήποτε οντότητα της βάσης, κόμβο ή γνώρισμα, και σε όλα τα επίπεδα ταξινόμησης. Αυτό ισχύει επειδή οι σχέσεις ταξινόμησης και απόδοσης γνωρίσματος μπορούν να εφαρμοσθούν με τον ίδιο ακριβώς τρόπο σε όλες τις οντότητες μιας βάσης που έχει κατασκευασθεί με το ΣΣΕ και σε όλα τα επίπεδα ταξινόμησης.

Στη συνέχεια θα εξετάσουμε τις τρείς δομές που συνιστούν το μοντέλο παρουσίασης.

3.3.2 Σημασιολογική δομή



Σχήμα 3.5: Οι κλάσεις αντικειμένων *View* και *Presentation*

Κάθε αντικείμενο βρίσκεται μέσα σε έναν κόσμο που προσδιορίζεται από τη βάση. Η παρουσίαση ενός αντικειμένου ορίζεται ως μια όψη (view) του κόσμου του αντικειμένου.

Η σημασιολογική δομή περιγράφει τη σημασιολογία της παρουσίασης και προσδιορίζεται από δύο παραμέτρους : την εφαρμογή μέρος της οποίας είναι το αντικείμενο που πρόκειται να παρουσιασθεί και την όψη, ή εφαρμογή, στα πλαίσια της οποίας θα γίνει η παρουσίαση.

Η παρουσίαση ενός αντικειμένου ουσιαστικά αποτελεί μια όψη του αντικειμένου σύμφωνα με κάποια κριτήρια παρουσίασης, όπως για παράδειγμα αν πρόκειται για έμπειρο ή άπειρο χρήστη. Για ένα αντικείμενο μπορούν να ορισθούν περισσότερες από μια όψεις ανεξάρτητα από το είδος διαμόρφωσης κειμένου που χρησιμοποιείται. Επιπλέον, σύμφωνα με μια όψη μπορούν να παρουσιασθούν περισσότερα του ενός αντικείμενα. Για την περιγραφή μιας όψης, ορίσαμε την κλάση *View* που βρίσκεται σε M1_Class επίπεδο ταξινόμησης, σχήμα 3.5.

Σε κάθε όψη μπορεί να αντιστοιχισθεί μια τυπική παρουσίαση των αντικειμένων, η οποία ακολουθείται όταν για κάποιο αντικείμενο δεν έχει δηλωθεί παρουσίαση. Η παραπάνω σχέση αποδίδεται μέσω του γνωρίσματος *default_presentation* που έχει ως αφετηρία την κλάση *View* και ως προορισμό την κλάση *Document*, σχήμα 3.5. Η κλάση *Document* περιγράφει το λογικό τμήμα της παρουσίασης και θα περιγραφεί στη συνέχεια.

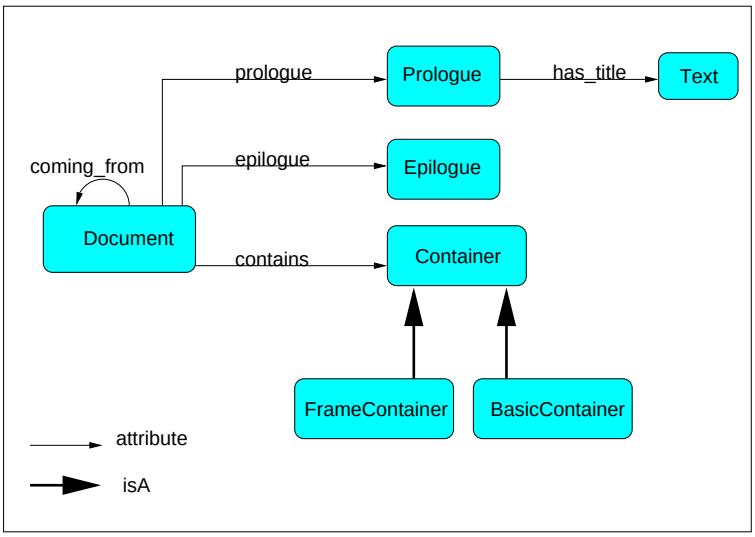
3.3.3 Φυσική δομή

Η φυσική δομή περιγράφει την τελική μορφοποίηση του κειμένου που θα χρησιμοποιηθεί για την παρουσίαση ενός ή περισσότερων αντικειμένων. Για την παρουσίαση μπορούν να χρησιμοποιηθούν οι μορφότυποι HTML, XML, Latex ή απλό κείμενο (παρουσίαση ASCII). Για την περιγραφή των διαφορετικών ειδών διαμόρφωσης κειμένου ορίσθηκε η κλάση *Presentation*, σχήμα 3.5. Περιπτώσεις της κλάσης *Presentation* είναι οι κλάσεις *HTML_Presentation*, *XML_Presentation*, *Latex_Presentation* και *ASCII_Presentation*. Η κλάση *Presentation* βρίσκεται σε M1_Class επίπεδο ταξινόμησης ενώ τα μέλη της σε S_Class επίπεδο ταξινόμησης.

Σε αυτό το σημείο πρέπει να τονίσουμε ότι η HTML εστιάζει κυρίως στην περιγραφή της δομής της φυσικής παρουσίασης ενώ οι SGML και XML περιγράφουν την λογική δομή ενός κειμένου. Στην προσέγγιση που ακολουθούμε επιτρέπουμε και στην περίπτωση της HTML την περιγραφή της λογικής δομής ενός κειμένου, προσέγγιση που μοιάζει με αυτή του προτύπου SGML.

3.3.4 Λογική δομή

Η λογική δομή περιγράφει την οργάνωση του κειμένου που πρέπει να παραχθεί, δηλαδή περιγράφει το περιεχόμενο της παρουσίασης και το πώς αυτό το περιεχόμενο θα ενσωματωθεί στη δομή του τελικού κειμένου. Σε αυτήν την ενότητα θα περιγράψουμε τη λογική δομή μιας σελίδας HTML. Αν χρησιμοποιήσουμε κάποιο άλλο είδος διαμόρφωσης



Σχήμα 3.6: Η κλάση *Document* και τα γνωρίσματα που της έχουν αποδοθεί

κειμένου, εκτός από HTML, τα περισσότερα τμήματα παραμένουν τα ίδια, υπάρχουν όμως και μερικές διαφορές η οποίες θα επισημανθούν στην πορεία.

Μια σελίδα HTML περιλαμβάνει τον πρόλογο (prologue), το κύριο μέρος και τον επίλογο (epilogue). Ο πρόλογος αποτελείται από τον τίτλο και αντιστοιχεί στην ετικέτα διαμόρφωσης <TITLE> της HTML. Το κύριο μέρος μπορεί να περιέχει πλαίσια (frames) ή όχι. Οι δύο αυτές περιπτώσεις αντιστοιχούν στις ετικέτες διαμόρφωσης <FRAMESET> και <BODY> που χρησιμοποιεί η HTML. Ο επίλογος έχει δηλωθεί για μελλοντική χρήση.

Η κλάση που περιγράφει τη λογική δομή είναι η *Document* , σχήμα 3.6, και βρίσκεται σε S_Class επίπεδο ταξινόμησης.

Στην κλάση *Document* έχουν ορισθεί τα γνωρίσματα *prologue*, *epilogue* και *contains*. Το γνώρισμα *prologue* έχει ως προορισμό την κλάση *Prologue*, η οποία έχει ένα γνώρισμα, το *has_title*, για τον καθορισμό του τίτλου της σελίδας. Το γνώρισμα *epilogue* έχει ως προορισμό την κλάση *Epilogue*, ενώ το γνώρισμα *contains* έχει ως προορισμό την κλάση *Container*.

Η κλάση *Container* προσδιορίζει τη συγκεκριμένη γεωμετρία παρουσίασης που εμφανίζεται στη φυσική δομή του κειμένου. Η ίδια λογική παράσταση μπορεί να παρουσιασθεί σε HTML χρησιμοποιώντας διαφορετική γεωμετρία παρουσίασης, δηλαδή διαφορετική όψη παρουσίασης. Επομένως, παρατηρούμε ότι όψεις μπορούμε να ορίσουμε και στις λογικές παραστάσεις. Οι όψεις μπορεί να προσδιορίζουν την παρουσίαση σε HTML χρησιμοποιώντας διαφορετική γεωμετρία παρουσίασης ή την παρουσίαση της λογικής παράστασης χρησιμοποιώντας διαφορετικό είδος διαμόρφωσης κειμένου.

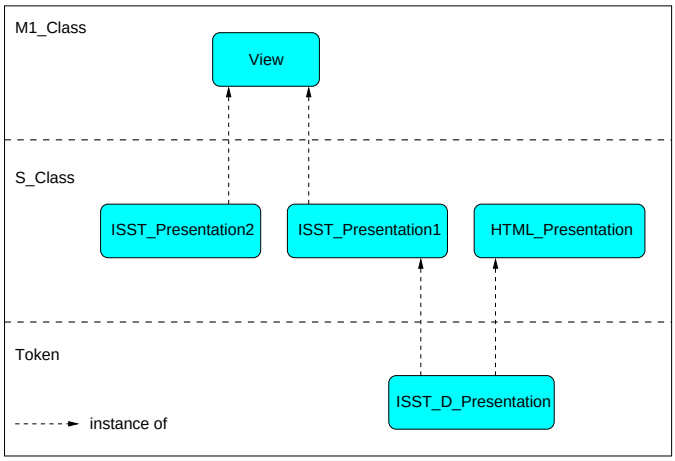
Όπως ήδη αναφέρθηκε το κύριο μέρος μιας σελίδας HTML μπορεί να περιέχει πλαίσια ή όχι. Η χρήση πλαισίων αποτελεί μια συγκεκριμένη γεωμετρία παρουσίασης. Επομένως, η κλάση *Container* έχει δύο υποκλάσεις : *FrameContainer*, όταν το κύριο

μέρος περιέχει πλαίσια και *BasicContainer*, όταν δεν έχει πλαίσια. Ο πρόλογος και ο επίλογος είναι δύο τμήματα που είναι κοινά για όλα τα είδη των παρουσιάσεων, αν και μπορεί κάποιο από αυτά τα τμήματα να είναι κενό. Το κύριο μέρος όμως δεν έχει την ίδια γεωμετρία σε όλα τα είδη παρουσιάσεων. Για παράδειγμα, σε μια παρουσίαση ASCII το κύριο μέρος δεν θα περιέχει πλαίσια, επειδή δεν υπάρχει φυσική σημασία των πλαισίων σε απλό κείμενο. Σε αυτήν την περίπτωση η δομή του κύριου μέρους δηλώνεται χρησιμοποιώντας την κλάση *BasicContainer*.

Σε πολλές περιπτώσεις η παρουσίαση ενός αντικειμένου επηρεάζεται από τα προηγούμενα αντικείμενα που παρουσιάσθηκαν, δηλαδή η παρουσίαση ενός αντικειμένου επηρεάζεται από την ιστορία του συστήματος. Αυτή η εξάρτηση είναι απαραίτητη κυρίως όταν υπάρχει μια εννοιολογική ή σημασιολογική σχέση μεταξύ των αντικειμένων και η σχέση αυτή πρέπει να αποδοθεί και κατά την παρουσίαση των αντικειμένων. Για την απόδοση της συσχέτισης της σειράς παρουσίασης των αντικειμένων με την παρουσίασή τους, έχει ορισθεί το γνώρισμα *coming_from* στην κλάση *Document*, σχήμα 3.6. Για να καταλάβουμε την σημασία της ιστορίας του συστήματος, ας δούμε την εφαρμογή του στο παράδειγμα. Η παρουσίαση ενός λογισμικού που έχει αναπτυχθεί στα πλαίσια ενός τομέα εξαρτάται από το αν ο χρήστης επισκέπτεται την παρουσίαση πρώτη φορά ή την έχει επισκευθεί αρκετές φορές και από το μονοπάτι που έχει ακολουθήσει. Την πρώτη φορά που ένας χρήστης επισκέπτεται την παρουσίαση ενός λογισμικού, περιμένει να βρεί γενικές πληροφορίες, όπως για παράδειγμα μια γενική περιγραφή της λειτουργίας του και το σκοπό ανάπτυξής του. Κάθε φορά που επανεπισκέπτεται την ιστοσελίδα, περιμένει να βρεί περισσότερες πληροφορίες σχετικά με την υλοποίηση του λογισμικού και όχι μια απλή και σύντομη περιγραφή του. Το γεγονός ότι επανεπισκέπτεται την ίδια σελίδα σημαίνει, τις περισσότερες φορές, ότι αναζητεί περισσότερες πληροφορίες. Επίσης, αν ο χρήστης επισκέπτεται την παρουσίαση ενός λογισμικού από την πρώτη σελίδα του τομέα, περιμένει γενικές πληροφορίες ενώ αν επισκεφθεί την παρουσίαση από το έργο στα πλαίσια του οποίου έχει αναπτυχθεί, περιμένει να βρεί πληροφορίες που θα συσχετίζουν το λογισμικό και με το υπόλοιπο λογισμικό που αναπτύχθηκε στα πλαίσια του έργου.

Στη συνέχεια περιγράφουμε το πώς συνδέονται οι τρείς δομές που εξετάσαμε και στη συνέχεια τη δομή του κυρίου μέρους μιας σελίδας HTML η οποία μπορεί να περιέχει πλαίσια ή όχι. Σε κάθε περίπτωση υπάρχουν κάποια γνωρίσματα που αφορούν όλο το κύριο μέρος, όπως η γεωμετρία των πλαισίων ή το φόντο της σελίδας. Επιπλέον, το κύριο μέρος μπορεί να θεωρηθεί ως μια διατεταγμένη ακολουθία μικρότερων τμημάτων που έχουν τα δικά τους χαρακτηριστικά.

3.3.4.1 Σύνδεση σημασιολογικής, φυσικής και λογικής δομής



Σχήμα 3.7: Μια λογική παράσταση ταξινομείται στην όψη και το είδος διαμόρφωσης κειμένου σύμφωνα με τα οποία θα παραχθεί η παρουσίαση

Μια λογική δομή ταξινομείται τόσο στην όψη σύμφωνα με την οποία θα παραχθεί η παρουσίαση όσο και στο είδος διαμόρφωσης κειμένου που θα χρησιμοποιηθεί.

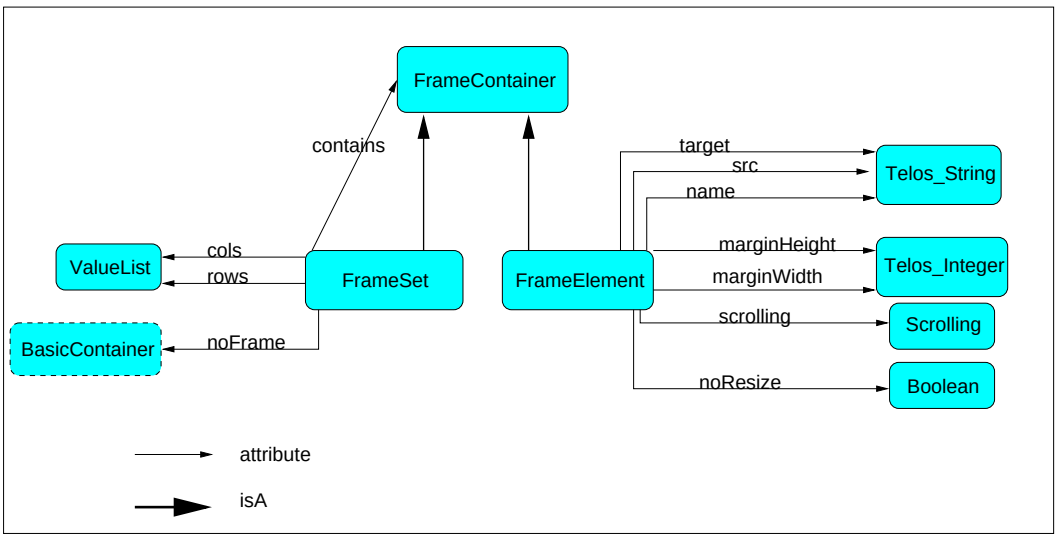
Ας δούμε με ένα παράδειγμα πώς ορίζουμε όψεις και πώς συσχετίζουμε τις τρείς δομές μεταξύ τους, δηλαδή τη σημασιολογική, τη φυσική και τη λογική δομή. Έστω ότι θέλουμε να προσδιορίσουμε μια παρουσίαση HTML για τον Τομέα Πληροφοριακών Συστημάτων και Τεχνολογίας Λογισμικού με διαφορετικό τρόπο ανάλογα με το αν ο χρήστης είναι μέλος του τομέα ή όχι. Δηλαδή, θέλουμε να ορίσουμε δύο όψεις στην πληροφορία που αφορά τον τομέα. Επομένως, ορίζουμε δύο διαφορετικά αντικείμενα/κλάσεις, περιπτώσεις της κλάσης *View*, τα *ISST_Presentation1* και *ISST_Presentation2*, σχήμα 3.7. Ο τομέας παριστάνεται στη βάση με το αντικείμενο *ISST_Division*. Στο σχήμα 3.4 είδαμε πώς συσχετίζουμε μια παρουσίαση με ένα αντικείμενο. Το αντικείμενο *ISST_D_Presentation* περιγράφει τη δομή της παρουσίασης σύμφωνα με την όψη *ISST_Presentation1* και για την παρουσίαση θα χρησιμοποιηθεί η HTML. Επομένως, σύμφωνα με αυτά που περιγράψαμε, το αντικείμενο *ISST_D_Presentation* πρέπει να ταξινομηθεί στις κλάσεις *ISST_Presentation1* και *HTML_Presentation*, σχήμα 3.7.

3.3.4.2 Η δομή FrameContainer

Η δομή FrameContainer διακρίνεται σε δύο υποσύνολα, τα πλαίσια *FrameElement*, και τα σύνολα πλαισίων *FrameSet*, σχήμα 3.8. Ένα σύνολο πλαισίων μπορεί να περιέχει πλαίσια και σύνολα πλαισίων. Για την απόδοση αυτής της σχεσης ορίζεται το

γνώρισμα *contains* που έχει ως αφετηρία την κλάση *FrameSet* και προορισμό την κλάση *FrameContainer*.

Ένα σύνολο πλαισίων χαρακτηρίζεται από τη γεωμετρία των πλαισίων που περιέχει, δηλαδή πόσες γραμμές και πόσες στήλες καταλαμβάνει το κάθε πλαίσιο. Έτσι, στην κλάση *FrameSet* έχουν ορισθεί τα γνωρίσματα *rows* και *cols* που έχουν ως προορισμό την κλάση *ValueList*, σχήμα 3.8.



Σχήμα 3.8: Οι κλάσεις *FrameContainer*, *FrameSet* και *FrameElement*

Η υποστήριξη πλαισίων δεν είναι δυνατή από όλους τους αναδιφητές του Παγκόσμιου Ιστού. Για αυτόν τον λόγο μπορεί να προσδιορισθεί η σελίδα HTML που πρέπει να παρουσιασθεί στη θέση των πλαισίων, όταν ο αναδιφητής του Παγκόσμιου Ιστού δεν υποστηρίζει τη χρήση πλαισίων. Αυτή η δυνατότητα υποστηρίζεται από το μοντέλο μέσω του γνωρίσματος *noFrame* που έχει ως προορισμό την κλάση *BasicContainer*, σχήμα 3.8. Όπως έχει ειπωθεί, η κλάση *BasicContainer* προσδιορίζει το κύριο μέρος μιας σελίδας HTML χωρίς τη χρήση πλαισίων.

Ένα πλαίσιο προσδιορίζεται από την πηγή από την οποία θα αντλήσει την πληροφορία που θα περιέχει, το όνομά του, το αν θα είναι μεταβλητό το μέγεθός του, το ύψος και το πάχος του περιθωρίου και το αν θα υπάρχει ράβδος ολίσθησης (scrollbar). Για κάθε ένα από τα παραπάνω χαρακτηριστικά έχει ορισθεί αντίστοιχο γνώρισμα στην κλάση *FrameElement*. Ένα επιπλέον γνώρισμα της κλάσης *FrameElement* είναι το όνομα του πλαισίου στο οποίο πρέπει να παρουσιασθεί η πληροφορία όταν επιλέγεται ένας υπερ-σύνδεσμος. Για τον προσδιορισμό της πηγής της πληροφορίας, χρησιμοποιούνται URL/URI. Επομένως, η πηγή πληροφορίας θα πρέπει να προσδιορίζει το αντικείμενο που θα παρουσιασθεί και την όψη σύμφωνα με την οποία θα γίνει η παρουσίαση, ή κάποιο άλλο URL αν για την παρουσίαση της πληροφορίας δεν απαιτείται η χρήση της πύλης διασύνδεσης.

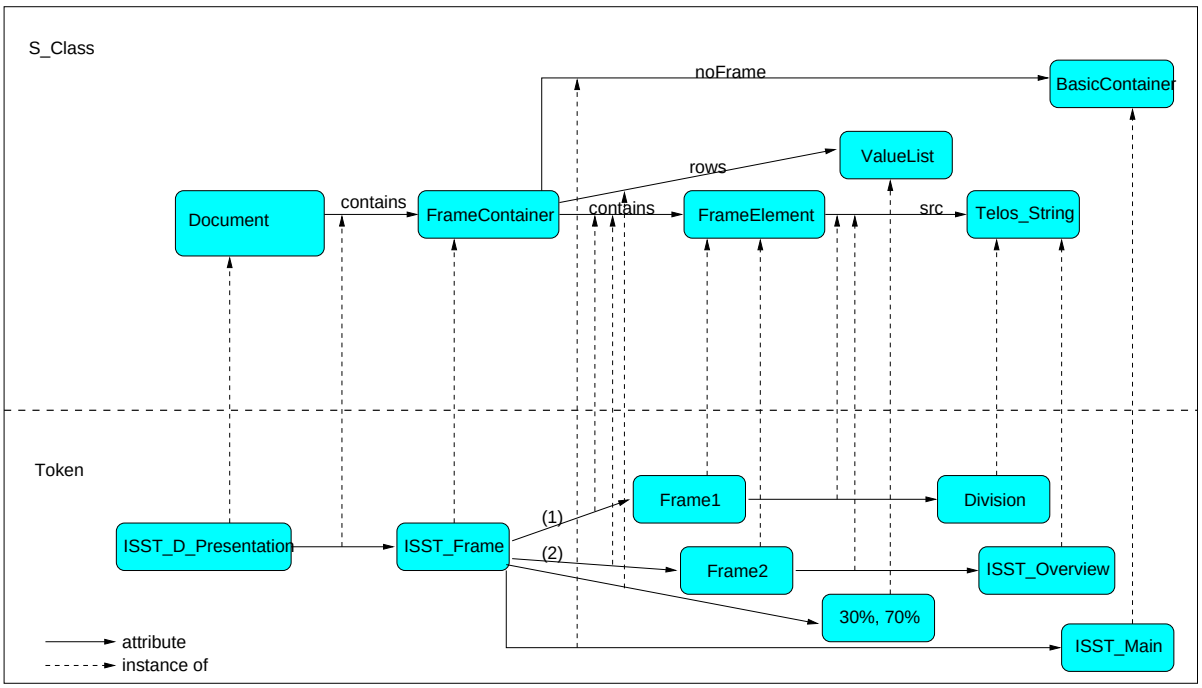
Έστω ότι θέλουμε να παρουσιάσουμε σε HTML το αντικείμενο *ISST_Division* σύμφωνα με την όψη *ISST_Presentation1*, με χρήση πλαισίων. Έχουμε ήδη ορίσει ότι το αντικείμενο *ISST_D_Presentation* είναι μια παρουσίαση του *ISST_Division* σε HTML, σύμφωνα με την όψη *ISST_Presentation1*. Το αντικείμενο *ISST_D_Presentation* είναι περίπτωση της κλάσης *Document* και περιγράφει τη δομή της παρουσίασης, σχήμα 3.9. Επιπλέον, έστω ότι η σελίδα HTML με τη χρήση πλαισίων θέλουμε να αποτελείται από δύο πλαίσια. Στο πρώτο πλαίσιο να παρουσιάζονται οι κατηγορίες *Δραστηριότητες*, *Συστήματα*, *Προσωπικό*, *Έργα*, *Εκδηλώσεις* και *Δημοσιεύσεις* ως υπερ-σύνδεσμοι. Στο δεύτερο πλαίσιο θέλουμε να παρουσιάζεται μια σύντομη περιγραφή του τομέα.

Οι κατηγορίες που πρέπει να εμφανισθούν ως υπερ-σύνδεσμοι είναι οι κλάσεις με τις οποίες συνδέεται η κλάση *Division* μέσω συνδέσμων απόδοσης γνωρίσματος και οι σύνδεσμοι αυτοί έχουν ως αφετηρία την κλάση *Division*. Επομένως, το περιεχόμενο του πρώτου πλαισίου είναι η παρουσίαση σε κείμενο HTML της κλάσης *Division*. Το περιεχόμενο του δεύτερου πλαισίου είναι η παρουσίαση σε κείμενο HTML της σύντομης περιγραφής του τομέα, που είναι το αντικείμενο *ISST_Overview*.

Στο αντικείμενο *ISST_D_Presentation* ορίζουμε ένα γνώρισμα που επιναι μέλος της κλάσης γνωρισμάτων *contains* με προορισμό το αντικείμενο *ISST_Frame*. Η σελίδα θέλουμε να αποτελείται από δύο πλαίσια. Επομένως, αντιστοιχίζουμε στο αντικείμενο *ISST_Frame* δύο μέλη της κλάσης *FrameElement* μέσω γνωρισμάτων που είναι μέλη της κλάσης γνωρισμάτων *contains* που έχει ως αφετηρία την κλάση *FrameSet*. Τα λογικά ονόματα των γνωρισμάτων είναι *(1)* και *(2)* και τα ονοματίζουμε για να προσδιορίσουμε τη σειρά με την οποία πρέπει να εμφανισθούν. Έστω ότι τα πλαίσια θα είναι έτσι τοποθετημένα ώστε να διαχωρίζουν την σελίδα σε δύο οριζόντια τμήματα. Για να προσδιορίσουμε τη γεωμετρία των πλαισίων, αποδίδουμε στο αντικείμενο *ISST_Frame* ένα γνώρισμα που είναι μέλος της κλάσης γνωρισμάτων *rows* με τιμή το αλφαριθμιτικό *30%, 70%*. Με την τιμή που δώσαμε, καθορίσαμε ότι το πρώτο πλαίσιο θα καταλαμβάνει το 1/3 και το δεύτερο τα 2/3 της σελίδας HTML. Τα παραπάνω φαίνονται στο σχήμα 3.9.

Στο σχήμα 3.9 διακρίνεται επίσης ότι στο αντικείμενο *Frame1* έχουμε αντιστοιχίσει μέσω γνωρίσματος, που είναι μέλος της κλάσης γνωρισμάτων *source*, το αλφαριθμιτικό *Division* και στο *Frame2* το *ISST_Overview*. Με αυτόν τον τρόπο προσδιορίζουμε ότι για το περιεχόμενο του πρώτου πλαισίου θα εκτελεστεί μια επερώτηση για την παρουσίαση του αντικειμένου *Division* και για τα περιεχόμενα του δεύτερου πλαισίου θα εκτελεστεί μια επερώτηση για την παρουσίαση του αντικειμένου *ISST_Overview*. Επιπλέον, στο ίδιο σχήμα φαίνεται ότι στο αντικείμενο *ISST_Frame* έχει αντιστοιχισθεί μέσω γνωρίσματος, που είναι μέλος της κλάσης γνωρισμάτων *noFrame*, το αντικείμενο *ISST_Main*. Το αντικείμενο αυτό είναι περίπτωση της κλάσης *BasicContainer* και η δομή του θα περιγραφεί παρακάτω.

Σε αυτό το σημείο πρέπει να τονίσουμε ότι η χρήση πλαισίων σε μια σελίδα HTML είναι ένας υψηλότερου επιπέδου τρόπος περιγραφής της λογικής δομής της σελίδας. Η περιγραφή των πλαισίων δεν καθορίζει και το περιεχόμενο του κάθε πλαισίου. Το



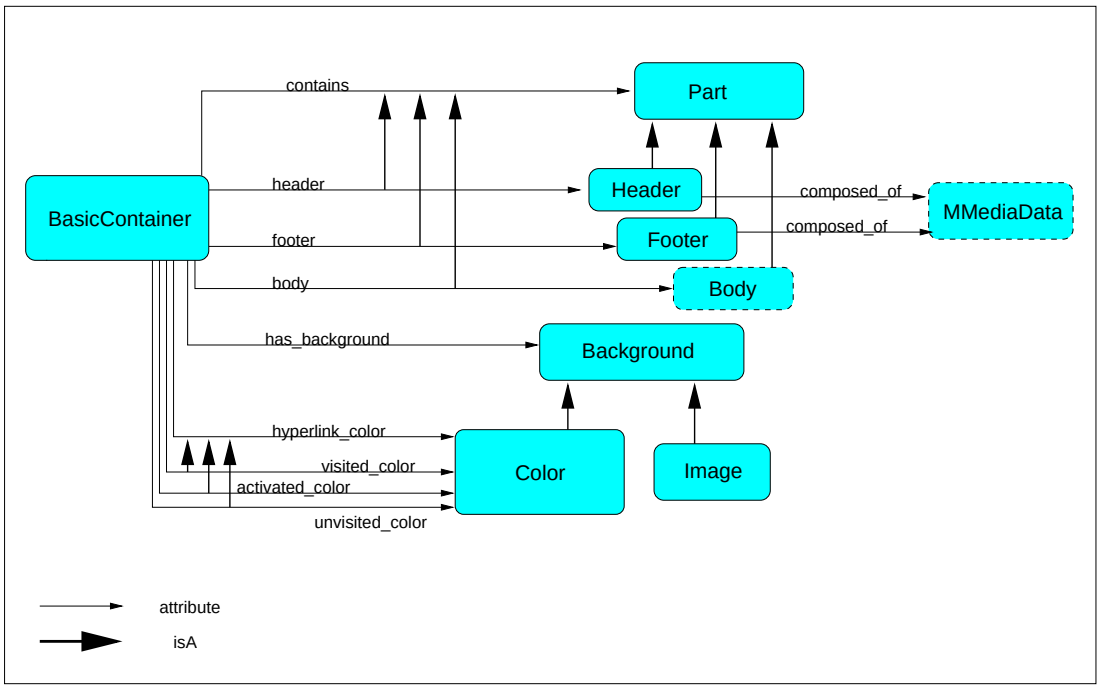
Σχήμα 3.9: Παράδειγμα χρήσης πλαισίων

περιεχόμενο του κάθε πλαισίου προσδιορίζεται μέσω ενός URL/URI.

3.3.4.3 Η δομή BasicContainer

Στην περίπτωση που το κύριο μέρος δεν περιέχει πλαίσια, τα γνωρίσματα προσδιορίζουν το φόντο της σελίδας και το χρώμα που θα έχουν οι υπερ-σύνδεσμοι. Το αντικείμενο που προσδιορίζει τη δομή του κύριου μέρους της σελίδας HTML σε αυτήν την περίπτωση είναι η κλάση *BasicContainer*. Σε αυτήν την κλάση έχουν ορισθεί τα γνωρίσματα *hyperlink_color* και *has_background*, που έχουν ως προορισμό αντίστοιχα τις κλάσεις *Color* και *Background*, που φαίνονται στο σχήμα 3.10. Οι υπερσύνδεσμοι διακρίνονται σε τρείς κατηγορίες : στους προσπελασθέντες, στους μη προσπελασθέντες και στους ενεργούς. Επομένως, η κλάση γνωρισμάτων *hyperlink_color* έχει τρείς υποκλάσεις, μία για κάθε είδος υπερ-συνδέσμου, που είναι οι *visited_color*, *unvisited_color* και *activated_color*, σχήμα 3.10.

Το κύριο μέρος μιας σελίδας HTML χωρίς τη χρήση πλαισίων, διακρίνεται σε μικρότερα τμήματα. Η σχέση αυτή αποτυπώνεται στη δομή μέσω του γνωρίσματος *contains* που έχει ως αφετηρία την κλάση *BasicContainer* και ως προορισμό την κλάση *Part*. Τα τμήματα από τα οποία αποτελείται μια σελίδα HTML είναι το άνω μέρος (header), το κύριο μέρος (body) και στο κάτω μέρος (footer). Επομένως, η κλάση γνωρισμάτων *contains* έχει τρείς υποκλάσεις που είναι *header*, *body* και *footer* που έχουν προορισμό



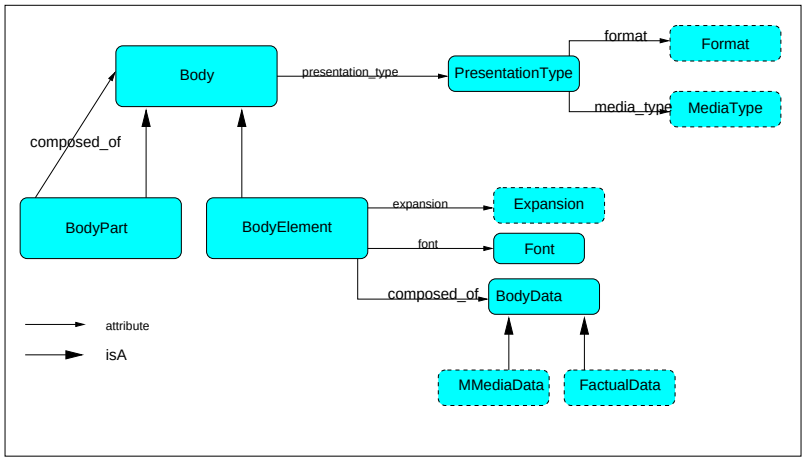
Σχήμα 3.10: Η κλάση *BasicContainer* και τα γνωρίσματά της.

αντίστοιχα τις κλάσεις *Header*, *Body* και *Footer*, σχήμα 3.10.

Το άνω και το κάτω μέρος μιας σελίδας HTML, μπορεί να θεωρηθεί ως μια διατεταγμένη ακολουθία μικρότερων τμημάτων που περιέχουν κείμενο σε συνδυασμό με δεδομένα πολυμέσων. Έτσι οι κλάσεις *Header* και *Footer* έχουν ένα γνώρισμα, το *composed_of* που έχει ως προορισμό την κλάση *MMediaData*, σχήμα 3.10.

Το κύριο μέρος (body) αποτελείται από μικρότερα τμήματα που περιέχουν δεδομένα πολυμέσων και δεδομένα που βρίσκονται στη βάση. Το μικρότερο τμήμα το οποίο μπορεί να περιέχεται σε μια σελίδα HTML, δηλαδή μια εικόνα ή μια γραμμή κειμένου, ονομάζεται *BodyElement*. Το κύριο μέρος διακρίνεται σε τμήματα που αποτελούνται από *BodyElements* στα οποία έχει αποδοθεί ένας κοινός τύπος παρουσίασης. Ο τύπος παρουσίασης προσδιορίζει το είδος των δεδομένων πολυμέσων και τον μορφότυπο της παρουσίασης. Έτσι, για παράδειγμα, μπορεί να ορισθεί ότι ένα σύνολο από *BodyElements* θα παρουσιασθεί ως γραμμές ενός πίνακα. Αυτή η σχέση προσδιορίζεται μέσω του γνωρίσματος *presentation_type* που έχει ως αφετηρία την κλάση *Body* και προορισμό την κλάση *PresentationType*. Τα σύνολα των *BodyElements* ονομάζονται στο μοντέλο *BodyPart*. Για τον προσδιορισμό των μικρότερων τμημάτων που περιέχει ένα *BodyPart* χρησιμοποιείται το γνώρισμα *composed_of* που έχει ως προορισμό την κλάση *Body*. Έτσι, ένα *BodyPart* μπορεί να περιέχει και μικρότερα τμήματα τύπου *BodyPart*.

Οι κλάσεις *BodyElement*, *MMediaData* και *PresentationType* θα περιγραφούν στη συνέχεια.

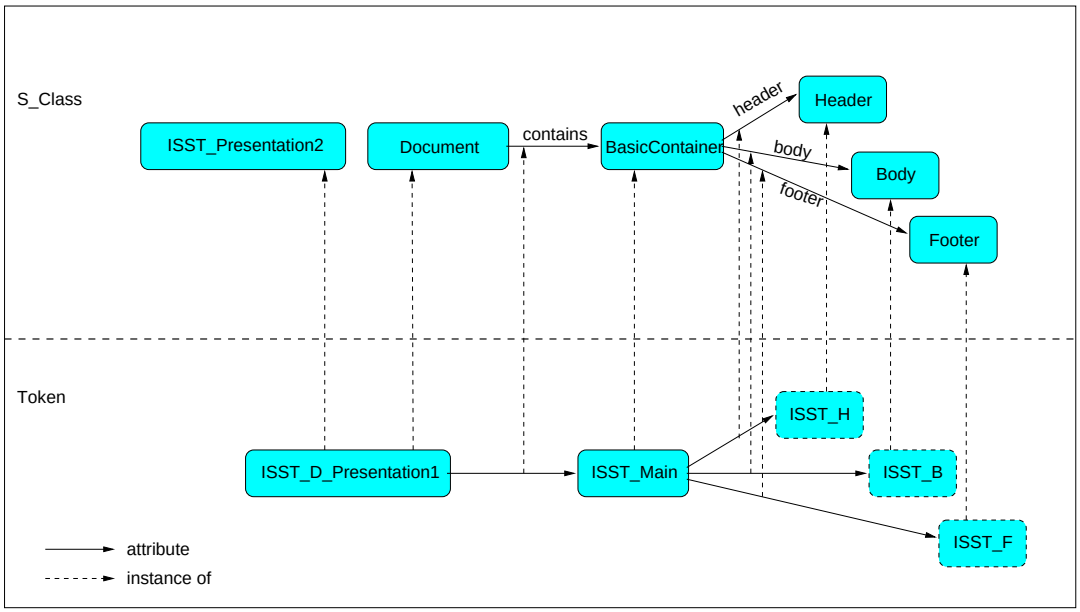


Σχήμα 3.11: Η κλάση *BodyElement* και οι κλάσεις με τις οποίες σχετίζεται

Έστω οτι θέλουμε να παρουσιάσουμε σε HTML το αντικείμενο *ISST_Division* χωρίς τη χρήση πλαισίων. Η παρουσίαση θα γίνει σύμφωνα με την όψη *ISST_Presentation2*. Αντιστοιχίζουμε ένα αντικείμενο τύπου *Document* στο *ISST_Division*, όπως ακριβώς έγινε η αντιστοίχιση του αντικειμένου *ISST_D_Presentation*, σχήμα 3.4. Έστω *ISST_D_Presentation1* αυτό το αντικείμενο, το οποίο είναι μέλος της κλάσης *Document*. Στο αντικείμενο *ISST_D_Presentation1* συσχετίζουμε μέσω του γνωρίσματος *contains* το αντικείμενο *ISST_Main* που είναι μέλος της κλάσης *BasicContainer*, σχήμα 3.12. Έστω ότι θέλουμε η σελίδα να περιέχει στο άνω μέρος το λογότυπο του τομέα Πληροφοριακών Συστημάτων και Τεχνολογίας Λογισμικου, ενώ στο κάτω μέρος το λογότυπο του Ινστιτούτου Πληροφορικής του ΙΤΕ. Μέσω των γνωρισμάτων *header*, *body* και *footer* συσχετίζουμε τα αντικείμενα *ISST_H*, *ISST_B* και *ISST_F* με το αντικείμενο *ISST_Main*. Το αντικείμενο *ISST_H* είναι περίπτωση της κλάσης *Header* και ορίζουμε ένα γνώρισμα της κλάσης γνωρισμάτων *composed_of* που έχει ως κόμβο προορισμού το αντικείμενο *ISST_Logo*. Το αντικείμενο *ISST_Logo* είναι μια εικόνα (περίπτωση της κλάσης *Image*) και είναι ο λογότυπος του Τομέα Πληροφοριακών Συστημάτων και Τεχνολογίας Λογισμικού. Αντίστοιχα, στο αντικείμενο *ISST_F*, που είναι περίπτωση της κλάσης *Footer*, έχει ορισθεί ένα γνώρισμα της κλάσης γνωρισμάτων *composed_of* που έχει ως προορισμό το αντικείμενο *ICS_Logo*. Το αντικείμενο *ICS_Logo* είναι μέλος της κλάσης *Image* και είναι ο λογότυπος του Ινστιτούτου Πληροφορικής, σχήμα 3.13. Το αντικείμενο *ISST_B* είναι μέλος της κλάσης *Body* και θα περιγραφεί στη συνέχεια.

3.3.4.4 Η δομή BodyElement

Η κλάση *BodyElement* περιγράφει τη δομή των μικρότερων τμημάτων τα οποία περιλαμβάνει το κύριο μέρος μιας σελίδας HTML. Για το κάθε στοιχείο (στοιχείων)



Σχήμα 3.12: Παράδειγμα χρήσης του *BasicContainer*

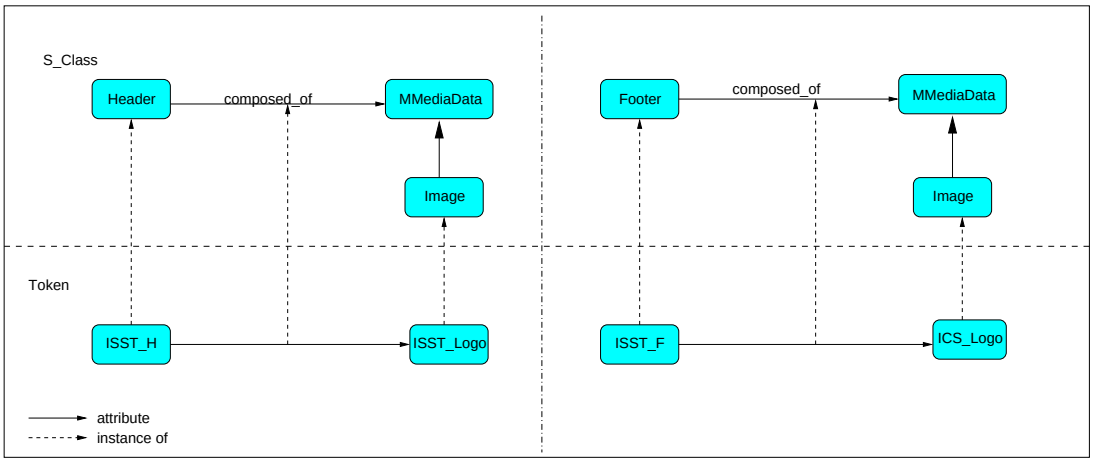
μπορεί να ορισθεί ποιά γραμματοσειρά θα χρησιμοποιηθεί και πώς θα επεκταθούν τα δεδομένα. Έτσι, στην κλάση *BodyElement* έχουν ορισθεί τα γνωρίσματα *font* και *expansion*, που έχουν ανίστοιχα ως προορισμό τις κλάσεις *Font*, και *Expansion*, σχήμα 3.11. Η κλάση *Font* περιέχει ως μέλη γραμματοσειρές, ενώ η κλάση *Expansion* θα περιγραφεί στη συνέχεια.

Κάθε στοιχείο μιας σελίδα HTML, μπορεί να περιέχει δεδομένα από τη βάση ή όχι. Τα δεδομένα που δεν προέρχονται από τη βάση θα τα ονομάζουμε στο εξής δεδομένα πολυμέσων. Συνεπώς η κλάση *BodyElement* έχει ένα γνώρισμα, το *composed_of*, που έχει ως προορισμό την κλάση *BodyData*. Η κλάση *BodyData* έχει δύο υποκλάσεις, τις κλάσεις *MMediaData* και *FactualData*, όπου η κλάση *MMediaData* περιέχει ως μέλη δεδομένα πολυμέσων ενώ η κλάση *FactualData* περιέχει ως μέλη δεδομένα που προέρχονται από τη βάση, σχήμα 3.11.

3.3.4.5 Η δομή τύπος παρουσίασης

Η κλάση τύπος παρουσίασης χρησιμοποιείται για τον καθορισμό του τύπου παρουσίασης. Ο τύπος παρουσίασης εξαρτάται τόσο από τον μορφότυπο όσο και από το μέσο, όπως εικόνα, ήχος. Αυτή η σχέση αποδίδεται στην κλάση μέσω των γνωρισμάτων *format* και *media_type*, που έχουν αντίστοιχα ως προορισμό τις κλάσεις *Format* και *MediaType*, σχήμα 3.11.

Για την παρουσίαση των δεδομένων πολυμέσων στην HTML δεν υπάρχει μια ένα



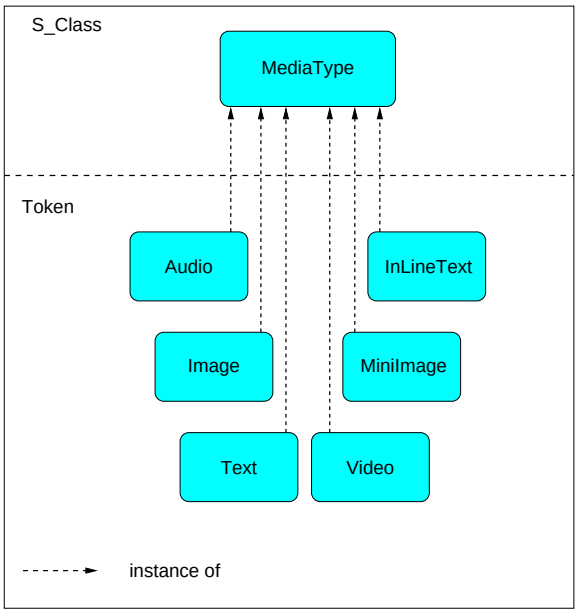
Σχήμα 3.13: Παράδειγμα *Header* και *Footer*

προς ένα αντιστοιχία με το μέσο που θα χρησιμοποιηθεί. Έτσι για παράδειγμα, μια εικόνα μπορεί να παρουσιασθεί όπως είναι, ή να παρουσιασθεί ως ένας υπερ-σύνδεσμος ο οποίος, αν επιλεγεί, έχει ως αποτέλεσμα την εμφάνιση της εικόνας. Συνεπώς, πρέπει ο προγραμματιστής εφαρμογής να μπορεί να δηλώσει το μέσο που θα χρησιμοποιηθεί. Για την παράσταση των διαφορετικών επιλογών χρησιμοποιείται η κλάση *MediaType* που έχει ως περιπτώσεις τις ατομικές οντότητες (σχήμα 3.14):

- *Audio*
- *Image*
- *Text*
- *Video*

Οι παραπάνω τύποι δεν μπορούν να εφαρμοσθούν σε οποιοδήποτε αντικείμενο πολυμέσων, όπως για παράδειγμα μια εικόνα δεν μπορεί να παρουσιασθεί ως ήχος ή ένα αρχείο κειμένου δεν μπορεί να παρουσιασθεί ως βίντεο ή ήχος. Έτσι, ο τύπος *Text* μπορεί να εφαρμοστεί σε όλα τα είδη δεδομένων επειδή μια εικόνα, ένα αρχείο κειμένου, ένα βίντεο και ο ήχος μπορούν να παρουσιασθούν ως υπερ-σύνδεσμοι. Ο τύπος *Audio* μπορεί να εφαρμοστεί μόνο σε ήχο, οι τύποι *Image* και *Mini Image* μόνο σε εικόνα και ο τύπος *Video* μόνο σε βίντεο. Τυπική τιμή του τύπου μορφότυπου παρουσίασης είναι *Text*.

Για τον προσδιορισμό του μορφότυπου παρουσίασης ενός συνόλου δεδομένων χρησιμοποιείται η κλάση *Format* (σχήμα 3.15). Ο μορφότυπος παρουσίασης δεν εξαρτάται άμεσα από το είδος της παρουσίασης, δηλαδή αν είναι παρουσίαση σε HTML ή απλό κείμενο. Ο μορφότυπος καθορίζει κάποια γενικά χαρακτηριστικά που είναι ανεξάρτητα του είδους παρουσίασης. Ο τρόπος μετάφρασης αυτών των χαρακτηριστικών είναι θέμα της υλοποίησης.



Σχήμα 3.14: Η κλάση *MediaType* και οι περιπτώσεις της.

Διακρίνονται δύο βασικές κατηγορίες μορφότυπου παρουσίασης, οι προδηλωμένοι μορφότυποι (*Predefined*) και οι οριζόμενοι από τον προγραμματιστή εφαρμογής (*UserDefined*). Τα παραπάνω φαίνονται στο σχήμα 3.15.

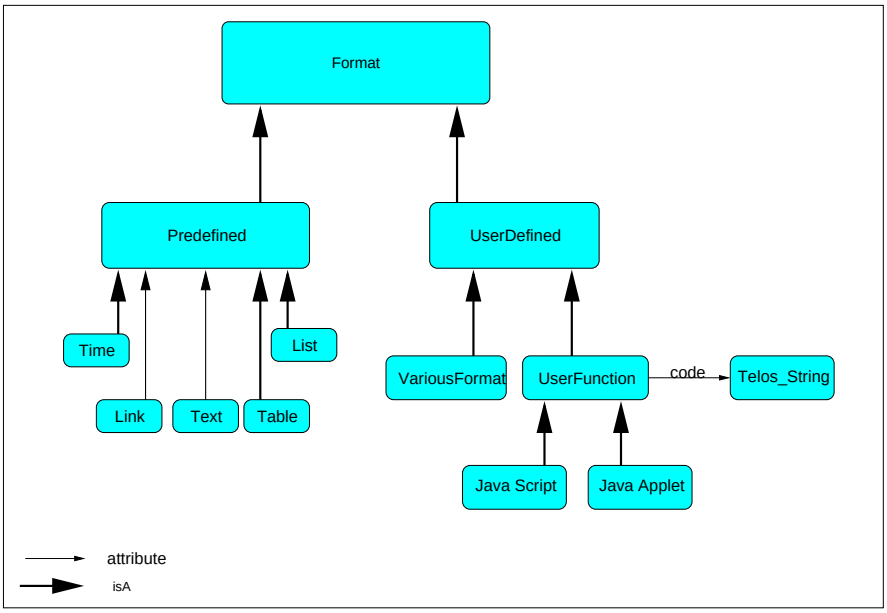
Ως προδηλωμένοι μορφότυποι έχουν ορισθεί οι εξής :

- *Time*
  Ο χρόνος μπορεί να παρασταθεί με πολλούς διαφορετικούς τρόπους, ανάλογα με τη γεωγραφική περιοχή. Χρησιμοποιώντας τον μορφότυπο *Time* μπορούν να γίνουν διακρίσεις στον τρόπο παρουσίασης της ημερομηνίας, περιόδων και χρόνου.

- *Table*
  Ο μορφότυπος *Table* διακρίνεται σε δύο υποκατηγορίες : *TableRows* και *TableColumns*. Αν χρησιμοποιηθεί ο μορφότυπος *TableRows* τότε τα αντικείμενα ενός συνόλου (σύνολο ορίζεται και με ένα στοιχείο) παρουσιάζονται ως γραμμές ενός πίνακα, ενώ αν χρησιμοποιηθεί ο μορφότυπος *TableColumns*, παρουσιάζονται ως στήλες ενός πίνακα.

- *List*
  Ο μορφότυπος *List* διακρίνεται στις υποκατηγορίες *UnorderedList*, *OrderedList* και *DefinitionList*. Τα μέλη ενός συνόλου παρουσιάζονται ως μέλη μιας μη αριθμημένης λίστας, αριθμημένης λίστας ή μιας λίστας με ορισμούς αντίστοιχα, όταν δηλωθεί ένας από τους παραπάνω μορφότυπους.

- *Text*



Σχήμα 3.15: Η κλάση *Format* και οι υποκλάσεις της.

Υποκατηγορίες του μορφότυπου *Text* είναι μορφότυποι που μπορούν να εφαρμοστούν σε ένα κείμενο, όπως λ.χ. να παρουσιασθεί ως μια παράγραφος. Μερικές υποκατηγορίες του μορφότυπου *Text* είναι *Centered*, *Paragraph*, *Bold*, *Italic*, *Emphasized*, *Header* κ.α.

- *Link*
  Ένα γνώρισμα προσδιορίζεται από τρία στοιχεία, την οντότητα αφετηρίας, το λογικό όνομα του συνδέσμου και την οντότητα προορισμού. Ο μορφότυπος *Link* προσδιορίζει ποιό στοιχείο από τα τρία και πώς θα παρουσιασθεί. Διακρίνεται στις υποκλάσεις,

  ▷ *HyperlinkAll*,
  όπου παρουσιάζονται όλα τα στοιχεία ως υπερ-σύνδεσμοι.

  ▷ *NoHyperlinkAll*,
  όπου παρουσιάζονται όλα τα στοιχεία ως απλό κείμενο.

  ▷ *HyperlinkFromValue*,
  όπου η οντότητα αφετηρίας παρουσιάζεται ως υπερ-σύνδεσμος.

  ▷ *NoHyperlinkFromValue*,
  όπου η οντότητα αφετηρίας παρουσιάζεται ως απλό κείμενο.

  ▷ *HyperlinkLabel*,
  όπου το λογικό όνομα του συνδέσμου παρουσιάζεται ως υπερ-σύνδεσμος.

- ▷ *NoHyperlinkLabel*,
  όπου το λογικό όνομα του συνδέσμου παρουσιάζεται ως απλό κείμενο.
- ▷ *HyperlinkToValue*,
  όπου η οντότητα προορισμού παρουσιάζεται ως υπερ-σύνδεσμος.
- ▷ *NoHyperlinkToValue*,
  όπου η οντότητα προορισμού παρουσιάζεται ως απλό κείμενο.

Σε μια παρουσίαση μπορούν να αντιστοιχισθούν περισσότεροι του ενός μορφότυποι παρουσίασης όταν δεν υπάρχουν συγκρούσεις. Για παράδειγμα, μπορεί να χρησιμοποιηθεί πίνακας (γραμμές ή στήλες) για την παρουσίαση ενός συνόλου αντικειμένων και ταυτόχρονα τα ονόματα των αντικειμένων να είναι πλάγια. Ομοίως μπορούν να χρησιμοποιηθούν περισσότεροι του ενός μορφότυποι τύπου *Link* αρκεί να μη συγκρούονται. Τυπική τιμή του μορφότυπου παρουσίασης είναι *DefinitionList* και για τα γνωρίσματα *HyperlinkAll*.

Επειδή οι παραπάνω μορφότυποι δεν μπορούν να καλύψουν όλους τους δυνατούς τρόπους παρουσίασης αντικειμένων και όσο ευρύς και να γίνει ο κατάλογος κάποιες περιπτώσεις δεν θα είναι διαθέσιμες, δίνεται η δυνατότητα στον προγραμματιστή της εφαρμογής να δημιουργήσει τον μορφότυπο που θέλει. Υπάρχουν δύο επιλογές, να χρησιμοποιήσει τον μορφότυπο *VariousFormat* προσδιορίζοντας την ετικέτα διαμόρφωσης που πρέπει να τυπωθεί στην αρχή και στο τέλος των προς παρουσίαση αντικειμένων καθώς και την ετικέτα διαμόρφωσης που πρέπει να τυπωθεί μεταξύ δύο διαδοχικών αντικειμένων ή να χρησιμοποιήσει τον τύπο *UserFunction*. Ο μορφότυπος *UserFunction* διακρίνεται σε δύο υποκατηγορίες *JavaScript* και *Java*. Χρησιμοποιώντας την πρώτη περίπτωση ο προγραμματιστής εφαρμογής μπορεί να προσδιορίσει ένα Java script το οποίο θα συμπεριληφθεί στην τελική σελίδα. Ομοίως αν χρησιμοποιήσει τον τύπο *Java*, θα συμπεριληφθεί ένα Java applet στην τελική σελίδα. Για την εισαγωγή του κώδικα χρησιμοποιείται το γνώρισμα *code* που έχει ως οντότητα προορισμού το *Telos_String*, σχήμα 3.15.

3.3.4.6 Επέκταση

Ο όρος *Επέκταση* αναφέρεται στην κατάσταση όπου για να είναι δυνατή η κατανόηση του ρόλου και της σημασίας μιας οντότητας πρέπει να υπάρξει μια σφαιρική κατανόηση της γειτονιάς της, δηλ. των οντοτήτων με τις οποίες συνδέεται άμεσα ή έμμεσα. Στην περίπτωση της Telos η γειτονιά μιας οντότητας υπολογίζεται χρησιμοποιώντας του συνδέσμους ταξινόμησης, γενίκευσης και απόδοσης γνωρίσματος.

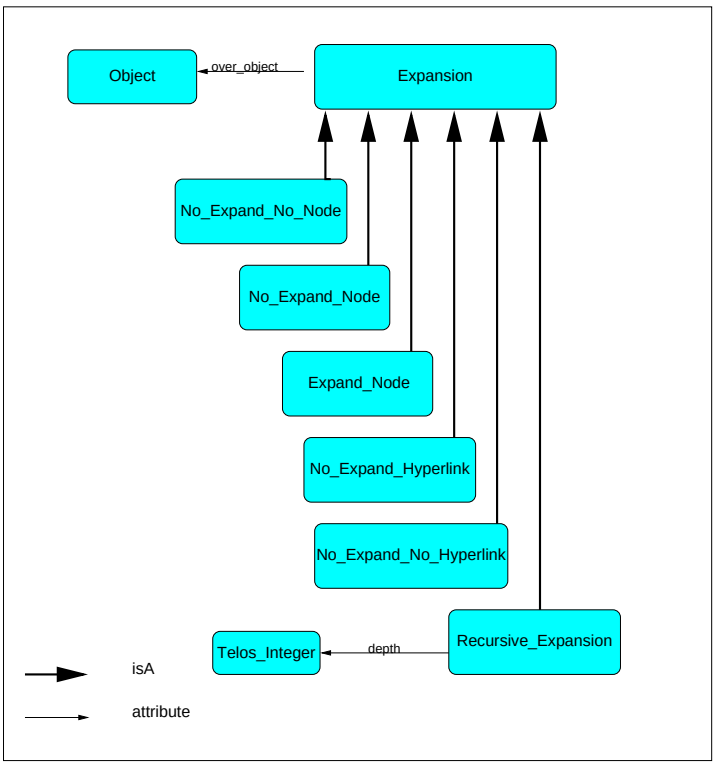
Θεωρητικά υπάρχουν δύο ειδών επεκτάσεις :

- Οριζόντια επέκταση,
  Η οριζόντια επέκταση αφορά τα γνωρίσματα των οντοτήτων. Τα γνωρίσματα έχουν

οντότητες ως κόμβους προορισμού οι οποίες μπορεί να έχουν δικά τους γνωρίσματα κ.ο.κ. Χρησιμοποιώντας οριζόντια επέκταση προσδιορίζεται το ‘‘βάθος’’ στο οποίο πρέπει να φτάσει η παρουσίαση των γνωρισμάτων.

- Κάθετη επέκταση,
  Η κάθετη επέκταση αφορά οντότητες με τις οποίες συνδέεται η τρέχουσα οντότητα μέσω συνδέσμων ταξινόμησης και γενίκευσης. Χρησιμοποιώντας κάθετη επέκταση προσδιορίζεται το αν και πόσο πρέπει να επεκταθεί η παρουσίαση ως προς την κατεύθυνση αυτή.

Στο μοντέλο που περιγράφεται η διάκριση αυτή γίνεται έμμεσα. Δεν υπάρχει κάποιο γνώρισμα ή κάποιες οντότητες που να προσδιορίζουν το είδος της επέκτασης αλλά ο προσδιορισμός γίνεται χρησιμοποιώντας το σύνολο των οντοτήτων στο οποίο αναφέρεται. Αν για παράδειγμα, η επέκταση έχει δηλωθεί στο σύνολο των οντοτήτων που είναι μέλη της τρέχουσας οντότητας, τότε η επέκταση είναι κάθετη. Αν η επέκταση δηλώνεται στο σύνολο των οντοτήτων που συνδέονται με τη τρέχουσα οντότητα μέσω συνδέσμων απόδοσης γνωρίσματος, τότε είναι οριζόντια.



Σχήμα 3.16: Η κλάση *Expansion* και οι υποκλάσεις της.

Η επέκταση αφορά σύνολο οντοτήτων ή μια οντότητα σύμφωνα με συγκεκριμένη όψη. Στα πλαίσια διαφορετικής όψης μπορεί να ορισθεί διαφορετική επέκταση στο ίδιο

σύνολο αντικειμένων. Για τον προσδιορισμό της επέκτασης σε ένα σύνολο οντοτήτων χρησιμοποιείται το γνώρισμα *expansion* που έχει ως αφετηρία την κλάση *BodyElement*, σχήμα 3.11. Η επέκταση μιας οντότητας μπορεί να γίνει με πολλούς τρόπους. Μερικοί τρόποι έχουν δηλωθεί στο μοντέλο ως υποκλάσεις της κλάσης *Expansion*, σχήμα 3.16.

Για την καλύτερη κατανόηση των διαφόρων ειδών επέκτασης, θα εξηγήσουμε τί σημαίνουν οι όροι που εμφανίζονται ως συνθετικά στα ονόματα των ειδών επεκτάσεων. Ο όρος *Node* σημαίνει ότι το αντικείμενο που εξετάζεται θα παρουσιασθεί. Αντίστοιχα, ο όρος *No_Node* σημαίνει ότι ο όρος που εξετάζεται δεν θα παρουσιασθεί. Ο όρος *Expand* σημαίνει ότι το αντικείμενο που εξετάζεται θα επεκταθεί, ενώ ο όρος *No_Expand* σημαίνει ότι το αντικείμενο που εξετάζεται δεν θα επεκταθεί. Ομοίως, ο όρος *Hyperlink* σημαίνει ότι το αντικείμενο που εξετάζεται θα παρουσιασθεί ως υπερ-σύνδεσμος, δηλαδή το όνομα του αντικειμένου θα είναι ένας υπερ-σύνδεσμος, ενώ ο όρος *No_Hyperlink* σημαίνει ότι το αντικείμενο που εξετάζεται θα παρουσιασθεί χρησιμοποιώντας απλό κείμενο, δηλαδή το όνομα του αντικειμένου θα εμφανισθεί ως απλό κείμενο.

Κατά την περιγραφή των διαφορετικών επεκτάσεων θα χρησιμοποιηθεί το παράδειγμα που περιγράφηκε στην αρχή του κεφαλαίου.

- *Recursive_Expansion*,
  Σε αυτήν την περίπτωση η επέκταση είναι αναδρομική και πρέπει να προσδιορισθεί το βάθος της επέκτασης, δηλαδή σε πόσα επίπεδα θα εφαρμοστεί η επερώτηση που έχει προσδιορισθεί στην τρέχουσα οντότητα. Δεν υπάρχει κανένας περιορισμός για το βάθος της επέκτασης. Αν η επέκταση είναι οριζόντια, δηλαδή η επερώτηση προσδιορίζει τα γνωρίσματα της τρέχουσας οντότητας, τότε οι κόμβοι προορισμού των γνωρισμάτων θα επεκταθούν στα δικά τους γνωρίσματα μέχρι το βάθος να μηδενιστεί ή να υπάρξει κάποια οντότητα που δεν έχει γνωρίσματα ώστε να μη μπορεί να συνεχιστεί η επέκταση. Παρόμοια αν η επέκταση είναι κάθετη.

  Ας δούμε με ένα παράδειγμα πώς μπορεί να εφαρμοσθεί αυτό το είδος επέκτασης. Ένα γνώρισμα της κλάσης *Project* είναι τα συστήματα που χρησιμοποιεί και ο οργανισμός που χρηματοδοτεί το έργο. Ένα σύστημα μπορεί να αναπτυχθεί στα πλαίσια ενός έργου. Εφαρμόζοντας αναδρομική επέκταση βάθους 2 πάνω στα γνωρίσματα που έχουν ως αφετηρία ένα έργο Α, μπορούμε να πάρουμε ως αποτέλεσμα όλα τα συστήματα που χρησιμοποιεί, ο οργανισμός που το χρηματοδοτεί και οι τους συμμετέχοντες στο συγκεκριμένο έργο. Επιπλέον, για κάθε σύστημα που χρησιμοποιεί το Α παρουσιάζονται το έργο στα πλαίσια του οποίου αναπτύχθηκε, τον οργανισμό που το χρηματοδότησε και τους συμμετέχοντες. Για κάθε συμμετέχοντα του έργου Α, παρουσιάζονται οι δραστηριότητές του, τα έργα στα οποία συμμετέχει, οι δημοσιεύσεις του, το προσωπικό και τα συστήματα που έχει αναπτύξει. Εφαρμόζοντας την αναδρομή σε μεγαλύτερο βάθος, το κάθε ένα από τα παραπάνω επεκτείνεται στα γνωρίσματά του.

- *No_Expand_No_Node*,

Στην περίπτωση αυτή δεν εμφανίζεται ούτε επεκτείνεται καμία οντότητα του συνόλου στο οποίο έχει δηλωθεί η συγκεκριμένη επέκταση. Η ύπαρξη τους δηλαδή αποκρύπτεται. Για παράδειγμα, αν θεωρήσουμε ότι κείμενα τα οποία δεν έχουν δημοσιευθεί είναι μόνο για τους εσωτερικούς χρήστες του τομέα, τότε θέτουμε στην κλάση *Internal* αυτό το είδος επέκτασης με αποτέλεσμα να αποκρύπτεται η ύπαρξή του από τους εξωτερικούς χρήστες.

- *No_Expand_No_Hyperlink*,
  Στην περίπτωση που έχει δηλωθεί αυτό το γνώρισμα, κάθε μία από τις οντότητες του συνόλου που προσδιορίζει η επερώτηση δεν επεκτείνεται, αλλά παρουσιάζεται ως απλό κείμενο. Στο παράδειγμα, αν θέλουμε να παρουσιάσουμε όλους τους φοιτητές που απασχολούνται στα πλαίσια ενός τομέα, αλλά θέλουμε τα ονόματά τους να εμφανισθούν ως απλό κείμενο και όχι υπερ-σύνδεσμοι, χρησιμοποιούμε αυτό το είδος επέκτασης.

- *No_Expand_Hyperlink*,
  Σε αυτήν την περίπτωση κάθε μία από τις οντότητες του συνόλου, στο οποίο έχει δηλωθεί η συγκεκριμένη επέκταση, παρουσιάζεται ως υπερ-σύνδεσμος. Αν εφαρμόσουμε αυτήν την επέκταση στην κλάση *Publication* του παραδείγματος, τότε αν ο χρήστης επιλέξει αυτήν την κατηγορία, παρουσιάζονται όλες οι κατηγορίες δημοσιεύσεων ως υπερ-σύνδεσμοι. Με αυτόν τον τρόπο μπορεί ο χρήστης να δεί όλες τις δημοσιεύσεις ανά κατηγορία.

- *Expand_No_Node*,
  Σε πολλές περιπτώσεις θέλουμε να μην εμφανισθούν τα αντικείμενα αλλά να επεκταθούν. Η επέκταση *Expand_No_Node* προσδιορίζει ότι τα αντικείμενα του συνόλου, στο οποίο προσδιορίζεται η επέκταση, θα επεκταθούν χωρίς να παρουσιασθούν.

  Για να γίνει κατανοητό αυτό το είδος επέκτασης, ας εξετάσουμε το παράδειγμα. Έστω ότι θέλουμε να παρουσιάσουμε τη διεύθυνση του Τομέα Πληροφοριακών Συστημάτων και Τεχνολογίας Λογισμικού. Στο σχήμα 3.2 φαίνεται ότι το αντικείμενο *Address* έχει ως γνωρίσματα την πόλη, την οδό, τη χώρα και τον ταχυδρομικό κώδικα. Για την παρουσίαση της διεύθυνσης αυτό που μας ενδιαφέρει είναι τα γνωρίσματά της και όχι το αντικείμενο *Address*. Επομένως, θα πρέπει να δηλώσουμε ότι θέλουμε να χρησιμοποιηθεί επέκταση τύπου *Expand_No_Node* στην κλάση *Address*

- *Expand_Node*,
  Η λειτουργία είναι ίδια με του *Expand_No_Node* με τη διαφορά ότι το κάθε μέλος του συνόλου παρουσιάζεται και επεκτείνεται. Η χρησιμότητα αυτού του είδους επέκτασης γίνεται κατανοητή αν θεωρήσουμε ότι η παρουσίαση του προσωπικού ενός τομέα γίνεται έτσι ώστε να ομαδοποιούνται οι εργαζόμενοι ανά κατηγορία στην οποία ανήκουν. Επομένως, αν θέσουμε στις υποκατηγορίες της κλάσης

Personnel αυτού του είδους την επέκταση, θα παρουσιασθούν οι υποκατηγορίες με τα μέλη τους.

Η επέκταση όπως έχει περιγραφεί ως τώρα αναφέρεται σε ένα σύνολο αντικειμένων. Υπάρχουν περιπτώσεις όπου είναι απαραίτητη η παρουσίαση ενός αντικειμένου με τρόπο σταθερό, ανεξαρτήτως του τί έχει δηλωθεί για το σύνολο στο οποίο ανήκει. Για τέτοιες περιπτώσεις μπορεί να χρησιμοποιηθεί το γνώρισμα *over_object* που έχει ως αφετηρία την κλάση *Expansion*. Ο προσδιορισμός επέκτασης πάνω σε ένα αντικείμενο υπερκαλύπτει την επέκταση που έχει δηλωθεί για το σύνολο στο οποίο ανήκει. Ας δούμε με ένα παράδειγμα την εφαρμογή του. Έστω ένα έργο, το *Aquarelle* για το οποίο υπάρχει επίσημος κόμβος στον Παγκόσμιο Ιστό (official web site). Στην παρουσίαση θέλουμε κάθε φορά που παρουσιάζεται το έργο *Aquarelle*, να εμφανίζεται ως υπερ-σύνδεσμος που ''δείχνει'' στην επίσημη διεύθυνση. Αυτό μπορεί να επιτευχθεί χρησιμοποιώντας επέκταση πάνω στο συγκεκριμένο αντικείμενο, τύπου *No_Expand_Hyperlink*.

Τυπική επέκταση είναι η *No_Expand_Hyperlink*.

3.3.5 Δεδομένα παρουσιάσεων

Η σημασιολογική δομή περιγράφει την πληροφορία που πρέπει να παρουσιασθεί σύμφωνα με τη λογική δομή χρησιμοποιώντας το είδος διαμόρφωσης κειμένου που προσδιορίζει η φυσική δομή.

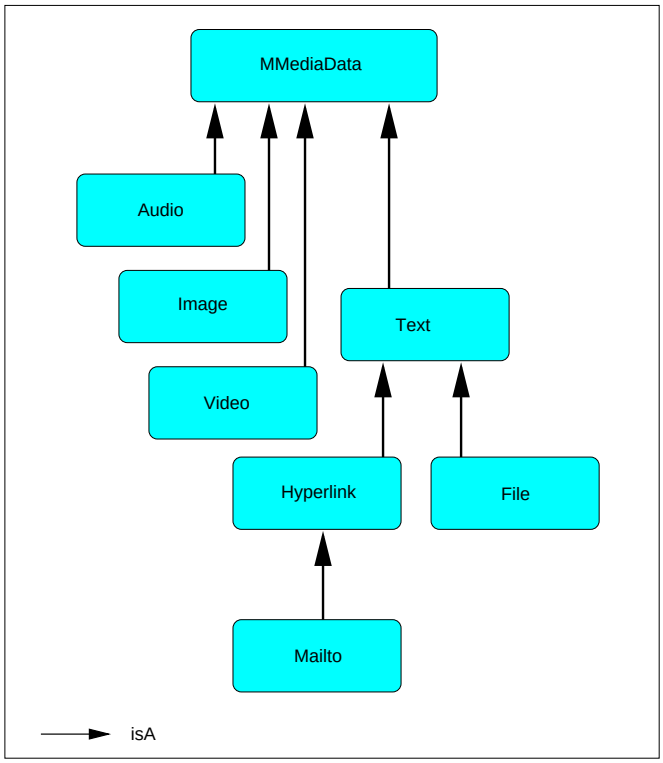
Για τον προσδιορισμό της πληροφορίας που περιέχει ένα τμήμα της HTML σελίδας χρησιμοποιείται η κλάση *BodyElement*. Αυτή η κλάση έχει δύο υποκλάσεις ανάλογα με το είδος των δεδομένων που μπορούν να περιέχονται σε ένα τμήμα μιας HTML σελίδας. Οι υποκλάσεις είναι *MMediaData*, που αντιστοιχεί στα δεδομένα πολυμέσων και *FactualData*, που αντιστοιχεί σε δεδομένα που βρίσκονται αποθηκευμένα στη βάση.

3.3.5.1 Δεδομένα πολυμέσων

Τα είδη των δεδομένων πολυμέσων είναι βίντεο, εικόνα, ήχος και κείμενο. Ενώ το κείμενο μπορεί να είναι υπερ-σύνδεσμος ή κείμενο που βρίσκεται αποθηκευμένο σε αρχείο συστήματος. Μια ειδική κατηγορία υπερ-συνδέσμου είναι ο υπερ-σύνδεσμος που προσδιορίζει μια ηλεκτρονική διεύθυνση.

Για την περιγραφή των παραπάνω στη σημασιολογική δομή, έχουν ορισθεί ως υποκλάσεις της κλάσης *MMediaData* οι κλάσεις, σχήμα 3.17 :

- *Audio*,
- *Image*,



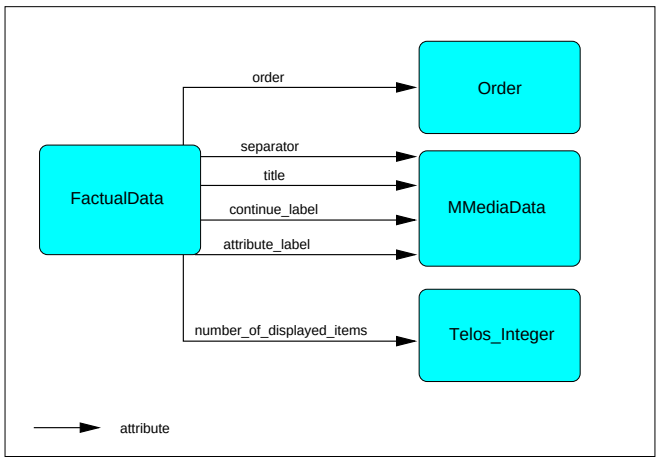
Σχήμα 3.17: Η κλάση *MMediaData* και οι υποκλάσιες της

- *Text*,
  Η κλάση *Text* έχει υποκλάσεις τις *Hyperlink*, και *File*.
- *Video*

3.3.5.2 Δεδομένα αποθηκευμένα στη βάση

Για τον προσδιορισμό των δεδομένων που είναι αποθηκευμένα στη βάση, πρέπει να προσδιορισθεί το σύνολο στο οποίο ανήκουν. Για την παρουσίαση των δεδομένων πρέπει να καθορισθεί η σειρά, δηλαδή αν θα παρουσιασθούν αλφαβητικά ή όπως τα επιστρέφει ο εξυπηρέτης της βάσης, αν μεταξύ δύο διαδοχικών αντικειμένων προς παρουσίαση πρέπει να συμπεριληφθεί κάποιο δεδομένο πολυμέσων και αν το τμήμα με το σύνολο των δεδομένων θα έχει κάποιον τίτλο. Για τον καθορισμό αυτών των χαρακτηριστικών, έχουν αποδοθεί στην κλάση *FactualData* τα γνωρίσματα *order*, *separator* και *title*. Το γνώρισμα *order* έχει ως προορισμό την κλάση *Order* ενώ τα γνωρίσματα *separator* και *title* έχουν ως προορισμό την κλάση *MMediaData*, σχήμα 3.18.

Τα ατομικά γνωρίσματα στην Telos δεν είναι απαραίτητο να έχουν λογικό όνομα. Αν το σύνολο των δεδομένων που προσδιορίσθηκε περιέχει γνωρίσματα χωρίς λογικό



Σχήμα 3.18: Η κλάση *FactualData* και τα γνωρίσματα που της έχουν αποδοθεί

όνομα, μπορούμε να ορίσουμε ένα δεδομένο πολυμέσων το οποίο θα χρησιμοποιηθεί, στη θέση του λογικού ονόματος που απουσιάζει, για την παρουσίαση των ατομικών γνωρισμάτων. Αυτή η δήλωση γίνεται μέσω του γνωρίσματος *attribute_label* που έχει ως προορισμό την κλάση *MMediaData*, σχήμα 3.18.

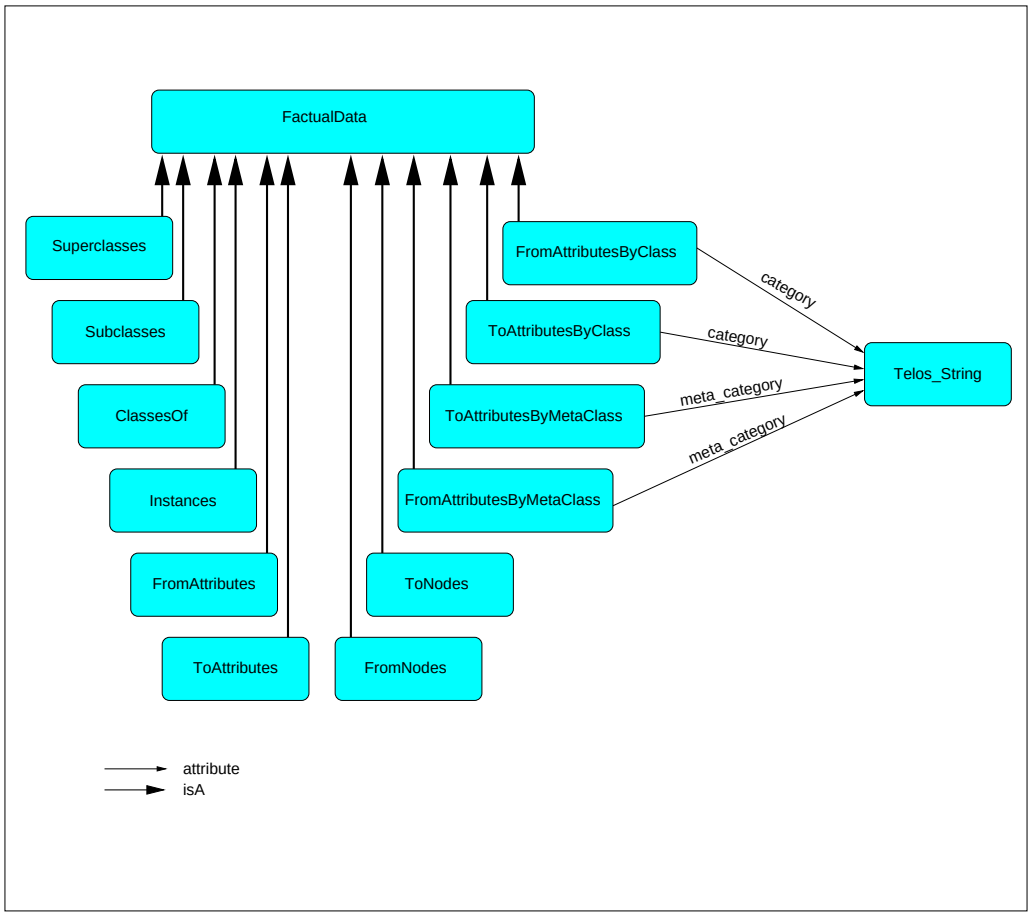
Τέλος, το σύνολο των δεδομένων μπορεί να έχει πολύ μεγάλο πληθικό αριθμό, δηλαδή να περιέχει πάρα πολλά δεδομένα. Σε αυτές τις περιπτώσεις πρέπει να παρέχεται η δυνατότητα στον προγραμματιστή εφαρμογής να δηλώσει έναν αριθμό αποτελεσμάτων που θα παρουσιασθούν, ενώ για την παρουσίαση όλων των αντικειμένων του συνόλου να μπορεί να προσδιορισθεί το συγκεκριμένο σύνολο και να παρουσιασθεί μόνο αυτό σε διαφορετική σελίδα HTML. Το γνώρισμα *number_of_displayed_items* προσδιορίζει τον αριθμό των αποτελεσμάτων που θα παρουσιασθούν. Το γνώρισμα *continue_label* προσδιορίζει ένα δεδομένο πολυμέσων που θα εμφανισθεί αν ο πληθικός αριθμός του συνόλου αποτελεσμάτων είναι μεγαλύτερος από τον αριθμό που προσδιορίζει το γνώρισμα *number_of_displayed_items* και θα είναι ένας υπερ-σύνδεσμος, που προσδιορίζει την επερώτηση που είχε ως αποτέλεσμα το συγκεκριμένο σύνολο αποτελεσμάτων. Αν ακολουθηθεί αυτός ο υπερ-σύνδεσμος, θα εμφανισθούν μόνο τα αντικείμενα του συγκεκριμένου συνόλου αποτελεσμάτων. σχήμα 3.18.

Ανάλογα με το είδος των οντοτήτων που σχετίζονται με την οντότητα στην οποία αναφέρεται η παράσταση, διακρίνονται οι εξής υποκατηγορίες, σχήμα 3.19 :

- *Superclasses*,
  Οι υπερ-κλάσεις της οντότητας.

- *Subclasses*,
  Οι υποκλάσεις της οντότητας.

- *ClassesOf*,



Σχήμα 3.19: Η κλάση *FactualData* και οι υποκλάσεις της

Οι κλάσεις στις οποίες είναι περίπτωση η οντότητα.

- *Instances*,
  Οι περιπτώσεις της οντότητας.
- *FromAttributes*,
  Οι σύνδεσμοι απόδοσης γνωρίσματος που έχουν ως αφετηρία την οντότητα.
- *ToAttributes*,
  Οι σύνδεσμοι απόδοσης γνωρίσματος που έχουν ως προορισμό την οντότητα.
- *FromAttributesByClass*,
  Οι σύνδεσμοι απόδοσης γνωρίσματος που έχουν ως αφετηρία την οντότητα και είναι περιπτώσεις μιας συγκεκριμένης κλάσης γνωρισμάτων. Δέχεται ως όρισμα το όνομα της κλάσης γνωρισμάτων.
- *ToAttributesByClass*,
  Οι σύνδεσμοι απόδοσης γνωρίσματος που έχουν ως προορισμό την οντότητα και

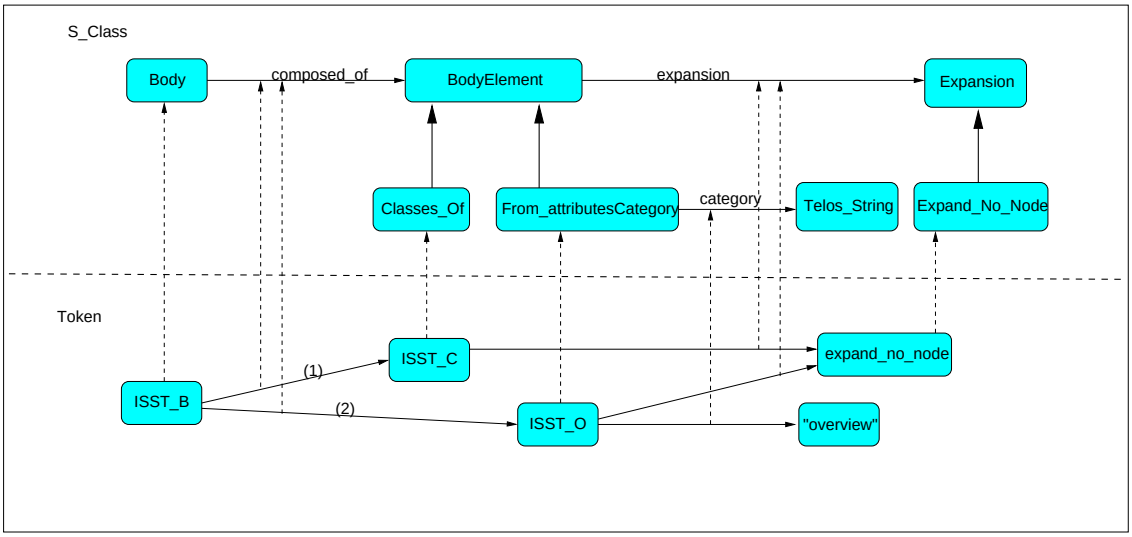
είναι περιπτώσεις μιας συγκεκριμένης κλάσης γνωρισμάτων. Δέχεται ως όρισμα το όνομα της κλάσης γνωρισμάτων.

- *FromAttributesByMetaClass*,
  Οι σύνδεσμοι απόδοσης γνωρίσματος που έχουν ως αφετηρία την οντότητα και είναι περιπτώσεις κλάσης γνωρισμάτων που είναι μέλος μιας συγκεκριμένης μετα-κλάσης γνωρισμάτων. Δέχεται ως όρισμα το όνομα της μετα-κλάσης γνωρισμάτων.

- *ToAttributesByMetaClass*,
  Οι σύνδεσμοι απόδοσης γνωρίσματος που έχουν ως προορισμό την οντότητα και είναι περιπτώσεις κλάσης γνωρισμάτων που είναι μέλος μιας συγκεκριμένης μετα-κλάσης γνωρισμάτων. Δέχεται ως όρισμα το όνομα της μετα-κλάσης γνωρισμάτων.

- *FromNodes*,
  Οι κόμβοι που συνδέονται με την τρέχουσα οντότητα μέσω συνδέσμων απόδοσης γνωρίσματος που έχουν ως αφετηρία την τρέχουσα οντότητα.

- *ToNodes*,
  Οι οντότητες που συνδέονται με την τρέχουσα οντότητα μέσω συνδέσμων απόδοσης γνωρίσματος που έχουν ως προορισμό την τρέχουσα οντότητα.



Σχήμα 3.20: Παράδειγμα κυρίου τμήματος μιας σελίδας HTML

Στο σημείο αυτό μπορούμε να συνεχίσουμε το παράδειγμα και να περιγράψουμε το αντικείμενο *ISST_B* του σχήματος 3.12. Στο κύριο μέρος της σελίδας HTML πρέπει να εμφανίζονται ως υπερ-σύνδεσμοι οι κατηγορίες *Δραστηριότητες*, *Συστήματα*, *Δημοσιεύσεις*, *Έργα* και *Προσωπικό* και μια σύντομη περιγραφή του τομέα. Οι κατηγορίες που πρέπει να εμφανισθούν ως υπερ-σύνδεσμοι είναι τα αντικείμενα με τα οποία συνδέεται η

κλάση *Division* μέσω συνδέσμων απόδοσης γνωρίσματος και οι σύνδεσμοι αυτοί έχουν ως αφετηρία την κλάση *Division*. Επιπλέον, η σύντομη περιγραφή του τομέα είναι συσχετισμένη με τον τομέα μέσω συνδέσμου απόδοσης γνωρίσματος της κατηγορίας *overview*. Με βάση τα παραπάνω, αποδίδουμε στο αντικείμενο *ISST_B*, που παριστάνει το κύριο μέρος της σελίδας, τα γνωρίσματα με λογικό όνομα *(1)* και *(2)*. Αυτά τα γνωρίσματα είναι περιπτώσεις της κλάσης γνωρισμάτων *composed_of*. Ο λόγος ονοματοδοσίας τους και μάλιστα με αριθμούς είναι επειδή θέλουμε στο τελικό αποτέλεσμα να εμφανιστούν πρώτα οι υπερ-σύνδεσμοι και μετά η σύντομη περιγραφή. Τα αντικείμενα προορισμού των γνωρισμάτων *(1)* και *(2)* είναι αντίστοιχα τα αντικείμενα *ISST_C*, που είναι περίπτωση της κλάσης *ClassesOf*, και *ISST_O*, που είναι περίπτωση της κλάσης *FromAttributesByClass*. Το αντικείμενο *ISST_O* έχει ένα γνώρισμα που είναι μέλος της κλάσης γνωρισμάτων *category* και έχει ως κόμβο προορισμού το αλφαριθμιτικό *overview*. Το γνώρισμα αυτό χρησιμοποιείται για τον προσδιορισμό της κλάσης γνωρισμάτων βάση της οποίας θα γίνει η επερώτηση. Τέλος θέτουμε ως επέκταση στα αντικείμενα *ISST_C* και *ISST_O* το είδος *Expand_No_Node*. Αυτό έχει ως αποτέλεσμα την επέκταση των αντικειμένων που προσδιορίζουν οι επερωτήσεις χωρίς την εμφάνισή τους. Ο τρόπος σύμφωνα με τον οποίο θα γίνει η επέκταση προσδιορίζεται από τα ίδια τα αντικείμενα. Οι επερωτήσεις που θέσαμε σ'αυτό το παράδειγμα, έχουν ως αποτελέσματα την κλάση *Division* και το αντικείμενο *ISST_Overview* αντίστοιχα. Τα παραπάνω φαίνονται στο σχήμα 3.20. Η παρουσίαση της κλάσης *Division* και του αντικειμένου *ISST_Overview* ορίζεται με αντίστοιχο τρόπο.

Στην ενότητα 3.3.3 τονίσαμε ότι μπορούμε να ορίσουμε όψεις στις λογικές παραστάσεις. Οι δύο παρουσιάσεις που περιγράφηκαν, δηλαδή με χρήση πλαισίων και χωρίς, είναι δύο όψεις των ίδιων λογικών παραστάσεων. Στο παράδειγμα με τη χρήση πλαισίων, το περιεχόμενο του πρώτου πλαισίου προσδιοριζόταν ως μια κλήση της πύλης διασύνδεσης για την παρουσίαση της κλάσης *Division*. Όμοια το περιεχόμενο του δεύτερου πλαισίου προσδιοριζόταν ως μια κλήση της πύλης διασύνδεσης για την παρουσίαση του αντικειμένου *ISST_Overview*. Συγκρίνοντας τα δύο παραδείγματα, παρατηρούμε ότι και στις δύο περιπτώσεις παρουσιάζεται η ίδια πληροφορία και χρησιμοποιούνται οι ίδιες λογικές παραστάσεις, δηλαδή οι παρουσιάσεις των *Division* και *ISST_Overview*, αλλά σύμφωνα με διαφορετική όψη.

3.3.5.3 Η κλάση This

Σε πολλές περιπτώσεις είναι αναγκαία η ύπαρξη ενός μηχανισμού προσδιορισμού του τρέχοντος κάθε φορά αντικειμένου, χωρίς να χρησιμοποιείται το αναγνωριστικό του, που στην περίπτωση της Telos είναι το λογικό όνομά του. Για την κάλυψη αυτών των περιπτώσεων έχει εισαχθεί η κλάση *This*, που προσδιορίζει το τρέχον κάθε φορά αντικείμενο. Η σημασία της δηλαδή, είναι όμοια με τον τελεστή this στη γλώσσα προγραμματισμού C++. Υποκλάσεις αυτής της κλάσης είναι :

- *LogicalName*,
  που προσδιορίζει το λογικό όνομα της τρέχουσας οντότητας.

- *LogicalNameFromNode*,
  που προσδιορίζει το λογικό όνομα της οντότητας αφετηρίας, αν η τρέχουσα οντότητα είναι γνώρισμα.

- *LogicalNameToNode*,
  που προσδιορίζει το λογικό όνομα της οντότητας προορισμού, αν η τρέχουσα οντότητα είναι γνώρισμα.

- *Type*,
  που προσδιορίζει τον τύπο της τρέχουσας οντότητας, δηλαδή αν είναι κόμβος ή γνώρισμα.

- *Level*,
  που προσδιορίζει το επίπεδο ταξινόμησης της τρέχουσας οντότητας.

Για να καταλάβουμε τη χρήση της κλάσης *This* θα ακολουθήσουμε πάλι το παράδειγμα. Έστω ότι ορίζουμε κοινό τρόπο παρουσίασης όλων των δημοσιεύσεων, όμως θέλουμε κάθε φορά να φαίνεται ως τίτλος της σελίδας HTML το είδος της δημοσίευσης, δηλαδή η αντίστοιχη υποκλάση της κλάσης *Publication*. Για να το πετύχουμε αυτό θέτουμε ως τίτλο στο αντίστοιχο αντικείμενο, ένα αντικείμενο που είναι περίπτωση της κλάσης *LogicalName* (που είναι υποκλάση της κλάσης *This*) και της κλάσης *Text*. Έτσι προσδιορίσαμε ότι ο τίτλος είναι κείμενο και η τιμή του είναι το λογικό όνομα της τρέχουσας οντότητας.

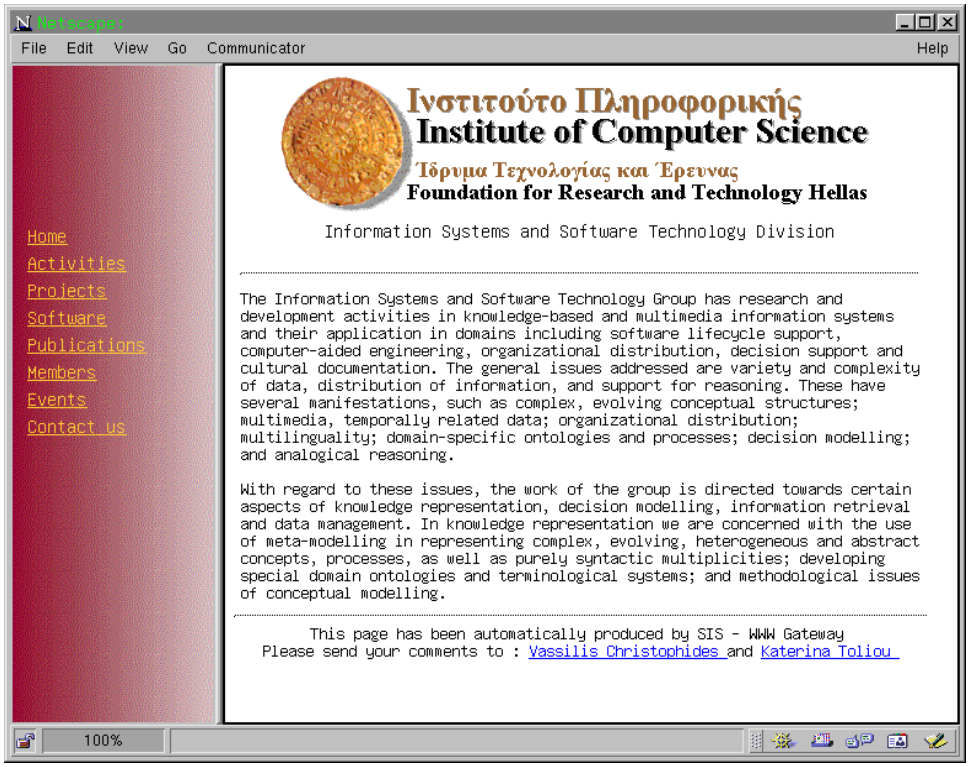
3.3.6 Προδηλωμένη Παρουσίαση

Για τις περιπτώσεις που δεν έχει δηλωθεί μοντέλο παρουσίασης για ένα αντικείμενο, δεν υπάρχει μοντέλο παρουσίασης που να κληρονομείται και δεν έχει ορισθεί τυπική (default) παράσταση για τη συγκεκριμένη όψη, υπάρχει μία προδηλωμένη παρουσίαση η οποία είναι αποθηκευμένη στη βάση.

Στην προδηλωμένη παρουσίαση χρησιμοποιείται ο προδηλωμένος μορφότυπος *Unordered List* και παρουσιάζονται όλα τα αντικείμενα που συνδέονται με το τρέχον κάθε φορά αντικείμενο, μέσω συνδέσμων ταξινόμησης, γενίκευσης και απόδοσης γνωρίσματος. Όλα τα αντικείμενα παρουσιάζονται ως υπερ-σύνδεσμοι για εύκολη πλοήγηση, δηλαδή χρησιμοποιείται το είδος επέκτασης *No_Expand_Hyperlink*. Επιπλέον παρουσιάζονται το επίπεδο ταξινόμησης και ο τύπος (Individual ή Attribute) κάθε αντικειμένου. Το λογικό όνομα του αντικειμένου εμφανίζεται ως τίτλος της παρουσίασης.

3.4 Παράδειγμα κατασκευής ιστοσελίδας

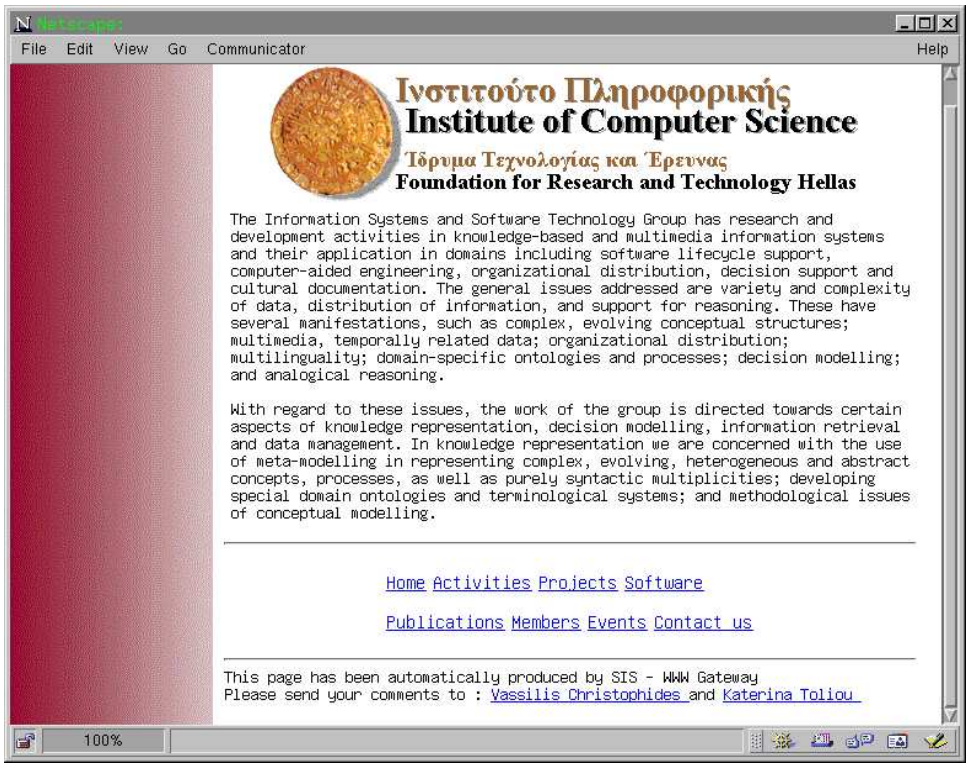
Με βάση το παράδειγμα και χρησιμοποιώντας το μοντέλο παρουσίασης που ορίσθηκε, κατασκευάσθηκαν δύο σελίδες HTML. Πρόκειται για την αρχική σελίδα παρουσίασης σε HTML του Τομέα Πληροφοριακών Συσημάτων και Τεχνολογίας Λογισμικού, με και χωρίς χρήση πλαισίων. Οι ιστοσελίδες παρουσιάζονται στα σχήματα 3.21 και 3.22, όπου στην πρώτη χρησιμοποιούνται πλαίσια ενώ στη δεύτερη όχι. Οι διαφορές από το μοντέλο που περιγράφηκε ως παράδειγμα είναι ότι το άνω μέρος της σελίδας περιέχει το λογότυπο του ΙΠ-ΙΤΕ και ένα κείμενο, το όνομα του τομέα, ενώ στο κάτω μέρος εμφανίζεται ένα κείμενο που δηλώνει ότι η ιστοσελίδα κατασκευάστηκε από το σύστημα. Όπως μπορούμε να παρατηρήσουμε και στο σχήμα 3.21, όταν χρησιμοποιού-



Σχήμα 3.21: Η πρώτη ιστοσελίδα με τη χρήση πλαισίων

νται πλαίσια, η ιστοσελίδα διακρίνεται σε δύο τμήματα αυτόνομα που έχουν τα δικά τους χαρακτηριστικά, όπως το χρώμα των υπερ-συνδέσμων και το φόνο της ιστοσελίδας. Το ένα τμήμα καταλαμβάνει το αριστερό μέρος της ιστοσελίδας, ενώ το δεύτερο τμήμα καταλαμβάνει το υπόλοιπο μέρος. Το κάθε τμήμα ενεργοποιείται, όταν επιλεγεί με τη βοήθεια του ποντικιού. Κάθε φορά ένα τμήμα μπορεί να είναι ενεργοποιημένο. Αντίθετα, η ιστοσελίδα χωρίς πλαίσια, που φαίνεται στο σχήμα 3.22, αποτελείται από ένα τμήμα.

Στην ιστοσελίδα με πλαίσια οι κατηγορίες πληροφορίας καταλαμβάνουν το αρι-



Σχήμα 3.22: Η πρώτη ιστοσελίδα χωρίς πλαίσια

στερό μέρος, που είναι και ξεχωριστό τμήμα. Η επιλογή μιας κατηγορίας πληροφορίας έχει ως αποτέλεσμα να εμφανιστεί η αντίστοιχη ιστοσελίδα στο δεξί τμήμα της ιστοσελίδας. Μ'αυτόν τον τρόπο, το αριστερό τμήμα της ιστοσελίδας παραμένει σταθερό, ενώ το δεξί τμήμα αλλάζει ανάλογα με την πληροφορία που παρουσιάζεται.

Στην ιστοσελίδα χωρίς πλαίσια οι κατηγορίες πληροφορίας βρίσκονται στο κάτω μέρος της σελίδας. Η επιλογή μιας κατηγορίας έχει ως αποτέλεσμα να αλλάξει όλη η ιστοσελίδα.

3.5 Παρατηρήσεις - Συμπεράσματα

Το μοντέλο παρουσίασης που περιγράφηκε επιτρέπει την αναδίφηση και παρουσίαση σε HTML των οντοτήτων μιας βάσης που έχει κατασκευασθεί με το ΣΣΕ. Δεν μπορεί να χρησιμοποιηθεί για παρουσίαση αποτελεσμάτων επερωτήσεων, αν και ένας τέτοιος μηχανισμός είναι εύκολο να υποστηριχθεί επεκτείνοντας το υπάρχον μοντέλο. Για την ανάκτηση πληροφορίας από τη βάση δεδομένων, θα μπορούσε να χρησιμοποιηθεί το πρωτόκολλο Z39.50[60]. Για το ΣΣΕ έχει κατασκευασθεί ένας εξυπηρέτης εφαρμογής που υποστηρίζει το πρωτόκολλο Z39.50[61]. Επομένως, συνδυάζοντας τα δύο

αυτά συστήματα μπορεί να υποστηριχθεί και η διαλογική ανάκτηση πληροφορίας από μια βάση που έχει κατασκευασθεί με το ΣΣΕ και η παρουσίασή της σε HTML.

Το μοντέλο πληρεί τα χαρακτηριστικά που θέσαμε στο τέλος του δεύτερου κεφαλαίου και συγκεκριμένα :

- Είναι ευέλικτο
  μπορεί να ορισθεί παρουσίαση σε κάθε αντικείμενο της βάσης, κόμβο ή γνώρισμα, ανεξάρτητα του επιπέδου ταξινόμησης στο οποίο βρίσκεται

- Είναι επεκτάσιμο
  η επεκτασιμότητα προσδιορίζεται από :

  ▷ τη δυνατότητα παρουσίασης πολλών διαφορετικών όψεων σε HTML του ίδιου αντικειμένου

  ▷ τη δυνατότητα παρουσίασης του ίδιου αντικειμένου χρησιμοποιώντας διαφορετικά είδη διαμόρφωσης κειμένου, όπως XML, Latex και απλό κείμενο

  ▷ τη δυνατότητα που παρέχεται στον προγραμματιστή εφαρμογής παράκαμψης των περιορισμών της HTML χρησιμοποιώντας Java applets και scripts

- Είναι δηλωτικό
  ο προγραμματιστής εφαρμογής περιγράφει με δηλωτικό τρόπο την λογική δομή της παρουσίασης.

- Προσφέρει τη δυνατότητα ορισμού όψεων πάνω στα αντικείμενα του μοντέλου παρουσίασης

- Υποστηρίζει ιστορία ενός βήματος, χρησιμοποιώντας το *coming_from*. Το σύστημα ‘‘θυμάται’’ την πρόσφατη ιστορία και ενεργεί σύμφωνα με αυτή

Κεφάλαιο 4

Πύλη Διασύνδεσης ΣΣΕ - Παγκόσμιου Ιστού

Σε αυτό το κεφάλαιο περιγράφεται η υλοποίηση της πύλης διασύνδεσης του ΣΣΕ με τον Παγκόσμιο Ιστό. Αρχικά, εξετάζουμε τον τρόπο εικοινωνίας μιας εξωτερικής εφαρμογής με το ΣΣΕ. Στη συνέχεια, περιγράφεται η υλοποίηση και παρατίθεται μια σύγκριση της υλοποίησης που πραγματοποιήσαμε με τα συστήματα που περιγράφηκαν στο δεύτερο κεφάλαιο. Η σύγκριση εστιάζεται σε συστήματα που εμφανίζουν ομοιότητες ως προς την αρχιτεκτονική ή τη λογική ανάπτυξης με την προσέγγισή μας.

4.1 Πύλη Διασύνδεσης ΣΣΕ-ΠΙ

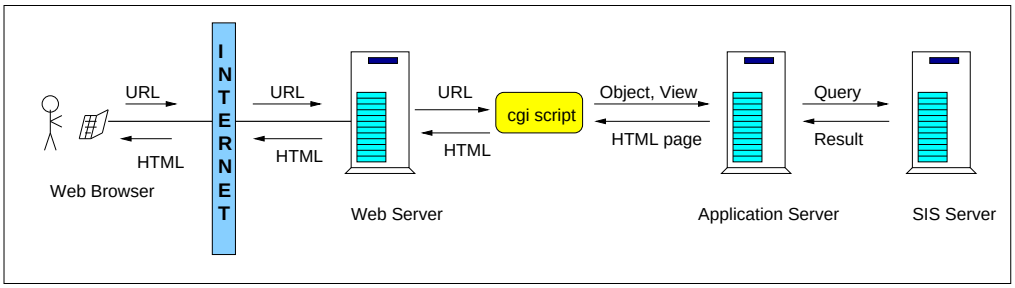
Με βάση τα όσα περιγράφησαν στο δεύτερο κεφάλαιο είναι κατανοητό ότι η αρχιτεκτονική τριών επιπέδων προσφέρει πολλά πλεονεκτήματα και είναι προτιμότερη από την αρχιτεκτονική δύο επιπέδων. Επιπλέον, κατά τη σύγκριση των διαθέσιμων αρχιτεκτονικών υλοποίησης μιας πύλης διασύνδεσης βάσεων δεδομένων με τον παγκόσμιο ιστό, αναφέρθηκε ότι ακολουθήσαμε την αρχιτεκτονική CGI Application Server λόγω ευελιξίας και εύκολης μελλοντικής μετάβασης σε αρχιτεκτονική Server API. Συνεπώς, η αρχιτεκτονική υλοποίησης που ακολουθήθηκε στο σύστημά μας είναι η αρχιτεκτονική CGI Application Server όπου ο ενδιάμεσος εξυπηρέτης είναι η πύλη διασύνδεσης. Στη συνέχεια θα περιγραφεί το τμήμα της υλοποίησης που πρέπει να αντικατασταθεί σε μία τέτοια μετάβαση.

Όπως φαίνεται και στο σχήμα 4.1, ο χρήστης στέλνει την αίτηση του με τη μορφή ενός URL μέσω ενός αναδιφητή του Παγκόσμιου Ιστού (Web Browser) στον εξυπηρέτη του Παγκόσμιου Ιστού. Το URL είναι της μορφής ‘‘http://<server-name>/<cgi-name>?<ObjectName>+<View>+[<Query>+<State>]’’, όπου <server-name> είναι η διεύθυνση του εξυπηρέτη του Παγκόσμιου Ιστού. Το <cgi-name> είναι το όνομα του εκτελέσιμου cgi προγράμματος που δέχεται ως όρισμα το λογικό όνομα του αντικειμέ-

νου, <ObjectName>, για το οποίο πρέπει να παραχθεί η παρουσίαση σε κείμενο HTML. Η παρουσίαση θα παραχθεί σύμφωνα με την όψη <View>. Τα ορίσματα <Query> και <State> είναι προαιρετικά. Το <Query> προσδιορίζει μια επερώτηση. Χρησιμοποιείται όταν ο προγραμματιστής εφαρμογής έχει δηλώσει στο μοντέλο παρουσίασης ότι μόνο ένας αριθμός αποτελεσμάτων πρέπει να παρουσιάζεται ως το αποτέλεσμα μιας επερώτησης, μέσω του γνωρίσματος *number_of_displayed_items*. Για την παρουσίαση όλων των αποτελεσμάτων της επερώτησης χρησιμοποιείται αυτός ο τρόπος. Το <State> χρησιμοποιείται όταν η παρουσίαση του αντικειμένου <ObjectName> εξαρτάται από το προηγούμενο αντικείμενο που παρουσιάσθηκε. Ουσιαστικά πρόκειται για την ‘‘μνήμη’’ του συστήματος.

Ο εξυπηρέτης του Παγκόσμιου Ιστού δημιουργεί μια εικόνα του εκτελέσιμου cgi το οποίο και αποκτά τον έλεγχο της εκτέλεσης. Το εκτελέσιμο cgi ανακτά από το URL τις παραμέτρους εκτέλεσης. Στη συνέχεια επικοινωνεί με τον εξυπηρέτη εφαρμογής (Application Server) όπου στέλνει μέσω του πρωτοκόλλου TCP/IP τα παραπάνω ορίσματα και αναμένει την απάντηση. Αν δεν υπάρχει κάποιο λάθος, ως απάντηση δέχεται το ζητούμενο κείμενο HTML.

Ο εξυπηρέτης εφαρμογής αφού λάβει την αίτηση από τον εξυπηρετούμενο, δηλαδή το εκτελέσιμο cgi, παράγει το ζητούμενο κείμενο HTML. Τμηματικά στέλνει το παραγόμενο κείμενο στον εξυπηρετούμενο ο οποίος το στέλνει στο χρήστη μέσω του εξυπηρέτη του Παγκόσμιου Ιστού. Ο εξυπηρέτης εφαρμογής για να παραγάγει τη ζητούμενη σελίδα HTML επικοινωνεί με τον εξυπηρέτη της βάσης (SIS Server).

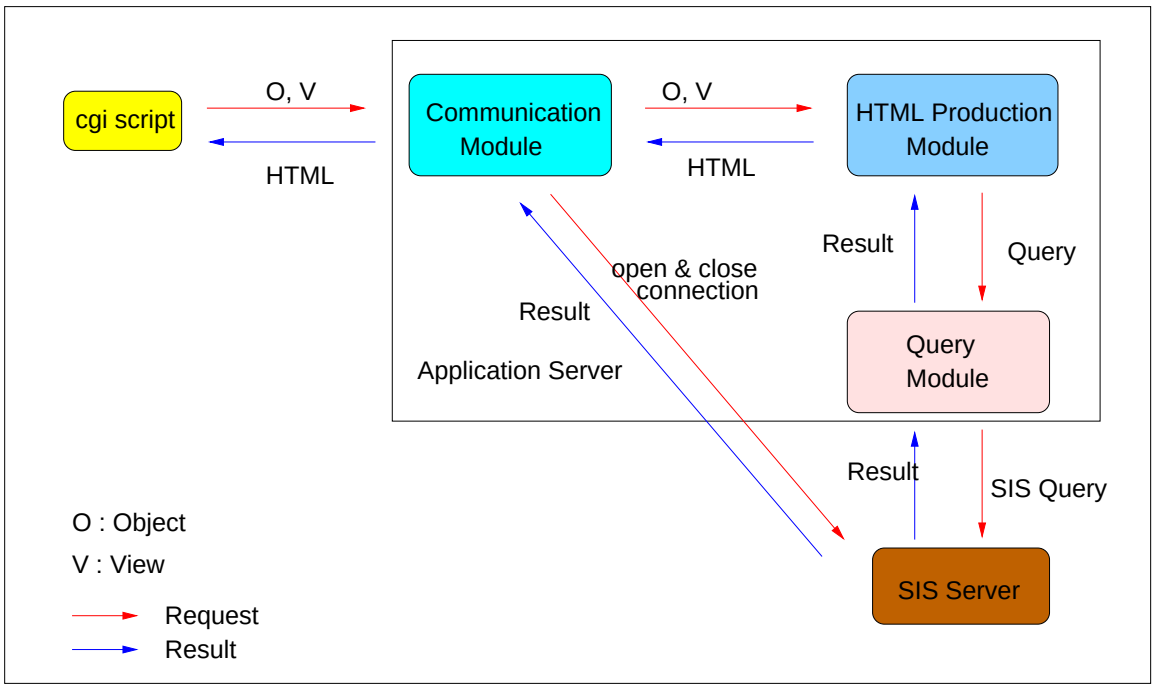


Σχήμα 4.1: Γενική άποψη της αρχιτεκτονικής που υλοποιήθηκε

Το τμήμα στο οποίο θα επικεντρώσουμε την προσοχή μας είναι ο εξυπηρέτης εφαρμογής.

4.1.1 Εξυπηρέτης εφαρμογής

Όπως φαίνεται και στο σχήμα 4.2 ο εξυπηρέτης εφαρμογής αποτελείται από τρία βασικά τμήματα : το τμήμα που επικοινωνεί με τους εξυπηρετούμενους μεταφέροντας



Σχήμα 4.2: Ο εξυπηρέτης εφαρμογής

τις αιτήσεις και απαντήσεις (Communication Module), το τμήμα που παράγει την HTML σελίδα (HTML Production Module) και το τμήμα που επικοινωνεί με τον εξυπηρέτη της βάσης (Query Module).

4.1.1.1 Τμήμα επικοινωνίας

Η επικοινωνία μεταξύ του εξυπηρέτη εφαρμογής και των εξυπηρετουμένων γίνεται με ανταλλαγή μηνυμάτων (Message Oriented Middleware). Τα μηνύματα ανταλάσσονται μέσω σύνδεσης τύπου TCP/IP. Ο εξυπηρετούμενος συνδέεται στον εξυπηρέτη και στέλνει την αίτησή του. Η αίτηση προσδιορίζει υποχρεωτικά το αντικείμενο, για το οποίο ζητείται η παρουσίαση, και την όψη, σύμφωνα με την οποία πρέπει να παραχθεί η παρουσίαση, και, προαιρετικά, την επερώτηση που πρέπει να εκτελεστεί (Query) και/ή την τρέχουσα κατάσταση (State). Στη συνέχεια ο εξυπηρετούμενος περιμένει την απάντηση του εξυπηρέτη. Επομένως, για την επικοινωνία μεταξύ του εξυπηρετούμενου και του εξυπηρέτη χρησιμοποιείται ο σύγχρονος τρόπος επικοινωνίας (synchronous communication mode). Ο εξυπηρέτης μόλις δεχθεί την αίτηση ενός εξυπηρετούμενου, ανακτά τις αντίστοιχες πληροφορίες και καλεί την αντίστοιχη διαδικασία του τμήματος παραγωγής της σελίδας HTML. Μόλις αρχίσει να παράγεται η ζητούμενη σελίδα HTML, στέλνεται τμηματικά στον εξυπηρετούμενο.

Για την εξυπηρέτηση ταυτόχρονων αιτήσεων χρησιμοποιούνται Light Weight

Processes (LWPs ή threads). Από εδώ και στο εξής με τον όρο ‘‘διεργασία’’ θα αναφερόμαστε σε LWP. Για κάθε νέα αίτηση δημιουργείται μια νέα διεργασία η οποία αναλαμβάνει την εξυπηρέτησή της. Επειδή όμως η δημιουργία, η αρχικοποίηση και ο τερματισμός μιας διεργασίας είναι διαδικασίες που κοστίζουν πολύ σε χρόνο, υπάρχει η δυνατότητα να παραμένει ένας αριθμός διεργασιών σε επαγρύπνηση για την εξυπηρέτηση νέων αιτήσεων. Δηλαδή, μόλις μια διεργασία εξυπηρετήσει μια αίτηση δεν τερματίζει, αλλά παραμένει σε κατάσταση αναμονής μέχρι να της ζητηθεί να εξυπηρετήσει μια νέα αίτηση. Ο αριθμός των διεργασιών που βρίσκονται σε κατάσταση αναμονής εξαρτάται από μια παράμετρο που προσδιορίζεται από τον προγραμματιστή εφαρμογής. Ο λόγος που δεν χρησιμοποιείται μια τυπική τιμή, είναι ότι ο αριθμός αυτός εξαρτάται από την εφαρμογή και μια ικανοποιητική τιμή μπορεί να προκύψει μόνο κατά τη διάρκεια λειτουργίας του συστήματος. Η δυναμική εύρεση της βέλτιστης τιμής αποτελεί μια επέκταση του συστήματος, που θα μας απασχολήσει στο τελευταίο κεφάλαιο.

Το τμήμα επικοινωνίας είναι υπεύθυνο για την έναρξη και τη λήξη της επικοινωνίας με το ΣΣΕ. Η επικοινωνία αποκαθίσταται από μια διεργασία κατά την έναρξη της λειτουργίας της και λήγει κατά τον τερματισμό της διεργασίας. Μια διεργασία που θα παραμείνει σε κατάσταση αναμονής διατηρεί την επικοινωνία με το ΣΣΕ.

Σε μια μελλοντική μετάβαση από την αρχιτεκτονική εξυπηρέτη εφαρμογής (Application Server) στην αρχιτεκτονική χρήσης της διεπιφάνειας προγραμματισμού εφαρμογών του εξυπηρέτη του Παγκόσμιου Ιστού (Server API), το τμήμα που πρέπει να αλλάξει είναι το τμήμα επικοινωνίας. Η επικοινωνία με τον εξυπηρέτη του Παγκόσμιου Ιστού δεν θα απαιτεί την χρήση μιας ενδιάμεσης διαδικασίας (εκτελέσιμο cgi). Το τμήμα επικοινωνίας θα ανακτά τα ορίσματα από το URL και θα καλεί την αντίστοιχη συνάρτηση του τμήματος παραγωγής της σελίδας HTML.

4.1.1.2 Τμήμα παραγωγής σελίδας HTML

Το τμήμα παραγωγής της σελίδας HTML έχει αναπτυχθεί ως ένα ξεχωριστό τμήμα από τον εξυπηρέτη εφαρμογής. Για την κλήση του παρέχεται μια διεπιφάνεια προγραμματισμού εφαρμογών που περιλαμβάνει διαδικασίες για την παραγωγή κειμένου HTML για ένα αντικείμενο. Ουσιαστικά αυτές οι διαδικασίες μεταφράζουν την λογική παρουσίαση σε παρουσίαση HTML. Αφού εντοπισθεί η όψη, εξετάζεται αν υπάρχει αντικείμενο παρουσίασης που είναι σύμφωνο με την όψη και συσχετισμένο με το αντικείμενο, έστω X, που πρέπει να παρουσιασθεί. Αν δεν υπάρχει, τότε εξετάζεται αν κληρονομείται μοντέλο παρουσίασης από τις υπερκλάσεις του X. Αν σε καμία υπερκλάση δεν έχει αντιστοιχισθεί μοντέλο παρουσίασης, τότε εξετάζονται οι κλάσεις των οποίων είναι περίπτωση η οντότητα X. Αν σε καμία κλάση της X δεν έχει ορισθεί μοντέλο παρουσίασης ούτε και κληρονομείται από τις υπερκλάσεις της, τότε εξετάζονται οι μετακλάσεις της οντότητας X. Σε περίπτωση που δεν βρεθεί κανένα μοντέλο παρουσίασης, εξετάζεται αν

έχει δηλωθεί τυπική παρουσίαση στην όψη, σύμφωνα με την οποία πρέπει να παραχθεί η παρουσίαση. Αν και τότε αποτύχει το σύστημα να βρεί κάποιο μοντέλο παρουσίασης, χρησιμοποιεί την τυπική τιμή που έχει ορισθεί για το είδος διαμόρφωσης κειμένου.

Αν η οντότητα Χ κληρονομεί περισσότερα από ένα μοντέλα παρουσίασης υπάρχει σύγκρουση και ενημερώνεται κατάλληλα ο προγραμματιστής εφαρμογής. Οι συγκρούσεις λύνονται με τη βοήθεια προτεραιοτήτων που ορίζονται σε ένα αρχείο συστήματος από τον προγραμματιστή εφαρμογής.

Το μοντέλο παρουσίασης, δηλαδή η λογική παρουσίαση, μπορεί να υπάρχει ήδη στη μνήμη του συστήματος. Ο αριθμός των λογικών παρουσιάσεων που διατηρούνται στη μνήμη του συστήματος είναι μια παράμετρος που προσδιορίζεται κατά την εκκίνηση του συστήματος από τον προγραμματιστή εφαρμογής. Αν δεν υπάρχει ήδη στην μνήμη του συστήματος, επιλέγεται η κατάλληλη λογική παρουσίαση που πρέπει να απομακρυνθεί από τη μνήμη και ανακτάται από τη βάση η ζητούμενη. Για την αντικατάσταση των λογικών παρουσιάσεων που βρίσκονται στη μνήμη, ακολουθείται ο αλγόριθος γήρανσης (aging). Ο αλγόριθμος αυτός είναι όπως ακριβώς ο αλγόριθμος γήρανσης για την αντικατάσταση σελίδας στη μνήμη. Σύμφωνα με τον αλγόριθμο γήρανσης, όταν πρέπει να διαγραφεί από τη μνήμη του συστήματος μια λογική παρουσίαση, επιλέγεται αυτή που έχει χρησιμοποιηθεί λιγότερο πρόσφατα.

Όπως έχει τονισθεί, το μοντέλο παρουσίασης περιγράφει τους λογικούς μορφότυπους και όχι τους μορφότυπους της HTML. Δηλαδή, στο μοντέλο παρουσίασης μπορεί να προσδιορίζεται ότι κάποιο κείμενο πρέπει να παρουσιασθεί στοιχισμένο στο κέντρο. Για την εύρεση της κατάλληλης ετικέτας διαμόρφωσης που πρέπει να χρησιμοποιηθεί αναζητείται ο λογικός μορφότυπος σε έναν πίνακα αντιστοιχίσεων. Ο πίνακας αντιστοιχίσεων είναι διαφορετικός για κάθε είδος διαμόρφωσης κειμένου και αντιστοιχίζει τους λογικούς μορφότυπους σε μορφότυπους του είδους διαμόρφωσης. Στην περίπτωση της HTML και στο παραπάνω παράδειγμα ο πίνακας αντιστοιχίσεων συσχετίζει τον λογικό μορφότυπο ‘‘στοίχιση στο κέντρο’’ με την ετικέτα διαμόρφωσης της HTML ‘‘<CENTER>’’.

Για την παραγωγή της παρουσίασης σε HTML ενός αντικειμένου ακολουθούνται τα παρακάτω βήματα :

1. Εύρεση της λογικής παρουσίασης
2. Προσδιορισμός των τμημάτων δεδομένων
3. Για κάθε ένα τμήμα δεδομένων :
   - 3.1 Προσδιορισμός των δεδομένων που περιέχει
   - 3.2 Προσδιορισμός του μορφότυπου και του είδους δεδομένου πολυμέσων που πρέπει να χρησιμοποιηθεί
   - 3.3 Προσδιορισμός της επέκτασης που έχει δηλωθεί για το συγκεκριμένο τμήμα δεδομένων

3.4 Για κάθε ένα από τα δεδομένα του τμήματος, ελέγχουμε αν έχει προσδιορισθεί επέκταση[1]. Αν έχει προσδιορισθεί επέκταση για το συγκεκριμένο δεδομένο, τότε αυτή υπερκαλύπτει την επέκταση που έχει ορισθεί στο σύνολο δεδομένων που ανήκει.

3.5 Παρουσίαση του τμήματος σε κείμενο HTML

Η λογική παρουσίαση μπορεί να χρησιμοποιηθεί για την παρουσίαση της πληροφορίας χρησιμοποιώντας και άλλα είδη διαμόρφωσης κειμένου εκτός από HTML, όπως για παράδειγμα XML. Σε αυτήν την περίπτωση πρέπει να δημιουργηθεί ένας πίνακας αντιστοιχίας μορφοτύπων, ανάλογος εκείνου για παρουσίαση σε HTML. Για την υποστήριξη παρουσίασης σε περισσότερα του ενός είδη διαμόρφωσης κειμένου απαιτείται μια συνάρτηση, που ελέγχοντας το είδος στο οποίο ανήκει η λογική παρουσίαση, καλεί την αντίστοιχη συνάρτηση παραγωγής παρουσίασης. Ο έλεγχος του είδους διαμόρφωσης κειμένου είναι απλός. Στο σχήμα 3.5 βλέπουμε ότι το αντικείμενο *Presentation* έχει ως μέλη τα αντικείμενα *HTML_Presentation*, *XML_Presentation*, *Latex_Presentation* και *Text_Presentation*. Η λογική παρουσίαση είναι μέλος μιας από τις παραπάνω κλάσεις. Επομένως, ελέγχοντας το αντικείμενο, στο οποίο είναι μέλος, μπορούμε να βρούμε το είδος διαμόρφωσης κειμένου, που πρέπει να χρησιμοποιηθεί, καθώς και τους αντίστοιχους πίνακες αντιστοίχισης. Συνεπώς, αν θέλουμε να επεκτείνουμε το σύστημα ώστε να παράγει και XML κείμενο εκτός από HTML, η αρχιτεκτονική που θα έχει το σύστημα είναι αυτή που φαίνεται στο σχήμα 4.3. Όπως παρατηρούμε και στο σχήμα, προστίθενται δύο επιπλέον τμήματα στον εξυπηρέτη εφαρμογής. Το τμήμα που βρίσκει ποιό είδος διαμόρφωσης κειμένου πρέπει να χρησιμοποιηθεί (Presentation Type Check Module), και το τμήμα παραγωγής της παρουσίασης XML (XML Production Module).

4.1.1.3 Τμήμα επερωτήσεων

Για την ανάκτηση της πληροφορίας από τη βάση χρησιμοποιούνται διαδικασίες του τμήματος επερωτήσεων (Query Module).

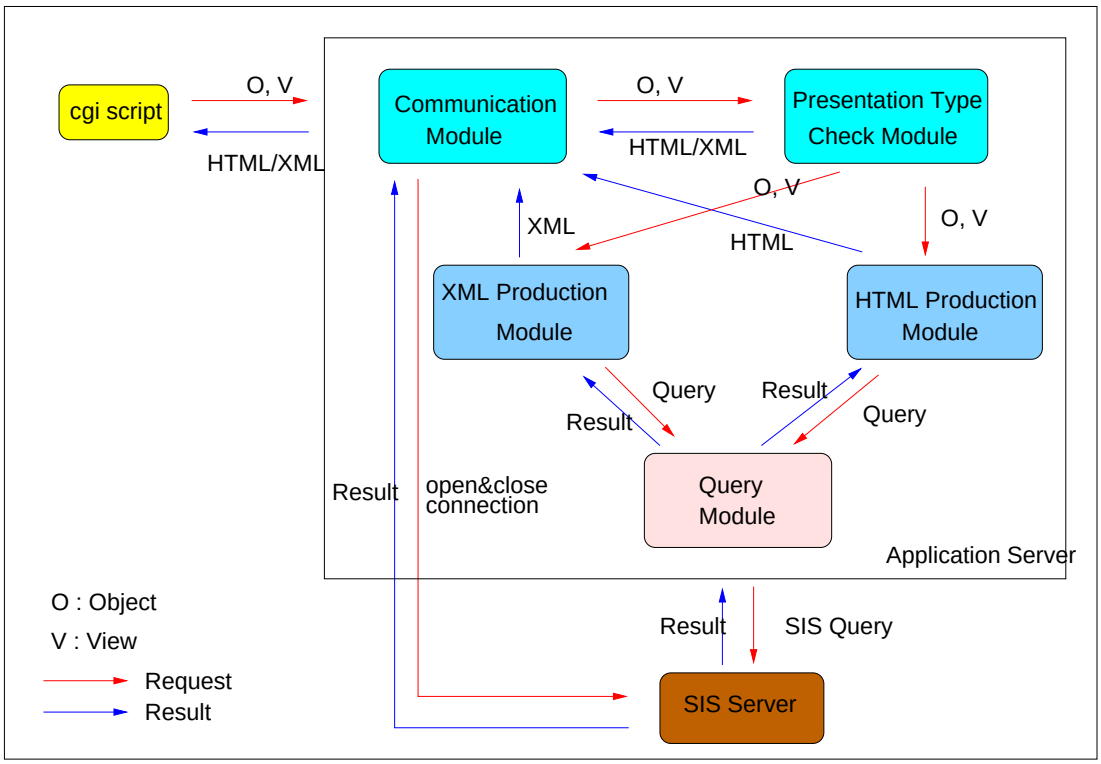
Για την ανάκτηση πληροφορίας από το ΣΣΕ μέσω μιας εξωτερικής εφαρμογής ακολουθούνται τα παρακάτω βήματα :

1. Τίθεται ως τρέχων κόμβος εκείνος στον οποίο θα εφαρμοστεί μια από τις συναρτήσεις αποστολής επερώτησης του SIS-API.

2. Χρησιμοποιούνται οι κατάλληλες συναρτήσεις για την αποστολή επερώτησης του SIS-API για την αναζήτηση πληροφορίας στη Βάση Γνώσης.

3. Αν είναι απαραίτητο, επαναλαμβάνονται τα βήματα 1 και 2, όσες φορές είναι αναγκαίο, και χρησιμοποιούνται κατάλληλες συναρτήσεις διαχείρισης συνόλων

[1] Μπορεί να προσδιορισθεί επέκταση σε ένα σύνολο αντικειμένων ή σε ένα αντικείμενο



Σχήμα 4.3: Ο εξυπηρέτης εφαρμογής υποστηρίζει παρουσίαση σε HTML και XML

αποτελεσμάτων του SIS-API.

Το τμήμα επερωτήσεων περιέχει διαδικασίες για την ανάκτηση πληροφορίας από τη βάση. Οι διαδικασίες χρησιμοποιούν την διεπιφάνεια προγραμματισμού εφαρμογών (SIS-API) που παρέχει το ΣΣΕ. Ουσιαστικά, μια διαδικασία του τμήματος επερωτήσεων είναι η εφαρμογή των παραπάνω βημάτων, όσες φορές είναι απαραίτητο, και η επεξεργασία των αποτελεσμάτων για την ανάκτηση της απαιτούμενης πληροφορίας.

4.2 Σύγκριση με άλλες υλοποιήσεις

Σε αυτήν την ενότητα γίνεται μια σύγκριση με κάποια από τα προϊόντα που περιγράφηκαν στο κεφάλαιο 2. Ο λόγος επιλογής ορισμένων μόνο προϊόντων και όχι όλων είναι ότι τα άλλα προϊόντα διαφέρουν τόσο πολύ από την προσέγγιση της παρούσας εργασίας, ώστε δεν τίθεται θέμα σύγκρισης. Η σύγκριση γίνεται με προϊόντα τα οποία παρουσιάζουν μερικές ομοιότητες με το δικό μας σύστημα είτε ως προς την αρχιτεκτονική υλοποίησης, είτε ως προς τη λογική ανάπτυξης.

Συγκρίνουμε το σύστημα που υλοποιήσαμε με τα προϊόντα που περιγράφηκαν στο δεύτερο κεφάλαιο και έχουν αναπτυχθεί σύμφωνα με την αρχιτεκτονική Application

Server. Τα προϊόντα αυτά είναι τα O_2Web, LINCKS και Oracle Web Application Server. Και τα τρία προϊόντα, όπως και το δικό μας σύστημα, αποθηκεύουν την πληροφορία που είναι απαραίτητη για την παρουσίαση των δεδομένων στη βάση. Στην περίπτωση του προϊόντος O_2Web υπάρχει μια μοντελοποίηση της σελίδας HTML και ο προγραμματιστής εφαρμογής συμπληρώνει τον κώδικα μεθόδων. Στην περίπτωση του Oracle Web Application Server, ο προγραμματιστής εφαρμογής συμπληρώνει τον κώδικα συναρτήσεων, οι οποίες αποθηκεύονται στη βάση (stored procedures), ενώ στην περίπτωση του προϊόντος LINCKS υπάρχει μια περισσότερο αφαιρετική μοντελοποίηση. Ο προγραμματιστής εφαρμογής δηλώνει το περιεχόμενο της σελίδας HTML με τρόπο λογικό και όχι συμπληρώνοντας κώδικα σε κάποια γλώσσα προγραμματισμού. Όμως στην περίπτωση του LINCKS ο προγραμματιστής εφαρμογής πρέπει να περιγράψει με μεγάλη λεπτομέρεια τις ενέργειες που πρέπει να γίνουν για την μετάβαση από τη λογική περιγραφή στην σελίδα HTML. Στο σύστημά μας ο προγραμματιστής εφαρμογής δηλώνει τα περιεχόμενα της σελίδας HTML αλλά δεν χρειάζεται να δηλώσει πώς θα γίνει η μετατροπή από τη λογική περιγραφή στην παραγωγή της σελίδας HTML.

Επίσης, σε αντίθεση με τα προϊόντα O_2Web και Oracle Web Application Server, στο σύστημά μας οι λογικές παρουσιάσεις μπορούν να εξειδικευθούν ακόμη και στο επίπεδο των δεδομένων (Token). Έτσι, ένα αντικείμενο μπορεί να έχει διαφορετική παρουσίαση από αυτή που έχει δηλωθεί στην κλάση, της οποίας είναι μέλος.

Επιπλέον, στις περιπτώσεις των προϊόντων O_2Web και Oracle Web Application Server δεν υπάρχει η δυνατότητα να παρασταθούν τα ίδια δεδομένα με διαφορετικό τρόπο ανάλογα με την εφαρμογή. Αυτό υποστηρίζεται στο LINCKS και στην παρούσα εργασία μέσω διαφορετικών όψεων. Επιπλέον, μπορούν να ορισθούν όψεις και στις λογικές παρουσιάσεις. Παρόλο που το μοντέλο παρουσίασης αποθηκεύεται στη βάση, τα άλλα συστήματα δεν χειρίζονται το μοντέλο παρουσίασης ως δεδομένα στα οποία μπορεί να καθορισθεί τρόπος παρουσίασης.

Η διαδικασία πρέπει να επαναληφθεί για την παράσταση της ίδιας πληροφορίας χρησιμοποιώντας κάποιον άλλο τρόπο διαμόρφωσης κειμένου εκτός από HTML, π.χ. XML. Για το Oracle Web Application Server πρέπει να ξαναγραφούν οι αποθηκευμένες συναρτήσεις, για το O_2Web πρέπει να επαναπροσδιορισθούν (overload) οι μέθοδοι, ενώ για το LINCKS πρέπει να συμπληρωθούν οι ενέργειες. Στην παρούσα εργασία το μόνο που χρειάζεται είναι να συμπληρωθεί ένας πίνακας αντιστοιχίας μεταξύ λογικών μορφότυπων και μορφότυπων του είδους διαμόρφωσης κειμένου, ενέργεια που θα γίνει μια φορά για κάθε είδος αποτελέσματος που παρέχεται και όχι για κάθε εφαρμογή.

Τέλος, στην περίπτωση των O_2Web και Oracle Web Application Server, η ύπαρξη διανεμητή αιτήσεων είναι ένα πολύ μεγάλο πλεονέκτημα. Η προσθήκη διανεμητή αιτήσεων αποτελεί μελλοντικό στόχο για το σύστημά μας.

Το προϊόν Oracle Web Application Server χρησιμοποιεί Web Request Broker. Με αυτόν τον τρόπο εκμεταλλεύεται με τον καλύτερο τρόπο τα πλεονεκτήματα που προσφέρει η αρχιτεκτονική τριών επιπέδων και η αρχιτεκτονική CORBA.

Η χρήση του μοντέλου παρουσίασης των ιστοσελίδων επιτρέπει τον σαφή διαχωρισμό της πληροφορίας που βρίσκεται μέσα σε μία βάση, από τη λογική όψη της πληροφορίας που θα παρουσιασθεί και από τη γραφική παρουσίαση σε HTML. Με βάση την περιγραφή του συστήματος Strudel, παρατηρούμε ότι η λογική ανάπτυξης του συστήματος μας πλησιάζει πολύ τη λογική ανάπτυξης του Strudel, που είναι ένα Σύστημα Διαχείρισης Κόμβων του Παγκόσμιου Ιστού (ΣΔΚΠΙ).

Κεφάλαιο 5

Εμπειρία χρήσης του συστήματος

Στο κεφάλαιο αυτό γίνεται μια αναφορά σε παρατηρήσεις που προέκυψαν από τη χρήση του συστήματος που αναπτύχθηκε. Το σύστημα που περιγράφεται χρησιμοποιήθηκε για την παραγωγή και διατήρηση των ιστοσελίδων του Τομέα Πληροφοριακών Συστημάτων και Τεχνολογίας Λογισμικού του ΙΠ-ΙΤΕ. Επιπλέον, χρησιμοποιήθηκε στα πλαίσια του Συστήματος Διαχείρισης Θησαυρών Όρων - ΣΔΘΟ (Thesaurus Management System - TMS) για την παραγωγή απλού κειμένου. Τέλος, με βάση τη χρήση του συστήματος έχουν προκύψει κάποιες παρατηρήσεις σε σχέση με τη χρήση του μοντέλου παρουσίασης ιστοσελίδων, με την ταχύτητα απόκρισης του συστήματος και το μέγεθος της απαιτούμενης μνήμης.

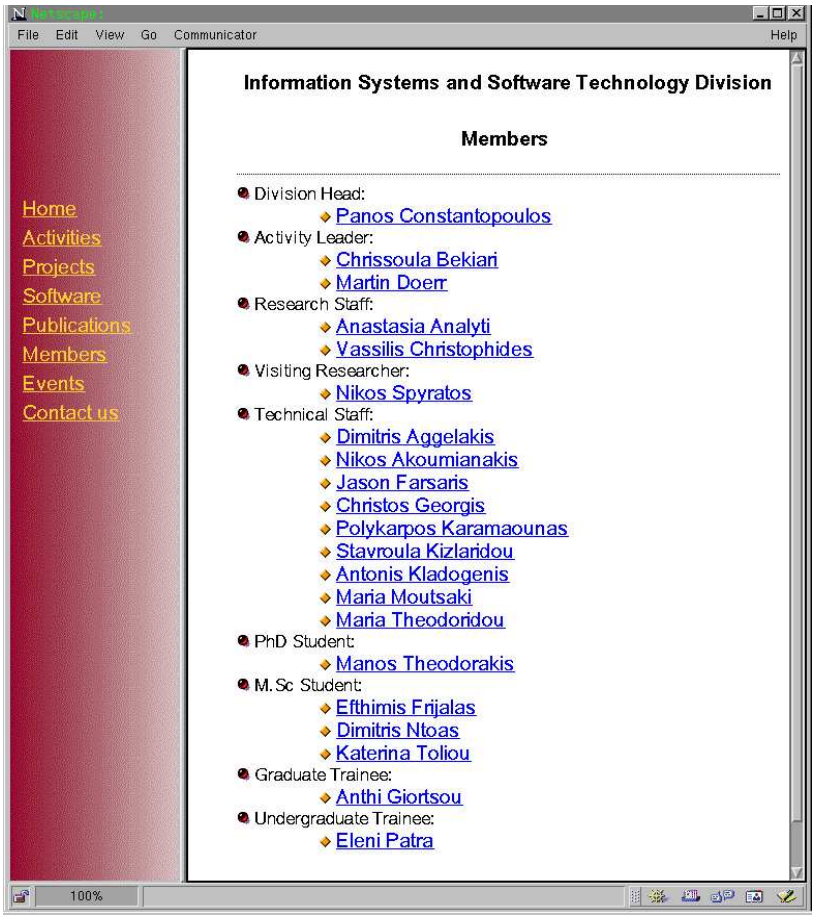
5.1 Ιστοσελίδες Τομέα Πληροφοριακών Συστημάτων και Τεχνολογίας Λογισμικού

Για την παρουσίαση της δομής του Τομέα Πληροφοριακών Συστημάτων και Τεχνολογίας Λογισμικού χρησιμοποιήθηκε το μοντέλο που περιγράφηκε στο τρίτο κεφάλαιο και εμπλουτίσθηκε με την πληροφορία που αφορά τον συγκεκριμένο τομέα.

5.1.1 Σενάριο παρουσίασης

Το σενάριο παρουσίασης που υλοποιήθηκε περιλαμβάνει την κατηγοριοποίηση της πληροφορίας που αφορά τον τομέα στις κατηγορίες ‘‘Δραστηριότητες’’, ‘‘Έργα’’, ‘‘Λογισμικό’’, ‘‘Δημοσιεύσεις’’, ‘‘Μέλη’’, ‘‘Εκδηλώσεις’’ και ‘‘Επικοινωνία’’. Στην αρχική σελίδα παρουσιάζεται μια σύντομη περιγραφή του τομέα.

Η κατηγορία ‘‘Δραστηριότητες’’ περιλαμβάνει μια αναφορά στις δραστηριότητες του τομέα κάθε μια από τις οποίες είναι ένας υπερσύνδεσμος για την παρουσίαση πληρο-



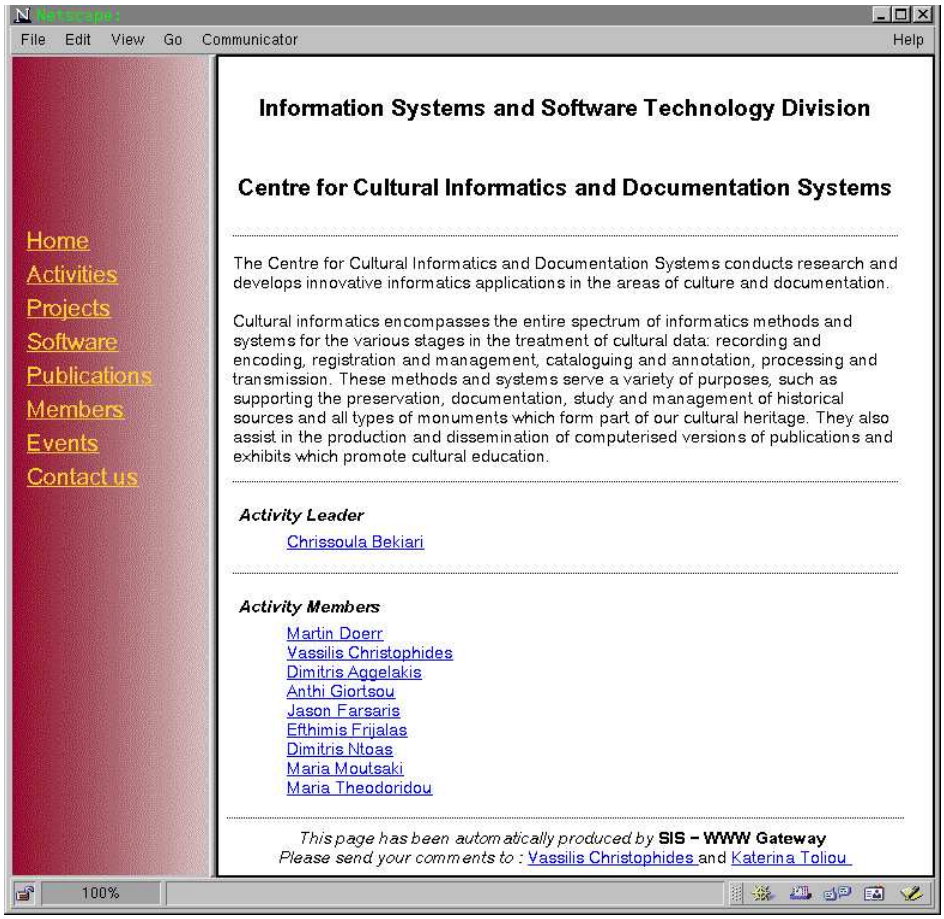
Σχήμα 5.1: Παρουσίαση των μελών με χρήση πλαισίων

φορίας σχετικά με την αντίστοιχη δραστηριότητα. Η παρουσίαση μιας δραστηριότητας περιλαμβάνει μια σύντομη αναφορά στη δραστηριότητα, τον υπεύθυνο και τα μέλη της.

Η κατηγορία ‘‘ Έργα’’ περιλαμβάνει αναφορά στα έργα στα οποία έχει συμμετάσχει ο τομέας. Τα έργα ομαδοποιούνται σε ενεργά και ολοκληρωμένα. Η κατηγορία ‘‘Λογισμικό’’ περιλαμβάνει μια αναφορά στο λογισμικό που έχει αναπτυχθεί στα πλαίσια του τομέα.

Οι δημοσιεύσεις διακρίνονται στα είδη, δηλαδή δημοσιεύσεις σε περιοδικά, σε συνέδρια κ.α, και στις χρονολογίες δημοσίευσης. Η κάθε κατηγορία δημοσίευσης είναι ένας υπερσύνδεσμος για την παρουσίαση των αντιστοιχων δημοσιεύσεων.

Τα μέλη του τομέα ομαδοποιούνται ανάλογα με τη θέση που κατέχουν στα πλαίσια του τομέα, όπως επικεφαλής τομέα, ερευνητής κ.α. Το κάθε μέλος εμφανίζεται ως υπερσύνδεσμος για την παρουσίαση περισσότερης πληροφορίας για το συγκεκριμένο μέλος. Η πληροφορία που αφορά ένα μέλος είναι η θέση του στον τομέα, ο αριθμός τηλεφώνου και το γραφείο, η ηλεκτρονική διεύθυνση, η δραστηριότητα στην οποία συμμετέχει και οι δημοσιεύσεις που έχει κάνει.



Σχήμα 5.2: Παρουσίαση της δραστηριότητας ‘‘Συστήματα Πολιτισμικής Πληροφορικής και Συστημάτων Τεκμηρίωσης’’ με χρήση πλαισίων

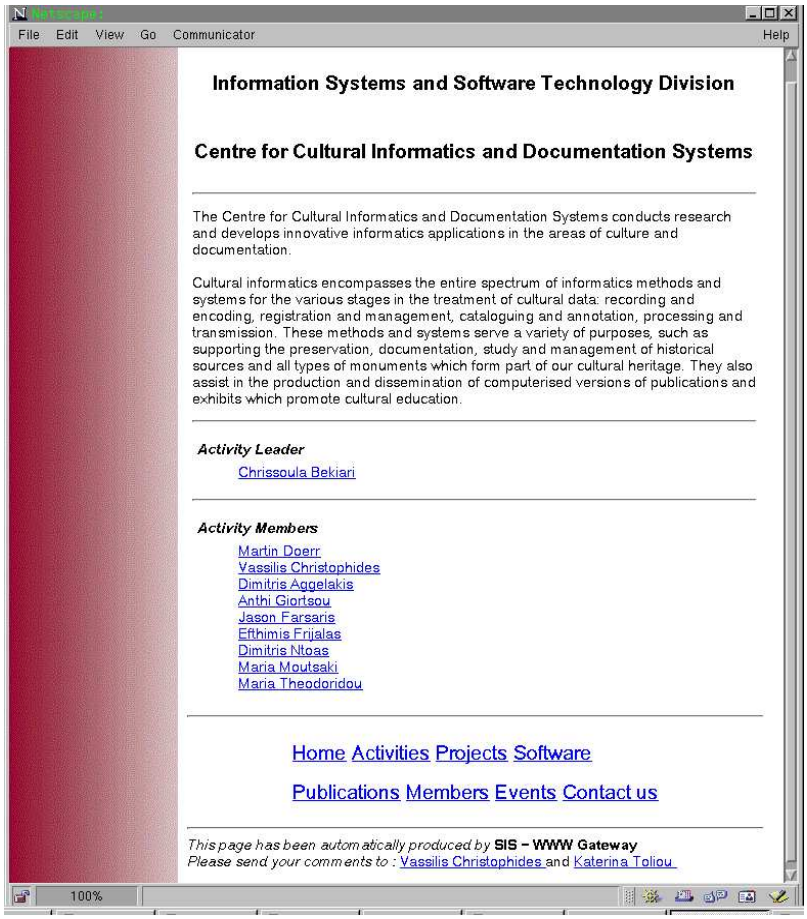
Η παρουσίαση των εκδηλώσεων είναι μια απλή καταγραφή των εκδηλώσεων τις οποίες έχει οργανώσει ή στις οποίες έχει συμμετάσχει ο τομέας.

Τέλος, η κατηγορία ‘‘Επικοινωνία’’ αφορά πληροφορίες για την επικοινωνία με τον επικεφαλής του τομέα και με τους υπεύθυνους των δραστηριοτήτων.

Για την παρουσίαση του τομέα χρησιμοποιήθηκαν δύο όψεις, όπου σύμφωνα με την πρώτη όψη η πληροφορία οργανώνεται σε πλαίσια, ενώ σύμφωνα με τη δεύτερη όψη η ίδια πληροφορία παρουσιάζεται χωρίς τη χρήση πλαισίων.

5.1.2 Αποτελέσματα

Με βάση το παραπάνω σενάριο δηλώθηκε για το κάθε αντικείμενο η ανάλογη παρουσίαση με τη χρήση του μοντέλου που περιγράφηκε στο τρίτο κεφάλαιο. Στο σχήμα 5.1 διακρίνεται η παρουσίαση των μελών του τομέα με τη χρήση πλαισίων. Στα σχήματα



Σχήμα 5.3: Παρουσίαση της δραστηριότητας ‘‘Συστήματα Πολιτισμικής Πληροφορικής και Συστημάτων Τεκμηρίωσης’’ χωρίς πλαίσια

5.2 και 5.3 διακρίνονται οι παρουσιάσεις της δραστηριότητας ‘‘Κέντρο Πολιτισμικής Πληροφορικής και Συστημάτων Τεκμηρίωσης’’ με χρήση πλαισίων και χωρίς.

Στις παραγόμενες ιστοσελίδες περιέχεται ένας συνδυασμός δεδομένων που είναι αποθηκευμένα στη βάση με δεδομένα που δεν είναι αποθηκευμένα στη βάση. Τα εικονίδια που βρίσκονται πριν από τις κατηγορίες μελών και τα μέλη, είναι δεδομένα που δεν είναι αποθηκευμένα στη βάση.

Σε όλες τις παρουσιάσεις η ιστοσελίδα αποτελείται από τέσσερα μέρη. Στις κατηγορίες πληροφορίας που είναι υπερσύνδεσμοι, στο όνομα του τομέα, στο κάτω μέρος της σελίδας που υπάρχει ένα κείμενο, το οποίο δηλώνει οτι η σελίδα κατασκευάστηκε από το σύστημα, και στο κύριο μέρος. Τα τρία πρώτα μέρη είναι κοινά σε όλες τις παρουσιάσεις, ενώ το κύριο μέρος διαφοροποιείται ανάλογα με την προς παρουσίαση πληροφορία. Στις ιστοσελίδες με πλαίσια οι κατηγορίες πληροφορίας εμφανίζονται στο αριστερό πλαίσιο, ενώ στις ιστοσελίδες χωρίς πλαίσια οι κατηγορίες πληροφορίας εμφανίζονται στο κάτω μέρος της ιστοσελίδας.

Το κάθε ένα από τα τέσσερα μέρη, από τα οποία αποτελείται μια ιστοσελίδα, αντιστοιχεί σε ένα τμήμα της λογικής παράστασης και είναι τύπου *MainBody*. Με αυτόν τον τρόπο στο κάθε τμήμα δηλώνεται ο μορφότυπος που πρέπει να χρησιμοποιηθεί και σελίδες που έχουν κοινά μέρη χρησιμοποιούν το ίδιο τμήμα της λογικής παράστασης. Το όνομα του τομέα και το κείμενο στο κάτω μέρος της σελίδας είναι κοινά σε όλες τις παρουσιάσεις σύμφωνα και με τις δύο όψεις. Επομένως, οι αντίστοιχες λογικές παραστάσεις χρησιμοποιούν το ίδιο τμήμα λογικής παράστασης. Στις ιστοσελίδες με πλαίσια, οι κατηγορίες πληροφορίας εμφανίζονται στο αριστερό πλαίσιο, το οποίο παραμένει σταθερό. Επομένως, οι αντίστοιχες λογικές παραστάσεις δεν περιέχουν το τμήμα της λογικής παράστασης που αντιστοιχεί στις κατηγορίες πληροφορίας. Αντίθετα, στις ιστοσελίδες χωρίς πλαίσια το τμήμα της λογικής παράστασης που αντιστοιχεί στις κατηγορίες πληροφορίας, είναι κοινό σε όλες τις λογικές παραστάσεις. Με αυτόν τον τρόπο αν αλλάξει το τμήμα, όλες οι ιστοσελίδες ενημερώνονται κατάλληλα χωρίς να χρειαστεί κάποια άλλη αλλαγή από τον προγραμματιστή εφαρμογής.

Το ίδιο ισχύει και για το τέταρτο μέρος που περιλαμβάνει την κύρια πληροφορία. Έχει δηλωθεί ως ένα αντικείμενο τύπου *MainBody* το οποίο χρησιμοποιείται από τις λογικές παρουσιάσεις της ίδιας πληροφορίας στις δύο όψεις. Συνεπώς, μια αλλαγή στο ίδιο το τμήμα έχει ως αποτέλεσμα να εφαρμοσθεί η αλλαγή και στις δύο όψεις.

Επομένως, το μοντέλο παρουσίασης χρησιμοποιείται για τον ορισμό όψεων όχι μόνο στα δεδομένα της βάσης, αλλά και στα αντικείμενα του μοντέλου παρουσίασης.

Το κύριο μέρος της παρουσίασης των μελών του τομέα περιλαμβάνει μια παράθεση των μελών ομαδοποιημένα σύμφωνα με τη θέση που κατέχουν στα πλαίσια του τομέα. Το κάθε μέλος εμφανίζεται ως υπερσύνδεσμος για την παρουσίαση περισσότερης πληροφορίας, σχήμα 5.1. Η κλάση *Personnel* έχει ως υποκλάσεις τις κατηγορίες εργαζομένων και ο κάθε εργαζόμενος είναι περίπτωση της κατηγορίας στην οποία ανήκει. Για την παρουσίαση των κατηγοριών εργαζομένων που υπάρχουν στα πλαίσια του τομέα, έχει ορισθεί στο μοντέλο παρουσίασης να παρουσιασθούν οι υποκλάσεις της κλάσης *Personnel* (αντικείμενο *Subclasses*). Για την παρουσίαση των εργαζομένων ανά κατηγορία, έχει ορισθεί στο μοντέλο παρουσίασης να επεκταθούν οι υποκλάσεις της κλάσης *Personnel*. Επειδή πρέπει να εμφανισθεί η κατηγορία εργαζομένων και να επεκταθεί, έχει χρησιμοποιηθεί ως είδος επέκτασης το *Expand_Node*. Για την επέκταση των κατηγοριών εργαζομένων χρησιμοποείται το μοντέλο παρουσίασης που έχει ορισθεί για την κάθε κατηγορία. Επειδή είναι κοινό το είδος της πληροφορίας και ο τρόπος με τον οποίο θα παρουσιασθεί για όλες τις κατηγορίες, έχει ορισθεί ένα κοινό μοντέλο παρουσίασης στο οποίο δηλώνεται ότι για την κάθε κατηγορία εργαζομένων θα παρουσιασθούν τα μέλη της ως υπερσύνδεσμοι (αντικείμενο *Instances*).

Το κύριο μέρος της παρουσίασης της δραστηριότητας ''Κέντρο Πολιτισμικής Πληροφορικής και Συστημάτων Τεκμηρίωσης'' περιλαμβάνει μια σύντομη περιγραφή της δραστηριότητας, τον επικαφαλή και τα μέλη της δραστηριότητας, σχήματα 5.2 και 5.3. Ο επικεφαλής και τα μέλη της δραστηριότητας παρουσιάζονται ως υπερσύνδεσμοι. Η

σύντομη περιγραφή, ο επικεφαλής και τα μέλη μιας δραστηριότητας έχουν δηλωθεί στο μοντέλο παράστασης της δομής του τομέα ως γνωρίσματα. Επομένως, για την παρουσίαση της σύντομης περιγραφής αρκεί να ακολουθηθεί το γνώρισμα της δραστηριότητας που είναι μέλος της κλάσης γνωρισμάτων *overview*. Ομοίως για την παρουσίαση του υπεύθυνου και των μελών. Έτσι, στο μοντέλο παρουσίασης έχει δηλωθεί ότι για την παρουσίαση μιας δραστηριότητας θα χρησιμοποιηθούν τα γνωρίσματα που ανήκουν στις κλάσεις *overview*, *activity leader* και *members*, χρησιμοποιώντας τα αντικείμενα *FromAttributesByClass* του μοντέλου παρουσίασης. Από τα γνωρίσματα θέλουμε να παρουσίασθεί μόνο η τιμή τους, στην περίπτωση της σύντομης περιγραφής ως απλό κείμενο ενώ για τον υπεύθυνο και τα μέλη ως υπερσύνδεσμος. Επομένως, ως μορφότυπος χρησιμοποιήθηκαν τα *NoHyperlinkToValue* και *HyperlinkToValue* αντίστοιχα.

Για την μορφοποίηση των επιμέρους τμημάτων που συνιστούν την ιστοσελίδα και του συνδυασμού των δύο τελευταίων τμημάτων, δηλαδή του ονόματος του τομέα και της κύριας πληροφορίας, έχει χρησιμοποιηθεί ο τύπος *TableRows* που δηλώνει ότι τα δεδομένα του κάθε τμήματος θα παρουσιασθούν ως γραμμές ενός πίνακα. Για την όψη παρουσίασης με χρήση πλαισίων έχει δηλωθεί ότι το πρώτο πλαίσιο θα καταλαμβάνει το 25% των στηλών της ιστοσελίδας ενώ το δεύτερο το υπόλοιπο. Για την όψη παρουσίασης χωρίς πλαίσια έχει δηλωθεί ότι όλη η σελίδα αποτελείται από δύο τμήματα που θα παρουσιασθούν ως στήλες ενός πίνακα. Το πρώτο τμήμα είναι οι κατηγορίες πληροφορίας ενώ το δεύτερο περιλαμβάνει το τμήμα του ονόματος του τομέα και το κύριο τμήμα των δεδομένων.

5.2 Σύστημα Διαχείρισης Θησαυρών Όρων

Το *Σύστημα Διαχείρισης Θησαυρών Όρων - ΣΔΘΟ*[62] είναι ένα πολυγλωσσικό σύστημα διαχείρισης θησαυρών όρων. Τα κύρια χαρακτηριστικά του είναι η ικανότητα αποθήκευσης, ανάπτυξης, παρουσίασης και ανάκτησης πολλών θησαυρών όρων και των μεταξύ τους σχέσεων σύμφωνα με ένα κοινό σχήμα βάσης δεδομένων. Επιπλέον, η δημιουργία γραφικών όψεων και η δυναμική εξειδίκευση οποιουδήποτε είδους σχέσεων σε νέες. Ακόμη, υποστηρίζει τον απαραίτητο έλεγχο εκδόσεων για ανάπτυξη και ανταλλαγή δεδομένων με συνεργαζόμενες εφαρμογές.

Το ΣΔΘΟ είναι μια εφαρμογή του ΣΣΕ και έχει αναπτυχθεί από τον Τομέα Πληροφοριακών Συστημάτων και Τεχνολογίας Λογισμικού του ΙΠ-ΙΤΕ[63].

Η αλληλεπίδραση του χρήστη με το ΣΔΘΟ γίνεται μέσω μιας γραφικής διεπιφάνειας χρήσης. Η γραφική διεπιφάνεια χρήσης υποστηρίζει αναδίφηση χωρίς την επιβολή περιορισμών μέσα σε έναν θησαυρό όρων και μεταξύ πολλών συνδεδεμένων θησαυρών όρων. Ο χρήστης μπορεί να ανακτήσει πληροφορία από έναν θησαυρό όρων χρησιμοποιώντας ένα σύνολο προκαθορισμένων επερωτήσεων. Το αποτέλεσμα των επερωτήσεων εμφανίζεται γραφικώς ή ως απλό κείμενο. Το ΣΔΘΟ έχει ένα τμήμα το οποίο εξάγει λίστες

όρων χρησιμοποιώντας απλό κείμενο.

Για την ολοκλήρωση του ΣΔΘΟ κρίνεται αναγκαία η ύπαρξη ενός ευέλικτου εργαλείου παραγωγής αναφορών (Report Writer). Ως πρώτο βήμα για τη δημιουργία του εργαλείου παραγωγής αναφορών, δημιουργήθηκε ένα εργαλείο που, δεδομένης μιας ιεραρχίας όρων, παρουσιάζει όλα τα μέλη της ιεραρχίας και για κάθε μέλος κάποιους όρους, με τους οποίους υπάρχουν συγκεκριμένου είδους συσχετίσεις. Για την παρουσίαση χρησιμοποιείται απλό κείμενο και το αποτέλεσμα αποθηκεύεται σε ένα αρχείο κειμένου που προσδιορίζεται από το χρήστη του συστήματος.

Για την κατασκευή του περιορισμένου εργαλείου παραγωγής αναφορών χρησιμοποιήθηκε το μοντέλο παρουσίασης που περιγράφηκε στο κεφάλαιο 3. Η υλοποίηση διαφέρει από εκείνη που περιγράφηκε στο κεφάλαιο 4 ως προς το είδος του κειμένου διαμόρφωσης που χρησιμοποιείται, δηλαδή απλό κείμενο και όχι HTML, και ως προς το μέσο που χρησιμοποιείται για τη παρουσίαση των αποτελεσμάτων, δηλαδή αρχείο κειμένου και όχι ένας αναδιφητής του Παγκόσμιου Ιστού.

Report Generated for hierarchy "MERIMEETopTerm"

---

1) Terme	<architecture agricole>
CA	<architecture agricole>, MERIMEETopTerm
NE	"Terme generique regroupant les ensembles, edifices ..."
to_RCHME_relation	<agriculture and sunsistence>
2) Terme	<architecture artisanale>
SYN	PARTIES ARTISANALES
CA	<architecture artisanale>, MERIMEETopTerm
NE	"Terme generique regroupant les edifices, edicules et parties ..."
3) Terme	<architecture commerciale>
SYN	PARTIES COMMERCIALES
CA	<architecture commerciale>, MERIMEETopTerm
NE	"Terme generique regroupant les ensembles ..."
to_RCHME_relation	<commercial>

Σχήμα 5.4: Παράδειγμα αρχείου αναφοράς

Το μοντέλο παρουσίασης αποτελείται από δύο βασικά αντικείμενα. Ένα χρησιμοποιείται για τον προσδιορισμό της λογικής παράστασης της ιεραρχίας που δίδεται ως αρχικό όρισμα και ένα για τον προσδιορισμό της λογικής παράστασης των μελών της ιεραρχίας. Η λογική παράσταση που αντιστοιχεί στην ιεραρχία, προσδιορίζει ότι πρέπει να παρουσιασθούν όλα τα μέλη του τρέχοντος αντικειμένου (χρησιμοποιώντας έναν κόμβο που είναι μέλος της κλάσης κόμβων *Instances*) ως γραμμές ενός πίνακα, δηλαδή ο μορφότυπος είναι τύπου *TableRows*. Για το κάθε μέλος πρέπει να παρουσιασθούν κάποιοι σχετιζόμενοι όροι. Για αυτόν τον λόγο δηλώνεται στη λογική παράσταση που αντιστοι-

χεί στον τρέχοντα κόμβο, ότι τα αντικείμενα που προσδιορίζονται με την επερώτηση που έχει δηλωθεί πρέπει να επεκταθούν. Για τον προσδιορισμό του είδους επέκτασης χρησιμοποιείται ένα μέλος της κατηγορίας κόμβων *Expand_Node*. Η δεύτερη λογική παράσταση που αναφέρεται στα μέλη του τρέχοντος αντικειμένου, περιγράφει ότι πρέπει να παρουσιασθούν οι συνώνυμοι, οι γενικότεροι, οι ειδικότεροι, οι ισοδύναμοι και οι σχετικοί όροι. Οι παραπάνω κατηγορίες προσδιορίζονται αντίστοιχα χρησιμοποιώντας μέλη των κόμβων *FromAttributesByClass* και *FromAttributesByMetaClass*.

Για την παραγωγή του απλού κειμένου χρησιμοποιήθηκε ένας πίνακας αντιστοιχίας από το λογικό μορφότυπο που περιγράφεται στο μοντέλο παράστασης σε μορφότυπο απλού κειμένου. Ο πίνακας αυτός χρησιμοποιήθηκε στη θέση του πίνακα αντιστοιχίας από λογικό μορφότυπο σε μορφότυπο της HTML. Ουσιαστικά για την παραγωγή του ζητούμενου απλού κειμένου άλλαξε ο πίνακας αντιστοιχίας.

Επιπλέον, το αποτέλεσμα αντί να αποστέλλεται στον εξυπηρέτη του Παγκόσμιου Ιστού, αποθηκεύεται σε ένα αρχείο. Ένα παράδειγμα αποτελέσματος διακρίνεται στο σχήμα 5.4. Οι λέξεις που είναι με έντονη γραμματοσειρά αποτελούν τμήμα της παρουσίασης και καθορίζονται στο μοντέλο παρουσίασης. Οι λέξεις/όροι που είναι με κανονική γραμματοσειρά είναι δεδομένα της βάσης. Η ετικέτα ‘‘CA’’ προσδιορίζει τις κλάσεις στις οποίες ανήκει ο όρος, η ετικέτα ‘‘NE’’ προσδιορίζει επεξηγηματικό κείμενο, η ετικέτα ‘‘SYN’’ συνώνυμο όρο και η ετικέτα ‘‘to_RCHME_relation’’ προσδιορίζει όρους που ανήκουν στον θησαυρό όρων RCHME και σχετίζονται με τον τρέχοντα.

Το παραγόμενο αποτέλεσμα ήταν το ζητούμενο. Επιπλέον, αποτέλεσε και την πρώτη δοκιμή για χρησιμοποίηση της λογικής περιγραφής για την παραγωγή αποτελέσματος μη HTML. Αποδείχθηκε ότι η μετάβαση σε άλλου είδους αποτέλεσμα είναι απλή και απαιτεί απλές επεκτάσεις της υλοποίησης.

5.3 Παρατηρήσεις

Το σύστημα χρησιμοποιήθηκε για τη συντήρηση των ιστοσελίδων του Τομέα Πληροφοριακών Συστημάτων και Τεχνολογίας Λογισμικού και για την κατασκευή ενός εργαλείου παραγωγής αναφορών για το Σύστημα Διαχείρισηςς Θησαυρών Όρων. Οι ιστοσελίδες του Τομέα Πληροφοριακών Συστημάτων και Τεχνολογίας Λογισμικού είναι αρκετά πολύπλοκες επιτρέποντας την εξαγωγή κάποιων παρατηρήσεων για το σύστημα. Ο λόγος που δεν χρησιμοποιήθηκε μια βάση δεδομένων είναι ότι συνήθως οι βάσεις δεδομένων έχουν μια αυστηρή δομή και ο τρόπος παρουσίασης των δεδομένων τους δεν παρουσιάζει μεγάλη ποικιλομορφία. Εν αντιθέσει, η χρησιμοποίηση του συστήματος για την κατασκευή και συντήρηση ενός κόμβου του Παγκόσμιου Ιστού προσφέρει τον απαραίτητο πλουραλισμό.

Από τη χρήση του συστήματος για την παραγωγή και συντήρηση των ιστοσελίδων του Τομέα Πληροφοριακών Συστημάτων και Τεχνολογίας Λογισμικού υπάρχουν ενδεί-

ξεις ότι η συντήρηση πολλών όψεων παρουσίασης της ίδιας πληροφορίας είναι απλή, εύκολη και με μικρό χρονικό κόστος. Αυτή η ευελιξία προκύπτει από τη δυνατότητα ορισμού τμημάτων πληροφορίας τα οποία αποτελούνται από μικρότερα και μπορούν να είναι κοινά σε πολλές παρουσιάσεις. Οι ιστοσελίδες ενός κόμβου του Παγκόσμιου Ιστού έχουν, συνήθως, κοινή δομή και κάποια κοινά τμήματα πληροφορίας. Αυτά τα τμήματα μπορούν να ορισθούν ως ανεξάρτητα τμήματα το καθένα, τύπου *MainBody*, τα οποία χρησιμοποιούνται από όλες τις παρουσιάσεις, όπως για παράδειγμα το τμήμα με τις κατηγορίες της διαθέσιμης πληροφορίας για τον Τομέα Πληροφοριακών Συστημάτων και Τεχνολογίας Λογισμικού. Αλλαγές στο κοινό τμήμα προκαλούν την αντίστοιχη ενημέρωση όλων των ιστοσελίδων που το χρησιμοποιούν. Επιπλέον, επειδή ο μορφότυπος που θα χρησιμοποιηθεί για την παρουσίαση ενός τμήματος είναι ανεξάρτητος των δεδομένων που περιέχει το τμήμα, διευκολύνονται και οι αλλαγές που αφορούν την μορφοποίηση των παραγόμενων ιστοσελίδων.

Ακόμη, η δυνατότητα χρησιμοποίησης Java scripts και applets, προσφέρει σημαντική βοήθεια στον προγραμματιστή εφαρμογής για την παράκαμψη των περιορισμών πουυ επιβάλει το πρότυπο HTML.

Σχετικά με το μοντέλο παρουσίασης αποτελεί σημαντικό μειονέκτημα ότι η κατασκευή και η ενημέρωσή του δεν γίνεται με γραφικό, διαλογικό τρόπο. Αυτό έχει ως αποτέλεσμα τη δυσκολία έκφρασης σε λογική παράσταση αυτού που θέλει να δηλώσει ο προγραμματιστής εφαρμογής. Από το αποτέλεσμα, που είναι οι σελίδες HTML που παράγονται, καταλαβαίνει αν η λογική παράσταση που δήλωσε είναι αυτή που θέλει.

Η ταχύτητα απόκρισης του συστήματος κρίνεται ικανοποιητική.

Κεφάλαιο 6

Επίλογος / Συμπεράσματα

6.1 Αποτελέσματα της εργασίας

Στην παρούσα εργασία ασχοληθήκαμε με τη σχεδίαση και την κατασκευή μιας πύλης διασύνδεσης μεταξύ του Συστήματος Σημασιολογικού Ευρετηριασμού και του Παγκόσμιου Ιστού. Το σύστημα που κατασκευάστηκε μπορεί να χρησιμοποιηθεί για την πλοήγηση σε οποιοδήποτε σημασιολογικό δίκτυο που έχει κατασκευαστεί με τη χρήση της Telos.

Για την υλοποίηση της πύλης διασύνδεσης ακολουθήθηκε η αρχιτεκτονική CGI Application Server. Η πύλη διασύνδεσης είναι ο εξυπηρέτης εφαρμογής που δέχεται αιτήσεις μέσω του Παγκόσμιου Ιστού και παράγει κείμενο σύμφωνο με το πρότυπο HTML.

Για τον προσδιορισμό της παρουσίασης ενός αντικειμένου σε HTML, κατασκευάστηκε ένα μοντέλο παράστασης ιστοσελίδων. Ο προγραμματιστής εφαρμογής χρησιμοποιεί το μοντέλο για να περιγράψει με δηλωτικό τρόπο την παρουσίαση ενός αντικειμένου σε κείμενο HTML. Το μοντέλο αποτελεί τμήμα του σημασιολογικού δικτύου και η πύλη διασύνδεσης λαμβάνοντας υπόψη την πληροφορία που περιέχει, παράγει την τελική ιστοσελίδα.

Τα κύρια πλεονεκτήματα που προκύπτουν από τη χρήση του μοντέλου παρουσίασης ιστοσελίδων είναι :

- Το μοντέλο παρουσίασης μπορεί να χρησιμοποιηθεί για οποιοδήποτε αντικείμενο της βάσης, κόμβο ή γνώρισμα, ανεξαρτήτως του επιπέδου ταξινόμησης στο οποίο ανήκει.
- Υποστηρίζεται η ύπαρξη διαφορετικών όψεων παρουσίασης σε HTML του ίδιου αντικειμένου.
- Η ίδια λογική παράσταση μπορεί να χρησιμοποιηθεί για την παρουσίαση του ίδιου αντικειμένου, χρησιμοποιώντας διαφορετικά είδη διαμόρφωσης κειμένου.

- Διαφορετικές όψεις μπορούν να ορισθούν και για τα αντικείμενα του μοντέλου παρουσίασης.
- Υποστηρίζεται η χρησιμοποίηση Java scripts και applets. Μ'αυτόν τον τρόπο παρέχεται η δυνατότητα στον προγραμματιστή εφαρμογής να παρακάμψει τους περιορισμούς που επιβάλει το πρότυπο HTML.
- Υποστηρίζεται ιστορία ενός βήματος.

Το σύστημα χρησιμοποιήθηκε για τη συντήρηση των ιστοσελίδων του Τομέα Πληροφοριακών Συστημάτων και Τεχνολογίας Λογισμικού και για την κατασκευή ενός εργαλείου παραγωγής αναφορών, όπου το παραγόμενο κείμενο ήταν απλό και όχι HTML.

6.2 Μελλοντικές κατευθύνσεις

Για την τελειοποίηση του συστήματος που αναπτύχθηκε στην παρούσα εργασία υπάρχει ένας αριθμός επεκτάσεων, οι οποίες μπορούν να υλοποιηθούν στο μέλλον.

6.2.1 Διαλογική ενημέρωση του μοντέλου παρουσίασης

Ένα βασικό μειονέκτημα της παρούσας εργασίας είναι ότι η δημιουργία και η ενημέρωση του μοντέλου παρουσίασης δεν γίνεται με τρόπο διαλογικό (interactive). Επομένως, το πρώτο βήμα για την τελειοποίηση του συστήματος που υλοποιήθηκε είναι η κατασκευή ενός εργαλείου για τη διαλογική ενημέρωση του μοντέλου παρουσίασης. Επειδή το σύστημα που περιγράφηκε χρησιμοποιεί τον Παγκόσμιο Ιστό ως διεπιφάνεια χρήσης, και το εργαλείο για τη διαλογική ενημέρωση του μοντέλου παρουσίασης πρέπει να χρησιμοποιεί τον Παγκόσμιο Ιστό ως διεπιφάνεια χρήσης. Επειδή η HTML παρέχει πολύ περιορισμένες δυνατότητες, θα ήταν προτιμότερο να υλοποιηθεί χρησιμοποιώντας Java.

Βασικό στοιχείο για τη διαλογική ενημέρωση του μοντέλου παρουσίασης είναι να αποικονίζεται με γραφικό τρόπο το μοντέλο και να δίνεται στο χρήστη μια εικόνα της λογικής παράστασης που έχει ορίσει. Όπως γίνεται τώρα η δημιουργία και ενημέρωση του μοντέλου παρουσίασης, ο χρήστης δεν έχει μια γραφική ή κατανοητή παράσταση της δομής που έχει ορίσει. Από το αποτέλεσμα καταλαβαίνει αν όντως αυτό ήθελε να δηλώσει. Επομένως, η γραφική διαλογική ενημέρωση θα βοηθήσει τον προγραμματιστή εφαρμογής να δηλώνει ευκολότερα αυτό που επιθυμεί.

6.2.2 Παρουσίαση σε XML - SGML

Ένα επόμενο βήμα είναι η επέκταση της υλοποίησης για την παράσταση της πληροφορίας σε XML και SGML.

Στο τέταρτο κεφάλαιο και συγκεκριμένα στην περιγραφή του τμήματος παραγωγής της σελίδας HTML, περιγράψαμε τις προσθήκες που χρειάζονται ώστε το σύστημα να παράγει αποτέλεσμα σύμφωνο με την XML.

Για την παραγωγή παρουσίασης σύμφωνης με το πρότυπο SGML, οι αλλαγές είναι αντίστοιχες με αυτές για XML αν το αποτέλεσμα πρέπει να παραχθεί σύμφωνα με ένα συγκεκριμένο DTD (ή με ένα πλήθος προκαθορισμένων DTD). Αν όμως πρέπει το μοντέλο παρουσίασης να καθορίζει και το DTD που θα ακολουθηθεί, τότε πρέπει να γίνουν πολύ περισσότερες αλλαγές στο μοντέλο παρουσίασης.

6.2.3 Υποστήριξη πολύπλοκων επερωτήσεων

Όπως περιγράφηκε στο τρίτο κεφάλαιο, το μοντέλο παρουσίασης έχει σχεδιαστεί για ένα σύνολο απλών επερωτήσεων. Το μοντέλο παρουσίασης μπορεί να επεκταθεί για την υποστήριξη πολύπλοκων επερωτήσεων. Οι επερωτήσεις αυτές μπορεί να προσδιορίζονται και από τον τελικό χρήστη του συστήματος. Το τελευταίο απαιτεί τη δημιουργία μιας διεπιφάνειας χρήσης από όπου ο τελικός χρήστης θα μπορεί να επηρεάζει κάποιες παραμέτρους που αφορούν στην τελική παρουσίαση των δεδομένων μιας βάσης. Ακόμη, επειδή προς το παρόν δεν υπάρχει δηλωτική γλώσσα επερώτησης για το ΣΣΕ, όταν αυτή κατασκευαστεί θα ήταν μια χρήσιμη επέκταση ο δυναμικός καθορισμός επερωτήσεων από το χρήστη του συστήματος χρησιμοποιώντας την γλώσσα επερώτησης.

6.2.4 Βέλτιστος αριθμός ενεργών διεργασιών και καταλαμβανόμενης μνήμης

Στην περιγραφή του τμήματος επικοινωνίας στο τέταρτο κεφάλαιο αναφέρθηκε ότι ο αριθμός των ενεργών διεργασιών, καθορίζεται από τον προγραμματιστή εφαρμογής κατά την εκκίνηση του συστήματος. Για την εύρεση του βέλτιστου αριθμού ενεργών διεργασιών κάθε χρονική στιγμή, πρέπει να εξετασθεί ένα σύνολο αλγορίθμων και με βάση τις επιδόσεις του κάθε αλγορίθμου να επιλεγεί ένας ή περισσότεροι, αν αυτό είναι απαραίτητο.

Επιπλέον, ένας δυναμικός αλγόριθμος είναι απαραίτητος για την εύρεση του βέλτιστου αριθμού λογικών παραστάσεων που πρέπει να παραμένουν στην μνήμη του συστήματος.

6.2.5 Διανεμητής Αιτήσεων

Σύμφωνα με την υλοποίηση που έχει γίνει, δεν μπορούν να συνεργαστούν εξυπηρέτες εφαρμογής μεταξύ τους για την εξυπηρέτηση αιτήσεων. Έτσι, μπορεί να υπάρξει εξυπηρέτης εφαρμογής που έχει πολύ μεγάλο φόρτο εργασίας ενώ κάποιος άλλος δεν έχει σχεδόν καθόλου αιτήσεις προς εξυπηρέτηση. Αυτό το πρόβλημα μπορεί να λυθεί με τη χρήση ενός διανεμητή αιτήσεων (dispatcher). Ο διανεμητής αιτήσεων θα είναι υπεύθυνος για την εξισορρόπηση του φόρτου εργασίας μεταξύ των εξυπηρετών εφαρμογής με τρόπο διαφανή προς το χρήστη του συστήματος.

Επιπλέον, κατά την εξισορρόπηση φόρτου εργασίας πρέπει να λαμβάνεται υπόψη και το ποιά διεργασία από τις ενεργές είναι η καταλληλότερη για την εξυπηρέτηση της αίτησης. Κάθε διεργασία κατά την εκκίνησή της αρχικοποιείται με κάποιες άδειες, όπως ως προς χρήστης και με ποιούς κωδικούς (passwords) επικοινωνεί με τον εξυπηρέτη της βάσης. Επομένως, είναι προτιμότερο να χρησιμοποιείται μια διεργασία η οποία έχει αρχικοποιηθεί με τις κατάλληλες άδειες από τη δημιουργία μιας νέας για αυτόν τον σκοπό.

Τέλος, με τη χρησιμοποίηση ενός διανεμητή αιτήσεων μπορούν να λειτουργούν ταυτόχρονα πολλοί εξυπηρέτες εφαρμογής, που ο καθένας εξυπηρετεί μια διαφορετική βάση δεδομένων. Σύμφωνα με την τρέχουσα υλοποίηση ο χρήστης πρέπει να γνωρίζει τη διεύθυνση του κάθε εξυπηρέτη εφαρμογής. Στην περίπτωση που υπάρχει διανεμητής αιτήσεων ο χρήστης δεν χρειάζεται να γνωρίζει πού βρίσκεται ο κάθε εξυπηρέτης εφαρμογής για κάθε βάση δεδομένων, με την οποία θέλει να συνδεθεί.

6.2.6 Αύξηση του αριθμού βημάτων ιστορίας

Όπως έχει ήδη αναφερθεί, η ιστορία που κρατάει το σύστημα είναι ενός βήματος μόνο. Δηλαδή, το σύστημα μπορεί να ‘‘θυμάται’’ μόνο το προηγούμενο αντικείμενο που παρουσιάσθηκε. Μια ακόμη επέκταση είναι η αύξηση των βημάτων ιστορίας που συγκρατεί το σύστημα.

6.2.7 Γραφική παρουσίαση των δεδομένων

Ταυτόχρονα με την παρούσα εργασία εκτελείται μια εργασία για την γραφική παρουσίαση των δεδομένων ενός σημασιολογικού δικτύου που έχει κατασκευαστεί με τη χρήση της Telos. Ένα σημαντικό βήμα θα είναι η σύνδεση σε ένα ενιαίο σύστημα των δύο συστημάτων. Με αυτόν τον τρόπο ο τελικός χρήστης θα μπορεί να πλοηγηθεί στο σημασιολογικό δίκτυο είτε γραφικά είτε χρησιμοποιώντας απλό κείμενο μέσω του Παγκόσμιου Ιστού.

6.2.8 Δημιουργία ενός ΣΔΚΠΙ χρησιμοποιώντας το ΣΣΕ

Για τη δημιουργία ενός ΣΔΚΠΙ τα απαραίτητα βήματα είναι ο διαχωρισμός των δεδομένων από την παρουσίασή τους και η υποστήριξη διαφορετικών παρουσιάσεων των δεδομένων σε HTML. Επιπλέον, απαιτείται ένας μηχανισμός άντλησης πληροφορίας από ετερογενείς πηγές δεδομένων.

Το μοντέλο παρουσίασης σε HTML, που περιγράφηκε στο τρίτο κεφάλαιο διαχωρίζει τα δεδομένα από την παρουσίασή τους σε HTML και υποστηρίζει την ύπαρξη διαφορετικών όψεων παρουσίασης των δεδομένων σε HTML. Επιπλέον, υποστηρίζει τη παρουσίαση των δεδομένων χρησιμοποιώντας διαφορετικά είδη διαμόρφωσης κειμένου.

Συνεπώς, για τη δημιουργία ενός ΣΔΚΠΙ χρησιμοποιώντας το ΣΣΕ είναι απαραίτητη η κατασκευή μεταφραστών (wrappers) για την επικοινωνία του ΣΣΕ με εξωτερικές πηγές δεδομένων, η κατασκευή μιας δηλωτικής γλώσσας επερώτησης και η υλοποίηση ενός μηχανισμού επιβολής και διατήρισης περιορισμών.

Παράρτημα Α

Συντομογραφίες

ΕΛ	Ενδιάμεσο Λογισμικό
ΕΛΒΔ	Ενδιάμεσο Λογισμικό για Βάσεις Δεδομένων
ΙΠ	Ινστιτούτο Πληροφορικής
ΙΤΕ	Ίδρυμα Έρευνας και Τεχνολογίας
ΠΙ	Παγκόσμιος Ιστός
ΣΔΘΟ	Σύστημα Διαχείρισης Θησαυρών Όρων
ΣΔΚΠΙ	Σύστημα Διαχείρισης Κόμβων του Παγκόσμιου Ιστού
ΣΣΕ	Σύστημα Σημασιολογικού Ευρετηριασμού

API	Application Programming Interface
CGI	Common Gateway Interface
COM	Component Object Model
CORBA	Common Object Request Broker Architecture
DCOM	Distributed Component Object Model Protocol
JDBC	Java DataBase Connectivity
HTML	HyperText Markup Language
HTTP	HyperText Transfer Protocol
HTS	HyperText Sybase
ODBC	Open DataBase Connectivity
OQL	Object Query Language
RPC	Remote Procedure Call
URI	Uniform Resource Identifier
URL	Uniform Resource Locator

SIS	Semantic Index System
SGML	Standard Generalized Markup Language
TMS	Thesaurus Management System
XML	Extensible Markup Language
WRB	Web Request Broker

Παράρτημα Β

Αγγλο-ελληνικό γλωσσάριο

API	Διεπιφάνεια Προγραμματισμού Εφαρμογών
Application pages	Σελίδες Εφαρμογής
Application server	Εξυπηρέτης Εφαρμογής
Background	Φόντο
Browser	Αναδιφητής
Buffer	Ενταμιευτής
Cache Memory	Κρυφή Μνήμη
Client	Εξυπηρετούμενος
Constraint	Περιορισμός
Database Middleware	Ενδιάμεσο Λογισμικό για Βάσεις Δεδομένων
Default	Τυπικός
Dispatcher	Διανεμητής Αιτήσεων
Format	Μορφότυπος
Frame	Πλαίσιο
Hyperlink	Υπερ-σύνδεσμος
Hypertext	Υπερ-κείμενο
Gateway	Πύλη Διασύνδεσης
Interactive	Διαλογικός
Internet	Διαδίκτυο
Load Balancing	Εξισορρόπηση Φορτίου
Macro file	Αρχείο Μακρο-εντολών
Margin	Περιθώριο
Match	Ταίριασμα

Mediator	Διαμεσολαβητής
Middleware	Ενδιάμεσο Επίπεδο Λογισμικού
Navigation	Πλοήγηση
Official Web Site	Επίσημος Κόμβος στον Παγκόσμιο Ιστό
Overloading (Method)	Υπερφόρτωση (μεθόδου)
Page-Oriented	Σελιδο-κεντρικό
Page-scheme	Σχήμα Σελίδων
Password	Συνθηματική Λέξη
Process Management	Διαχείριση Διεργασιών
Query	Επερώτηση
Query Language	Γλώσσα Επερώτησης
Relational Data Model	Σχεσιακό Μοντέλο Δεδομένων
Reaction	Αντίδραση
Report Writer	Εργαλείο Παραγωγής Αναφορών
Scaleable	Κλιμακούμενος
Scrollbar	Ράβδος Ολίσθησης
Server	Εξυπηρέτης
Source	Πηγή
Standard Interfaces	Πρότυπες Διεπιφάνειες (Χρήσης)
Standard Data Formats	Πρότυποι Μορφότυποι Δεδομένων
Structured	Δομημένος
Tag	Ετικέτα Διαμόρφωσης
Template	Σκελετός
Transaction Monitor	Επιτηρητής Δοσοληψιών
Tuple	Πλειάδα
Web Browser	Αναδιφητής του Παγκόσμιου Ιστού
Web Site	Κόμβος του Παγκόσμιου Ιστού
Web Site Management System	Σύστημα Διαχείρισης Κόμβων του Παγκόσμιου Ιστού
World Wide Web	Παγκόσμιος Ιστός
Wrapper	Μεταφραστής

Παράρτημα Γ

Μοντέλο παρουσίασης ιστοσελίδων σε Telos

```
TELL Attribute html_presentation
  from: Individual
  to: Individual
in S_Class
end

TELL Individual View in M1_Class with
attribute
  default_presentation: Document;
end

TELL Individual Presentation in M1_Class
end

TELL Individual HTML_Presentation in S_Class, Presentation
end

TELL Individual XML_Presentation in S_Class, Presentation
end

TELL Individual Latex_Presentation in S_Class, Presentation
end

TELL Individual ASCII_Presentation in S_Class, Presentation
end
```

```
TELL Individual Document in S_Class with
attribute
  prologue : Prologue;
  epilogue : Epilogue;
  contains : Container;
  coming_from : Document;
end

TELL Individual Prologue in S_Class with
attribute
  has_title : Text;
end

TELL Individual Epilogue in S_Class
end

TELL Individual Container in S_Class
end

TELL Individual BasicContainer in S_Class isA Container with
attribute
  header : Header;
  footer : Footer;
  body : Body;
  has_background : Background;
  visited_color : Color;
  unvisited_color : Color;
  activated_color : Color;
end

TELL Individual FrameContainer in S_Class isA Container
end

TELL Individual FrameSet in S_Class isA FrameContainer with
attribute
  contains : FrameContainer;
  rows : ValueList;
  cols : ValueList;
  noFrame : BasicContainer;
```

```
end

TELL Individual FrameElement in S_Class isA FrameContainer with
attribute
  target : Telos_String;
  src : Telos_String;
  obj : Telos_String;
  view : Telos_String;
  name : Telos_String;
  marginHeight : Telos_Integer;
  marginWidth : Telos_Integer;
  scrolling : Boolean;
  resize : Boolean;
end

TELL Individual ValueList in S_Class
end

TELL Individual Boolean in S_Class
end

TELL Individual Color in S_Class isA Background
end

TELL Individual Background in S_Class
end

TELL Individual Header in S_Class with
attribute
  composed_of : MMediaData;
end

TELL Individual Footer in S_Class with
attribute
  composed_of : MMediaData;
end

TELL Individual Body in S_Class with
attribute
  presentation_type : PresentationType;
```

```
end

TELL Individual BodyPart in S_Class isA Body with
attribute
  composed_of : Body;
end

TELL Individual BodyElement in S_Class isA Body with
attribute
  expansion : Expansion;
  font : Font;
  composed_of : BodyData;
end

TELL Individual Font in S_Class with
attribute
  size : Size;
  color : Color;
  face : Face;
end

TELL Individual Size in S_Class
end

TELL Individual Face in S_Class
end

TELL Individual BodyData in S_Class
end

TELL Individual MMediaData in S_Class isA BodyData
end

TELL Individual Audio in S_Class isA MMediaData
end

TELL Individual Video in S_Class isA MMediaData
end

TELL Individual Text in S_Class isA MMediaData
```

```
end

TELL Individual File in S_Class isA Text
end

TELL Individual Hyperlink in S_Class isA Text with
attribute
  label : Text;
  src : Text;
  obj : Telos_String;
  view : Telos_String;
  state : Telos_String;
end

TELL Individual Mailto in S_Class isA Hyperlink
end

TELL Individual Separator in S_Class isA Text
end

TELL Individual Image in S_Class isA MMediaData, Background
end

TELL Individual FactualData in S_Class isA BodyData with
attribute
  separator : MMediaData;
  order : Order;
  title : MMediaData;
  attribute_label : MMediaData;
  number_of_displayed_items : Telos_Integer;
  continue_label : MMediaData;
end

TELL Individual Superclasses in S_Class isA FactualData
end

TELL Individual Subclasses in S_Class isA FactualData
end

TELL Individual ClassesOf in S_Class isA FactualData
```

```
end

TELL Individual Instances in S_Class isA FactualData
end

TELL Individual FromAttributes in S_Class isA FactualData
end

TELL Individual ToAttributes in S_Class isA FactualData
end

TELL Individual FromNodes in S_Class isA FactualData
end

TELL Individual ToNodes in S_Class isA FactualData
end

TELL Individual FromAttributesByClass in S_Class isA FactualData
end

TELL Individual FromAttributesByMetaClass in S_Class isA FactualData
end

TELL Individual ToAttributesByClass in S_Class isA FactualData
end

TELL Individual ToAttributesByMetaClass in S_Class isA FactualData
end

TELL Individual PresentationType in S_Class with
attribute
  format : Format;
  media_type : MediaType;
end

TELL Individual Format in S_Class
end

TELL Individual UserDefined in S_Class isA Format
end
```

```
TELL Individual Predefined in S_Class isA Format
end

TELL Individual TimeFormat in S_Class isA Format
end

TELL Individual LinkFormat in S_Class isA Format
end

TELL Individual TextFormat in S_Class isA Format
end

TELL Individual Table in S_Class isA Predefined
end

TELL Individual List in S_Class isA Predefined
end

TELL Individual VariousFormat in S_Class isA UserDefined
end

TELL Individual UserFunction in S_Class isA UserDefined with
attribute
  code : Telos_String;
end

TELL Individual JavaScript in S_Class isA UserFunction
end

TELL Individual JavaApplet in S_Class isA UserFunction
end

TELL Individual HyperlinkAll in Token, LinkFormat
end

TELL Individual HyperlinkFromValue in Token, LinkFormat
end

TELL Individual HyperlinkToValue in Token, LinkFormat
```

```
end

TELL Individual HyperlinkLabel in Token, LinkFormat
end

TELL Individual NoHyperlinkAll in Token, LinkFormat
end

TELL Individual NoHyperlinkFromValue in Token, LinkFormat
end

TELL Individual NoHyperlinkToValue in Token, LinkFormat
end

TELL Individual NoHyperlinkLabel in Token, LinkFormat
end

TELL Individual MediaType in S_Class
end

TELL Individual AudioType in Token, MediaType
end

TELL Individual ImageType in Token, MediaType
end

TELL Individual TextType in Token, MediaType
end

TELL Individual VideoType in Token, MediaType
end

TELL Individual Expansion in S_Class, View with
attribute
  over_object: Individual;
end

TELL Individual No_Expand_No_Node in S_Class isA Expansion
end
```

```
TELL Individual Expand_Node in S_Class isA Expansion
end

TELL Individual Expand_No_Node in S_Class isA Expansion
end

TELL Individual No_Expand_No_Hyperlink in S_Class isA Expansion
end

TELL Individual No_Expand_Hyperlink in S_Class isA Expansion with
attribute
  label : Text;
  src : Text;
  object : Telos_String;
  view : Telos_String;
end

TELL Individual Recursive_Expansion in S_Class isA Expansion with
attribute
  depth : Telos_Integer;
end

TELL Individual Expand_From_Node in S_Class isA Expansion
end

TELL Individual Expand_To_Node in S_Class isA Expansion
end

TELL Individual This in S_Class
end

TELL Individual LogicalName in S_Class isA This
end

TELL Individual LogicalNameFromNode in S_Class isA This
end

TELL Individual LogicalNameToNode in S_Class isA This
end
```

```
TELL Individual Level in S_Class isA This
end

TELL Individual Type in S_Class isA This
end

TELL Individual Order in S_Class
end
```

Βιβλιογραφία

[1] Software Technology Review. Two Tier Software Architectures. http://www.sei.cmu.edu/str/descriptions/twotier.html.

[2] Software Technology Review. Three Tier Software Architectures. http://www.sei.cmu.edu/str/descriptions/threetier.html.

[3] Software Technology Review. Client/Server Software Architectures--An Overview. http://www.sei.cmu.edu/str/descriptions/clientserver_body.html.

[4] Network Computing Online. Planning and Troubleshooting Middleware Implementations. http://www.nwc.com/netdesign/cdmwoutline.htm, by Bruce Robertson.

[5] Network Computing Online. The Ups And Downs Of Analyzing Middleware. http://www.nwc.com/902/902ws2.html, by Barry Nance.

[6] Network Computing Online. Middleware Should Play The Name Game. http://www.nwc.com/814/814colrobertson.html, by Bruce Robertson.

[7] Information Week OnLine : News in Review. Middleware Evolution. http://www.informationweek.com/690/90iumid.htm, by Alan Radding.

[8] International Systems Group and Inc. Middleware White Paper - Middleware-The Essential Component for Enterprise Client/Server Applications. http://www.openvms.digital.com/openvms/whitepapers/middleware/isgmidware.html, 1997.

[9] OMG Home Page. http://www.omg.org/.

[10] Microsoft COM. http://www.microsoft.com/com/.

[11] Apple and IBM Technologies. OpenDOC. http://www.opendoc.apple.com/.

[12] Microsoft. ODBC 3.0. http://www.microsoft.com/products/prodref/264_ov.htm.

[13] The $JDBC^{TM}$ Database Access API. http://java.sun.com/products/jdbc/index.html.

[14] World Wide Web Consortium. HTTP - HyperText Transfer Protocol. http://www.w3.org/Protocols/.

[15] World Wide Web Consortium. HyperText Markup Language. http://www.w3.org/MarkUp/.

[16] P. Constantopoulos and M. Doerr. The SIS System: A Brief Presentation. ICS-FORTH, http//www.csi.forth.gr/isst, May 1993.

[17] J. Mylopoulos, A. Borgida, M. Jarke, and M. Koubarakis. Telos: Representing Knowledge About Information Systems. *ACM Transactions on Information Systems*, 8(4):325--362, 1990.

[18] World Wide Web Consortium. Extensible Markup Language - XML. http://www.sil.org/sgml/xml.html.

[19] Charles F. Goldfarb. *The SGML Handbook*. 1990.

[20] Apache HTTP Server Project. Apache Project. http://www.apache.org/.

[21] World Wide Web Consortium. HTML 3.2 Reference Specification. http://www.w3.org/TR/REC-html32.html.

[22] Wayne W. Eckerson. Three-Tier Client/Server Architecture. *OPEN INFORMATION SYSTEMS*, 10(1), 1995.

[23] Craig Thompson. Requirements for OO + Web Integration. http://www.objs.com/survey/Rqmts.htm.

[24] Pyung-Chul Kim. A Taxonomy on the Architecture of Database Gateways for the Web. http://grigg.chungnam.ac.kr/projects/UniWeb/documents/taxonomy/text.html.

[25] CGI: Common Gateway Interface. http://www.w3.org/cgi.

[26] Naming and Addressing: URIs. http://www.w3.org/addressing.

[27] Tam Nguyen and V. Srinivasan. Accessing Relational Databases from the World Wide Web. In *SIGMOD '96*, Montreal, Canada, June 1996.

[28] John Gaffney. Illustra's Web DataBlade Module. *SIGMOD Record*, 25(1), March 1996.

[29] PHP/FI. http://www.iserver.com/servers/extensions/phpfi/.

[30] WebinTool Home Page. http://www.ri.bbsrc.ac.uk/webintool.html.

[31] GSQL a Mosaic-SQL gateway. http://www.ncsa.uiuc.edu/sdg/people/jason/pub/gsql/starthere.html.

[32] Web/Genera. http://gdbdoc.gdb.org/letovsky/genera/genera.html.

[33] WDB. http://venus.dtv.dk/~bfr/wdb/.

[34] SQLGate Home Page. http://think.ucdavis.edu/~cgi/sqlgate/.

[35] O2 Technologies. *O_2Web User Manual*, release 4.6 edition, January 1996.

[36] Oracle Corporation. Oracle Web Application Server 3.0. http://www.oracle.com/, May 1997.

[37] LINCKS. http://www.ida.liu.se/~lincks/.

[38] SIBYLLA. http://www.ariadne.it/sibylla/sibylla.en.html.

[39] Charles Lamb, Gordon Landis, Jack Orenstein, and Dan Weinreb. The ObjectStore System. *CACM*, 10(34):50--63, March 1991.

[40] NetDynamics. http://www.netdynamics.com/products/.

[41] O2 Technologies. http://www.o2tech.fr/.

[42] NSAPI. http://developer.netscape.com:80/docs/manuals/enterprise/nsapi.

[43] Microsoft Internet Information Server's API. http://www.microsoft.com/win32dev/apiext/isalegal.htm.

[44] Sybase web.sql. http://www.sybase.com/products/internet/websql/.

[45] Sybase. http://www.sybase.com.

[46] AOL Server. http://www.aolserver.com/server/index.html.

[47] UniSQL. http://www.unisql.com.

[48] Mary Fernandez, Daniela Florescu, Jaewoo Kang, Alon Levy, and Dan Suciu. STRUDEL: A Web Site Management System. http://www.research.att.com/ mff/strudel/doc/, 1997.

[49] Mary Fernandez, Daniela Florescu, Alon Levy, and Dan Suciu. Reasoning About Web-Site Structure. http://www.research.att.com/ mff/strudel/doc/, 1998.

[50] Mary Fernandez, Daniela Florescu, Alon Levy, and Dan Suciu. A Query Language for a Web-Site Management System. http://www.research.att.com/ mff/strudel/doc/, 1998.

[51] G. Mecca, P. Atzeni, A. Masci, P. Merialdo, and G. Sindoni. The ARANEUS Web-Base Management System. In *XI ERCIM Database Working Group Workshop: "Metadata in Web Databases"*, May 1998.

[52] G. Mecca, P. Atzeni, P. Merialdo, A. Masci, and G. Sindoni. From Databases to Web-Bases: The ARANEUS Experience. Rt-dia-34-1998, University of Rome III, Italy, May 1998.

[53] G. Mecca, A. Mendelzon, and P. Merialdo. Efficient Queries over Web Views. Rt-dia-31-1998, University of Rome III, Italy, January 1998.

[54] P. Atzeni, G. Mecca, and P. Merialdo. Semistructured and Structured Data in the Web: Going Back and Forth. In *Sigmod Record, Special Issue on the Workshop on the Management of Semistructured Data*, 1997.

[55] P. Atzeni, G. Mecca, P. Merialdo, and G. Sindoni. A Logical Model for Metadata in Web Bases. In *Exhibits Program of ACM SIGMOD '98*, 1998.

[56] Jerome Simeon and Sophie Cluet. Using YATT to Build a Web Server. In *International Workshop on the Web and Databases - WebDB'98*, Valencia, Spain, March 1998.

[57] Y. Papakonstantinou, H. Garcia-Molina, and J. Widom. Object exchange accross heterogeneous information sources. In *Proc. IEEE International Conference on Deductive and Object-Oriented Databases*, 1995.

[58] Institute of Computer Science, Foundation for Research and Technology, Hellas (ICS-FORTH). *SIS - Appication Programmatic Interface: Reference Manual*, v2.1.1 edition, 1997.

[59] Institute of Computer Science, Foundation for Research and Technology, Hellas (ICS-FORTH). *SIS - Query Interpreter: An Interactive Program to Use PQI functions*, v2.0 edition, 1997.

[60] W. E. Moen. A Guide to the ANSI/NISO Z39.50 Protocol: Information Retrieval in the Information Infrastructure. NISO Press, http://www.cni.org/pub/NISO/docs/Z39.50-brochure/50.brochure.toc.html, 1995.

[61] Ιωάννης Ε. Βελεγράκης. Δηλωτικός προσδιορισμός μεταφραστών Z39.50 με χρήση Περιγραφικής Λογικής. Μεταπτυχιακή Εργασία, Τμήμα Επιστήμης Υπολογιστών, Πανεπιστήμιο Κρήτης, Ιούνιος 1998.

[62] Martin Doerr and Irini Fundulaki. SIS - TMS : A Thesaurus Management System for Distributed Digital Collections. In *Proc. 2nd European Conference on Digital Libraries - EuroDL'98*, Heraclion, Crete, Greece, September 1998.

[63] Insitute of Computer Science, Foundation for Research and Technology, HELLAS. SIS - TMS : The Thesaurus Management System by ICS-FORTH. http://www.ics.forth.gr/proj/isst/Systems/TMS/, May 1998.