



Πανεπιστήμιο Αιγαίου
Σχολή Κοινωνικών Επιστημών
Τμήμα Γεωγραφίας

Μεταπτυχιακό Πρόγραμμα Σπουδών:

«ΓΕΩΓΡΑΦΙΑ ΚΑΙ ΣΧΕΔΙΑΣΜΟΣ ΣΤΗΝ ΕΥΡΩΠΗ ΚΑΙ ΤΗ ΜΕΣΟΓΕΙΟ»

Διπλωματική Εργασία του *Παντελέλη Μιχαήλ* για το ΜΔΕ με τίτλο:

«Διαχείριση και Ανάλυση Χωροχρονικών Δεδομένων:
Παράδειγμα Εφαρμογής στην Αέρια Ρύπανση Αστικών Περιοχών»

Τριμελής Επιτροπή:

Επιβλέπων:

Επικ.Καθ. Σουλακέλλης Νικόλαος

Μέλη:

Καθ. Σελλής Τίμος

Λέκ. Βαΐτης Μιχαήλ

Μυτιλήνη, Δεκέμβριος 2003

Δήλωση Πρωτοτυπίας

Εγώ, ο Μιχαήλ Παντελέλης του Γρηγορίου, δηλώνω υπεύθυνα, ότι η παρούσα εργασία αποτελεί προϊόν δικής μου εργασίας και ότι όπου έχει χρησιμοποιηθεί εργασία τρίτων τους έχει αναγνωριστεί στο σύνολό της.

31 Δεκεμβρίου 2003
Μιχαήλ Γ. Παντελέλης

Δικαιώματα

Τα δικαιώματα χρήσης της παρούσας διπλωματικής εργασίας διατηρούνται από το συγγραφέα, Μιχαήλ Γ. Παντελέλη και το Τμήμα Γεωγραφίας του Πανεπιστημίου Αιγαίου. Οι ιδέες που περιέχονται στην παρούσα εργασία αποτελούν πνευματική ιδιοκτησία του συγγραφέα και του επιβλέποντα την εργασία, εκτός εάν αναφέρεται διαφορετικά.

Διατηρούνται όλα τα δικαιώματα. Η χρήση οποιουδήποτε τμήματος της παρούσας εργασίας ανατυπωμένο, μεταδιδόμενο με οποιοδήποτε τρόπο, ηλεκτρονικά, μηχανικά, με φωτοτυπία, αναπαραγωγή ή με άλλο τρόπο, ή αποθηκευμένο σε οποιοδήποτε σύστημα ανάκτησης, χωρίς την προηγούμενη έγγραφη άδεια του συγγραφέα και του Τμήματος Γεωγραφίας του Πανεπιστημίου Αιγαίου, απαγορεύεται.

Πρόλογος

Ήδη από τα χρόνια της φοίτησής μου στο Εθνικό Μετσόβιο Πολυτεχνείο, η επαφή μου με την τεχνολογία των συστημάτων διαχείρισης βάσεων δεδομένων μου είχε κεντρίσει το ενδιαφέρον. Λίγα πράγματα είχα ακούσει και για τα συστήματα γεωγραφικών πληροφοριών και σχεδόν έψαχνα, σαν από διαίσθηση ένα τρόπο να ασχοληθώ και με το αντικείμενο αυτό. Η ευκαιρία μου δόθηκε και μάλιστα στην ιδιαίτερή μου πατρίδα τη Μυτιλήνη, αμέσως μετά την ολοκλήρωση της στρατιωτικής μου θητείας, όταν πρωτοξεκίνησε η λειτουργία του μεταπτυχιακού προγράμματος σπουδών στο Τμήμα Γεωγραφίας. Το αντικείμενο ήταν ανέλπιστα, ενδιαφέρον αφού τελικά μπορέσα να εξερευνήσω τόσο τα σύγχρονα συστήματα γεωγραφικών πληροφοριών, αλλά και να επεκτείνω τις γνώσεις μου στο πεδίο των βάσεων δεδομένων και μάλιστα στην πιο ενδιαφέρουσα μορφή τους.

Μπορώ να πώ ότι στάθηκα τυχερός τόσο χωρικά, αφού μπόρεσα να μείνω όλο αυτό τον καιρό στη Μυτιλήνη, όπου δύσκολα βρίσκει κανείς ευκαιρίες απασχόλησης αλλά και ενασχόλησης γενικότερα, όσο και χρονικά αφού συνέπεσε να μελετήσω ένα αντικείμενο στο οποίο τα τελευταία πέντε χρόνια έχει γίνει σημαντική πρόοδος τεχνολογικά και αποτελεί ένα ευρύ πεδίο έρευνας.

Τμήμα της παρούσας εργασίας το οποίο αφορά στη μοντελοποίηση της βάσης δεδομένων έχει παρουσιαστεί στο 13^ο Πανελλήνιο συνέδριο χρηστών ArcView - ArcGIS το οποίο έλαβε χώρα στις 13 και 14 Νοεμβρίου 2003 στην Θεσσαλονίκη υπό διοργάνωση της Marathon Data Systems.

Το σύνολο της παρούσας εργασίας πρόκειται να παρουσιαστεί υπό την μορφή δημοσίευσης στην «Ημερίδα Νέων Ερευνητών» που διοργανώνει η HellasG.I.s στην Αθήνα τον Ιανουάριο του 2004.

Το πλήρες κείμενο της παρούσας διπλωματικής εργασίας στοιχειοθετήθηκε με την χρήση του συστήματος L^AT_EX, σε λειτουργικό σύστημα Linux. Η επεξεργασία εικόνων έγινε με το GIMP.

Ευχαριστίες

Με την ολοκλήρωση της παρούσας διπλωματικής εργασίας θα ήθελα να ευχαριστήσω όλους εκείνους τους οποίους με τον ένα ή τον άλλο τρόπο με βοήθησαν.

Με ιδιαίτερη εκτίμηση θέλω να αναφερθώ στο Μιχάλη Ζαχαράκη για την ιδανική συνεργασία που έχουμε στην Μονάδα Πληροφορικής του Πανεπιστημίου Αιγαίου, όπου εργαζόμουν παράλληλα με την παρακολούθηση του μεταπτυχιακού. Αν και η καθυστέρηση της ολοκλήρωσης της διπλωματικής αυτής, οφείλεται εν μέρει και στην απασχόλησή μου εκεί, μπόρεσα έτσι να δώ πώς είναι κάποια πράγματα στη πράξη, χάρη στην εμπιστοσύνη που από την αρχή μου έδειξε.

Επίσης θέλω να ευχαριστήσω ιδιαίτερα το καθηγητή μου κ. Νικόλαο Σουλακέλλη όχι μόνο γιατί με παρότρυνε στο ξεκίνημα του μεταπτυχιακού αλλά, γιατί συνεχίζει ακόμα να με υποστηρίζει και να με συμβουλεύει.

Οι συμφοιτητές μου έχουν ιδιαίτερη θέση σε αυτή την μικρή αναφορά: η Χριστίνα, ο Ηλίας και η Ντίνα, ο Ιγνάτης, ο Θοδωρής, η Άρτεμις και η Βίκη. Με άλλους περισσότερο και με άλλους λιγότερο, οι συνεργασίες και οι συζητήσεις που κάναμε ήταν καθοριστικές στο να καταλάβω πράγματα και να δώ πολλές φορές την πραγματικότητα από μία άλλη οπτική.

Τέλος, θέλω να ευχαριστήσω την τριμελή επιτροπή, που δέχτηκε να εξετάσει το αποτέλεσμα της παρούσας εργασίας, μαζί με όλους όσους μπήκαν στο κόπο να την διαβάσουν και με τις παρατηρήσεις τους να συμβάλουν στην βελτίωσή της.

Μυτιλήνη, 31 Δεκεμβρίου 2003

Μιχαήλ!

στη Μητέρα μου

Περιεχόμενα

Δήλωση Πρωτοτυπίας	i
Δικαιώματα	ii
Πρόλογος	iii
Ευχαριστίες	iv
Περιεχόμενα	vi
Κατάλογος Σχημάτων	ix
Γλωσσάρι	xii
Περίληψη	xiii
Abstract	xiv
1 Εισαγωγή	1
1.1 Σκοπός	1
1.2 Η βάση δεδομένων	3
1.3 Αντικειμενοστραφής σχεδιασμός της βάσης δεδομένων με χρήση της UML	4
1.4 Περιγραφή των δεδομένων με χρήση της XML	4
1.5 Υλοποίηση	6
1.6 Παρουσίαση	6
2 Συστήματα Γεωγραφικών Πληροφοριών	8
2.1 Γενικά	8
2.2 Ιστορική Αναδρομή	9
2.3 Λειτουργία	10
2.4 Επεκτάσεις	11

3 Χωροχρονικές Βάσεις Δεδομένων	14
3.1 Αρχιτεκτονική βάσεων δεδομένων	14
3.2 Δεδομένα σε μία βάση δεδομένων	17
3.3 Μοντέλα δεδομένων	18
3.4 Πρόσβαση στα δεδομένα	19
3.5 Ερωτήσεις στη βάση δεδομένων	20
3.5.1 Τελεστές	21
3.5.2 Γλώσσες ερωτήσεων	22
3.5.3 Νέες τεχνικές ανάλυσης	22
3.6 Ποιότητα δεδομένων	23
4 Το σύστημα Icaros.NET	24
4.1 Εισαγωγή	24
4.2 Σκοπός	24
4.3 Αντίστοιχες εργασίες	26
4.4 Απαιτήσεις υλοποίησης	27
4.4.1 Λειτουργία	28
4.4.2 Περιβάλλον εργασίας	28
4.4.3 Δεδομένα	29
4.5 Αρχιτεκτονική	30
4.6 Εφαρμογή	31
4.6.1 Προκλήσεις	32
5 Ενοποιημένη Γλώσσα Μοντελοποίησης (UML)	34
5.1 Τί είναι;	34
5.2 Ιστορικά	35
5.3 Γιατί μοντελοποιούμε;	36
5.3.1 Η σημασία ενός μοντέλου	36
5.3.2 Αρχές Μοντελοποίησης	37
5.3.3 Αντικειμενοστραφής Προσέγγιση	37
5.3.4 Η UML σαν εργαλείο οπτικοποίησης - παρουσίασης	39
5.4 Τα διαγράμματα της UML	39
5.5 UML και Γεωγραφικά Συστήματα Πληροφοριών	40
5.6 Εφαρμογές	42
5.7 Πώς αξιοποιείται στο Icaros.NET;	43
6 Επεκτάσιμη Γλώσσα Σήμανσης (XML)	44
6.1 Τί είναι;	44
6.2 Ιστορικά	44
6.3 Πού χρησιμοποιείται;	46
6.4 Πώς υλοποιείται;	47

6.4.1	Ο XML parser	47
6.4.2	SAX και DOM	48
6.4.3	Αποθήκευση αρχείων	48
6.5	Τι δυνατότητες δίνει;	49
6.6	Πώς αξιοποιείται στο <i>Icaros.NET</i> ;	50
7	Υλοποίηση	52
7.1	Εισαγωγή	52
7.2	Αρχιτεκτονική του Επιπέδου Διαχείρισης Δεδομένων	52
7.3	Η γεωγραφική βάση δεδομένων	53
7.3.1	Η επιλογή της UML για τον ορισμό του σχήματος της Βάσης.	55
7.3.2	Σχεδιαστικά ζητήματα	56
7.3.3	Στατιστικά στοιχεία	58
7.4	Το σχήμα περιγραφής των αρχείων XML	59
7.4.1	DTD ή XML Schema	59
7.4.2	Από το σχήμα της βάσης στο XML Schema	60
7.5	Η βιβλιοθήκη υλοποίησης	61
7.5.1	Απαιτήσεις	61
7.5.2	Η βιβλιοθήκη	61
7.6	Είσοδος Δεδομένων	63
7.7	Πλεονεκτήματα προσέγγισης	64
8	Αποτελέσματα – Συμπεράσματα	65
8.1	Αρχικοί Στόχοι	65
8.2	Τι Υλοποιήθηκε	66
8.3	Συμπεράσματα	67
8.4	Μελλοντική Εργασία	67
	Παράρτημα	69
A'	Το Σχήμα της Γεωγραφικής Βάσης Δεδομένων	70
A'.1	Εισαγωγή	70
A'.2	Παρουσίαση	70
A'.2.1	Γενικά Βοηθητικά Αντικείμενα	71
A'.2.2	Τύποι Κωδικοποίησης (Coded Value Domains)	73
A'.2.3	Τύποι Περιοχής Τιμών (Range Value Domains)	78
A'.2.4	Περιγραφή Επίγειων Σταθμών Μέτρησης	84
A'.2.5	Περιγραφή Ατμοσφαιρικών Μοντέλων	95

Β' Το σχήμα ορισμού των αρχείων της XML	105
Β'.1 Εισαγωγή	105
Β'.2 Παρουσίαση	105
Γ' Η υλοποίηση της βιβλιοθήκης	106
Γ.1 Εισαγωγή	106
Γ.2 Παρουσίαση	106
Γ.2.1 Η κλάση CIcarosData	107
Γ.2.2 Η κλάση CIcarosDataSource	109
Γ.2.3 Η κλάση CMonitoringStation	113
Γ.2.4 Η κλάση CIcarosHelper	116
Γ.2.5 Η κλάση CMeasuredQuantities	117
Γ.2.6 Η κλάση CModelGrid	121
Γ.2.7 Η κλάση CModelResult	124
Γ.2.8 Η κλάση CModelResults	126
Γ.2.9 Η κλάση Cmeasurement	130
Γ.2.10 Συνοπτικό μοντέλο Visual Basic - UML	132
Δ' Ο συνοδευτικός οπτικός δίσκος	133
Δ'.1 Εισαγωγή	133
Δ'.2 UML και XML	133
Δ'.3 Ο πηγαίος Κώδικας	133
Βιβλιογραφία	134

Κατάλογος Σχημάτων

4.1	Η αρχιτεκτονική του συστήματος και οι ισχυρές εξαρτήσεις από το επίπεδο διαχείρισης δεδομένων	31
7.1	Η αρχική οθόνη της πλατφόρμας <i>Icaros.NET</i>	53
7.2	Η λειτουργία της βιβλιοθήκης	54
A'.1	Ορισμός γενικών αντικειμένων	71
A'.2	Τύποι κωδικοποίησης A'	76
A'.3	Τύποι κωδικοποίησης B'	77
A'.4	Τύποι περιοχής τιμών A'	80
A'.5	Τύποι περιοχής τιμών B'	81
A'.6	Τύποι περιοχής τιμών Γ'	82
A'.7	Τύποι περιοχής τιμών Δ'	83
A'.8	Το μοντέλο για τους επίγειους σταθμούς μέτρησης	84
A'.9	Υπότυποι μετεωρολογικών μετρήσεων A'	88
A'.10	Υπότυποι μετεωρολογικών μετρήσεων B'	89
A'.11	Υπότυποι μετρήσεων ρύπων A'	90
A'.12	Υπότυποι μετρήσεων ρύπων B'	92
A'.13	Υπότυποι μετρήσεων ρύπων Γ'	93
A'.14	Υπότυποι μετρήσεων ρύπων Δ'	94
A'.15	Μοντέλο για την περιγραφή των ατμοσφαιρικών μοντέλων	95
A'.16	Υπότυποι ατμοσφαιρικών μοντέλων A'	98
A'.17	Υπότυποι ατμοσφαιρικών μοντέλων B'	99
A'.18	Υπότυποι ατμοσφαιρικών μοντέλων Γ'	100
A'.19	Υπότυποι ατμοσφαιρικών μοντέλων Δ'	101
A'.20	Υπότυποι ατμοσφαιρικών μοντέλων E'	102
A'.21	Υπότυποι ατμοσφαιρικών μοντέλων ΣΤ'	103
Γ.1	Η κλάση <i>CIcarosData</i>	107
Γ.2	Η κλάση <i>CIcarosDataSource</i>	110
Γ.3	Η κλάση <i>CMonitoringStation</i>	113

Γ.4	Η κλάση CCarosHelper	116
Γ.5	Η κλάση CMeasuredQuantities	120
Γ.6	Η κλάση CModelGrid	121
Γ.7	Η κλάση CModelResult	124
Γ.8	Η κλάση CModelResults	129
Γ.9	Η κλάση Cmeasurement	130
Γ.10	Συνοπτικό μοντέλο Visual Basic - UML	132

Γλωσσάρι

AOT Aerosol Optical Thickness

ΣΓΠ Σύστημα Γεωγραφικών Πληροφοριών

BLOB Binary Large Object

BNF Backus Naur Form

CASE Computer Aided Software Engineering

COM Common Object Model

DOM Document Object Model

DTD Document Type Definition

EER Enhanced Entity Relationship model

ESRI Environmental Systems Research Institute

GML Geographic Markup Language

GMT Global Mean Time

GOLAP Geographical Online Analytical Processing

GPS Global Positioning System

HTML Hypertext Markup Language

Icaros.NET Integrated Computational Assessment of Air Quality via Remote Observations Systems Network

MDD Multidimensional Discrete DATA

OGC Open GIS Consortium

OID Object Identifier

PVL Plug-in for Visual Languages

RAD Rapid Application Development

RTF Revision Task Force

SAX Simple API for XML

ΣΔΒΔ Σύστημα Διαχείρισης Βάσης Δεδομένων

SFS Simple Feature Specification

SGML Structured Generalized Markup Language

SVG Scalable Vector Graphics

UML Unified Modeling Language

USGS United States Geological Survey

WMS Web Map Server

XML eXtensible Markup Language

Περίληψη

Η περιγραφή αποθήκευση και μεταφορά δεδομένων, τα οποία εκτός των άλλων εμπεριέχουν και γεωγραφική πληροφορία, αποτελεί αναπόσπαστο κομμάτι κάθε συστήματος γεωγραφικών πληροφοριών(ΣΓΠ). Είτε πρόκειται για ένα γενικευμένο ΣΓΠ, είτε για μία εξειδικευμένη γεωγραφική εφαρμογή υπάρχει πάντα η απαίτηση της ορθής περιγραφής των δεδομένων, της δομημένης αποθήκευσής και της μεταφοράς τους με αποτελεσματικό τρόπο, όσον αφορά στην ακρίβεια αλλά και στην ταχύτητα. Τα παραπάνω αποκτούν ιδιαίτερη σημασία όταν έχουμε να κάνουμε με την υλοποίηση ενός συστήματος το οποίο απαιτεί αλληλεπίδραση με ένα μεγάλο αριθμό ανεξάρτητων μεταξύ τους πηγών πληροφορίας διαφορετικής δομής. Στην παρούσα εργασία εξετάζεται η περίπτωση του συστήματος *Icaros.NET* το οποίο αποσκοπεί στην καταγραφή, την μελέτη και την οπτικοποίηση της αέριας ρύπανσης αστικών περιοχών. Η λειτουργία του συστήματος υποστηρίζεται από μία βάση γεωγραφικών δεδομένων, η οποία τροφοδοτείται από μία πληθώρα εξωτερικών, γεωγραφικών και μη δεδομένων. Σκοπός είναι η μοντελοποίηση της γεωγραφικής βάσης δεδομένων με τεχνικές αντικειμενοστραφούς σχεδίασης (Unified Modeling Language-UML) και η δημιουργία ενός μοντέλου περιγραφής και ανταλλαγής δεδομένων με την χρήση της γλώσσας Extensible Markup Language-XML.

Λέξεις Κλειδιά: Βάσεις Γεωγραφικών Δεδομένων, XML, UML, *Icaros.NET*, Επεξεργασία δεδομένων, Αέρια Ρύπανση.

Abstract

The description and the exchange of data, which consists, among others, of geographic information is a basic module of every Geographic Information System (GIS). Whether dealing with a general purpose GIS, or a specialized GIS application, there is always the need to correctly describe the data, to store them in a structured fashion and to exchange them in an effective way, as far as the accuracy and the efficiency are concerned. These are aspects of main interest, especially when dealing with the design and the implementation of a system, which interacts with a lot of independent information sources of different structure. In this work the case of *Icaros.NET* platform is described, which aims to the monitoring and visualization of the air pollution in urban areas. *Icaros.NET* bases its operation on a geographic database which stores a great volume of external geographic and thematic data. The main objective is the geographic database modeling using Object Oriented Design methods, namely the Unified Modeling Language-UML and the creation of a XML Schema definition and the description and exchange of information, making use of the Extensible Markup Language-XML.

Key Words: Geographic Databases, XML, UML, *Icaros.NET*, Data processing, Air pollution monitoring

Κεφάλαιο 1

Εισαγωγή

Η παρούσα διπλωματική εργασία εξετάζει κάποιες από τις δυνατότητες που παρέχονται από τις σύγχρονες τεχνολογίες της Πληροφορικής για τη διαχείριση και την ανάλυση χωροχρονικών δεδομένων. Αφειτηρία της αναζήτησης αυτής αποτέλεσε η ανάγκη για την ανάπτυξη ενός Συστήματος Γεωγραφικών Πληροφοριών (ΣΓΠ) ικανού για την αποθήκευση και επεξεργασία δεδομένων σχετικών με την αέρια ρύπανση: του συστήματος *Icaros.NET*¹. Τέτοια συστήματα πρέπει να είναι ικανά να διαχειριστούν μεγάλη ποικιλία δεδομένων σε δομή και ποιότητα αλλά και μεγάλη ποσότητα δεδομένων, τα οποία μάλιστα έχουν άρρηκτα συνδεδεμένη μαζί τους εκτός από την αριθμητική τιμή τους, μία χωρική και μία χρονική συνιστώσα.

1.1 Σκοπός

Το σύστημα *Icaros.NET* έχει ως αντικείμενο την καταγραφή, τη μελέτη και την οπτικοποίηση της αέριας ρύπανσης, συνδυάζοντας μία πληθώρα πρωτογενών και όχι μόνο δεδομένων. Συγκεντρώνει πληροφορίες και δεδομένα από επίγειους σταθμούς μέτρησης, δορυφορικές εικόνες και μοντέλα προσομοίωσης της συμπεριφοράς της ατμόσφαιρας τα οποία δίνουν αποτελέσματα σε ένα πλέγμα προκαθορισμένων σημείων. Απαιτείται η αποδοτική αποθήκευ-

¹Αναπτύσσεται στα πλαίσια του Ευρωπαϊκού προγράμματος *Icaros.NET* (IST-2000-29264) στο Εργαστήριο Συστημάτων Γεωγραφικών Πληροφοριών του Πανεπιστημίου Αιγαίου.

σή τους σε ένα Σύστημα Διαχείρισης Βάσης Δεδομένων (ΣΔΒΔ), ώστε να είναι εύκολη η πρόσβαση στην συλλεγόμενη πληροφορία και η περαιτέρω επεξεργασία της, για τη διεξαγωγή υπολογισμών και την εξαγωγή συμπερασμάτων, που σχετίζονται με την αέρια ρύπανση μίας συγκεκριμένης περιοχής. Από τη φύση της εφαρμογής, απαιτείται η επεξεργασία δεδομένων, τα οποία προϋποθέτουν ένα πλήθος συνοδευτικής πληροφορίας, όπως είναι για παράδειγμα οι μονάδες μέτρησης των διαφόρων μεγεθών. Επίσης σε αρκετές περιπτώσεις απλές μετρήσεις συνοδεύονται απαραίτητα από γεωγραφική πληροφορία, όπως είναι για παράδειγμα οι συντεταγμένες ενός σταθμού μέτρησης.

Τα προηγούμενα είναι ήδη αρκετά σημαντικά ζητήματα αφού η αντιμετώπισή τους επηρεάζει άμεσα τις σχεδιαστικές αποφάσεις που αφορούν στην αρχιτεκτονική της πλατφόρμας. Υπάρχουν δύο ακόμα σημεία στα οποία πρέπει να δοθεί ιδιαίτερη προσοχή:

- Το μεγάλο πλήθος δεδομένων και η αυξημένη πιθανότητα λάθους κατά την εισαγωγή αυτών στη βάση δεδομένων:
Η λειτουργία της πλατφόρμας σε πραγματικό χρόνο και η απαίτηση για την αυτοματοποιημένη εισαγωγή δεδομένων εγκυμονεί μεγάλους κινδύνους όσον αφορά στην αξιοπιστία των εξαγόμενων αποτελεσμάτων. Εκτός από την εσωτερική λειτουργία της η οποία μπορεί να θεωρηθεί ελεγχόμενη, η εξάρτηση της από ένα μεγάλο πλήθος εξωτερικών δεδομένων μπορεί να αποτελέσει ισχυρό παράγοντα εισαγωγής σφαλμάτων. Απαιτείται έτσι η υιοθέτηση ενός κοινού τρόπου περιγραφής των δεδομένων, τα οποία περιέχονται στα αρχεία που πρόκειται να εισαχθούν στην βάση δεδομένων της πλατφόρμας, ο οποίος να είναι ακριβής αλλά και ευέλικτος.
- Η λειτουργία της πλατφόρμας σε διαφορετικές γεωγραφικές περιοχές:
Η πρόθεση για την λειτουργία του συστήματος σε διαφορετικές γεωγραφικές περιοχές, οι οποίες ανήκουν σε διαφορετικές χώρες, εισάγει μία ακόμα παράμετρο η οποία είναι δυνατόν στην πορεία να επηρεάσει την σωστή λειτουργία της εφαρμογής. Έχουμε να αντιμετωπίσουμε εκτός από τα διαφορετικά προβολικά συστήματα συντεταγμένων που χαρακτηρίζουν κάθε περιοχή, καθώς και τη διαφορά ώρας που αφορά

στη χρονική στιγμή των μετρήσεων ή τη λήψη των δορυφορικών εικόνων αφού, η μέτρηση του χρόνου εκφράζεται άλλοτε σε τοπική ώρα και άλλοτε σε παγκόσμια ώρα (GMT). Επιπλέον κάθε γεωγραφική περιοχή είναι πιθανό να καλύπτεται από διαφορετικής τεχνολογίας δίκτυα μετρήσεων και διαφορετικά μοντέλα προβλέψεων, γεγονός που σημαίνει ότι για κάθε περιοχή η ίδια πληροφορία πιθανώς να βρίσκεται αποθηκευμένη με διαφορετική μορφή. Ενώ αυτό δεν διαφοροποιεί καθόλου την λειτουργία της πλατφόρμας από την στιγμή που τα δεδομένα θα εισαχθούν στην γεωγραφική βάση δεδομένων, δημιουργεί μία ιδιόμορφη κατάσταση μέχρι τη στιγμή που τα δεδομένα θα φτάσουν εκεί.

Στόχο της διπλωματικής εργασίας αποτελεί η σχεδίαση του υποσυστήματος διαχείρισης δεδομένων του συστήματος *Icaros.NET* σύμφωνα με της απαιτήσεις της τελικής εφαρμογής. Όπως μπορεί να υποψιαστεί κανείς το υποσύστημα αυτό πρόκειται να δεχτεί το μεγαλύτερο μέρος της πληροφορίας που θα επεξεργάζεται το σύστημα και ο σωστός σχεδιασμός της αποτελεί κατά ένα μεγάλο ποσοστό κριτήριο επιτυχίας για το σύνολο του συστήματος.

1.2 Η βάση δεδομένων

Αρχικά έγινε καταγραφή των δεδομένων που απαιτούνται για τη λειτουργία του συστήματος, τα οποία συνοψίζονται σε αποτελέσματα μετρήσεων από επίγειους σταθμούς μέτρησης, αποτελέσματα μοντέλων της ατμόσφαιρας τα οποία δίνονται υπό μορφή πλέγματος σημείων και δορυφορικές εικόνες. Από αυτά, οι δορυφορικές εικόνες δεν θα αποθηκευτούν στην γεωγραφική βάση δεδομένων λόγω της ιδιομορφίας και του όγκου τους. Για τα υπόλοιπα δεδομένα πρέπει να υπάρχει η δυνατότητα της αποθήκευσης κατ' αρχάς της γεωγραφικής πληροφορίας που σχετίζεται με τις θέσεις των σταθμών και των σημείων του πλέγματος των αποτελεσμάτων των μοντέλων. Παράλληλα πρέπει να υποστηρίζεται η περιγραφή σχέσεων τύπου *Είναι(Is-A)* και *Έχει(Has-A)* μεταξύ των αντικειμένων που πρόκειται να αποθηκευτούν στην γεωβάση. Για παράδειγμα ένας «σταθμός μέτρησης» *Έχει* πραγματοποιήσει μία «μέτρηση» η οποία *Είναι* «μετρούμενη ποσότητα NO_x ».

1.3 Αντικειμενοστραφής σχεδιασμός της βάσης δεδομένων με χρήση της UML

Ο σχεδιασμός της βάσης δεδομένων έγινε με την βοήθεια του Visio και με βάση τα αντικείμενα που παρέχονται από την ESRI. Η χρήση της UML μας επέτρεψε να περιγράψουμε τα αντικείμενα εκείνα που σχετίζονται άμεσα με την υλοποίηση της πλατφόρμας *Icaros.NET* και τις μεταξύ τους συσχετίσεις. Επίσης μας δόθηκε η δυνατότητα στα πρώτα στάδια της υλοποίησης της πλατφόρμας για την εύκολη επέκταση και αλλαγή, όποτε αυτό ήταν απαραίτητο, του σχήματος της βάσης. Χαρακτηριστική είναι και η ευκολία με την οποία μπορούμε να επεκτείνουμε την εφαρμογή προσθέτοντάς της την υποστήριξη νέων τύπων, για παράδειγμα, ρύπανσης μέσω της κληρονομικότητας που προσφέρουν τα συστήματα τα οποία βασίζονται σε μία αντικειμενοστραφή αρχιτεκτονική. Έχουμε έτσι, ένα ευέλικτο και επεκτάσιμο σχήμα για την βάση δεδομένων της εφαρμογής μας.

1.4 Περιγραφή των δεδομένων με χρήση της XML

Εκτός από τη δομή της βάσης δεδομένων, όπως προαναφέραμε, ένα εξίσου σημαντικό κομμάτι είναι και η μέθοδος που ακολουθείται για την εισαγωγή των στοιχείων σε αυτή. Ενώ αντιμετωπίζουμε μία ποικιλομορφία δεδομένων πριν αυτά εισαχθούν στην πλατφόρμα, εσωτερικά η πλατφόρμα ουσιαστικά αναγνωρίζει από τη γεωγραφική βάση δεδομένων μόνο μετρήσεις από επίγειους σταθμούς και αποτελέσματα από ατμοσφαιρικά μοντέλα. Έτσι για παράδειγμα, μπορεί να έχουμε πέντε επίγειους σταθμούς μέτρησης από τους οποίους κάθε ένας μετράει διαφορετικές ομάδες ρύπων. Επίσης έχουμε αποτελέσματα από τέσσερα διαφορετικά μοντέλα. Κάθε ένα από τα προηγούμενα, μπορεί να δίνει αποτελέσματα σε διαφορετική μορφή, όσον αφορά στη δομή του αρχείου που παράγεται, είτε από το σταθμό μέτρησης, είτε από το ατμοσφαιρικό μοντέλο. Είναι σημαντικό εξάλλου να αναφερθεί ότι συνήθως τα αρχεία των μετρήσεων είναι απλουστευμένα υπό την έννοια ότι δεν συμπεριλαμβάνουν στοιχεία σχετικά με τις μονάδες των μετρήσεων που τα αφορούν

ή άλλες τέτοιες πληροφορίες, οι οποίες παρέχονται συγκεντρωτικά υπό την μορφή συνοδευτικών εντύπων.

Η ενσωμάτωση στην πλατφόρμα διαδικασιών για τη διαχείριση όλων αυτών των περιπτώσεων των αρχείων προκειμένου να εισαχθούν τα απαραίτητα δεδομένα στην βάση δεδομένων, είναι ζωτικής σημασίας, ενώ υπάρχει ο κίνδυνος να γίνει αυτοσκοπός της εφαρμογής. Εκτός των άλλων εισάγει επιπλέον πολυπλοκότητα στην εφαρμογή και επιδρά αρνητικά στο σχεδιασμό και την αρχιτεκτονική της. Για να δοθεί λύση στο πρόβλημα αυτό έγινε η χρήση της XML για την περιγραφή του συνόλου των δεδομένων που μπορεί να δεχθεί η πλατφόρμα και αφορά τους επίγειους σταθμούς μέτρησης, τα αποτελέσματα των ατμοσφαιρικών μοντέλων και όλων των άλλων στοιχείων που σχετίζονται με αυτά. Σχεδιάστηκε έτσι ένα κατάλληλο σχήμα για την περιγραφή των δεδομένων σε XML (XML-Schema) το οποίο ακολουθεί το σχήμα της βάσης δεδομένων της πλατφόρμας. Με αυτό πετυχαίνουμε την εισαγωγή δεδομένων στην πλατφόρμα μόνο στην περίπτωση που αυτά είναι στην μορφή που υπαγορεύει το XML-Schema. Παρέχεται έτσι μία μεγάλη ευελιξία αφού μπορούμε να καθορίσουμε τον τύπο των δεδομένων που απαιτούνται, ποιά δεδομένα είναι απαραίτητα και ποιά προαιρετικά, ακόμα και αλληλοσυσχετίσεις των αντικειμένων μεταξύ τους και να περιλάβουμε όλη αυτή την πληροφορία σε ένα απλό αλλά δομημένο αρχείο κειμένου. Η πλατφόρμα το μόνο που καλείται τώρα να κάνει είναι να επιβεβαιώσει την ορθότητα του προς εισαγωγή XML αρχείου και να το εισάγει στη βάση δεδομένων. Απλοποιείται έτσι σημαντικά η δομή και η λειτουργία της στο χαμηλό επίπεδο της λήψης δεδομένων.

Στην ιδανική περίπτωση όπου οι εξωτερικές πηγές δεδομένων μπορούν να παρέχουν τη μορφή των αρχείων που εμείς επιθυμούμε, δεν μένει παρά να απαιτήσουμε τη μορφή που υπαγορεύει το XML-Schema. Σε αντίθετη περίπτωση, που είναι και η συχνότερη, πρέπει να γίνει η μετατροπή των αρχείων που παρέχονται από τις εξωτερικές πηγές στην προκαθορισμένη από το XML-Schema μορφή. Αυτό όμως είναι μία σχετικά απλή διαδικασία γιατί μπορεί να υλοποιηθεί μέσα από μία ανεξάρτητη εφαρμογή-φίλτρο, που θα κατασκευαστεί για το λόγο αυτό, χωρίς να επηρεαστεί η δομή και η λειτουργία της πλατφόρμας. Στην πραγματικότητα, η χρήση της XML μας επέτρεψε να ανεξαρτητοποιήσουμε την πλατφόρμα από την μορφή των εξωτερικών πηγών

δεδομένων που απαιτούνται.

1.5 Υλοποίηση

Η ανάπτυξη του συστήματος *Icaros.NET* βασίζεται στην αρχιτεκτονική που προτείνει η ESRI™ με το ArcGIS 8.1 και τη βιβλιοθήκη ArcObjects. Η τελευταία βασίζεται στην αρχιτεκτονική COM της Microsoft™. Παρά τους περιορισμούς που μπορεί να μας επιβάλει η υποταγή σε μία συγκεκριμένη αρχιτεκτονική, στην περίπτωση μας μάλλον τα πλεονεκτήματα ήταν περισσότερα. Το ΣΔΒΔ που αξιοποιήθηκε για τις ανάγκες της υλοποίησης είναι το ESRI - PersonalGeodatabase, ενώ για το προγραμματισμό χρησιμοποιήθηκε η Visual Basic. Η τελευταία, ως εργαλείο RAD μας επιτρέπει να μείνουμε συγκεντρωμένοι στο τελικό στόχο μας χωρίς να μπερδευτούμε με τις ιδιομορφίες γλωσσών προγραμματισμού όπως η C++. Ακόμα, το γενικής χρήσης περιβάλλον εργασίας του ArcGIS παρέχει αρκετά έτοιμα εργαλεία, η αξιοποίηση των οποίων μπορεί να επιταχύνει αρκετά τη διαδικασία ανάπτυξης. Για την υλοποίηση του XML τμήματος της εφαρμογής αξιοποιείται ο MSXML4.0 της Microsoft.

1.6 Παρουσίαση

Μετά από τη σύντομη εισαγωγή σχετικά με τους στόχους της παρούσας διπλωματικής εργασίας, στο κεφάλαιο 2 παρουσιάζονται κάποια στοιχεία για τα ΣΓΠ καθώς και τα δεδομένα που διαχειριζόμαστε με αυτά. Στο κεφάλαιο 3 γίνεται μία περιγραφή για τα συστήματα διαχείρισης βάσεων δεδομένων από τα κλασικά, μέχρι και τα πιο σύνθετα χωροχρονικά. Στο κεφάλαιο 4 παρουσιάζεται το σύστημα *Icaros.NET* και οι απαιτήσεις που αφορούν στο επίπεδο διαχείρισης δεδομένων. Στα κεφάλαια 5 και 6 παρουσιάζονται οι τεχνολογίες UML και XML και οι εφαρμογές που μπορεί να έχουν στα ΣΓΠ, τόσο γενικά αλλά και συγκεκριμένα για την περίπτωση μας. Στο κεφάλαιο 7 παρουσιάζεται η υλοποίηση για τη διαχείριση των δεδομένων του συστήματος, ενώ τα συμπεράσματά από την συνολική προσέγγιση παρουσιάζονται στο κεφάλαιο

8. Τέλος στο παράρτημα Α' περιγράφεται το σχήμα της γεωγραφικής βάσης δεδομένων, στο παράρτημα Β' περιγράφεται το σχήμα ορισμού των αρχείων XML, στο παράρτημα Γ' περιγράφεται το στατικό διάγραμμα κλάσεων που χρησιμοποιήθηκε για την υλοποίηση της βιβλιοθήκης IcarosXML.dll ενώ στο παράρτημα Δ' περιγράφονται τα περιεχόμενα του συνοδευτικού οπτικού δίσκου της εργασίας.

Κεφάλαιο 2

Συστήματα Γεωγραφικών Πληροφοριών

2.1 Γενικά

Τα ΣΓΠ, αποτελούν μία ειδικότερη κατηγορία των πληροφοριακών συστημάτων με έμφαση στη διαχείριση γεωγραφικής πληροφορίας. Μπορεί να πει κανείς ότι είναι η φυσική επέκταση ενός πληροφοριακού συστήματος οδηγούμενη από τη ραγδαία τεχνολογική εξέλιξη που συντελείται στις μέρες μας. Σύγχρονα λειτουργικά συστήματα, βάσεις δεδομένων, και μεθοδολογίες ανάπτυξης εφαρμογών σε συνδυασμό με την τάση για γραφική παρουσίαση και επικοινωνία ανθρώπου-υπολογιστή σε ένα περιβάλλον δικτυακής επικοινωνίας, μας δίνουν απεριόριστες δυνατότες ([14],[16], [3]).

Από μία άλλη οπτική γωνία, είναι η φυσική εξέλιξη των στατικών τυπωμένων χαρτών σε μία αναπόφευκτη συνάντηση με την τεχνολογία. Σκοπός δεν είναι πλέον, για παράδειγμα, η κατασκευή ενός ναυτικού χάρτη ακριβείας, διαδικασία που παλιότερα ήταν επίπονη και χρονοβόρα. Αντίθετα στόχος είναι η συλλογή και οργάνωση δεδομένων σε ένα πληροφοριακό σύστημα με τη δυνατότητα εκτός των άλλων, για τη δημιουργία χαρτών σε εκτυπώσιμη μορφή.

Ποτέ άλλοτε δεν υπήρχαν δεδομένα με τόση μεγάλη διαθεσιμότητα και μάλιστα με τόσο μικρό κόστος. Η δορυφορική τηλεπισκόπηση μπορεί μας

παρέχει μαζικά εικόνες οποιουδήποτε σημείου του πλανήτη. Πλήθος οργανισμών συλλέγουν και διαθέτουν δεδομένα, τα οποία μπορεί να είναι χρήσιμα σε πλήθος εφαρμογών. Είναι λοιπόν μία πρόκληση η αξιοποίηση ακόμα κι ενός μικρού μέρους της διαθέσιμης αυτής πληροφορίας. Από εκεί πηγάζει και η ανάγκη για την ανάπτυξη εργαλείων με σκοπό τη βοήθεια στην κατανόηση των γεωγραφικών διαδικασιών, τόσο στο πεδίο του χώρου όσο και του χρόνου, αλλά και την αποδοτική αξιοποίηση του μεγάλου πλήθους των διαθέσιμων χωροχρονικών δεδομένων [74]. Την απόσταση από τη συλλογή και την οργάνωση μεγάλου όγκου δεδομένων, μέχρι την παραγωγή χαρτών έρχονται να καλύψουν τα ΣΓΠ, όχι όμως αποκλειστικά σαν εργαλεία παραγωγής χαρτών αλλά σαν ολοκληρωμένα εργαλεία ανάλυσης δεδομένων και λήψης αποφάσεων.

2.2 Ιστορική Αναδρομή

Ένας από τους πρώτους χάρτες ανακαλύφθηκε κοντά στην περιοχή του ποταμού Δνειπέρου σκαλισμένος σε κόκαλο από μαμούθ πριν από 15000 χρόνια. Η πρώτη απόπειρα απεικόνισης του γεωγραφικού χώρου σε χάρτη ηλεκτρονικού υπολογιστή έγινε στα τέλη της δεκαετίας του 1950 από τον Waldo Tobler. Το πρώτο σύστημα GIS DIME χρησιμοποιήθηκε το 1970 στην απογραφή των Ηνωμένων Πολιτειών για την παροχή αυτοματοποιημένης κωδικοποίησης διευθύνσεων. Τέλος κατά την διάρκεια της δεκαετίας του 1970 κυριάρχησε το τοπολογικό διανυσματικό μοντέλο για χωρικά δεδομένα [74].

Επιγραμματικά τρία ήταν τα σημαντικότερα στάδια της εξέλιξης των ΣΓΠ[46]:

- Το πρώτο επιτυχημένο σύστημα γεωγραφικών πληροφοριών CGIS – Canadian Geographic Information System, κατάφερε να λειτουργήσει αποδοτικά την εποχή των διάτρητων καρτών και των mainframes
- Κατά τη δεκαετία του 1970 η επανάσταση στα συστήματα γεωγραφικών πληροφοριών ήρθε με το ARC/INFO, το οποίο βασίστηκε σε μία ψευδοσχεσιακή βάση δεδομένων (INFO) η οποία εμπλουτίστηκε με επεκτάσεις για την τοπολογική διαχείριση των δεδομένων (ARC).

- Η σύγχρονη πραγματικότητα που συνίσταται σε μεγάλες επενδύσεις στην παρούσα τεχνολογία και στις προοπτικές που ανοίγονται με τις σύγχρονες βάσεις δεδομένων και τις αντικειμενοστραφείς μεθοδολογίες.

Η εξέλιξη των ΣΓΠ συμβαδίζει με τις εξελίξεις στους υπόλοιπους κλάδους της πληροφορικής. Ωστόσο ιδιαίτερη προσοχή στην ανάπτυξή τους ως αυτόνομα συστήματα έχει δοθεί κυρίως από την προηγούμενη δεκαετία. Ήδη από τις αρχές της δεκαετίας του 1990 εξετάζονται οι δυνατότητες ανασχεδιασμού των γεωγραφικών συστημάτων πληροφοριών με σκοπό τη δυνατότητα μοντελοποίησης του πραγματικού κόσμου με πιο άμεσο τρόπο[75]. Έμφαση δίνεται στην ανάγκη για μία αντικειμενοστραφή προσέγγιση, με την οποία μπορούν να περιγραφούν με ένα συστηματικό τρόπο οντότητες του πραγματικού κόσμου όπως αυτές που καλείται να διαχειριστεί ένα σύστημα γεωγραφικών πληροφοριών.

2.3 Λειτουργία

Ένα γενικής χρήσης γεωγραφικό σύστημα πληροφοριών αποτελείται από τέσσερα κύρια υποσυστήματα ([46],[88],[109]) για την:

- εισαγωγή δεδομένων από υπάρχοντες χάρτες, τηλεπισκόπηση, δορυφορικές εικόνες κ.λ.π.
- αποθήκευση και ανάκτηση χωρικών δεδομένων σε μία μορφή η οποία υποστηρίζει γρήγορη ανάκτηση και ταχύτατες ενημερώσεις
- συστήματα διαχείρισης δεδομένων για την επεξεργασία των χαρακτηριστικών των οντοτήτων με γεωγραφική πληροφορία
- παρουσίαση χαρτών και αναφορές από τα αποθηκευμένα δεδομένα

Κάθε ένα από τα υποσυστήματα αυτά είναι εξίσου σημαντικό και μπορεί να αντιστοιχηθεί σε αντίστοιχους κλάδους της πληροφορικής. Το υποσύστημα που σχετίζεται με την εισαγωγή δεδομένων έχει να κάνει με τεχνολογίες ανταλλαγής και περιγραφής δεδομένων. Η περιγραφή των δεδομένων με ένα μηχανιστικό τρόπο και η συμπερίληψη μαζί μετα-δεδομένων (meta-data), με

δυνατότητες που μας δίνει για παράδειγμα η XML, παρέχουν ένα νέο τρόπο αντίληψης του τρόπου που ένα σύστημα αλληλεπιδρά με τις εξωτερικές πηγές πληροφορίας.

Η αποθήκευση και ανάκτηση δεδομένων σχετίζεται άμεσα με τις εξελίξεις στο πεδίο των συστημάτων διαχείρισης βάσεων δεδομένων ([29], [104]). Ενώ παλιότερα ένα σύστημα γεωγραφικών πληροφοριών μπορεί να βασιζόταν σε απλά αρχεία για την αποθήκευση των δεδομένων του, σήμερα ένα σύστημα βάσης δεδομένων το οποίο μάλιστα να υποστηρίζει επεκτάσεις για την διαχείριση γεωγραφικών δεδομένων, θεωρείται αναπόσπαστο κομμάτι.

Κάθε ΣΓΠ πρέπει να παρέχει εργαλεία για την επεξεργασία και ανάλυση της πληροφορίας που διαχειριζόμαστε. Έτσι μπορεί να βοηθήσει στην παραγωγή νέας γνώσης ή στην ανάδειξη της ήδη υπάρχουσας ([102], [65],[87]). Μέθοδοι χωρικής στατιστικής, εξόρυξης δεδομένων ή ακόμα και τρισδιάστατης μοντελοποίησης μπορούν να εφαρμοστούν σε γεωγραφικά δεδομένα. Επιπλέον αν υποθέσουμε ότι η εισαγωγή των δεδομένων γίνεται δυναμικά σε πραγματικό χρόνο, για παράδειγμα με δεδομένα τα οποία προέρχονται από κάποιο δορυφόρο, μπορούμε να έχουμε και κάποιας μορφής αυτοματοποιημένης επεξεργασίας.

Το τελευταίο υποσύστημα που αφορά στην παρουσίαση χαρτών και αποτελεσμάτων είναι ιδιαίτερα σημαντικό, αφού σχετίζεται άμεσα με το τρόπο που ο χρήστης ενός ΣΓΠ αλληλεπιδρά με την χωρική πληροφορία [20]. Όσο πιο εύκολη είναι αυτή η αλληλεπίδραση, τόσο περισσότερο χρήσιμο μπορεί να είναι ένα τέτοιο σύστημα. Οι δυνατότητες που παρέχονται μέσα από τις γραφικές γλώσσες ερωτήσεων για βάσεις δεδομένων καθώς και τα σύγχρονα περιβάλλοντα ανάπτυξης λογισμικού, συμβάλλουν δυναμικά προς τη βελτίωση κι αυτής της λειτουργίας.

2.4 Επεκτάσεις

Ιδιαίτερη βαρύτητα έχει δοθεί τελευταία στη λειτουργία των ΣΓΠ σε σχέση με το διαδίκτυο. Εδώ πλέον το μέσο αλληλεπίδρασης του χρήστη με το σύστημα, είναι δυναμικές δικτυακές ιστοσελίδες. Παρέχεται έτσι πολύ μεγάλη ευελιξία και νέα πεδία εφαρμογών.

Για παράδειγμα στο [105], παρουσιάζονται οι Υπερ-Χάρτες (Hypermaps) σε αντιστοιχία με το Υπερ-Κείμενο (Hypertext), σε μία εφαρμογή που σχετίζεται με την παρουσίαση γεωλογικών φαινομένων. Συνοδεύεται από μία περιγραφή για την ανάπτυξη ενός συστήματος παραγωγής γεωλογικών χαρτών. Αποτελεί μία προσέγγιση η οποία επεκτείνει τον κλασικό χάρτη και το ψηφιακό χάρτη που έχει προκύψει από ένα γεωγραφικό σύστημα πληροφοριών σε μία μορφή δυναμικού χάρτη με τη δυνατότητα να παρουσιάζει δεδομένα ανάλογα με τις ανάγκες του χρήστη.

Επίσης στο [70] εξετάζονται αρχιτεκτονικά ζητήματα διακομιστών κινούμενων αντικειμένων (MOS - Moving Object Servers) οι οποίοι παρέχουν την δυνατότητα της αντιστοίχισης δεδομένων από GPS με τις τροχιές κινούμενων αντικειμένων και την έγερση ειδοποιήσεων σχετικών με διάφορα γεγονότα στους αποδέκτες της υπηρεσίας, μέσω διαδικτύου. Η υλοποίηση γίνεται με χρήση τεχνολογιών βασισμένων στην XML, όπως η SVG. Περιγράφεται επίσης και η έννοια των διαρκών ερωτήσεων (Continuous Queries). Πρόκειται για ερωτήσεις που μπορεί να απευθυνθούν σε ένα ΣΓΠ και να παράγουν δυναμικά απαντήσεις σε ένα μεγάλο χρονικό εύρος.

Στο [93] γίνεται μία αναφορά στα διαθέσιμα λογισμικά για την έκδοση θεματικών χαρτών στο διαδίκτυο, ενώ παρουσιάζεται και το Interactice Map Applet, μία εφαρμογή Java για το σκοπό αυτό. Αντίστοιχα στο [92], παρουσιάζονται οι δυνατότητες ανάπτυξης ενός αντικειμενοστραφούς περιβάλλοντος στην γλώσσα PERL, για τη διαχείριση χωρικών δεδομένων, με σχεδίαση βασισμένη στη UML. Επίσης στο [106] γίνεται συζήτηση για την ανάπτυξη ενός κατανεμημένου συστήματος γεωγραφικών πληροφοριών βασισμένου στην αρχιτεκτονική CORBA.

Στο [63] γίνεται προσπάθεια για την εξαγωγή γεωγραφικών χαρακτηριστικών από διαδικτυακές διευθύνσεις και η ανάπτυξη ενός εργαλείου για την αναζήτηση στο διαδίκτυο με κριτήρια γεωγραφικής γειννίας. Η δυνατότητα αυτή μπορεί είναι ιδιαίτερα χρήσιμη σε «έξυπνες» εφαρμογές ηλεκτρονικού εμπορίου.

Τέλος στο [89] περιγράφεται η εφαρμογή τεχνολογιών WMS και μίας GML συμβατής εφαρμογής παρουσίασης για την κατασκευή ενός δικτυακού συστήματος χαρτογραφίας. Οι τεχνολογίες αυτές αποτελούν ένα υπό εξέλιξη

ΚΕΦΑΛΑΙΟ 2. ΣΥΣΤΗΜΑΤΑ ΓΕΩΓΡΑΦΙΚΩΝ ΠΛΗΡΟΦΟΡΙΩΝ

πρότυπο του OGC, για την ανταλλαγή γεωγραφικών πληροφοριών και την ανάπτυξη ΣΓΠ στο δίκτυο.

Κεφάλαιο 3

Χωροχρονικές Βάσεις Δεδομένων

Σε επέκταση ενός παραδοσιακού ΣΔΒΔ, μία χωροχρονική βάση δεδομένων περιλαμβάνει χωρικές, χρονικές αλλά και χωροχρονικές οντότητες και απεικονίζει ταυτόχρονα τόσο χωρικές, όσο και χρονικές όψεις των δεδομένων [69]. Η ζωή και η κίνηση ανεξάρτητων οντοτήτων φυσικών ή νοητών κυριαρχεί στην ύπαρξή μας και στην αντίληψη που έχουμε για το περιβάλλον μας [81]. Εξάλλου το ογδόντα τοις εκατό του συνόλου των ανθρωπίνων αποφάσεων εμπεριέχουν κάποιο χωρικό προσδιορισμό, ενώ οι περισσότερες εφαρμογές βάσεων δεδομένων περιλαμβάνουν την έννοια του χρόνου [56]. Η ανάγκη για τη μελέτη και την κατανόηση των οντοτήτων αυτών που μας περιβάλλουν και επηρεάζουν τις αποφάσεις μας, οι οποίες καταλαμβάνουν μία θέση στο χώρο και χαρακτηρίζονται γενικότερα από μία μεταβολή στο χρόνο, με τη βοήθεια των μεθόδων που παρέχει η πληροφορική, οδήγησαν στην επέκταση αυτή.

3.1 Αρχιτεκτονική βάσεων δεδομένων

Χρειάστηκαν περίπου είκοσι χρόνια από το πρώτο GIS για την αναπαράσταση και την ανάλυση ταυτόχρονα της χρονικής μαζί με την χωρική διάσταση [74]. Αυτό μπορεί να οφείλεται από την μία στο πρότυπο των κλασικών στατικών χαρτών κι από την άλλη στην έμφαση που δίνεται συνήθως για την ανάπτυξη εφαρμογών με στόχο την επίλυση ενός συγκεκριμένου προβλήματος σε μικρό χρονικό διάστημα. Οι πρώτες προσεγγίσεις είχαν κατεύθυνση από την πράξη

στη θεωρία και δεν ξεκινούσαν από ένα δομημένο θεωρητικό πλαίσιο. Αυτό πηγάζει και από το γεγονός ότι αρχικά η τεχνολογία των υπολογιστών όσο και οι τεχνικές μοντελοποίησης των δεδομένων, ήταν ακόμα ανώριμες. Ακόμα, η χαμηλή υπολογιστική δύναμη και οι περιορισμένοι οικονομικοί πόροι ωθούσαν τους χρηματοδότες οργανισμούς για εφαρμογές που κάλυπταν μόνο τις τρέχουσες απαιτήσεις τους.

Τα πρώτα συστήματα διαχείρισης χωρικών δεδομένων ήταν μονολιθικά υπό την έννοια ότι μπορούσαν να διαχειριστούν μόνο διανυσματικά ή ψηφιδωτά δεδομένα (Vector/Raster) [74]. Για παράδειγμα αρχικά η ESRI διέθετε τα PIOS και GRID αντίστοιχα. Στην δεκαετία του 1970 άρχισε να διαφαίνεται η δυνατότητα για τη χρήση σχεσιακών βάσεων δεδομένων σαν εναλλακτική για την αναπαράσταση χωρικών δεδομένων, όμως οι λειτουργίες GIS και DBMS παρέμεναν ανεξάρτητες. Οι Nagy και Wagle στο [66], κάνουν μία ανασκόπηση ήδη από το 1979 περιγράφοντας την μέχρι τότε προσέγγιση στην επεξεργασία γεωγραφικών δεδομένων. Αν και πολλά έχουν αλλάξει από τότε, οι γενικές αρχές και οι εφαρμογές παραμένουν σε γενικές γραμμές ίδιες.

Οι αρχιτεκτονικές που έχουν προταθεί κατά καιρούς για την υλοποίηση χωροχρονικών βάσεων δεδομένων συνοψίζονται στη συνέχεια [69]:

- Ένα υπάρχον σύστημα βάσης δεδομένων (σχεσιακό, αντικειμενοστραφές ή αντικειμενο-σχεσιακό) με ένα επιπλέον επίπεδο για τη διαχείριση των χωροχρονικών δεδομένων (thin or thick layer). Στην περίπτωση αυτή η βάση δεδομένων χρησιμοποιείται σαν μία «αποθήκη αντικειμένων». Αυτή την προσέγγιση ακολουθεί και η ESRI στο ArcGIS που χρησιμοποιήθηκε στην παρούσα εφαρμογή.
- Μία βάση δεδομένων σε συνδυασμό με το σύστημα αρχείων. Στην περίπτωση αυτή κάποια δεδομένα αποθηκεύονται στη βάση και κάποια άλλα παραμένουν στο σύστημα αρχείων, με το μειονέκτημα ότι απαιτείται ο συγχρονισμός των δύο αυτών ανεξάρτητων υποσυστημάτων. Η ανάπτυξη του συστήματος *Icaros.NET* βασίζεται σε αυτή την προσέγγιση, αφού τα διανυσματικά δεδομένα αποθηκεύονται σε μία Personal GeoDatabase, ενώ τα Raster δεδομένα οργανώνονται κάτω από φακέλους στο σύστημα αρχείων.

- Επεκτάσιμες βάσεις δεδομένων. Εισαγωγή νέων τύπων χωρικών και χρονικών δεδομένων στο σύστημα διαχείρισης της βάσης δεδομένων, μέσω μηχανισμών επέκτασης που παρέχονται. Για παράδειγμα η PostgreSQL υποστηρίζει την προσέγγιση αυτή.
- Η αρχιτεκτονική Secondo, που βασίζεται στην υλοποίηση ενός γενικευμένου ΣΔΒΔ το οποίο μπορεί να ενσωματώσει ένα μεγάλο φάσμα μοντέλων δεδομένων.
- Η αρχιτεκτονική Concert η οποία εστιάζεται σε σχεδιαστικά ζητήματα του φυσικού επιπέδου της βάσης δεδομένων παρέχοντας μηχανισμούς επέκτασης και ενσωμάτωσης, με δυνατότητες για την αξιοποίηση συστημάτων που δεν βασίζονται αποκλειστικά σε ένα ΣΔΒΔ, όπως για παράδειγμα παλιότερες εφαρμογές ή WWW Servers

Οι δύο πρώτες πρώτες προσεγγίσεις υποστηρίζουν μία ασθενή σύνδεση με τη βάση δεδομένων, ενώ οι τρεις υπόλοιπες μία προσέγγιση που βασίζεται εξ ολοκλήρου στις δυνατότητες που αυτή μας παρέχει [82].

Στις επεκτάσεις που αναφέρουμε, όσον αφορά στα χωρικά δεδομένα περιλαμβάνονται [48]:

- Χωρικοί τύποι δεδομένων, τόσο στο μοντέλο δεδομένων που υποστηρίζει το ΣΔΒΔ, όσο και στην γλώσσα ερωτήσεων που παρέχει
- Χωρικά ευρετήρια και αποδοτικούς αλγορίθμους χωρικών συνενώσεων (Spatial Indexing/Spatial Joins)

Στο [21] παρουσιάζεται ο τρόπος με τον οποίο ένα γενικής χρήσεως σχεσιακό σύστημα διαχείρισης βάσης δεδομένων, μπορεί να διαχειριστεί ένα μεγάλο όγκο δεδομένων εικόνων και ότι οι απλοί δικτυακοί φυλλομετρητές μπορούν να χρησιμοποιηθούν ικανοποιητικά για την προβολή και παρουσίαση χωρικών εικόνων. Ο Microsoft TerraServer αποτελεί το μεγαλύτερο Online γεωγραφικό ατλαντα-αποθήκη πληροφοριών που συνδιάζει δεδομένα από το USGS και το Spin-2 με συνολικό όγκο 5 terraBytes δεδομένων εικόνων, τα οποία παρέχονται στους χρήστες χωρίς την ανάγκη ειδικού λογισμικού ή υλικού εκτός, του δικτυακού φυλλομετρητή.

3.2 Δεδομένα σε μία βάση δεδομένων

Οι τύποι δεδομένων που υποστηρίζει μία χωροχρονική βάση δεδομένων περιλαμβάνουν [77]:

- Χωρικά δεδομένα
- Χρονικά δεδομένα
- Χωροχρονικά δεδομένα
- Χωρική, χρονική ή χωροχρονική μεταβολή στη θεματική πληροφορία
- Χωρικά, χρονικά ή χωροχρονικά σύνθετα δεδομένα
- Αντικείμενα ή συσχετίσεις που μεταβάλλονται ανάλογα με τη χρονική ή τη χωρική μεταβολή

Ο όρος χωρικά δεδομένα έχει να κάνει με οντότητες που είναι συνδεδεμένες με μία τοποθεσία ή μία σειρά τοποθεσιών πάνω, μέσα ή στην επιφάνεια της γης. Τα χωροχρονικά δεδομένα περιλαμβάνουν επιπλέον και μία χρονική συνιστώσα [74]. Η αναπαράσταση του χώρου και του χρόνου είναι από μόνα τους δύσκολα προβλήματα. Η αναπαράστασή τους δε ταυτόχρονα είναι ακόμα προβληματική.

Τα περισσότερα ΣΔΒΔ χειρίζονται δύο όψεις του χρόνου [81], το πραγματικό χρόνο και τον εσωτερικό χρόνο της βάσης δεδομένων. Ειδικότερα, διακρίνουμε τέσσερις κατηγορίες συστημάτων ανάλογα με τις αναπαραστάσεις του χρόνου που υποστηρίζονται:

- Στατικά, τα οποία δεν υποστηρίζουν καμία χρονική αναπαράσταση
- Ιστορικά, τα οποία υποστηρίζουν μόνο τον έγκυρο-πραγματικό χρόνο
- «Rollback», τα οποία υποστηρίζουν μόνο την εσωτερική αναπαράσταση του χρόνου από το ΣΔΒΔ
- «Bitemporal», τα οποία καλύπτουν και τις δύο προηγούμενες περιπτώσεις

Η χρονική πληροφορία σε μία βάση δεδομένων αναπαρίσταται συνήθως με τη χρήση timestamps. Αυτά μπορεί να χαρακτηρίζουν είτε σημειακά χρονικά γεγονότα (1/1/2001 11:00:00.00), είτε χρονικά διαστήματα (η διάρκεια του φετινού καλοκαιριού), είτε κάποια προκαθορισμένη χρονική διάρκεια χωρίς συγκεκριμένη έναρξη ή τέλος (μία εβδομάδα) [81]. Στο [24] γίνεται αναλυτική παρουσίαση παρόμοιων ζητημάτων. Η προσέγγιση αυτή ξεκίνησε στις αρχές της δεκαετίας του 1980 με την προσθήκη χρονικών δυνατοτήτων στις βάσεις δεδομένων [74].

3.3 Μοντέλα δεδομένων

Πολλές διαφορετικές προσεγγίσεις υπάρχουν για την αναπαράσταση χωρο-χρονικών δεδομένων και διαδικασιών [77], [42]:

- Μοντέλα βασισμένα σε οντότητες ή στο πεδίο (object vs. field)
- Γραμμικά ή μή μοντέλα του χρόνου
- Συνεχείς ή διακριτή αντιμετώπιση των αλλαγών
- Πολλαπλές διαστάσεις του χώρου και του χρόνου

Ειδικά η θεώρηση οντότητας/πεδίου ([33],[109]), χαρακτηρίζει δύο κυρίαρχα και συχνά αντικρουόμενα μοντέλα γεωγραφικών δεδομένων. Το μοντέλο που βασίζεται σε οντότητες είναι περισσότερο προσανατολισμένο στην αντικειμενοστραφή προσέγγιση και αντιμετωπίζει τα χωρικά δεδομένα περιγράφοντας την γεωμετρία σε χαρακτηριστικό των οντοτήτων που απαρτίζουν το χώρο. Αντίθετα η θεώρηση πεδίου αντιμετωπίζει τα χωρικά δεδομένα σε χαρακτηριστικό της θέσης στο χώρο.

Στο [113] προτείνεται ένα μοντέλο δεδομένων όπου κάθε γεωγραφικό αντικείμενο ή χαρακτηριστικό μπορεί να συντεθεί από θεματική πληροφορία, γεωμετρία, τοπολογία και χρονικούς προσδιορισμούς. Τα σύγχρονα συστήματα GIS δεν μπορούν να διαχειριστούν εξίσου καλά όλα τα προηγούμενα και ιδιαίτερο πρόβλημα συναντάται στο κομμάτι της χρονικής πληροφορίας. Η χρονική πληροφορία μπορεί να αποθηκευτεί στα υπάρχοντα συστήματα

γεωγραφικών πληροφοριων χρησιμοποιώντας ένα μοντέλο «snapshot» [74]. Αυτό αν και οδηγεί στην αποθήκευση επιπλέον περιττής πληροφορίας, είναι μία βολική προσέγγιση.

Αντικειμενοστραφείς τεχνικές έχουν συχνά χρησιμοποιηθεί για την υλοποίηση ενός λογικού διανυσματικού και ψηφιδωτού μοντέλου σε συνδυασμό με τη χρήση ενός σχεσιακού ΣΔΒΔ. Αυτό συχνά υποβαθμίζει την πραγματική δύναμη του αντικειμενοστραφούς μοντέλου το οποίο μπορεί να παρέχει τις απαραίτητες χωροχρονικές αναπαραστάσεις. Η επέκταση του υπάρχοντος σχεσιακού μοντέλου για την αναπαράσταση του χώρου και του χρόνου οδήγησε σε υλοποιήσεις ογκώδεις και πολύπλοκες. Επίσης, η απλή επέκταση ενός χωρικού μοντέλου δεδομένων ώστε να συμπεριλάβει και χρονικά δεδομένα ή το αντίστροφο, έχει σαν αποτέλεσμα αναποτελεσματικές και μη ευέλικτες αναπαραστάσεις χωροχρονικών δεδομένων [74].

Μία άλλη προσέγγιση συναντά κανείς στο Schneider ([85], [84]), ο οποίος παρουσιάζει την έννοια των ασαφών γεωγραφικών αντικειμένων (Fuzzy Spatial Objects) και μία σειρά ιδιοτήτων και κατηγορημάτων τέτοιων ώστε να μπορούν να εισαχθούν σε μία βάση δεδομένων και να ανακτηθούν μέσα από μία γλώσσα ερωτήσεων. Η προσέγγιση αυτή έρχεται σε αντίθεση με την παραδοχή ότι όλα τα αντικείμενα σε ένα γεωγραφικό σύστημα πληροφοριών έχουν σαφή όρια, η οποία απέχει πολύ από την πραγματικότητα σε ορισμένες περιπτώσεις αφού κανείς δεν μπορεί να ορίσει με ακρίβεια τα σύνορα ενός βουνού με μία πεδιάδα ή ακόμα και την κατανομή των πετρωμάτων σε μία περιοχή. Η χωρική ασάφεια είναι εγγενής ιδιότητα πολλών γεωγραφικών αντικειμένων και δεν μπορούμε να την εξαλείψουμε.

3.4 Πρόσβαση στα δεδομένα

Η αλληλεπιδραστική εργασία με τόσο μεγάλο όγκο χωρικών δεδομένων είναι μία δύσκολη εργασία και για το λόγο αυτό παραδοσιακά τα ΣΓΠ και πολλές γεωγραφικές βάσεις δεδομένων λειτουργούν σαν αυτόνομα συστήματα (stand-alone). Έχουν αναπτυχθεί ειδικές δομές για τη δημιουργία γεωγραφικών ευρητηρίων όπως τα K-d trees, Quad-trees και R-trees [90]. Οι δομές αυτές χρησιμοποιούνται από τα γεωγραφικά ΣΔΒΔ με σκοπό τη δημιουργία

ευρετηρίων αντιστοιχών με αυτά που δημιουργούνται για συνήθη δεδομένα (indexing).

Στο μέλλον προβλέπεται να υπάρξει μία μεγάλη ανάγκη για την υποστήριξη από τις βάσεις δεδομένων, αποδοτικής διαχείρισης δεδομένων Raster σε ένα δικτυακό περιβάλλον [108]. Προς το παρόν η υπάρχουσα γενική δυνατότητα για την αποθήκευση μεγάλων δυαδικών αντικειμένων (BLOB) δεν παρέχει ικανοποιητικές υλοποιήσεις.

Η έρευνα για την ανάπτυξη αποδοτικών συστημάτων βασισμένων στο διαδίκτυο για πρόσβαση σε βάσεις γεωγραφικών δεδομένων γίνεται προς την κατεύθυνση ([96], [35], [97]):

- της ανάπτυξης συστημάτων διαχείρισης βάσεων δεδομένων ικανών για την αποδοτική διαχείριση μεγάλων και ανομοιογενών συνόλων γεωγραφικών δεδομένων
- της υποστήριξης μεθόδων ανάλυσης των χωρικών δεδομένων
- της ανάπτυξης μεθόδων δεικτοδότησης (Indexing) χωρικών δεδομένων.

Παρά όλα αυτά, τα ΣΓΠ και οι εφαρμογές τους, που βασίζονται σε Web τεχνολογίες υστερούν αφού :

- υπάρχει αδυναμία των βάσεων δεδομένων για την αποδοτική διαχείριση μεγάλων και διαφορετικών συνόλων γεωγραφικών δεδομένων
- υπάρχει τάση για σύνθετη τεχνολογία υποστήριξης του WWW η οποία κληρονομείται και στις αντίστοιχες GIS εφαρμογές
- σημαντική είναι η υποβάθμιση της λειτουργίας πολλών συστημάτων GIS λόγω κακών ρυθμίσεων

3.5 Ερωτήσεις στη βάση δεδομένων

Έχοντας οργανώσει τα δεδομένα μας σε ένα ΣΔΒΔ, είναι πολύ σημαντικό να μπορούμε να εξάγουμε χρήσιμη πληροφορία από αυτά. Η διαδικασία αυτή

γίνεται συνήθως μέσα από μία αλληλεπιδραστική γλώσσα ερωτήσεων που συνοδεύει κάθε ΣΔΒΔ. Επειδή μία χωροχρονική βάση δεδομένων περιλαμβάνει κυρίως δεδομένα τα οποία γίνονται αντιληπτά σε έμας μέσω κάποιας μορφής οπτικοποίησης, προέκυψε η ανάγκη για την ανάπτυξη γραφικών γλωσσών ερωτήσεων, ως επέκταση των ήδη υπαρχόντων. Επιπλέον, είναι απαραίτητο να μπορούν να γίνονται ερωτήσεις που αφορούν στην χρονική εξέλιξη των οντοτήτων που βρίσκονται αποθηκευμένα στη βάση δεδομένων. Η ενιαία αντιμετώπιση τόσο χωρικών όσο και χρονικών δεδομένων αποτελεί σήμερα μία πρόκληση ([34], [26] και [25]).

3.5.1 Τελεστές

Οι ερωτήσεις που μπορεί να απευθύνει κάποιος σε ένα χωροχρονικό ΣΔΒΔ συνοψίζονται σε χωρικές ερωτήσεις (περιλαμβάνουν τουλάχιστον ένα χωρικό κριτήριο), χρονικές ερωτήσεις (περιλαμβάνουν τουλάχιστον ένα χρονικό κριτήριο) και χωροχρονικές ερωτήσεις (περιλαμβάνουν τουλάχιστον ένα χωρικό και ένα χρονικό κριτήριο ή είναι αποτέλεσμα κριτηρίων που αφορούν την εξέλιξη στο χρόνο) [107]. Η τελευταία μορφή απαντάται συχνότερα στα γεωγραφικά συστήματα πληροφοριών.

Μεταξύ των χωρικών οντοτήτων μπορεί να ορίσει κανείς δυδικούς χωρικούς τελεστές όπως η «επαφή», η «ταυτότητα (σύμπτωση)», η «συμπερίληψη», το «κοινό τμήμα», ή και η «μη συσχέτιση». Αντίστοιχα μεταξύ δύο χρονικών στιγμών ή διαρκειών μπορούν να οριστούν τελεστές όπως «πριν», «μαζί», «τελειώνει όταν ξεκινά», «αλληλοκαλύπτεται», «κατά την διάρκεια», «ξεκινά μαζί», «τελειώνει μαζί». Επίσης μπορούν να οριστούν τελεστές εξέλιξης των διαφορών οντοτήτων όπως «δημιουργία», «παύση-καταστροφή», «διαχωρισμός», «ένωση», «διόγκωση» και «συρρίκνωση». Ο συνδιασμός των προηγούμενων τελεστών με τις χωροχρονικές οντότητες και τα χαρακτηριστικά τους, τα οποία έχουν αποθηκευτεί σε ένα ΣΔΒΔ συνιστά στη δημιουργία ενός μεγάλου πλήθους διαθέσιμων ερωτήσεων.

3.5.2 Γλώσσες ερωτήσεων

Κατά καιρούς έχουν προταθεί διάφορες γλώσσες ερωτίσεων για την αλληλεπίδραση με ένα ΣΔΒΔ το οποίο εμπλέκει και χωροχρονικά δεδομένα. Η Constraint-based χωρική επέκταση της SQL με δυνατότητες χειρισμού γεωγραφικών δεδομένων περιγράφεται στο [59]. Στο [86] γίνεται ανάλυση ενός αποδοτικού τρόπου εφαρμογής ερωτήσεων που αφορούν κινούμενα αντικείμενα με προκαθορισμένες τροχιές σε ένα κατανεμημένο περιβάλλον. Στο [107] περιγράφεται μία γλώσσα ερωτήσεων για αντικειμενοστραφή ΣΓΠ με δυνατότητα για τοπολογικές συσχετίσεις (όπως: περιέχει, επικαλύπτει, ταυτίζεται), μετρικές σχέσεις (όπως: αποστάσεις και κατευθύνσεις), και ονομαστικούς προσδιορισμούς (όπως: βόρεια, νότια, αριστερά, δεξιά).

Μία μέθοδος για την επεξεργασία ερωτήσεων σε κατανεμημένα συστήματα γεωγραφικών βάσεων δεδομένων, ανομοιογενών με (χωρικές ασυμφωνίες) προτείνεται στο [53] και βρίσκει εφαρμογή στην περίπτωση που απαιτείται συνδυασμός δεδομένων από διαφορετικούς οργανισμούς, κάθε ένας από του οποίους μπορεί να έχει τη δική του πολιτική για τη συλλογή και διαχείριση δεδομένων, με διαφορετικές κλίμακες ή ακρίβεια. Στο [108] παρουσιάζεται ένα σύστημα βάσεων δεδομένων για την αποθήκευση ψηφιδωτών δεδομένων με επεκτάσεις στη γλώσσα ερωτήσεων για αναζητήσεις πολυδιάστατων διακριτών δεδομένων (MDD). Τέλος στο [71] γίνεται μία προσπάθεια για την τυποποίηση της χρονικής εξέλιξης θεματικών χαρτών με σκοπό την απάντηση ερωτήσεων που σχετίζονται με την χρονική εξέλιξη στοιχείων που χαρακτηρίζουν το θεματικό περιεχόμενο του χάρτη.

3.5.3 Νέες τεχνικές ανάλυσης

Η αυξανόμενη σημασία της χωρικής ανάλυσης δεδομένων, και η ανάγκη για αποδοτική και οικονομική δημιουργία αποθηκών δεδομένων σε συνδυασμό με τη ραγδαία ανάπτυξη των συστημάτων γεωγραφικών πληροφοριών έφερε στο προσκήνιο εφαρμογές που παλιότερα συναντούσε κανείς μόνο στα κλασικά ΣΔΒΔ [54]. Οι βάσεις δεδομένων εξάλλου παρουσιάζουν μία αλματώδη ανάπτυξη και η ανθρώπινη ικανότητα για την ερμηνεία του όγκου των διαθέσιμων δεδομένων και την μετατροπή του σε χρήσιμη πληροφορία οδηγεί στην

ανάπτυξη τεχνικών εξόρυξης δεδομένων (Spatial Data Mining) ([39], [49]). Για παράδειγμα στο [64] παρουσιάζεται μία τεχνική GOLAP η οποία αποτελεί επέκταση στο OLAP με προσθήκη χωρικών ευρετηρίων και διαστάσεων. Επίσης στο [60] μπορεί να βρει κανείς εφαρμογές ΣΓΠ για την υποστήριξη αποφάσεων βασισμένα στην ασαφή λογική και τα νευρωνικά δίκτυα. Τέλος στο [80] γίνεται παρουσίαση των δυνατοτήτων επέκτασης των ΣΓΠ για Spatiotemporal-Reasoning (ArcGis81, VB, Sicstus Prolog).

3.6 Ποιότητα δεδομένων

Ιδιαίτερο ρόλο στις βάσεις δεδομένων παίζει και η ποιότητα δεδομένων. Παρόλο που η διασφάλισή της δεν είναι εύκολη διαδικασία, παίζει καθοριστικό ρόλο στη σωστή λειτουργία του τελικού συστήματος. Οι χωροχρονικές βάσεις δεδομένων περιλαμβάνουν οντότητες οι οποίες χαρακτηρίζονται από ασάφεια (= εγγενής ανακρίβεια) και αβεβαιότητα (= έλλειψη πληροφορίας) [74]. Αυτό συμβαίνει ακόμα περισσότερο όταν έχουμε να κάνουμε με την ενοποίηση γεωγραφικών δεδομένων που προέρχονται από διαφορετικές πηγές [19], γεγονός που αποτελεί κανόνα για το μεγαλύτερο ποσοστό των γεωγραφικών εφαρμογών.

Η εισαγωγή γεωγραφικών δεδομένων σε ένα ΣΔΒΔ είναι η διαδικασία που συνήθως εισάγει και τα περισσότερα λάθη ενώ άλλες πηγές εισαγωγής λαθών μπορεί να συναντήσει κανείς κατά την συλλογή των δεδομένων, κατά την αποθήκευση των δεδομένων, κατά την επεξεργασία τους, κατά την εξαγωγή-παρουσίαση και τέλος κατά τη χρήση των αποτελεσμάτων και την ερμηνεία τους. Η ποιότητα των δεδομένων γενικά περιλαμβάνει [42] στοιχεία που αφορούν την πληρότητα των δεδομένων, την θεματική ακρίβεια, την χρονική ακρίβεια, την χωρική ακρίβεια, την λογική συνέπεια και την ύπαρξη μεταδεδομένων που περιγράφουν τα ίδια τα δεδομένα. Ο συνδιασμός όσο το δυνατόν περισσότερων από τα προηγούμενα μπορεί να συμβάλει στην σωστότερη αξιοποίηση της υπάρχουσας πληροφορίας και την αξιοπιστία των παραγόμενων αποτελεσμάτων.

Κεφάλαιο 4

Το σύστημα *Icaros.NET*

4.1 Εισαγωγή

Το σύστημα *Icaros.NET* περιγράφεται αναλυτικά στο [83]. Στο κεφάλαιο αυτό θα περιγραφεί συνοπτικά η σκοπιμότητα της ανάπτυξης της πλατφόρμας και η συνεισφορά της παρούσας διπλωματικής εργασίας προς την κατεύθυνση αυτή.

4.2 Σκοπός

Σκοπός του συστήματος *Icaros.NET*, το οποίο επικεντρώνεται στη μελέτη και την καταγραφή αέριας ρύπανσης, όπως προκύπτει από τις τεχνικές προδιαγραφές είναι επιγραμματικά :

- Η γρήγορη πρόβλεψη της κατανομής αερίων ρύπων στη τροπόσφαιρα
- Η παροχή στατιστικών δεδομένων σε ετήσια βάση σε σχέση με τη συγκέντρωση των αερίων ρύπων και τη συχνότητα υπέρβασης των ανεκτών ορίων
- Η βελτιστοποιημένη ανάπτυξη δικτύου καταγραφής των σωματιδιακών ρύπων $PM_{2.5}$ και PM_{10} (σωματίδια μεγέθους 2.5 και 10 μm αντίστοιχα)

- Η συμβολή στην ολοκλήρωση και λειτουργία ενός πολυκριτηριακού συστήματος επιτήρησης και διαχείρισης εκτάκτων καταστάσεων με έμφαση σε παράγοντες που επηρεάζουν την δημόσια υγεία, όπως $PM_{2.5}$, PM_{10} και O_3
- Η συνοπτική εκτίμηση των περιβαλλοντικών επιπτώσεων που οφείλονται στην αέρια ρύπανση
- Η παραγωγή χαρτών

Η αέρια ρύπανση είναι σημαντικός παράγοντας για τους κατοίκους μίας περιοχής σε σχέση με την ποιότητα της ζωής [55]. Για τους εκλεγμένους αντιπροσώπους και τα κυβερνητικά στελέχη είναι πολύ σημαντικό εργαλείο μία εφαρμογή η οποία να μπορεί σε σχεδόν πραγματικό χρόνο να αποδώσει μία σαφή εικόνα της αέριας ρύπανσης ώστε με ακρίβεια και συνέπεια να παρθούν τα κατάλληλα μέτρα σε περιπτώσεις υπέρβασης των ορίων των συγκεντρώσεων. Τόσο για τη λήψη βραχυπρόθεσμων αποφάσεων, αλλά περισσότερο για ένα μακροχρόνιο σχεδιασμό, μία τέτοια εφαρμογή μπορεί να αναδείξει την εξέλιξη της ρύπανσης σε μία περιοχή ανατρέχοντας σε ιστορικά δεδομένα και επιπλέον στην πρόβλεψη καταστάσεων με εφαρμογή υποθετικών σεναρίων. Σχετικά στο [51], παρουσιάζεται μία διαδικασία ενοποίησης κρατικών διαδικασιών και υπηρεσιών εξυπηρέτησης πολιτών μέσα από μία εφαρμογή Web GIS η οποία συμπεριλαμβάνει και ένα σύστημα λήψης αποφάσεων μέσα από κανόνες.

Η βασική ιδέα είναι ο συνδυασμός του συνόλου των υπάρχουσών τεχνικών για την παραγωγή των παραπάνω αποτελεσμάτων, μέσα από ένα ενιαίο σύστημα και με μεγάλο βαθμό αυτοματισμού. Τεχνικές που συνήθως παράγουν ένα μεγάλο ποσό χρήσιμης πληροφορίας, για την οποία απαιτείται ένα ικανό σύστημα μεταφοράς, αποθήκευσης και διαχείρισης, ώστε να μπορέσουν να προκύψουν όλα τα παραπάνω, με ένα μεγάλο βαθμό ακρίβειας, εγκυρότητας και αξιοπιστίας.

4.3 Αντίστοιχες εργασίες

Αντίστοιχη προσπάθεια έχει γίνει και από τον Folgert στο [41], ο οποίος ανέπτυξε ένα σύστημα πρόβλεψης αέριων ρύπων για την περιοχή του Billings, Montana, USA, με έμφαση στην μελέτη ρύπων SO_2 . Ωστόσο εκεί δεν δίνεται τόσο μεγάλη έμφαση στα εργαλεία ανάπτυξης της εφαρμογής και της μοντελοποίησης των δεδομένων. Το σίγουρο είναι ότι ενώ πρόκειται για μία πλήρη εφαρμογή, έχει να αντιμετωπίσει μικρότερο όγκο δεδομένων αφού εστιάζεται σε ένα μόνο ρύπο και μερικές μόνο μεθόδους υπολογισμού των συγκεντρώσεων του στην ατμόσφαιρα. Είναι δε σημαντικό ότι η προσέγγισή του βασίζεται στην προηγούμενης γενιάς πλατφόρμα GIS της ESRI που αξιοποιεί την Avenue.

Μία εφαρμογή κινούμενης χαρτογραφίας (animated cartography) για την μελέτη της καταγραφής του θορύβου σε ένα αστικό περιβάλλον παρουσιάζεται στο [55]. Σκοπός της εφαρμογής είναι η ανάπτυξη ενός εξειδικευμένου ΣΓΠ για τη μελέτη της κατανομής, την αναπαράσταση, τη διαχείριση και την εξομοίωση του θορύβου σε μία πόλη. Η συγκεκριμένη εφαρμογή έχει αρκετές ομοιότητες με το σύστημα *Icaros.NET*, αν η χρήση της στραφεί προς την καταγραφή και διαχείριση αέριας ρύπανσης. Στη περίπτωση αυτή οι τελικοί χρήστες και οι απαιτήσεις τους παραμένουν ίδιες.

Το σύστημα GeoWorlds [58], αποτελεί μία πλατφόρμα ανάπτυξης εφαρμογών GIS με χρήση τεχνολογιών διαδικτύου, ψηφιακών βιβλιοθηκών και συστημάτων γεωγραφικών πληροφοριών. Στόχος είναι η δημιουργία ομάδων χρηστών με κοινή αντίληψη για μία γεωγραφική περιοχή και την ικανότητα για διαχείριση και οργάνωση όλης της διαθέσιμης πληροφορίας σε σχέση με το χώρο και το χρόνο. Η πλατφόρμα εντάσσεται στην γενικότερη κατηγορία εφαρμογών με στόχο την κατανόηση και διαχείριση καταστάσεων. Η αρχιτεκτονική του συστήματος αποτελεί ένα γενικότερο πλαίσιο στο οποίο μπορεί να ενταχθεί και το σύστημα *Icaros.NET*.

Στο [62], γίνεται μία παρουσίαση για τις δυνατότητες και τις προοπτικές συνδυασμού των περιβαλλοντικών μοντέλων με τα ΣΓΠ με σκοπό την ανάπτυξη συστημάτων περιβαλλοντικής διαχείρισης και λήψης αποφάσεων. Ενώ η αναφορά εστιάζεται σε παγκοσμίου κλίμακας συστήματα, πολλά από

τα συμπεράσματα βρίσκουν εφαρμογή και στην περίπτωση της πλατφόρμας *Icaros.NET*. Αντίστοιχα στο [101], δίνεται η αρχιτεκτονική και παρουσιάζεται η λειτουργία ενός συστήματος για την παρουσίαση ενός υδροδυναμικού μοντέλου μεταφοράς της ρύπανσης με την χρήση ΣΓΠ.

4.4 Απαιτήσεις υλοποίησης

Για την ανάπτυξη μίας εφαρμογής ΣΓΠ, απαιτούνται δύο στάδια [103]:

- Η προετοιμασία των δεδομένων
- Ο σχεδιασμός της αλληλεπίδρασης του χρήστη με την εφαρμογή

Και τα δύο αυτά στάδια είναι εξίσου σημαντικά λόγω της διαρκούς αλληλεπίδρασης επιπέδου δεδομένων και επιπέδου παρουσίασης. Επίσης όπως προέκυψε από την μεθοδολογία ISOLA [20], ορίζονται οι ακόλουθες βασικές απαιτήσεις:

- Ο τελικός χρήστης πρέπει να μπορεί να εκφράσει ευθέως την υπολογιστική διαδικασία που πρόκειται να ακολουθηθεί μέσα από μία υψηλού επιπέδου γλώσσα
- Η λογική αναπαράσταση των δεδομένων (Conceptual Data Model), πρέπει να είναι επαρκής ώστε ο χρήστης να είναι αποκομμένος από την φυσική δομή και την κωδικοποίησή τους (Physical Data Model), αλλά να έχει μία περισσότερο αφηρημένη αντίληψη
- Η περιγραφή των διαδικασιών πρέπει να είναι διαχωρισμένη από την εκτέλεση των διαδικασιών ώστε να επιτρέπεται η σύνδεση με τα δεδομένα μόνο στο τελικό στάδιο της εκτέλεσης
- Η περιγραφή των διαδικασιών πρέπει να είναι ανεξάρτητη από τη τεχνολογία GIS που αξιοποιείται ώστε να μπορεί αυτή να αντικατασταθεί από άλλη

4.4.1 Λειτουργία

Το σύστημα *Icaros.NET* πρέπει να είναι ευέλικτο ώστε να επιτρέπεται η λειτουργία του τόσο σε ένα κατανεμημένο περιβάλλον όσο και σε αυτόνομη κατάσταση τοπικά. Ειδικότερα υποστηρίζεται:

- Δυνατότητα λειτουργίας μέσω διαδικτύου: Η βάση δεδομένων είναι εγκατεστημένη κεντρικά σε κάποιο διακομιστή και όλες οι λειτουργίες που αφορούν τα δεδομένα γίνεται εκεί ενώ οι λειτουργίες παρουσίασης γίνονται σε απομακρυσμένα μηχανήματα κατά περίπτωση.
- Δυνατότητα αυτόνομης λειτουργίας: Τόσο η βάση δεδομένων όσο και οι λειτουργίες παρουσίασης βρίσκονται και εκτελούνται στο ίδιο μηχάνημα.

Στην πρώτη περίπτωση το σύστημα λειτουργεί σε δικτυακό περιβάλλον επιτρέποντας σε περισσότερους από ένα χρήστη, να αντλεί πληροφορίες ενώ υπάρχει και η δυνατότητα εισαγωγής νέων δεδομένων σε πραγματικό χρόνο. Στην δεύτερη περίπτωση η λειτουργία περιορίζεται σε ένα απομονωμένο περιβάλλον, εξίσου χρήσιμο όμως, αφού έχει μικρότερες απαιτήσεις υλοποίησης και είναι ιδανικό για τις ανάγκες ανάπτυξης και επιδείξεων. Σε κάθε περίπτωση η λειτουργία του συστήματος πρέπει να είναι ανεξάρτητη από την κατάσταση λειτουργίας, τόσο σε επίπεδο διαχείρισης δεδομένων, όσο και σε επίπεδο χρήσης.

4.4.2 Περιβάλλον εργασίας

Ένα εύχρηστο γραφικό περιβάλλον εργασίας, τόσο για την εισαγωγή δεδομένων όσο και για την παρουσίαση αποτελεσμάτων είναι καθοριστικής σημασίας για την επιτυχή λειτουργία του *Icaros.NET*. Τα εμπορικά ΣΓΠ, όπως αυτό που εμείς χρησιμοποιούμε για την ανάπτυξη του *Icaros.NET* [98], είναι δύσκολα στη χρήση εκτός εάν προϋπάρχει επαρκής γνώση γεωγραφίας, χαρτογραφίας και συστημάτων διαχείρισης βάσεων δεδομένων, είναι προσανατολισμένα στην τεχνολογία των πληροφοριακών συστημάτων και απαιτούν μεγάλο χρόνο για την εξοικείωση με τις διαδικασίες που αντιστοιχούν στην αρχιτεκτονική

του συστήματος. Μέσα στο περιβάλλον εργασίας του *Icaros.NET* προσπαθούμε να κρύψουμε όλη αυτή τη προαπαιτούμενη γνώση, ώστε ο τελικός χρήστης να επικεντρωθεί στο στόχο του: Την μελέτη της αέριας ρύπανσης.

Ως τελικός χρήστης (End User) ορίζεται ένας χρήστης ο οποίος:

- χρησιμοποιεί έναν υπολογιστή ως μέσο με σκοπό να ολοκληρώσει μία εργασία στο πεδίο της ειδικότητάς του
- έχει λίγες ή καθόλου γνώσεις προγραμματισμού
- δεν έχει εγγενές ενδιαφέρον στους υπολογιστές
- προτιμάει να επικεντρωθεί στο δικό του πεδίο ενδιαφέροντος
- δεν έχει κάποια ειδίκευση στη γεωγραφία

Στην κατηγορία αυτή εντάσσονται τόσο κυβερνητικοί υπάλληλοι και οργανισμοί, οι οποίοι είναι δυνάμει χρήστες της πλατφόρμας αλλά και απλοί κάτοικοι μίας περιοχής στην οποία έχει εγκατασταθεί η πλατφόρμα *Icaros.NET*.

4.4.3 Δεδομένα

Το σύστημα στην πλήρη ανάπτυξή του θα πρέπει να μπορεί να διαχειριστεί μία ποικιλία δεδομένων, πολλά από τα οποία σχετίζονται μεταξύ άλλων και με γεωγραφικά χαρακτηριστικά. Εκτός από την πληροφορία εκείνη που παραμένει σχεδόν αμετάβλητη κατά την λειτουργία της πλατφόρμας και δεν θα μας απασχολήσει εδώ, όπως για παράδειγμα το οδικό δίκτυο της περιοχής μελέτης, άλλες πηγές παρέχουν πληροφορία με μεγαλύτερους ρυθμούς. Εξάλλου για τη σωστή χρήση και παρακολούθηση των επίγειων και περιβαλλοντικών πόρων για την ποιότητα της ζωής και την αειφόρο ανάπτυξη απαιτούνται πυκνά χωροχρονικά και ακριβή δεδομένα [103].

Επιγραμματικά τέτοιες πηγές περιλαμβάνουν:

- Μειρήσεις από επίγειους σταθμούς, οι οποίοι καταγράφουν σε τακτά χρονικά διαστήματα τις μέσες τιμές διαφόρων ρύπων
- Μοντέλα συμπεριφοράς της ατμόσφαιρας, με αποτελέσματα τα οποία παρέχονται συνήθως σε ημερήσια βάση

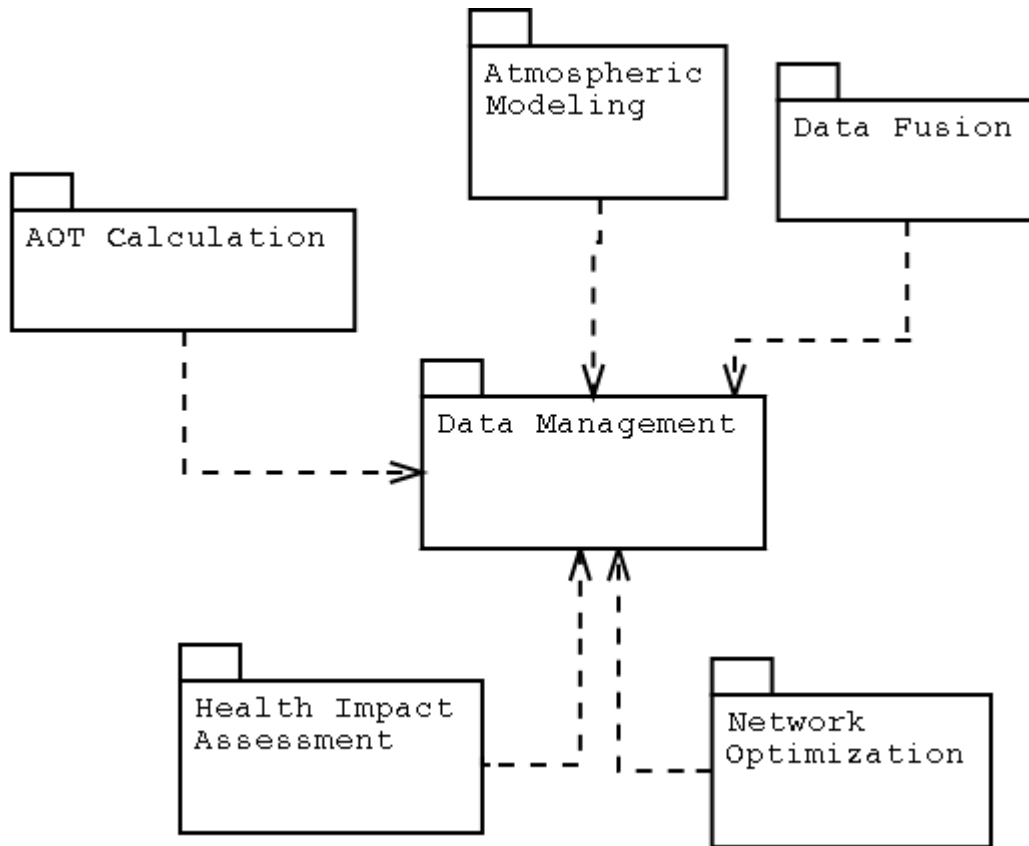
- Δορυφορικές εικόνες, οι οποίες παρέχονται ανά δύο ή τρεις ή και περισσότερες ημέρες ανάλογα με την διαθεσιμότητα από το δορυφόρο και τις οικονομικές δυνατότητες του χρηματοδότη

Τα δεδομένα που μόλις περιγράψαμε είναι καθοριστικά τόσο για την εγκατάσταση όσο και για την λειτουργία του συστήματος και η σωστή διαχείρισή τους επηρεάζει άμεσα την ποιότητα και την αξιοπιστία του παραγόμενου αποτελέσματος. Δεν πρέπει να παραβλέψουμε το γεγονός ότι η ανάλυση γεωγραφικών δεδομένων είναι τόσο καλή όσο τα δεδομένα στα οποία στηρίζεται [36].

4.5 Αρχιτεκτονική

Η αρχιτεκτονική σχεδίασης του συστήματος *Icaros.NET* αποτελείται από πέντε ανεξάρτητα υποσυστήματα, συνοψίζεται στο σχήμα 4.1 και περιγράφεται πιο αναλυτικά στη συνέχεια :

- Διαχείριση Δεδομένων (Data Management): Υλοποιεί την εισαγωγή δεδομένων στη βάση δεδομένων της πλατφόρμας, από όπου στη συνέχεια θα ανακτηθούν πληροφορίες για τις άλλες λειτουργίες της εφαρμογής
- Μοντέλα Συμπεριφοράς της Ατμόσφαιρας (Atmospheric Modeling): Υπολογισμός της συμπεριφοράς της ατμόσφαιρας με βάση επίκαιρα μετεωρολογικά δεδομένα και το μοντέλο εκπομπής ρύπων της υπό εξέταση περιοχής
- Υπολογισμός AOT (AOT Calculation): Υπολογισμός του AOT από τις δορυφορικές εικόνες
- Διαδικασία «Σύντηξης» Δεδομένων (Data Fusion): Σύνθεση της υπάρχουσας πληροφορίας
- Βελτιστοποίηση της Εγκατάστασης (Network Optimization): Επανυπολογισμός των παραμέτρων της λειτουργίας του συστήματος
- Εκτίμηση Κινδύνου (Health Impact Assesement): Παραγωγή χαρτών επικινδυνότητας



Σχήμα 4.1: Η αρχιτεκτονική του συστήματος και οι ισχυρές εξαρτήσεις από το επίπεδο διαχείρισης δεδομένων

Κάθε υποσύστημα του *Icaros.NET* υλοποιεί μία συγκεκριμένη εργασία, ωστόσο η διαχείριση δεδομένων αποτελεί κεντρικό σημείο με το οποίο αλληλεπιδρούν όλα τα άλλα υποσυστήματα. Για το λόγο αυτό έχει δοθεί ιδιαίτερη σημασία κατά την ανάπτυξη της πλατφόρμας στο υποσύστημα αυτό.

4.6 Εφαρμογή

Το σύστημα κατά την διάρκεια της υλοποίησης του δοκιμάζεται σε τέσσερις διαφορετικές περιοχές της Ευρωπαϊκής Ένωσης στις οποίες περιλαμβάνονται οι:

- Αθήνα

- Μόναχο
- Λομβαρδία
- Βουδαπέστη

Σε κάθε μία από τις περιοχές αυτές γίνονται εκστρατείες μαζικής συλλογής δεδομένων διάρκειας μίας έως δύο εβδομάδων. Τα δεδομένα αυτά για τα συγκεκριμένα χρονικά διαστήματα αντικατοπτρίζουν τον όγκο και τον τύπο των δεδομένων που πρέπει να διαχειριστεί το σύστημα υπό συνθήκες πραγματικής λειτουργίας. Έχοντας μία πληθώρα πραγματικών δεδομένων μπορούμε να αξιολογήσουμε και να βελτιώσουμε τόσο τη συμπεριφορά του συστήματος γενικότερα όσο και την λειτουργία του επιπέδου διαχείρισης δεδομένων, το οποίο και εξετάζουμε στη παρούσα διπλωματική εργασία.

Είναι σημαντικό να επισημάνουμε ότι ενώ και στις τέσσερις περιοχές πρόκειται να γίνει εγκατάσταση του ίδιου συστήματος *Icaros.NET*, καμία από τις περιοχές αυτές δεν έχει κάτι κοινό με κάποια άλλη. Έχουμε να αντιμετωπίσουμε διαφορετικό προβολικό σύστημα συντεταγμένων, διαφορετικού τύπου σταθμούς μέτρησης και διαφορετικής μορφής αρχεία δεδομένων. Η ενιαία αντιμετώπιση από το σύστημα του συνόλου της πληροφορίας και για τις τέσσερις περιοχές μπορεί να μας οδηγήσει σε ένα ομοιόμορφο σχεδιασμό του επιπέδου διαχείρισης δεδομένων παρέχοντας την δυνατότητα για την εγκατάσταση του μελλοντικά και σε άλλες γεωγραφικές περιοχές.

4.6.1 Προκλήσεις

Για την επιτυχή υλοποίηση του συστήματος πρέπει να λάβουμε υπ' όψη τα ακόλουθα που το διαφοροποιούν από άλλες εφαρμογές ΣΓΠ:

- Απαιτεί πολλούς τύπους διαφορετικών δεδομένων: Πρέπει να γίνει κατά το δυνατό ενιαία διαχείριση του συνόλου των δεδομένων ώστε το σύστημα διαχείρισης δεδομένων της πλατφόρμας να μην είναι ιδιαίτερα πολύπλοκο
- Εμπλέκει δεδομένα με γεωγραφική πληροφορία: Πρέπει εκτός από τα υπόλοιπα χαρακτηριστικά τους να ληφθεί υπ' όψη και το σύστημα

γεωγραφικής αναφοράς που σχετίζεται με την γεωγραφική πληροφορία

- Εγκαθίσταται σε διαφορετικές γεωγραφικές περιοχές εφαρμογής: Όπως είδαμε στην προηγούμενη παράγραφο το γεγονός αυτό περιπλέκει την διαδικασία ανάπτυξης
- Διαχειρίζεται δεδομένα με διαφορετικές ζώνες αναφοράς ώρας: Έχοντας να συνδυάσουμε δορυφορικά δεδομένα με επίγειες μετρήσεις πρέπει να λάβουμε υπ' όψη την ακριβή χρονική στιγμή που αφορά την συλλογή των διαφόρων κομματιών πληροφορίας που εισάγονται στο σύστημά μας
- Απαιτεί μεγάλο όγκο δεδομένων: Τα οποία όμως είναι απαραίτητα για την λειτουργία της πλατφόρμας
- Απαιτεί μεγάλο όγκο συνοδευτικής πληροφορίας για τα δεδομένα: Η οποία είναι απαραίτητη για την λειτουργία της πλατφόρμας και καλό θα ήταν να ενσωματωθεί με κάποιο τρόπο στην λογική λειτουργίας της
- Πρέπει να είναι ευέλικτη και επεκτάσιμη: Η πλατφόρμα πρέπει να είναι επεκτάσιμη κατά το δυνατόν, τόσο σε σχέση με την εφαρμογή της σε διαφορετικές περιοχές αλλά ακόμα περισσότερο σε σχέση με την υποστήριξη νέων ρύπων και ατμοσφαιρικών μοντέλων
- Απαιτεί εύκολη και γρήγορη εισαγωγή και ανάκτηση των δεδομένων: ώστε η πλατφόρμα να μπορεί να δώσει γρήγορα αποτελέσματα
- Απαιτεί αξιοπιστία και έλεγχο στα δεδομένα: ώστε και τα αποτελέσματα που προέρχονται από την πλατφόρμα να είναι κατά το δυνατόν αξιόπιστα
- Απαιτεί δυνατότητα αναζήτησης στα δεδομένα που έχουν εισαχθεί με κριτήρια χρονικά, ποιοτικά και ποσοτικά: με σκοπό την εξαγωγή χρήσιμων συμπερασμάτων

Κεφάλαιο 5

Ένοποιημένη Γλώσσα Μοντελοποίησης (UML)

5.1 Τί είναι;

The UML is a standard language for writing software blueprints. It may be used to visualize, specify, construct and document the artifacts of a software intensive system [11].

Το παραπάνω περιγράφει συνοπτικά το σκοπό ύπαρξης της UML. Αποτελεί μία γλώσσα μοντελοποίησης που διέπεται από σαφείς κανόνες κατασκευής μοντέλων που αφορούν συνήθως συστήματα λογισμικού, αλλά μπορεί να περιγράψει το ίδιο καλά και γενικότερα συστήματα ή διαδικασίες και είναι στενά συνδεδεμένη με μία αντικειμενοστραφή αντίληψη της πραγματικότητας. Όσον αφορά την τεχνολογία λογισμικού, η χρήση της UML πετυχαίνει αυτό που σε όλους τους άλλους χώρους κατασκευής συστημάτων είναι αυτονόητο: την παραγωγή σχεδίων που αντικατοπτρίζουν το σύνολο της ύπαρξης ενός συστήματος όσον αφορά στα στατικά αλλά και στα δυναμικά χαρακτηριστικά του. Η ύπαρξη δε κατάλληλων εργαλείων και μεθοδολογιών μπορεί να απλοποιήσει σε ένα μεγάλο βαθμό την ανάπτυξη σύνθετων συστημάτων.

5.2 Ιστορικά

Καθώς τα συστήματα λογισμικού γίνονταν ολοένα και περισσότερο πολύπλοκα, κατά την δεκαετία του 1970 και του 1980 εμφανίστηκαν διάφορες μεθοδολογίες μοντελοποίησης βασισμένες στο αντικειμενοστραφές μοντέλο προγραμματισμού [28]. Στη διάρκεια των ετών 1989 - 1994 ξεκινώντας με λιγότερες από 10 μεθοδολογίες εμφανίστηκαν περισσότερες από 50 αντικειμενοστραφείς μέθοδοι σχεδίασης. Από αυτές τρεις ήταν οι επικρατέστερες:

- Η μέθοδος του Booch
- του Jacobson - OOSE (Object Oriented Software Engineering)
- και του Rumbaugh - OMT (Object Modeling Technique)

Κάθε μέθοδος είχε τα δικά της πλεονεκτήματα και μειονεκτήματα ενώ η πολυφωνία παρά την ελεύθερη διακίνηση ιδεών στο πεδίο της μοντελοποίησης απέτρεπε την υιοθέτηση μίας κοινώς αποδέκτης μεθοδολογίας. Η λύση δόθηκε όταν οι Booch, Jacobson και Rumbaugh αποφάσισαν να συνεργαστούν, με σκοπό την ενοποίηση των μεθοδολογιών τους, αξιοποιώντας τα πλεονεκτήματα της κάθε μίας. Ο στόχος της ενοποίησης ήταν η δημιουργία μίας γλώσσας μοντελοποίησης:

- για μοντελοποίηση συστημάτων από την στιγμή της σύλληψης μέχρι την υλοποίηση του παραδοτέου με χρήση αντικειμενοστραφών τεχνικών
- για μοντελοποίηση μεγάλων και πολύπλοκων συστημάτων
- κατανοητής τόσο από ανθρώπους όσο και από μηχανές

Η πρώτη έκδοση της UML έκανε την εμφάνισή της τον Ιανουάριο του 1997. Το OMG (στο δίκτυο [9]) μέσα από ομάδες εργασίας (RTF) εξελίσσει διαρκώς τις προδιαγραφές της UML, η οποία βρίσκεται σήμερα στην έκδοση 2 [13].

5.3 Γιατί μοντελοποιούμε ;

5.3.1 Η σημασία ενός μοντέλου

Υπάρχουν αρκετοί λόγοι για να κατασκευάσουμε το μοντέλο του συστήματος που πρόκειται να υλοποιήσουμε [28]:

- Η μοντελοποίηση είναι μία γενικώς αποδεκτή τεχνική.
- Ένα μοντέλο είναι απλούστευση της πραγματικότητας.
- Κατασκευάζουμε μοντέλα για να κατανοήσουμε καλύτερα τα συστήματα που κατασκευάζουμε.
- Κατασκευάζουμε μοντέλα σύνθετων συστημάτων γιατί δεν μπορούμε να κατανοήσουμε τέτοια συστήματα στο ενιαίο σύνολό τους.

Το τελευταίο είναι ίσως το σημαντικότερο αν λάβουμε υπ' όψη ότι ειδικά τα συστήματα λογισμικού είναι εγγενώς πολύπλοκα [27]:

- υπάρχει πολυπλοκότητα στο πεδίο της εφαρμογής
- υπάρχει δυσκολία στη διαχείριση της διαδικασίας ανάπτυξης
- υπάρχει μεγάλη ευελιξία μέσω του λογισμικού
- υπάρχουν προβλήματα στο χαρακτηρισμό της συμπεριφοράς των διακριτών συστημάτων

Επειδή δεν υπάρχει τρόπος να εξαλείψουμε αυτής της μορφής την πολυπλοκότητα, η μοντελοποίηση μας παρέχεται σαν ένα εργαλείο για την συστηματική αντιμετώπισή της. Επιπλέον μπορούμε να εξασφαλίσουμε ένα αποδεκτό τρόπο επικοινωνίας μέσω του μοντέλου μεταξύ του συνόλου των εμπλεκόμενων στην υλοποίηση του συστήματος, ώστε να είναι κατανοητό από όλους τι και πώς πρόκειται να κατασκευαστεί.

5.3.2 Αρχές Μοντελοποίησης

Ενώ η μοντελοποίηση υπόσχεται να λύσει κάποια προβλήματα δεν πρέπει να ξεχνάμε ότι:

- Η επιλογή των μοντέλων που θα κατασκευάσουμε έχουν σαφή επίδραση στο τρόπο με τον οποίο αντιμετωπίζεται ένα πρόβλημα και στη λύση που υλοποιείται.
- Κάθε μοντέλο έχει πολλά επίπεδα ακριβείας.
- Τα καλύτερα μοντέλα έχουν άμεση σχέση με την πραγματικότητα
- Κανένα μοντέλο δεν είναι από μόνο του επαρκές. Κάθε πολύπλοκο σύστημα προσεγγίζεται καλύτερα από ένα μικρό σύνολο σχεδόν ανεξάρτητων μεταξύ τους μοντέλων.

Έτσι ένα μοντέλο το οποίο δεν κατασκευάζεται σωστά μπορεί να δίνει λύση σε ένα διαφορετικό πρόβλημα, οπότε δεν μας εξυπηρετεί, ή ακόμα χειρότερα να περιπλέξει το πρόβλημα που ήδη αντιμετωπίζουμε. Η μοντελοποίηση δεν πρέπει να γίνεται αυτοσκοπός: Σκοπός είναι να κατανοήσουμε το σύστημα ή το πρόβλημα που εξετάζουμε και όχι να κατασκευάσουμε πολλά και σύνθετα μοντέλα. Η UML και η αντικειμενοστραφής προσέγγιση υποστηρίζουν εξελικτική κατασκευή των μοντέλων των διαφόρων συστημάτων. Έτσι μπορούμε ξεκινώντας από ένα απλό και περισσότερο αφηρημένο μοντέλο, μέσα από σταδιακές επεκτάσεις να προσεγγίσουμε ένα ιδιαίτερα πολύπλοκο και σύνθετο σύστημα.

5.3.3 Αντικειμενοστραφής Προσέγγιση

Σε αντίθεση με άλλα μοντέλα προγραμματισμού όπως ο προστακτικός, ο λογικός, ο συναρτησιακός κ.λ.π., που έχουν προταθεί στο παρελθόν, το αντικειμενοστραφές μοντέλο ανάλυσης και σχεδιασμού εφαρμογών [27], θεωρεί ότι ένα σύστημα μπορεί να αποδομηθεί σε επιμέρους αυτοτελή αντικείμενα, τα οποία μπορούμε να αντιμετωπίσουμε ανεξάρτητα μεταξύ τους και μέσω της αλληλεπίδρασής τους, να προσεγγίσουμε τη λειτουργία του αρχικού συστήματος. Ένα αντικείμενο περιγράφεται από την κατάστασή του (State) η οποία

αντικατοπτρίζεται στις τιμές που έχουν διάφορες ιδιότητές του, τη συμπεριφορά του (Behavior) που αντικατοπτρίζεται στις μεθόδους που υλοποιεί και την ταυτότητά του (Identity), η οποία το διαχωρίζει από άλλα αντικείμενα.

Για παράδειγμα ένα ξυπνητήρι δείχνει την ώρα, η οποία αντιστοιχεί στην εσωτερική του κατάσταση, μπορούμε να προγραμματίσουμε πότε θα χτυπήσει, γεγονός που αντιστοιχεί στην συμπεριφορά του, αφού μας παρέχεται με κάποιο τρόπο η δυνατότητα του προγραμματισμού και τέλος αν πρόκειται για το δικό μου ξυπνητήρι έχει μία ταυτότητα που το διαχωρίζει από τα υπόλοιπα.

Η αντικειμενοστραφής προσέγγιση βασίζεται στις ακόλουθες έννοιες:

- Αφαιρετικότητα (abstraction): Κάθε αντικείμενο μπορεί να περιγραφεί στη βάση γενικότερων αρχών που διέπουν τη λειτουργία του
- Ενθυλάκωση (encapsulation): Τα αντικείμενα δεν επιτρέπουν στον έξω κόσμο να αλληλεπιδράσουν με τον μηχανισμό που διέπει την εσωτερική τους λειτουργία
- Τμηματικότητα (modularity): Κάθε αντικείμενο μπορεί να αποτελεί το αποτέλεσμα της σύνθεσης άλλων απλούστερων αντικειμένων
- Ιεραρχία (hierarchy): Τα αντικείμενα μπορούν να καταταχθούν σε κλάσεις ιεραρχίας στις οποίες ομαδοποιούνται τα κοινά χαρακτηριστικά τους
- Τύπους (typing): Κάθε αντικείμενο μπορεί να έχει ένα τύπο
- Ταυτοχρονισμό (concurrency): Κάθε αντικείμενο μπορεί να λειτουργεί παράλληλα με άλλα αντικείμενα
- Διατήρηση (persistence): Κάθε αντικείμενο μπορεί να αποθηκεύει την κατάστασή του ώστε να παρατείνει τη διάρκεια της ζωής του πέρα από τα στενά πλαίσια της εκτέλεσης ενός προγράμματος

Μπορούμε έτσι να κατατάξουμε τα αντικείμενα που αφορούν τη λειτουργία ενός συστήματος σε ιεραρχίες κλάσεων. Κάθε κλάση περιγράφει ομοιόμορφα τα χαρακτηριστικά και την συμπεριφορά μίας ομάδας ομοειδών αντικειμένων πετυχαίνοντας έτσι μία ομοιόμορφη περιγραφή του συστήματος. Οι κλάσεις

μπορούν να σχετίζονται μεταξύ τους με διάφορους τρόπους κατά περίπτωση. Συναντούμε έτσι σχέσεις:

- Γενίκευσης/Ειδίκευσης - Κληρονομικότητας (Generalization/specialization – Inheritance “is a”): Μία κλάση περιλαμβάνει τα χαρακτηριστικά μίας άλλης με κάποιες ενδεχομένως αλλαγές: για παράδειγμα η κλάση «Αυτοκίνητα» είναι μία ειδίκευση της κλάσης «Οχήματα» η οποία περιλαμβάνει μόνο τα οχήματα που έχουν πόρτες και περισσότερες από τρεις ρόδες...
- Μέρους/Συνόλου (whole/part – Aggregation “part of”): Όταν τα αντικείμενα μίας κλάσης απαιτούν την ενσωμάτωση ενός αριθμού αντικειμένων από μία άλλη κλάση τότε μιλάμε για σχέσεις περιεχομένου. Για παράδειγμα ένα αυτοκίνητο έχει μία μηχανή και τέσσερις ρόδες. Η μηχανή και οι ρόδες ανήκουν στις αντίστοιχες κλάσεις τους, όμως κανείς δεν μπορεί να ισχυριστεί ότι μία μηχανή ή μία ρόδα «είναι» από μόνα τους αντικείμενα της κλάσης «Αυτοκίνητο»
- Συσχέτισης (Association): Υποδηλώνει κάποιου είδους εξάρτηση. Εδώ μπορεί να υποδηλωθεί μία οποιοδήποτε άλλου τύπου εξάρτηση μεταξύ των αντικειμένων της ίδιας ή διαφορετικών κλάσεων.

5.3.4 Η UML σαν εργαλείο οπτικοποίησης - παρουσίασης

Η οπτικοποιημένη μοντελοποίηση ενός συστήματος γενικότερα, αλλά και του σχήματος μίας βάσης δεδομένων ειδικότερα, μας βοηθάει να κατανοήσουμε και να περιγράψουμε με μεγαλύτερη ακρίβεια το περιεχόμενο του [22]. Μας βοηθάει επίσης να δαμάσουμε την πολυπλοκότητα του προβλήματος, προωθεί την ανταλλαγή και την αξιολόγηση ιδεών, βοηθάει στη διαδικασία προγραμματισμού και στην ευκολία της συντήρησης του τελικού συστήματος.

5.4 Τα διαγράμματα της UML

Στο [28] περιγράφονται αναλυτικά και τα εννιά διαγράμματα που υποστηρίζονται από την UML. Κάθε ένα από τα διαγράμματα αυτά μπορεί να απει-

κονίσει μία όψη του υπό κατασκευή συστήματος και σε συνδυασμό μεταξύ τους, το σύστημα στο σύνολό του. Τα διαγράμματα χωρίζονται σε δύο ομάδες από τις οποίες, όπως φαίνεται και στη συνέχεια, η πρώτη σχετίζεται με τα διαγράμματα που αφορούν τη στατική εικόνα του συστήματος ενώ η δεύτερη σχετίζεται με τα δυναμικά χαρακτηριστικά του:

- Δομικά Διαγράμματα (Structural Diagrams):
 - Ιεραρχίας Κλάσεων (Class Diagram)
 - Αντικειμένων (Object Diagram)
 - Συστατικών (Component Diagram)
 - Παράταξης-Ανάπτυξης (Deployment Diagram)
- Δυναμικά Διαγράμματα (Behavioral Diagrams):
 - Περιπτώσεων Χρήσης (Use Case Diagram)
 - Διαδοχής (Sequence Diagram)
 - Συνεργασίας (Collaboration Diagram)
 - Κατοστάσεων (Statechart Diagram)
 - Δράσης (Activity Diagram)

Η UML περιγράφει εκτός των άλλων πώς πρέπει να κατασκευάζονται τα διαγράμματα αυτά καθώς και τη σημασία των συστατικών κάθε διαγράμματος. Η ύπαρξη ενός κατάλληλου εργαλείου για την κατασκευή και την οπτικοποίηση των διαγραμμάτων αυτών είναι σημαντική, αλλά πολλές φορές για απλά συστήματα αρκεί χαρτί και μολύβι για τον σχεδιασμό τους.

5.5 UML και Γεωγραφικά Συστήματα Πληροφοριών

Από την πλευρά της κατασκευής πληροφοριακών συστημάτων, τα γεωγραφικά συστήματα πληροφοριών δεν αποτελούν παρά ένα ακόμα πολύπλοκο

σύστημα λογισμικού. Είναι φυσικό λοιπόν, οι κλασικές τεχνικές μοντελοποίησης που έχουν εφαρμοστεί γενικότερα στα πληροφοριακά συστήματα να βρουν εφαρμογή και στο πεδίο αυτό. Οι ιδιαιτερότητες βέβαια ενός γεωγραφικού πληροφοριακού συστήματος εστιάζονται στην ανάγκη για μοντελοποίησης κυρίως της γεωγραφικής πληροφορίας και την απαίτηση για διαχείριση μεγάλου όγκου δεδομένων. Οι τεχνικές μοντελοποίησης που υπάρχουν για τα κλασικά πληροφοριακά συστήματα, όπως για παράδειγμα το μοντέλο οντοτήτων συσχετίσεων με το οποίο μπορεί να περιγραφεί αρκετά καλά το σχήμα μίας σχεσιακής βάσης δεδομένων, δεν μπορεί αυτούσιο να χρησιμοποιηθεί για την περιγραφή του σχήματος μίας γεωγραφικής βάσης δεδομένων χωρίς κάποιες αναγκαίες επεκτάσεις.

Προς την κατεύθυνση αυτή έχουν γίνει πολλές προσπάθειες από τις οποίες άλλες βασίζονται σε παλιότερες τεχνικές και άλλες σε επεκτάσεις που αφορούν την ίδια την UML. Για παράδειγμα στο [88] προτείνεται το GISER σαν επέκταση του EER ώστε να συμπεριλαμβάνει συνεχή πεδία (Continuous Fields). Στα [30] και [31] παρουσιάζεται μία προσέγγιση ενός αντικειμενοστραφούς μοντέλου για γεωγραφικές εφαρμογές βασισμένο στο OMT, το OMT-G. Επίσης γίνεται μία εκτενής αναφορά σε ζητήματα κανόνων ορισμού της ακεραιότητας των χωρικών δεδομένων, με σκοπό τη βελτίωση της ποιότητας των δεδομένων που εισάγονται και αποθηκεύονται σε μία γεωγραφική βάση δεδομένων.

Στο [73] προτείνεται η τεχνική MADS για την μοντελοποίηση χωροχρονικών οντοτήτων στο λογικό επίπεδο (Conceptual Level). Αναφέρεται ότι ένα καλό μοντέλο λογικού επιπέδου πρέπει να περιλαμβάνει ισχυρές δομές, απλό γραφικό συμβολισμό, ένα επίσημο ορισμό και μία γλώσσα διαχείρισης δεδομένων. Και τα τρία προηγούμενα παραπέμπουν στην αξιοποίηση της UML. Στο [68] περιγράφεται ένα εργαλείο και κάποιες τεχνικές για την μετατροπή και αντιστοίχιση των δεδομένων χαρτών από μία διανυσματική αναπαράσταση σε μία αντικειμενοστραφή αναπαράσταση. Στο [22] γίνεται μία αναδρομή στις διάφορες μεθοδολογίες μοντελοποίησης χωρικών βάσεων δεδομένων και η εισαγωγή της έννοιας Spatial PVL. Στο [100] προτείνεται μία επέκταση του E-R model για την μοντελοποίηση χωροχρονικών εφαρμογών, ενώ στο [78] αντίστοιχα μία επέκταση της UML για την περιγραφή χωροχρονικών δεδομένων στο εννοιολογικό επίπεδο. Τέλος προγενέστερες προσεγγίσεις βρίσκει

κανείς στο [99].

Μία άλλη προσέγγιση που σχετίζεται με τη μοντελοποίηση συναντούμε στα [45], [44], [43] και [76]. Εκεί περιγράφεται η αξιοποίηση σχεδιαστικών προτύπων (Design Patterns) για τη μοντελοποίηση και την ανάπτυξη εφαρμογών γεωγραφικών συστημάτων πληροφοριών. Εδώ δεν αναγνωρίζονται μόνο κλάσεις αντικειμένων που απαρτίζουν το υπό σχεδιασμό σύστημα, αλλά επιπλέον πρότυπα που αφορούν τη δομή και την λειτουργία του συστήματος γενικότερα. Αντίστοιχα, στα [40] και [72] παρουσιάζεται μία αντίστοιχη τεχνική βασισμένη σε πρότυπα ανάλυσης (Analysis Patterns) για το σχεδιασμό χωρικών και χωροχρονικών βάσεων δεδομένων. Γενικά πάντως δεν υπάρχει μία συνολική και ευρύτερα αποδεκτή προσέγγιση για την εννοιολογική μοντέλοποίηση του γεωγραφικού χώρου [56].

5.6 Εφαρμογές

Όπως είδαμε η UML επιτρέπει την μοντελοποίηση συστημάτων λογισμικού. Στο [67], γίνεται μία πλήρης περιγραφή ενός συστήματος παρακολούθησης ασθενών, με χρήση της UML. Ουσιαστικά πρόκειται για μία πλήρη αξιοποίηση των δυνατοτήτων της γλώσσας μοντελοποίησης για το σχεδιασμό και την ανάπτυξη ενός πληροφοριακού συστήματος, χωρίς όμως να περιλαμβάνει αυτό γεωγραφική πληροφορία.

Στο [79] περιγράφεται η χρήση του Rational Rose, ενός εργαλείου σχεδίασης αντίστοιχου με το Visio που χρησιμοποιούμε για τη σχεδίαση της βάσης δεδομένων στην πλατφόρμα *Icaros.NET*, ενώ στο [28] περιγράφεται η εφαρμογή και η χρήση της UML.

Σύμφωνα με το [32], υπάρχουν πολλοί λόγοι για την υιοθέτηση της UML σαν γλώσσας περιγραφής της βάσης δεδομένων: Αναγνωρίζεται σαν *de facto* πρότυπο στο πεδίο της τεχνολογίας λογισμικού και αποτελεί πρότυπο για το OMG. Επίσης η ομάδα εργασίας ISO/TC 211 η οποία ασχολείται με θέματα προτυποποίησης σχετικών με τα γεωγραφικά συστήματα πληροφοριών ([6]) καθώς και το OGC ([10]) έχουν επιλέξει την UML σαν γλώσσα για την μοντελοποίηση γεωγραφικών δεδομένων. Επιπλέον η UML προσφέρει εκτός των άλλων και ένα μηχανισμό επέκτασής της, ο οποίος είναι χρήσιμος στη

περίπτωση των ΣΓΠ.

Στο [32], περιγράφεται συνοπτικά η πρόοδος που έχει γίνει στον τομέα των γεωγραφικών πληροφοριακών συστημάτων σχετικά με τα πρότυπα ορισμού της περιγραφής χωροχρονικών δεδομένων και τη χρησιμότητα των αποθηκών (Repositories) για την προσωρινή αποθήκευση της περιγραφής του σχήματος μίας γεωγραφικής βάσης δεδομένων, με εφαρμογή στο εργαλείο CASE Perceptory. Το συγκεκριμένο εργαλείο περιγράφεται πλήρως στο [23].

5.7 Πώς αξιοποιείται στο *Icaros.NET*;

Για τις ανάγκες της κατασκευής του επιπέδου δεδομένων της πλατφόρμας *Icaros.NET*, χρησιμοποιήσαμε στατικά διαγράμματα κλάσεων για την περιγραφή του σχήματος της βάσης δεδομένων. Η ESRI παρέχει τη δυνατότητα για την κατασκευή του σχήματος μίας γεωγραφικής βάσης δεδομένων εξολοκλήρου στο γραφικό περιβάλλον του Visio, με χρήση της UML. Εκτεταμένη περιγραφή γίνεται στα [1] και [61].

Από το στατικό διάγραμμα κλάσεων παράγεται αυτοματοποιημένα μία κενή γεωγραφική βάση δεδομένων έτοιμη να δεχτεί τα αντικείμενα των κλάσεων που περιγράφονται στο UML μοντέλο. Είναι σημαντικό ότι αν κάποιος γνωρίζει τη σημασιολογία των διαγραμμάτων, μπορεί να κατανοήσει τα δεδομένα που απαιτεί και πρόκειται να διαχειριστεί η εφαρμογή κατά τη λειτουργία της. Επίσης η UML μας εξασφαλίζει ένα τρόπο περιγραφής και τεκμηρίωσης του σχήματος της βάσης δεδομένων ανεξάρτητο από το ΣΔΒΔ στο οποίο πρόκειται να γίνει τελικά η υλοποίηση.

Κεφάλαιο 6

Επεκτάσιμη Γλώσσα Σήμανσης (XML)

6.1 Τί είναι;

Η XML είναι, όπως υποδηλώνει και το πλήρες όνομά της (eXtensible Markup Language), είναι μία γλώσσα σήμανσης - περιγραφής των δεδομένων που περιέχονται σε ένα αρχείο. Ειδικότερα μπορεί κανείς να ορίσει τη δομή που θα έχει ένα αρχείο XML, προσδιορίζοντας τη μορφή των ετικετών και το τύπο των δεδομένων τους. Παρέχεται έτσι ένας πολύ ευέλικτος τρόπος για την ανταλλαγή δεδομένων βασισμένων σε μία δομή αρχείου XML, η οποία ανταποκρίνεται στην εκάστοτε εφαρμογή. Στην περίπτωση μάλιστα που έχουμε να αντιμετωπίσουμε πολλές διαφορετικές πηγές πληροφορίας, η υιοθέτηση τεχνικών της XML, αποτελεί μία καλή λύση για την ενοποίηση και την ομοιόμορφη διαχείριση των δεδομένων.

6.2 Ιστορικά

Η XML αποτελεί υποσύνολο της γενικότερης και περισσότερο πολύπλοκης SGML. Υποσύνολο της SGML είναι και η γλώσσα HTML η οποία χρησιμοποιείται για την κατασκευή δικτυακών σελίδων και μοιάζει πολύ στη δομή της με την XML. Δύο όμως είναι τα κυριότερα χαρακτηριστικά που διαφορο-

ποιούν τις δύο αυτές διαλέκτους της SGML [89]:

- Η πρώτη έχει προκαθορισμένες ετικέτες, ενώ η δεύτερη προϋποθέτει τον ορισμό από τον προγραμματιστή του είδους και της μορφής των ετικετών που θα χρησιμοποιηθούν για τη δόμηση των δεδομένων.
- Η πρώτη περιγράφει την μορφή με την οποία θα απεικονιστούν τα δεδομένα σε ένα φυλλομετρητή (Web Browser) με στόχο την ομοιόμορφη παρουσίαση τους, ενώ η δεύτερη περιγράφει τα ίδια τα δεδομένα σε μία ευέλικτη και «Self-describing» μορφή ανεξάρτητη από την εφαρμογή που πρόκειται να τα αξιοποιήσει.

Τόσο η XML όσο και άλλες Markup Languages αναπτύσσονται από το World Wide Web Consortium [12]. Εκεί μπορεί να βρει κανείς τις τελευταίες εξελίξεις αλλά και τις πλήρεις περιγραφές για σχετικές τεχνολογίες και ειδικότερα για την XML [4].

Μία εφαρμογή της XML που αφορά τα γεωγραφικά συστήματα πληροφοριών αποτελεί η GML, η οποία αποτελεί ουσιαστικά μία κωδικοποίηση XML του OGC για το SFS. Το [38], αποτελεί τον ορισμό για την GML. Προτείνεται σαν το πρότυπο για την περιγραφή τέτοιου είδους πληροφορίας και στις μέρες μας λαμβάνει όλο και μεγαλύτερης αποδοχής από διαφορετικούς κατασκευαστές λογισμικού.

Στο [113] μάλιστα, γίνεται μία προσπάθεια επέκτασης της GML ώστε να συμπεριλάβει και χρονικές δομές. Περιγράφεται πώς σύνθετες χρονικές δομές μπορούν να περιγραφούν με τη χρήση ενός XML-Schema και πώς μπορεί αυτό να συνδυαστεί με την GML. Η GML πρόκειται να αλλάξει την ικανότητα των οργανισμών να μοιράζονται γεωγραφικές πληροφορίες μεταξύ τους και να ενεργοποιήσουν συνδεδεμένα γεωγραφικά σύνολα δεδομένων [89]. Θέτει τις βάσεις για ένα ενιαίο τρόπο περιγραφής και ανταλλαγής γεωγραφικών δεδομένων.

Άλλες προσεγγίσεις εκτός της XML για την περιγραφή χωρικών δεδομένων βασίζονται σε μοντέλα «ημιδομημένων» δεδομένων (Semi-structured Data Models) [95]. Είναι δενδρικές δομές, παρόμοιες με αυτές της XML, όμως απλούστερες και στηρίζονται σε αντιστοιχίσεις ετικετών - τιμών.

6.3 Πού χρησιμοποιείται;

Περίπου πέντε χρόνια μετά την εμφάνιση της XML, οι εφαρμογές που βασίζονται σε συναφείς τεχνολογίες είναι απεριόριστες. Η γλώσσα έχει μία μεγάλη δυναμική και είναι ευρύτατα αποδεκτή με αποτέλεσμα να επηρεάζει την πορεία πολλών πεδίων που σχετίζονται με τη διαχείριση δεδομένων [81]. Ξεκινώντας από τα συστήματα διαχείρισης βάσεων δεδομένων πολλά από τα οποία έχουν ενσωματώσει δυνατότητες επεξεργασίας και παραγωγής αρχείων και δομών XML, έως και εφαρμογές γραφείου που βασίζονται σε αυτή για την περιγραφή των αρχείων τους, ο κατάλογος δεν έχει τέλος. Τεχνολογίες όπως τα Web Services βασίζονται εξ' ολοκλήρου για τη λειτουργία τους σε αρχεία XML. Επίσης το ηλεκτρονικό εμπόριο έχει πάρει μία μεγάλη ώθηση με την αξιοποίηση τεχνολογιών XML.

Η δυνατότητα των αρχείων XML να μπορούν να είναι αυτο-περιγραφικά (στο ίδιο αρχείο βρίσκεται και ο ορισμός της δομής των δεδομένων αλλά και τα ίδια τα δεδομένα), οδηγεί στην αξιοποίησή της όπου παλιότερα είχαμε απλά αρχεία κειμένου ή ακόμα και αρχεία που ακολουθούσαν μία πιο σύνθετη κωδικοποίηση. Έτσι, για παράδειγμα, αρχεία με πληροφορίες για την αρχειοποίηση διαφόρων εφαρμογών έχουν αντικατασταθεί από άλλα τα οποία βασίζονται στην XML. Το σημαντικότερο πλεονέκτημα είναι ότι αντί να υπάρχει μία, πολλές φορές, ασαφής περιγραφή των περιεχομένων ενός αρχείου σε κάποιο εγχειρίδιο, αυτή αντικαθίσταται από τον έναν ορισμό βασισμένο στην XML, με βάση τον οποίο μπορούν να γίνουν έλεγχοι ορθότητας χωρίς την ύπαρξη αμφιβολιών για το ποιόν των δεδομένων ή ακόμα και να εξαχθεί με ένα συστηματικό τρόπο χρήσιμη πληροφορία. Μπορούμε έτσι εύκολα να εξαλείψουμε όλα τα προβλήματα που συναντούσε κανείς σε ένα αρχείο κειμένου με τιμές χωρισμένο με κόμμα (CSV - Comma Separated Values αρχείο), ή ακόμα και σε ένα αρχείο προκαθορισμένου μήκους στηλών (Fixed Column Width αρχείο).

Ένα ακόμα σημαντικό πλεονέκτημα από τη χρήση της XML είναι ότι μία σωστή επιλογή για την περιγραφή των δεδομένων μπορεί να δώσει πληροφορίες για τα περιεχόμενα ενός αρχείου σε κάποιον που το επεξεργάζεται με έναν απλό text editor, αφού να ονόματα των ετικετών που επιλέγονται μπορεί

να ακολουθούν κανόνες παρόμοιους με εκείνους που έχουμε συνηθίσει για τις μεταβλητές ενός προγράμματος, για παράδειγμα σε C.

6.4 Πώς υλοποιείται;

Η περιγραφή των δεδομένων που υπάρχουν σε ένα XML αρχείο είναι δυνατή λόγω της ύπαρξης του DTD – Document Type Definition ή του XML Schema[89]. Το DTD ή το XML – Schema είναι πάλι αρχεία XML τα οποία όμως υπακούουν σε συγκεκριμένους κανόνες με σκοπό την περιγραφή της δομής άλλων XML αρχείων. Η βασική διαφορά τους είναι ότι η πρώτη μορφή υποστηρίζεται από την αρχή της ύπαρξης της XML και είναι απλούστερης μορφής. Η δεύτερη υποστηρίζει εκτός από τον ορισμό ετικετών και την επιβολή του τύπου των δεδομένων που μπορεί να υποστηρίξει μία ετικέτα (π.χ.: συμβολοσειρά ή αριθμός). Η τελευταία αυτή δυνατότητα είναι ιδιαίτερα χρήσιμη στην περίπτωση που έχουμε να κάνουμε με μεγάλο όγκο δεδομένων και θέλουμε να εξασφαλίσουμε την ορθότητά τους σχετικά με το τύπο τους.

Τον έλεγχο ενός αρχείου XML τον υλοποιεί ο XML parser. Αυτός είναι στην ουσία μία βιβλιοθήκη διαδικασιών που μπορεί να συνεργαστεί με διάφορες γλώσσες προγραμματισμού, όπως Visual Basic, Java, C++ κ.λ.π., έτσι ώστε να μπορέσουν οι γλώσσες αυτές να επεξεργαστούν τα δεδομένα αρχείων XML. Για παράδειγμα, στο [52] δίνεται μία πλήρης περιγραφή της γλώσσας XML, με έμφαση τον προγραμματισμό σε Java και τον ορισμό μέσω ενός DTD. Δίνονται επίσης διάφορα παραδείγματα εφαρμογής της XML.

6.4.1 Ο XML parser

Υπάρχουν γενικά δύο μεγάλες κατηγορίες XML parsers:

- Validating Parsers: Στηρίζονται στην ύπαρξη ενός ορισμού για το XML αρχείο DTD/Schema και για την επιτυχή επεξεργασία απαιτείται το αρχείο να συμμορφώνεται με τον ορισμό αυτό.
- Non Validating Parsers: Δεν απαιτείται αντίστοιχο αρχείο ορισμού και γίνεται απλώς έλεγχος, αν το υπό εξέταση αρχείο συμμορφώνεται με

τους γενικότερους κανόνες δόμησης που υπαγορεύονται από την XML

Στην πρώτη περίπτωση λέμε ότι το αρχείο μετά από επιτυχή επεξεργασία είναι Well-formed και Valid ενώ στη δεύτερη περίπτωση μπορούμε να ισχυριστούμε μόνο το δεύτερο.

Ο MSXML parser παρέχεται από την Microsoft [7] και η υλοποίηση υποστηρίζει το XML-Schema [5] που χρησιμοποιούμε για την περιγραφή των δεδομένων μας στη πλατφόρμα *Icaros.NET*. Στο [91] παρουσιάζονται συνοπτικά οι δυνατότητες του Microsoft MSXML 4.0 parser.

6.4.2 SAX και DOM

Δύο είναι η περισσότερο διαδεδομένες τεχνικές για την επεξεργασία ενός αρχείου XML [89]:

- SAX Είναι ιδιαίτερα αποδοτική στην περίπτωση που αρκεί ένα απλό διάβασμα των δεδομένων
- DOM Αξιοποιείται καλύτερα όταν απαιτείται διάβασμα πολλών παραματάων των αρχείων XML. Στην περίπτωση αυτή το δέντρο που αντιστοιχεί στην δομή της XML ανακατασκευάζεται στην μνήμη και οι επεμβάσεις, αναζητήσεις και μεταβολές γίνονται ταχύτερα.

Εμείς αξιοποιούμε την δεύτερη τεχνική διότι ενδείκνυται για την περίπτωση που απαιτούνται ερωτήσεις που απαιτούν πολλαπλά διαβάσματα για τα δεδομένα ή την σύνθεση πολλών αρχείων.

6.4.3 Αποθήκευση αρχείων

Εκτός από την περιγραφή των δεδομένων, γίνεται έρευνα και στο επίπεδο της ανάκτησης δεδομένων. Για παράδειγμα στο [37], γίνεται προσπάθεια για την ανάπτυξη μία γλώσσας χωρικών ερωτήσεων πάνω σε αρχεία GML/XML και μελέτη αποδοτικών τρόπων αποθήκευσης των αρχείων αυτών.

Οι τεχνικές που υπάρχουν για την αποθήκευση αρχείων XML σε μία βάση δεδομένων χωρίζονται σε δύο μεγάλες κατηγορίες :

- δομημένης αντιστοίχισης (Structured-mapping), όπου ο σχεδιασμός της βάσης δεδομένων αντιστοιχεί στο DTD ή το XML-Schema
- αντιστοίχισης μοντέλου (Model-mapping), όπου ο σχεδιασμός της βάσης αντιστοιχεί σε ένα σχήμα ικανού να αποθηκεύσει οποιοδήποτε αρχείο XML

Η χρήση ενός υπάρχοντος συστήματος διαχείρισης βάσεων δεδομένων για την αποθήκευση των GML/XML αρχείων αξιοποιεί όλη την υπάρχουσα υποδομή για την διαχείριση των δεδομένων. Το μειονέκτημα είναι ότι οι απαιτήσεις για την επεξεργασία αρχείων XML είναι λίγο διαφορετικές από αυτές που απαιτούνται για τα δομημένα σχεσιακά δεδομένα.

6.5 Τί δυνατότητες δίνει;

Η βασική δυνατότητα από τη χρήση της XML είναι η κατασκευή ανοιχτών συστημάτων. Στο [50] γίνεται αναφορά σε συστήματα ανοιχτών προδιαγραφών και στις τεχνολογίες παροχής δικτυακών δυνατοτήτων (Open Standards and Enabling Technologies). Με το τελευταίο εννοούνται τεχνολογίες οι οποίες παρέχουν προκαθορισμένους τρόπους για την πρόσβαση σε δεδομένα, την ανάκτηση αυτών και τη λειτουργία μέσω ενός δικτύου. Πρόκειται για μία αναφορά από την ESRI, αρκετά ενδιαφέρουσα γιατί αφορά και την δική μας πλατφόρμα ανάπτυξης για το *Icaros.NET*, αφού ήδη από το 1994 γίνεται ένας προΐδεασμός για τις δυνατότητες αξιοποίησης της UML και της XML στα ΣΓΠ. Συνοπτικά αναφέρονται ότι σαν οδηγός για την ανάπτυξη των συστημάτων που παρέχει η ESRI ακολουθούνται οι απαιτήσεις των χρηστών και αναλυτικότερα:

- Οι χρήστες πρέπει με ευκολία να εισάγουν γεωγραφικά δεδομένα στο γεωγραφικό σύστημα πληροφοριών που χρησιμοποιούν.
- Η χρήστες πρέπει να μπορούν εύκολα να βρίσκουν και να εισάγουν γεωγραφικά δεδομένα από εθνικές ή άλλες πηγές στο γεωγραφικό σύστημα πληροφοριών που χρησιμοποιούν.

- Οι χρήστες πρέπει να μπορούν να διαθέσουν τα δεδομένα που χρησιμοποιούν στο δίκτυο.
- Οι διάφοροι οργανισμοί χρειάζονται προδιαγραφές για να καταλογογραφήσουν και να οργανώσουν τα δεδομένα που κατέχουν.
- Οι χρήστες πρέπει να μπορούν να ενσωματώσουν τα δικά τους γεωγραφικά μοντέλα με τα μοντέλα του γεωγραφικού συστήματος πληροφοριών που χρησιμοποιούν.
- Οι χρήστες απαιτούν τα τρέχοντα συστήματα γεωγραφικών πληροφοριών που χρησιμοποιούν και τα μελλοντικά να συνεργάζονται καλύτερα μεταξύ τους.

Όλα τα προηγούμενα, άλλα λιγότερο και άλλα περισσότερο, αποτελούν απαιτήσεις και της εφαρμογής *Icaros.NET* που αναπτύσσουμε.

Η ίδια η ESRI χρησιμοποιεί την XML για την περιγραφή χαρτογραφικών εφαρμογών στο διαδίκτυο, οι οποίες βασίζονται στο ArcIMS [18], ενώ περισσότερες πληροφορίες για το μοντέλο αυτό δίνονται στο [17].

6.6 Πώς αξιοποιείται στο *Icaros.NET* ;

Στην πλατφόρμα *Icaros.NET* χρησιμοποιήθηκε η XML για την περιγραφή των δεδομένων που προέρχονται από εξωτερικές πηγές. Με το τρόπο αυτό εξασφαλίστηκε η ανεξαρτησία της πλατφόρμας από τις πηγές αυτές και είναι δυνατή η επέκτασή της με βάση το υπάρχον σχήμα. Αναπτύχθηκε ένα XML Schema που αντικατοπτρίζει το σχήμα της βάσης δεδομένων στο οποίο βασίζεται η πλατφόρμα.

Δεν αξιοποιήθηκε το υπάρχον μοντέλο της GML γιατί :

- Είναι ακόμα υπό ανάπτυξη και είναι περισσότερο γενικευμένο από ότι απαιτείται για τη δική μας εφαρμογή
- Σκοπός της ανάπτυξης είναι η εξοικείωση με την XML και η ανάπτυξη από την αρχή ενός XML Schema παρείχε τη δυνατότητα του πειραματισμού

- Το XML Schema ανταποκρίνεται όσο το δυνατό πιο πιστά στο σχήμα της βάσης δεδομένων γεγονός που το καθιστά μικρό και κατανοητό

Έγινε η επιλογή του DOM για την επεξεργασία των αρχείων XML, ενώ τα τελευταία αποτελούν τη γέφυρα μεταξύ της βάσης δεδομένων και του έξω κόσμου. Μετά την εισαγωγή τους στην βάση δεν είναι επιπλέον αξιοποιήσιμα. Χρησιμοποιείται έτσι ένα σχεσιακό σύστημα βάσεων δεδομένων με αντικειμενοστραφείς επεκτάσεις για την αποδοτική αποθήκευση των δεδομένων, μέσω ενός φίλτρου XML το οποίο εξασφαλίζει την ορθότητα της μορφής των δεδομένων, μειώνοντας την απαίτηση ελέγχων από την εφαρμογή στο ελάχιστο.

Κεφάλαιο 7

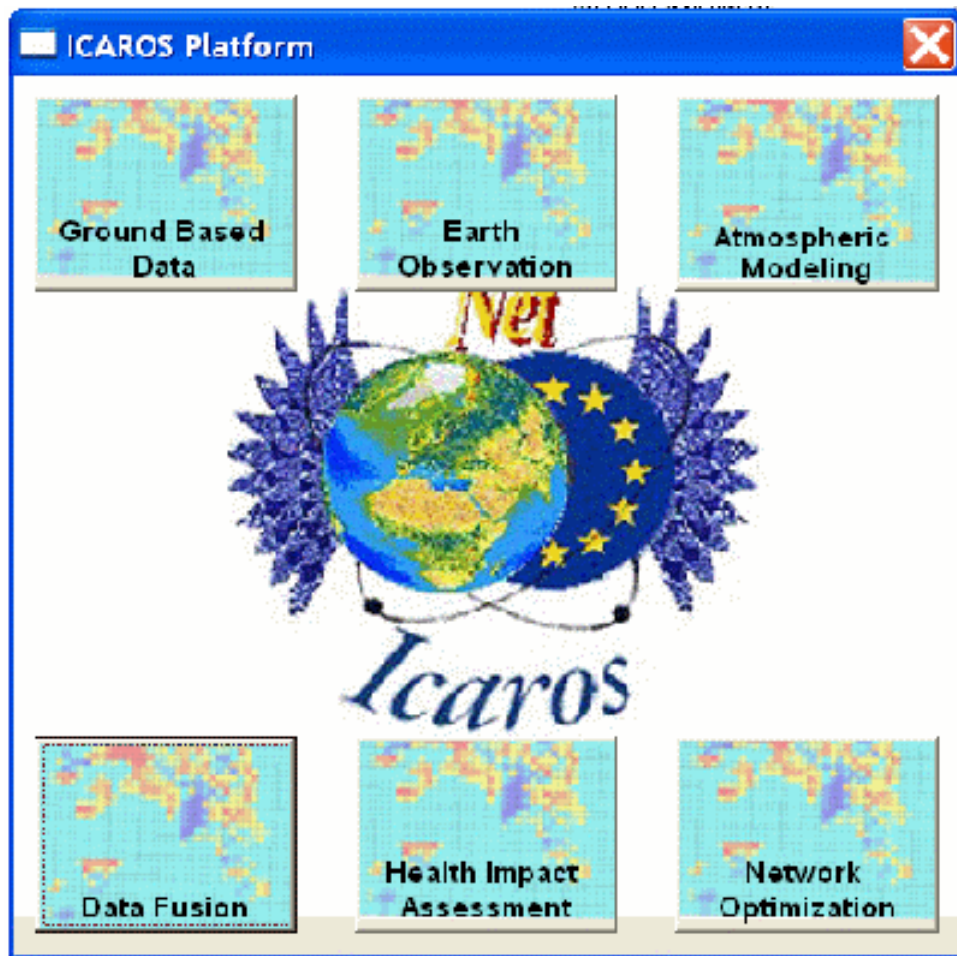
Υλοποίηση

7.1 Εισαγωγή

Στο κεφάλαιο αυτό παρουσιάζονται αναλυτικά όλα τα ζητήματα υλοποίησης τα οποία είτε προέκυψαν από τις απαιτήσεις της εφαρμογής, είτε από την επιλογή της πλατφόρμας υλοποίησης. Δίνονται επίσης οι απαραίτητες επεξηγήσεις για τις σχεδιαστικές επιλογές. Αναλυτικότερα περιγράφεται το σχήμα της γεωγραφικής βάσης δεδομένων, το σχήμα περιγραφής των αρχείων XML, η δημιουργία της βιβλιοθήκης σε μορφή Windows .dll, ο τρόπος εισόδου δεδομένων στην εφαρμογή και τέλος τα πλεονεκτήματα της προσέγγισης που ακολουθήθηκε.

7.2 Αρχιτεκτονική του Επιπέδου Διαχείρισης Δεδομένων

Η αρχιτεκτονική του υποσυστήματος διαχείρισης δεδομένων για το *Icaros.NET* αποδίδεται συνοπτικά στο σχήμα 7.2. Μπορεί να δει κανείς ότι από το UML μοντέλο της γεωγραφικής βάσης δεδομένων παράγεται η λειτουργική γεωγραφικών βάση δεδομένων για το *Icaros.NET* καθώς και το αντίστοιχο XML-Schema που περιγράφει την μορφή των δεδομένων την οποία μπορεί να δεχτεί η βάση. Η διαδικασία μετατροπής αρχείων δεδομένων στην μορφή XML καθώς και η εισαγωγή αυτών στην βάση δεδομένων γίνεται μέσω της βιβλιο-

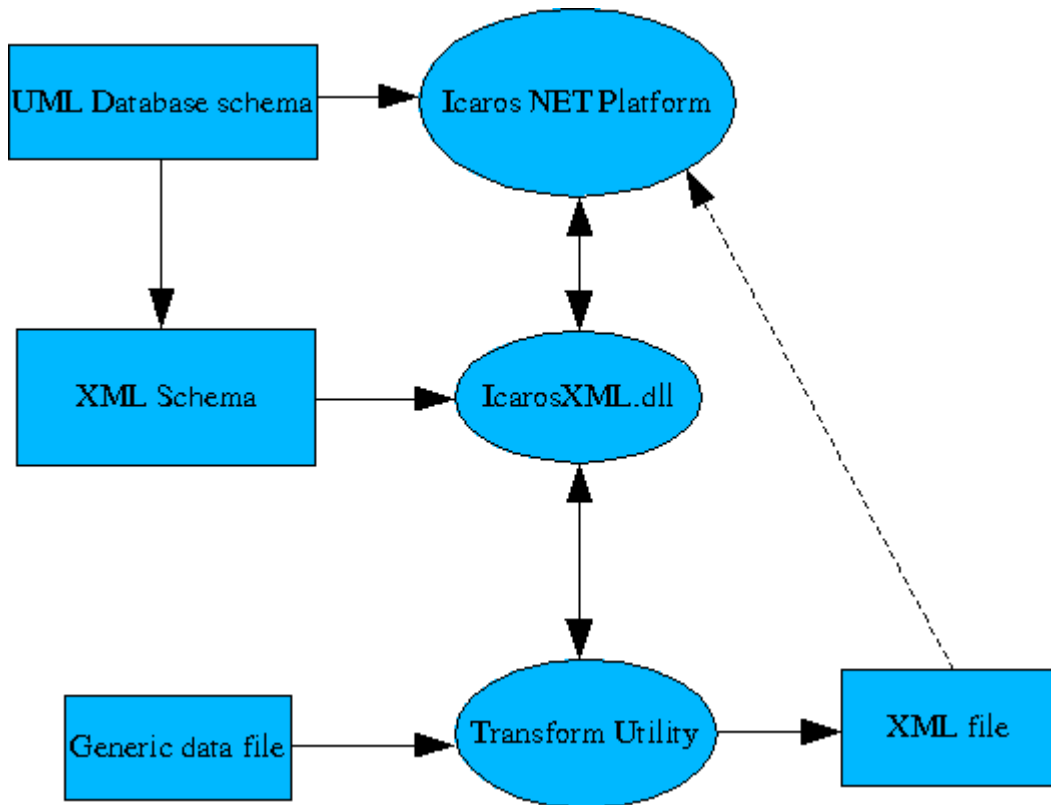


Σχήμα 7.1: Η αρχική οθόνη της πλατφόρμας *Icaros.NET*

θήκης *IcarosXML.dll*. Τα επιμέρους συστατικά που αφορούν το υποσύστημα διαχείρισης δεδομένων καθώς η μεταξύ τους αλληλεπιδράσεις περιγράφονται στην συνέχεια του κεφαλαίου.

7.3 Η γεωγραφική βάση δεδομένων

Η επιλογή της πλατφόρμας της ESRI για την ανάπτυξη του *Icaros.NET* επιβλήθηκε εκ των πραγμάτων όχι σαν καλύτερη δυνατή (μπορεί και να είναι) αλλά λόγω διαθεσιμότητας στο εργαστήριο. Επιπλέον παρέχει ένα περιβάλλον εργασίας με το οποίο είναι εξοικειωμένοι οι συνεργάτες, οι οποίοι εμπλέκον-



Σχήμα 7.2: Η λειτουργία της βιβλιοθήκης

ται στην ανάπτυξη της πλατφόρμας. Σε ότι αφορά το επίπεδο διαχείρισης δεδομένων το ArcGIS παρέχει:

- Πλούσια βιβλιοθήκη αντικειμένων COM – ArcObjects για προγραμματισμό.
- Δυνατότητες προγραμματισμού μέσα από την Visual Basic.
- Δυνατότητα συνεργασίας με διάφορες βάσεις δεδομένων.

Αξιοποιήθηκε η δυνατότητα της ανάπτυξης της γεωγραφικής βάσης δεδομένων σε τοπικό επίπεδο όπως παρέχεται από την Personal Geodatabase που είναι βασισμένη σε επεκτάσεις της ESRI πάνω στο ΣΔΒΔ της Microsoft Access. Παρά τους περιορισμούς που εισάγει η τελευταία είναι ιδανική για τις ανάγκες μας διότι:

- Είναι ικανοποιητική για πιλοτική υλοποίηση και μελλοντική επέκταση με την χρήση του ArcSDE.
- Υποστηρίζει αντικειμενοστραφή σχεδίαση με χρήση του Visio και τα πρότυπα UML που παρέχει η ESRI.
- Είναι εύκολος ο προγραμματισμός της με χρήση της Visual Basic και των αντικειμένων COM ArcObjects.
- Είναι δυνατή η αποθήκευση διανυσματικών δεδομένων (Vector) σε επιθυμητό σύστημα συντεταγμένων.
- Υποστηρίζει τον ορισμό συσχετίσεων μεταξύ δεδομένων.
- Υποστηρίζει υπότυπους (Subtypes).
- Υποστηρίζει Πεδία Ορισμού (Domains).
- Είναι εύκολη η δημιουργία του σχήματος της γεωβάσης μέσα από τα εργαλεία σχεδιασμού και το ArcCatalog.

Ο βασικότερος περιορισμός που εισάγεται με την χρήση μίας Personal Geodatabase είναι ότι δεν υποστηρίζεται η αποθήκευση δεδομένων τύπου Raster, τα οποία ούτως ή άλλως για τις ανάγκες του *Icaros.NET*, αποθηκεύονται στο σύστημα αρχείων.

7.3.1 Η επιλογή της UML για τον ορισμό του σχήματος της Βάσης.

Η UML όπως είδαμε στο κεφάλαιο 5 αποτελεί μία γλώσσα μοντελοποίησης η οποία κερδίζει συνέχεια έδαφος και έχει ωριμάσει αρκετά. Παρέχει ένα σύγχρονο μέσο για την σχεδίαση και την τεκμηρίωση λογισμικών συστημάτων και μπορεί να δώσει ευανάγνωστα διαγράμματα για το σχήμα της γεωγραφικής βάσης δεδομένων που πρόκειται να υλοποιήσουμε. Το σημαντικότερο πλεονέκτημα είναι ότι υποστηρίζει το αντικειμενοστραφές μοντέλο ανάπτυξης και διέπεται από μία εξελικτική φιλοσοφία σχεδιασμού.

Για την περίπτωση της πλατφόρμας *IcarosNET* η οποία αναπτύσσεται πάνω στο περιβάλλον ArcGIS υπάρχουν έτοιμα από την ESRI κατάλληλα templates και διαδικασίες για τον πλήρη σχεδιασμό μίας γεωγραφικής βάσης δεδομένων από το UML διάγραμμα που θα κατασκευάσουμε. Αυτό μας δίνει πολύ μεγάλη ευελιξία γιατί από τη γραφική απεικόνιση του μοντέλου της βάσης δεδομένων μπορούμε να περάσουμε αλλαγές στο τελικό σχήμα της γεωγραφικής βάσης δεδομένων.

7.3.2 Σχεδιαστικά ζητήματα

Για την κατασκευή μιας γεωγραφικής βάσης δεδομένων, στην περίπτωση που αξιολοπούμε την πλατφόρμα της ESRI, η διαδικασία περιλαμβάνει [110]:

- Μοντελοποίηση των δεδομένων από την οπτική του τελικού χρήστη: Πρόκειται για το στάδιο εκείνο της σχεδίασης όπου καταγράφονται οι απαιτήσεις από τον τελικό χρήστη της εφαρμογής
- Ορισμός των αντικειμένων και των συσχετίσεών τους: Από την καταγραφή των απαιτήσεων και δεδομένης μίας αντικειμενοστραφούς θεώρησης γίνεται εξαγωγή των αντικειμένων και των συσχετίσεων τους, που θα αποθηκευτούν στην βάση δεδομένων
- Επιλογή της γεωγραφικής αναπαράστασης: Για κάθε αντικείμενο επιλέγεται η εσωτερική αναπαράσταση για παράδειγμα αν θα απεικονιστεί σαν σημείο ή πολύγωνο
- Αντιστοίχιση στα στοιχεία που παρέχονται από την γεωγραφική βάση δεδομένων: Τα αντικείμενα αντιστοιχίζονται σε οντότητες που υποστηρίζονται από την βάση δεδομένων όπως πίνακες ή κλάσεις χαρακτηρισικών (Feature Classes)
- Οργάνωση της δομής της βάσης δεδομένων: Τέλος με τα κατάλληλα εργαλεία υλοποιείται το σχήμα της βάσης δεδομένων

Το σχήμα της βάσης δεδομένων που παρατίθεται στο παράρτημα Α' κατασκευάστηκε έτσι ώστε να αντικατοπτρίζει την πραγματικότητα που έχει να

αντιμετωπίζει πλατφόρμα *Icaros.NET* όσο έχει να κάνει τουλάχιστον με το επίπεδο των δεδομένων. Έχουμε δύο κεντρικά στατικά διαγράμματα κλάσεων. Το πρώτο αφορά τις κλάσεις για την διαχείριση των δεδομένων που αφορούν τους επίγειους σταθμούς μέτρησης και το δεύτερο τις κλάσεις που αφορούν τα αποτελέσματα από τα ατμοσφαιρικά μοντέλα.

Επίγειοι Σταθμοί Μέτρησης

Υλοποιούνται οι κλάσεις:

- **MonitoringStation**: Περιλαμβάνει σημειακή γεωγραφική πληροφορία και αποθηκεύει τα γενικά χαρακτηριστικά ενός σταθμού μέτρησης. Ο σταθμός αυτός μπορεί να πραγματοποιεί μία ή περισσότερες μετρήσεις (σχέση περιεχομένου)
- **MeasuredQuantities**: Αποθηκεύουν για κάθε σταθμό τα επιμέρους στοιχεία για τα όργανα μετρήσεων που περιλαμβάνει. Ουσιαστικά κάθε σταθμός έχει μία **MeasuredQuantity** για κάθε όργανο μέτρησης. Με κληρονομικότητα χωρίζονται σε
 - **Pollutant**: Για τα όργανα μέτρησης ρύπων.
 - **Meteorological** Για τα όργανα μέτρησης μετεωρολογικών δεδομένων.

Η κλάση **MeasureQuantities** είναι **Abstract**.

- **measurement**: Είναι η κλάση που συνδέεται με σχέση περιεχομένου με την προηγούμενη και αφορά τις επιμέρους μετρήσεις που γίνονται για από κάθε όργανο μέτρησης σε κάθε σταθμό, για μία δεδομένη χρονική στιγμή.

Τόσο η κλάση **MeasuredQuantities** όσο και η **Measurement** συνδέονται με τις απαραίτητες υποκλάσεις για την διαφοροποίησή τους σε σχέση με τις διαφορετικές μετρούμενες ποσότητες, όσον αφορά τα επιτρεπτά όρια, τις μονάδες μέτρησης κ.λ.π.

Ατμοσφαιρικά μοντέλα

Αντίστοιχα για τα ατμοσφαιρικά μοντέλα υλοποιούνται οι κλάσεις:

- **ModelGrid**: Περιλαμβάνει ένα σύνολο σημείων (γεωγραφικά σημεία) τα οποία ανήκουν στο πλέγμα των σημείων στα οποία δίνει αποτελέσματα ένα μοντέλο συμπεριφοράς της ατμόσφαιρας. Κάθε τέτοιο πλέγμα μπορεί να έχει τύπους μετρήσεων τα χαρακτηριστικά των οποίων αποθηκεύονται στην κλάση `GridMeasurement`.
- **GridMeasurement**: Η κλάση αυτή είναι `Abstract` και κληρονομεί στην `ModelForecast`. Εδώ αποθηκεύονται οι τύποι των αποτελεσμάτων των υπολογισμών των ατμοσφαιρικών μοντέλων. Ορίζονται για τη κλάση αυτή και οι κατάλληλοι υπότυποι ώστε να καλύπτεται όλο το φάσμα των δεδομένων που απαιτεί η πλατφόρμα *Icaros.NET*. Επίσης εδώ αποθηκεύεται και η χρονική στιγμή η οποία αφορά τα αποτελέσματα του ατμοσφαιρικού μοντέλου.
- **GridMeasurementValue**: Η κλάση αυτή αποθηκεύει για κάθε σημείο στο `ModelGrid` και κάθε πρόβλεψη από τα ατμοσφαιρικά μοντέλα στο `ModelForecast` την τιμή που υπολογίστηκε από το μοντέλο.

Παρατηρούμε ότι όλες οι κλάσεις που υλοποιήσαμε σχετίζονται με κάποιο από τα φυσικά αντικείμενα που διέπουν την πλατφόρμα *Icaros.NET*: Σταθμοί μέτρησης, αποτελέσματα από μοντέλα, σημεία του πλέγματος κ.λ.π. Αυτό μας διευκολύνει ιδιαίτερα όταν πρόκειται να επικοινωνήσουμε με άλλους επιστήμονες του κλάδου, κατά την σχεδίαση του συστήματος ώστε να καλύψουμε καλύτερα τις απαιτήσεις που προέρχονται από τα δεδομένα που εκείνοι μας παρέχουν.

7.3.3 Στατιστικά στοιχεία

Το σχήμα της γεωγραφικής βάσης δεδομένων όπως σχεδιάστηκε τελικά περιλαμβάνει:

- Τύπους Κωδικοποίησης: 6

- Τύπους Εύρους ζώνης: 19
- Γονικές Κλάσεις: 2
- Κλάσεις: 11
- Υπότυποι: 114

Το βασικό κομμάτι του σχήματος, αφορά τις κλάσεις οι οποίες είναι σχετικά λίγες. Αυτό είναι όμως αναμενόμενο αφού το σύστημα περιλαμβάνει λίγα αντικείμενα με πολλούς υποτύπους. Για παράδειγμα έχουμε την κλάση `MeasuredQuantities` η οποία μπορεί να έχει πολλούς υπότυπους όπως είναι οι διαφορετικοί ρύποι που υποστηρίζει η πλατφόρμα *Icaros.NET*. Έτσι για παράδειγμα διαφορετικοί ρύποι αποθηκεύονται στον ίδιο πίνακα στην βάση δεδομένων αλλά έχουν διαφορετικό υπότυπο, έτσι μπορούν να απεικονιστούν με διαφορετικό χρώμα.

7.4 Το σχήμα περιγραφής των αρχείων XML

Η κατεξοχήν σύγχρονη γλώσσα περιγραφής δεδομένων όπως είδαμε στο κεφάλαιο 6 είναι η XML. Παρέχει την δυνατότητα να ορίσουμε ένα σχήμα για τα αρχεία δεδομένων που απαιτεί η πλατφόρμα *Icaros.NET* ώστε να τα αναγνωρίζει αμέσως. Αυτό επιτυγχάνεται μέσα από το αρχείο ορισμού της XML που κατασκευάζουμε εμείς σύμφωνα με τις ανάγκες μας και μπορούμε να το διαθέσουμε σε όλους όσους προτίθενται να μας εφοδιάσουν με δεδομένα.

7.4.1 DTD ή XML Schema

Υπάρχουν δύο τρόποι για να κατασκευάσουμε το αρχείο αυτό. Ο πρώτος τρόπος οδηγεί στην δημιουργία ενός αρχείου DTD ενώ ο δεύτερος στο ορισμός ενός αρχείου XML Schema. Στην πρώτη περίπτωση έχουμε να κάνουμε με την δημιουργία ενός συντακτικού βασισμένου στην BNF ενώ στην δεύτερη με μία πιο σύνθετης μορφής περιγραφή. Η τελευταία εξυπηρετεί καλύτερα τις ανάγκες μας διότι εκτός από την δομική ορθότητα των αρχείων τα οποία βασίζονται σε αυτή ελέγχει και την εγκυρότητα των περιεχόμενων δεδομένων.

Ορίζεται δηλαδή για κάθε ετικέτα (TAG) και ο τύπος του δεδομένου (π.χ.: ημερομηνία ή ακέραιος), έτσι η εύρεση ενός διαφορετικού τύπου δεδομένου σημαίνει και σφάλμα επεξεργασίας. Ειδικά για την εφαρμογή μας, αυτό μας εξυπηρετεί ιδιαίτερα αφού πρόκειται να εισάγουμε δεδομένα αριθμητικά, κειμένου, και ημερομηνίας στην βάση δεδομένων και τα όποια σφάλματα καλό είναι να εντοπίζονται νωρίς κατά τη διαδικασία της εισαγωγής.

7.4.2 Από το σχήμα της βάσης στο XML Schema

Έχοντας ολοκληρώσει το σχήμα της βάσης δεδομένων, γνωρίζοντας τι δεδομένα απαιτεί η πλατφόρμα για την λειτουργία της μπορούμε να κατασκευάσουμε το επιθυμητό XML Schema. Το σχήμα της XML πρέπει να συμβαδίζει με το σχήμα της βάσης δεδομένων. Τελικά το σχήμα της XML που δημιουργήθηκε δίνεται στο παράρτημα Β', η απλουστευμένη δομή του δίνεται στο σχήμα Γ.10. Το σχήμα Γ.10 είναι φυσικά το ίδιο με το μοντέλο των κλάσεων της Visual Basic που αναπτύχθηκε για την επεξεργασία των XML αρχείων.

Το σκεπτικό με το οποίο δημιουργήθηκε το σχήμα για τα αρχεία XML της πλατφόρμας Icaros είναι ότι κάθε αρχείο XML αποτελεί ένα αρχείο με δεδομένα για την πλατφόρμα. Τα δεδομένα αυτά μπορεί να περιλαμβάνουν είτε στοιχεία για σταθμούς μέτρησης, είτε στοιχεία για αποτελέσματα μοντέλων, είτε στοιχεία για πλέγματα σημείων στα οποία δίνουν αποτελέσματα τα ατμοσφαιρικά μοντέλα. Κάθε σταθμός μέτρησης μπορεί να περιλαμβάνει μετρήσεις για ένα ή περισσότερα όργανα και κάθε όργανο μπορεί να περιλαμβάνει μετρήσεις για μία ή περισσότερες χρονικές στιγμές. Όμοια κάθε ατμοσφαιρικό μοντέλο μπορεί να περιλαμβάνει μετρήσεις για περισσότερα από ένα μεγέθη. Τα περισσότερα από τα προηγούμενα αντιστοιχούν σε ετικέτες της XML. Έχουμε έτσι την κεντρική ετικέτα IcarosData η οποία περικλείει ετικέτες του τύπου IcarosDataSource, που περιλαμβάνουν στοιχεία για την προέλευση των δεδομένων (ενδεχομένως πληροφορίες για το αρχείο ASCII από όπου προήλθαν τα δεδομένα). Αυτή με τη σειρά της περιλαμβάνει ετικέτες όπως MonitoringStation για τα στοιχεία περιγραφής ενός σταθμού μετρήσεων, ModelResults για τα αποτελέσματα ενός ατμοσφαιρικού μοντέλου, ModelGrid για τα στοιχεία ενός πλέγματος κ.λ.π.

7.5 Η βιβλιοθήκη υλοποίησης

7.5.1 Απαιτήσεις

Για την αξιοποίηση της βιβλιοθήκης υλοποίησης η βασική απαίτηση είναι η ύπαρξη του Microsoft XML parser στην έκδοση 4.0 ή νεότερη. Δεν απαιτείται δηλαδή η εγκατάσταση του περιβάλλοντος ArcGIS. Το τελευταίο απαιτείται μόνο στην περίπτωση που θέλουμε να εισάγουμε δεδομένα στην γεωγραφική βάση. Το γεγονός αυτό παρέχει την δυνατότητα της δημιουργίας ανεξάρτητων εφαρμογών-Command Line Utilities οι οποίες να μετατρέπουν τα διάφορα αρχεία τα οποία απαιτεί η πλατφόρμα *Icaros.NET* στο επιθυμητό XML format.

7.5.2 Η βιβλιοθήκη

Η βιβλιοθήκη αναπτύχθηκε στο περιβάλλον της Visual Basic, η οποία είναι μία γλώσσα εύκολη στην χρήση ενώ παρέχει πολλές δυνατότητες, γεγονός που την κάνει ιδανική για την γρήγορη ανάπτυξη πρωτοτύπων εφαρμογών. Είναι μία βιβλιοθήκη βασισμένη στο μοντέλο COM και μπορεί συνεπώς να χρησιμοποιηθεί και από άλλα περιβάλλοντα προγραμματισμού που υποστηρίζουν την συγκεκριμένη αρχιτεκτονική.

Το σύνολο των αντικειμένων που υλοποιούνται παρουσιάζονται αναλυτικά στο παράρτημα Γ'. Τα αντικείμενα αυτά προήλθαν σύμφωνα με τις απαιτήσεις που υπαγορεύει το αρχείο ορισμού XML που υιοθετήσαμε για την πλατφόρμα *Icaros.NET* και παρατίθεται στο παράρτημα Β'. Για την αξιοποίηση των δυνατοτήτων του Microsoft XML parser χρησιμοποιήθηκε η XML Wrapper Template όπως περιγράφεται στο [47], με κάποιες μικροαλλαγές που εξυπηρετούν της ανάγκες μας όπως για παράδειγμα την υποστήριξη στοιχείων χωρίς παιδιά (XML elements without children). Η τεχνική που περιγράφεται μας επιτρέπει την υλοποίηση σύνθετων δομών XML με ένα συστηματικό τρόπο, εύκολα και αποδοτικά. Στο [94] υπάρχει μία εναλλακτική προσέγγιση για την επεξεργασία δομών XML η οποία τελικά δεν ακολουθήθηκε στην πορεία της ανάπτυξης γιατί δεν ήταν κατάλληλη για την περίπτωσή μας.

Αξίζει να σημειωθεί ότι έχουν ληφθεί υπ' όψη οι διαδικασίες που αφορούν

την μετατροπή της ώρας (χρονική στιγμή) κατά την οποία λαμβάνουν χώρα οι διάφορες μετρήσεις ή υπολογισμοί των δεδομένων που αφορούν την πλατφόρμα και πρόκειται να εισαχθούν στην γεωγραφική βάση, τόσο σε σχέση με τις παγκόσμιες ζώνες ώρας, όσο και με την διαφορά που προκύπτει μεταξύ χειμερινής και θερινής ώρας. Τα αρχεία XML τα οποία βασίζονται σε ένα XML Schema απαιτείται από τα πεδία τους, που χαρακτηρίζονται σαν πεδία ημερομηνίας (dateTime), να ακολουθούν το πρότυπο ISO-8601-tz, βλέπε [57] και [8]. Το πρότυπο αυτό ορίζει το τρόπο που πρέπει να αποθηκεύονται πεδία ημερομηνίας-ώρας μέσα σε συμβολοσειρές ή αρχεία, χωρίς το κίνδυνο εισαγωγής διφορούμενων ημερομηνιών (όπως συνέβη με το γνωστό πρόβλημα του 2000). Επιπλέον μαζί με την ημερομηνία-ώρα είναι δυνατό να αποθηκευτεί και η ζώνη ώρας στην οποία αναφέρεται η συγκεκριμένη στιγμή. Επιπλέον στο [2] δίνονται πληροφορίες για το τρόπο εφαρμογής θερινής ώρας σε διάφορες περιοχές του κόσμου. Η βιβλιοθήκη μας παρέχει την κλάση CalendarHelper το οποία υλοποιεί μεθόδους για την μετατροπή από την μορφή του ISO-8601 στο τύπο date της VisualBasic και αντίστροφα. Σημειώνεται ότι όλα τα πεδία εντός της γεωγραφική βάσης δεδομένων, τα οποία αφορούν ημερομηνίες μετατρέπονται σε GMT χειμερινή ώρα κατά την εισαγωγή τους.

Η βιβλιοθήκη υποστηρίζει έτσι μία διπλή λειτουργία :

1. Διάβαση ενός αρχείου XML στην μνήμη
2. Δημιουργία ενός δέντρου XML στη μνήμη και αποθήκευσή του σε αρχείο XML

Στην πρώτη περίπτωση μπορούμε να διαβάσουμε ένα αρχείο XML και να το εισάγουμε στην μνήμη του υπολογιστή μέσα σε αντικείμενα τα οποία αντικατοπτρίζουν την δομή του. Έτσι μπορούμε στην συνέχεια από τα αντικείμενα αυτά να εισάγουμε στοιχεία στην γεωγραφική βάση της πλατφόρμας. Στη δεύτερη περίπτωση δημιουργούμε αντικείμενα στην μνήμη του υπολογιστή, καθώς επεξεργαζόμαστε αρχεία δεδομένων ASCII. Στη συνέχεια μπορούμε να αποθηκεύσουμε την δομή XML που αντιστοιχεί στα αντικείμενα που βρίσκονται στη μνήμη σε ένα XML αρχείο.

Περισσότερες πληροφορίες για την χρήση και τον προγραμματισμό των αντικειμένων των ESRI - ArcObjects μπορεί να αναζητήσει κανείς στα [15],[111]

και [112].

7.6 Είσοδος Δεδομένων

Για την σωστή είσοδο δεδομένων στην πλατφόρμα υπάρχουν δύο εναλλακτικοί τρόποι:

1. Απευθείας εισαγωγή αρχείου XML: Αν τα δεδομένα που πρόκειται να εισαχθούν ακολουθούν τον ορισμό XML, που ακολουθεί την τρέχουσα έκδοση της βάσης δεδομένων τότε η εισαγωγή μπορεί να γίνει χωρίς καμία προεργασία. Ο XML parser θα αναλάβει να ελέγξει την ορθότητα και την εγκυρότητα του αρχείου με βάση τον ορισμό της XML που απαιτείται και να εισάγει τελικά τα δεδομένα από το αρχείο στην βάση.
2. Μετατροπή αρχείου διαφορετικού τύπου: Για την εισαγωγή αρχείων δεδομένων διαφορετικού τύπου όπως για παράδειγμα αρχείων ASCII ή Microsoft Excel (.xls) απαιτείται πρώτα η ύπαρξη πρόσθετης πληροφορίας σχετικής με το αρχείο που πρόκειται να εισαχθεί. Η πληροφορία αυτή αφορά το περιεχόμενο του αρχείου και την γραμμογράφηση του. Στην συνέχεια πρέπει να δημιουργηθεί μία διαδικασία - εφαρμογή η οποία να αναλάβει την μετατροπή του αρχείου αυτού στο επιθυμητό XML format. Στην συνέχεια εισάγεται το αρχείο XML στην βάση.

Το σύστημα *Icaros.NET* στην παρούσα υλοποίησή του (σχήμα 7.1), δεν δέχεται από κανένα παροχό, πρωτογενή δεδομένα σε μορφή XML. Δεδομένα από τους σταθμούς μέτρησης λαμβάνονται σε μορφή ASCII ή .xls, με μεγάλη ποικιλία γραμμογραφήσεων, ενώ για τα ατμοσφαιρικά μοντέλα, παίρνουμε πάλι αρχεία ASCII με μία πιο σταθεροποιημένη μορφή. Θα ήταν ιδανικό, να λαμβάνονται τα δεδομένα αυτά απευθείας σε μορφή XML. Προς το παρόν όμως οι διαδικασίες μετατροπής έχουν ενσωματωθεί στην πλατφόρμα και η εισαγωγή για τις μορφές των αρχείων που έχουν προκαθοριστεί γίνεται αυτοματοποιημένα μέσα από ένα FTP Server.

7.7 Πλεονεκτήματα προσέγγισης

Η προσέγγιση που ακολουθήσαμε κατά την ανάπτυξη της πλατφόρμας *Icaros.NET* στο επίπεδο διαχείρισης δεδομένων μας δίνει αρκετά πλεονεκτήματα κυρίως στο τομέα της επεκτασιμότητας και της ευελιξίας.

Η σχεδίαση με χρήση UML μας επιτρέπει να κάνουμε αλλαγές στο σχήμα της βάσης και ιδιαίτερα προς την κατεύθυνση της προσθήκη νέων υπότυπων αντικειμένων και στην επέκταση των κωδικοποιήσεων. Αυτό είναι πιθανότερο να απαιτηθεί σε αντιδιαστολή με ένα ριζικότερο ανασχεδιασμό της βάσης δεδομένων.

Ο ορισμός της μορφής των αρχείων δεδομένων με χρήση της XML είναι απλός και μπορεί εύκολα να μεταβληθεί ώστε να συμβαδίζει με το σχήμα της βάσης δεδομένων. Αποτελεί εξάλλου τον σύγχρονο τρόπο ανταλλαγής και μεταφοράς δεδομένων μέσα από το δίκτυο. Υλοποιείται μέσα από απλά αρχεία ASCII, ανοιχτού format. Παρέχει έλεγχο της εγκυρότητας των δεδομένων με βάση το XML Schema και οδηγεί σε μία μελλοντική υιοθέτηση τεχνικών WebServices για την εισαγωγή δεδομένων στην βάση.

Κεφάλαιο 8

Αποτελέσματα – Συμπεράσματα

Από την έως τώρα παρουσίαση φάνηκε η αδιάσπαστη σχέση των ΣΓΠ με τις τεχνολογίες που σχετίζονται με τις βάσεις δεδομένων. Τα σύγχρονα ΣΓΠ βασίζονται εξολοκλήρου σε ΣΔΒΔ για την αποθήκευση των δεδομένων τους. Παράλληλα τα ΣΔΒΔ επεκτείνονται με τέτοιο τρόπο ώστε να καλύπτουν ανάγκες που πηγάζουν από την διαχείριση χωροχρονικής πληροφορίας, πληροφορία η οποία παράγεται κατά ένα μεγάλο βαθμό σαν αποτέλεσμα της λειτουργίας ΣΓΠ.

Στην περίπτωση του συστήματος *Icaros.NET* ασχοληθήκαμε με την ανάπτυξη του επιπέδου διαχείρισης δεδομένων το οποίο αποτελεί το βασικό υποσύστημα στο οποίο στηρίζεται η υπόλοιπη λειτουργία του. Πρόκειται για μία γεωγραφική βάση δεδομένων στην οποία αποθηκεύονται μετρήσεις επίγειων σταθμών και αποτελέσματα μοντέλων ατμοσφαιρικής συμπεριφοράς, με σκοπό την επεξεργασία τους για την μελέτη και την απεικόνιση της αέριας ρύπανσης σε μία ευρύτερη γεωγραφική περιοχή.

8.1 Αρχικοί Στόχοι

Από τις πρώτες φάσεις του σχεδιασμού και της ανάπτυξης του συστήματος *Icaros.NET* ήταν φανερό ότι απαιτεί για την λειτουργία του ένα μεγάλο όγκο δεδομένων με ποικιλία τόσο στη μορφή τους όσο και στο περιεχόμενό τους. Η μορφή αφορά τον τρόπο με τον οποίο τα δεδομένα αυτά είναι διαθέσιμα

ενώ το περιεχόμενο αφορά το τι ακριβώς περιγράφουν. Η εύρεση ενός ομοιόμορφου τρόπου για την διαχείριση του μεγαλύτερου ποσού της διαθέσιμης πληροφορίας αποτέλεσε έναν από τους βασικότερους στόχους. Αυτό μάλιστα έπρεπε να γίνει εμπλεκοντας το τελικό χρήστη του συστήματος στο μικρότερο δυνατό βαθμό και παρέχοντας μεγάλα ποσοστά αυτοματοποίησης των διαδικασιών.

8.2 Τι Υλοποιήθηκε

Συνοπτικά δύο πράγματα καταφέραμε να υλοποιήσουμε σαν αποτέλεσμα της παρούσας διπλωματικής εργασίας:

- Την γεωγραφική βάση δεδομένων
Στη βάση αυτή μπορούν να αποθηκευτούν τα αποτελέσματα από τις επίγειες μετρήσεις και τα ατμοσφαιρικά μοντέλα που απαιτούνται για την λειτουργία του συστήματος *Icaros.NET*. Η αντικειμενοστραφής σχεδίαση με χρήση της UML μας επέτρεψε να δημιουργήσουμε μία επεκτάσιμη περιγραφή των οντοτήτων που θέλουμε να αποθηκεύσουμε στη βάση δεδομένων. Ακόμα και αν μελλοντικά επιλέξουμε μία διαφορετική πλατφόρμα υλοποίησης μπορούμε να αξιοποιήσουμε το μοντέλο δεδομένων που παρουσιάστηκε με τις όποιες σχεδιαστικές επιλογές.
- Το XML Schema για την περιγραφή των δεδομένων
Μία ενιαία περιγραφή του συνόλου των δεδομένων που διαχειρίζεται το σύστημα *Icaros.NET* με χρήση της XML η οποία από τη μία παρέχει ένα σαφή ορισμό των δεδομένων που μπορούν να εισαχθούν και από την άλλη μπορεί να αποτελέσει τον συνδετικό κρίκο για την αλληλεπίδραση με την πλατφόρμα μέσω του διαδικτύου.

Το σημείο στο οποίο πρέπει να εστιάσει κανείς είναι ότι τόσο η UML όσο και η XML είναι δύο νέες τεχνολογίες, με μεγάλη δυναμική, ευρέως αποδεκτές και με εξαιρετικές δυνατότητες ευελιξίας. Η ευελιξία αφορά την δυνατότητα να εξελιχθεί το σύστημα που παρουσιάζουμε ώστε να συμπεριλάβει νέες παραμέτρους οι οποίες δεν είναι διαθέσιμες ή υπαρκτές κατά τον σχεδιασμό. Για παράδειγμα η ανάγκη για την προσθήκη ενός νέου τύπου ρύπου

δεν μπορεί να είναι γνωστή εκ των προτέρων. Όμως είναι δυνατή με μικρές τροποποιήσεις στο UML μοντέλο, το XML σχήμα και την βιβλιοθήκη .dll που αναπτύχθηκε.

8.3 Συμπεράσματα

Η ανάπτυξη ενός εξειδικευμένου ΣΓΠ είναι εφικτή με την χρήση των σύγχρονων εργαλείων προγραμματισμού τα οποία είναι στη διάθεσή μας. Τόσο οι διάφορες τεχνολογίες όσο και τα διαθέσιμα εργαλεία σχεδίασης είναι αρκετά ώριμα ώστε να προσφέρουν ένα πλούσιο περιβάλλον ανάπτυξης, χωρίς όμως να μπορεί να υποστηρίξει κανείς ότι έχουμε φτάσει στο τέλος. Είμαστε ακόμα στην αρχή. Η δική μας προσέγγιση, η οποία βασίζεται στις δυνατότητες επέκτασης που μας παρέχει ένα εμπορικό ΣΓΠ, μας έδωσε το πλεονέκτημα να μπορέσουμε να επικεντρωθούμε στο τελικό μας στόχο, το σύστημα *Icaros.NET*.

8.4 Μελλοντική Εργασία

Η ανάπτυξη του συστήματος έγινε με βάση της αρχιτεκτονική της επέκτασης πάνω σε ένα υπάρχον σχεσιακό ΣΔΒΔ. Θα είχε ενδιαφέρον η ανάπτυξη της εφαρμογής σε ένα πλήρως αντικειμενοστραφές ΣΔΒΔ. Ο τρόπος της σχεδίασης του φυσικού επιπέδου της βάσης δεδομένων καθώς και η επιλογή της XML για την περιγραφή των δεδομένων που διαχειρίζεται η πλατφόρμα μας επιτρέπουν να έχουμε ένα μεγάλο βαθμό ευελιξίας στην επιλογή της βάσης δεδομένων στην οποία θα βασιστεί η υλοποίηση τελικά.

Επίσης ένα μεγάλο κομμάτι που έμεινε ανοιχτό, είναι η διαχείριση των δεδομένων που προέρχονται από τις δορυφορικές εικόνες. Στην παρούσα προσέγγιση αγνοήθηκαν εντελώς τα προβλήματα και οι απαιτήσεις που πηγάζουν από την ανάγκη για την επεξεργασία των δορυφορικών δεδομένων. Κρίνεται σκόπιμο να εξετάσουν οι προοπτικές και οι δυνατότητες για την ένταξη των δορυφορικών δεδομένων στο μοντέλο δεδομένων της εφαρμογής και την δημιουργία μίας ενιαίας βάσης δεδομένων η οποία να μπορεί να αποθη-

κεύσει αποδοτικά το σύνολο της απαιτούμενης από το σύστημα *Icaros.NET* πληροφορίας.

Τέλος είναι σημαντικό να εξετάσουμε την αποδοτικότητα και τα όρια που μας παρέχονται από την υλοποίηση που επιλέξαμε ιδίως σε σχέση με την επιλογή της XML σαν μέσου περιγραφής των δεδομένων. Το επίπεδο διαχείρισης δεδομένων της πλατφόρμας *Icaros.NET* το οποίο υλοποιήθηκε τελικά, ενώ έχει δοκιμαστεί με πειραματικά δεδομένα δεν έχει λειτουργήσει σε πραγματικές συνθήκες κάτι που θα γίνει με την ολοκλήρωση της πλατφόρμας, αλλά μόνο στη βάση της *PersonaGeoDatabase*.

Παράρτημα

Παράρτημα Α΄

Το Σχήμα της Γεωγραφικής Βάσης Δεδομένων

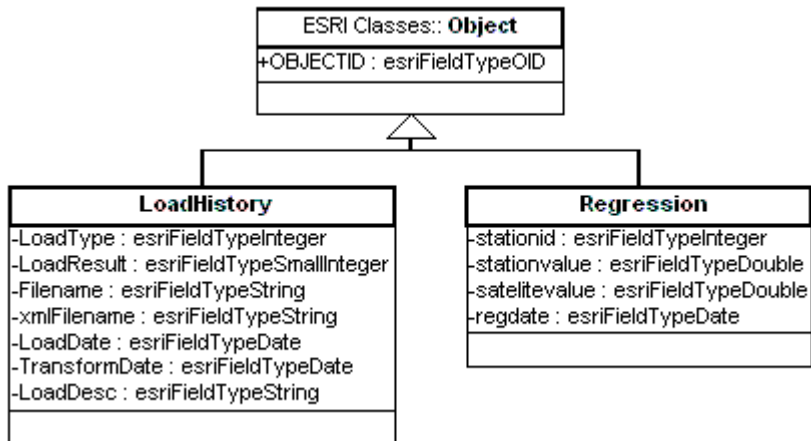
Α΄.1 Εισαγωγή

Στο παράρτημα αυτό παρουσιάζεται το σχήμα της γεωγραφικής βάσης δεδομένων όπως σχεδιάστηκε μέσα από το VISIO™ με τη χρήση των UML templates που παρέχει η ESRI. Από το σχήμα αυτό μπορεί να δημιουργηθεί η βάση δεδομένων, στην οποία θα αποθηκευτούν τα στοιχεία που αφορούν το δίκτυο των επίγειων σταθμών μετρήσεων και τα αποτελέσματα από τα μοντέλα εξομοίωσης της συμπεριφοράς της ατμόσφαιρας

Α΄.2 Παρουσίαση

Το σχήμα της βάσης γεωγραφικών δεδομένων μπορεί να αποτυπωθεί συνολικά σε ένα ενιαίο φύλλο. Για τις ανάγκες όμως της παρουσίασης εδώ χωρίστηκε σε επιμέρους μικρότερα τμήματα, τα οποία περιγράφονται αναλυτικά. Από τα επιμέρους σχήματα είναι εύκολο να ανακατασκευαστεί το συνολικό μοντέλο της βάσης γεωγραφικών δεδομένων. Στην συνέχεια παρουσιάζονται τα τμήματα αυτά.

Α'.2.1 Γενικά Βοηθητικά Αντικείμενα



Σχήμα Α'.1: Ορισμός γενικών αντικειμένων

Πρόκειται για κλάσεις οι οποίες αποθηκεύονται σαν απλοί πίνακες στη βάση γεωγραφικών δεδομένων, χωρίς γεωγραφική πληροφορία. Οι πληροφορίες που αποθηκεύουν τα αντικείμενα των κλάσεων αυτών, είναι βοηθητικές για τη λειτουργία της τελικής εφαρμογής. Στο σχήμα Α'.1 περιγράφονται οι κλάσεις LoadHistory και Regression.

Η πρώτη κλάση αποθηκεύει στοιχεία σχετικά με τα αρχεία που εισάγονται στην γεωγραφική βάση δεδομένων αυτοματοποιημένα.

- LoadType: Ο τύπος του αρχείου που εισάγεται.
- LoadResult: Το αποτέλεσμα της εισαγωγής.
- Filename: Το όνομα του αρχικού αρχείου.
- xmlFilename: Το όνομα αρχείου XML.
- LoadDate: Η ημερομηνία εισαγωγής του αρχείου.
- Transformdate: Η ημερομηνία μετατροπής του αρχείου.
- LoadDesc: Γενική περιγραφή.

ΠΑΡΑΡΤΗΜΑ Α'. ΤΟ ΣΧΗΜΑ ΤΗΣ ΓΕΩΓΡΑΦΙΚΗΣ ΒΑΣΗΣ ΔΕΔΟΜΕΝΩΝ

Η δεύτερη κλάση αποθηκεύει βοηθητικές πληροφορίες σχετικές με τις εσωτερικές διαδικασίες υπολογισμού που εκτελεί η εφαρμογή.

- stationid: Αναγνωριστικό OID για ένα σταθμό μέτρησης.
- stationvalue: Τιμή που προήλθε από το σταθμό.
- satelitevalue: Τιμή που προήλθε από την δορυφορική εικόνα.
- regdate: Ημερομηνία υπολογισμού.

Α'.2.2 Τύποι Κωδικοποίησης (Coded Value Domains)

Οι τύποι κωδικοποίησης ορίζουν μία αντιστοιχία ακεραίου - περιγραφής για διάφορες σταθερές οι οποίες μας εξυπηρετεί να εμφανίζονται σας συμβολοσειρές. Έτσι, ενώ εσωτερικά στη βάση δεδομένων αποθηκεύεται η ακέραια τιμή μπορούμε μέσα από κατάλληλες διαδικασίες να αναζητήσουμε την περιγραφή που της αντιστοιχεί. Στο σχήμα Α'.2 φαίνονται οι τύποι που περιγράφονται στην συνέχεια :

- Units Περιγράφονται όλες οι μονάδες μέτρησης που υποστηρίζει η πλατφόρμα *Icaros.NET* , και μπορεί να αποθηκεύει μεγέθη στις μονάδες αυτές.
 - mg_m3: χιλιοστά του γραμμαρίου ανά κυβικό μέτρο
 - Degrees: μοίρες
 - m_s: μέτρα ανά δευτερόλεπτο
 - oC: βαθμοί Κελσίου
 - percent: επί τα εκατό
 - hPa: εκατό-Πασκάλ
 - w_m2: Βαττ ανά τετραγωνικό μέτρο
 - oK: βαθμοί Κέλβιν
 - ug_m3: μικρογραμμάρια ανά κυβικό μέτρο
 - kg_kg: χιλιόγραμμα σε χιλιόγραμμα
 - m: μέτρα
 - m_sec_degrees: μέτρα ανά δευτερόλεπτο και μοίρες
 - ppb: μόρια στο εκατομμύριο
- ModelType Περιγράφονται οι τέσσερις διαφορετικοί τύποι μοντέλων ατμοσφαιρικής συμπεριφοράς που υποστηρίζει η πλατφόρμα
 - UAM
 - REMSAD

- CALPUFF
- MM5
- ModelDimensions Περιγράφει το τύπο του μοντέλου ατμοσφαιρικής συμπεριφοράς από πλευράς γεωμετρικών διαστάσεων.
 - e2d: Δισδιάστατο μοντέλο: δίνει αποτελέσματα σε ένα συγκεκριμένο ύψος
 - e3d: Τρισδιάστατο μοντέλο: δίνει αποτελέσματα για διάφορα ύψη

Στο σχήμα Α.3 φαίνονται οι τύποι που περιγράφονται στην συνέχεια :

- MeasurementStationDesc Περιγράφεται ο τύπος ενός σταθμού μετρήσεων σε σχέση με τη θέση του στον αστικό ιστό και περιλαμβάνει τις εξής περιπτώσεις:
 - STUnknown
 - Urban
 - UrbanBackground
 - UrbanIndustrial
 - UrbanCentre
 - Rural
 - BackGroun
 - SubUrban
 - Traffic
 - UrabnTraffic
- MeteoMeasurementPrinciple Περιγράφεται η μέθοδος που χρησιμοποιείται από το συγκεκριμένο όργανο για τη διεξαγωγή μετρήσεων μετεωρολογικών μεγεθών και μπορεί να πάρει τις ακόλουθες τιμές:
 - vane_cup_anemometer
 - platinum_resistance

- hair_hydrometer
- aneroid_barometer
- thermotension
- **PolutantMeasurementPrinciple** Περιγράφεται η μέθοδος που χρησιμοποιείται από το συγκεκριμένο όργανο για τη διεξαγωγή μετρήσεων ρυπαντών και μπορεί να πάρει τις ακόλουθες τιμές:
 - UV_fluorocence
 - IR_absortion
 - chemoluminescence
 - UV_absortion
 - FID_with_separation_column
 - thermodesorption
 - beta_absortion

«CodedValueDomain» Units
+FieldType : esriFieldType = esriFieldTypeInteger
+MergePolicy : esriMergePolicyType = esriMPTDefaultValue
+SplitPolicy : esriSplitPolicyType = esriSPTDefaultValue
+mg_m3 : <unspecified> = 1
+Degrees : <unspecified> = 2
+m_s : <unspecified> = 3
+oC : <unspecified> = 4
+percent : <unspecified> = 5
+hPa : <unspecified> = 6
+w_m2 : <unspecified> = 7
+oK : <unspecified> = 8
+ug_m3 : <unspecified> = 9
+kg_kg : <unspecified> = 10
+m : <unspecified> = 11
+m_sec_degrees : <unspecified> = 12
+ppb : <unspecified> = 13

«CodedValueDomain» ModelType
+FieldType : esriFieldType = esriFieldTypeInteger
+MergePolicy : esriMergePolicyType = esriMPTDefaultValue
+SplitPolicy : esriSplitPolicyType = esriSPTDefaultValue
+UAM : esriFieldTypeInteger = 1
+REMSAD : esriFieldTypeInteger = 2
+CALPUFF : esriFieldTypeInteger = 3
+MMS : esriFieldTypeInteger = 4

«CodedValueDomain» ModelDimensions
+FieldType : esriFieldType = esriFieldTypeInteger
+MergePolicy : esriMergePolicyType = esriMPTDefaultValue
+SplitPolicy : esriSplitPolicyType = esriSPTDefaultValue
+e2d : esriFieldTypeInteger = 1
+e3d : esriFieldTypeInteger = 2

Σχήμα Α'.2: Τύποι κωδικοποίησης Α'

«CodedValueDomain» MeasurementStationDesc
+FieldType : esriFieldType = esriFieldTypeInteger +MergePolicy : esriMergePolicyType = esriMPTDefaultValue +SplitPolicy : esriSplitPolicyType = esriSPTDefaultValue +STUnknown : esriFieldTypeInteger = 1 -Urban : esriFieldTypeInteger = 2 +UrbanBackground : esriFieldTypeInteger = 3 +UrbanIndustrial : esriFieldTypeInteger = 4 -UrbanCentre : esriFieldTypeInteger = 5 -Rural : esriFieldTypeInteger = 6 -BackGround : esriFieldTypeInteger = 7 -SubUrban : esriFieldTypeInteger = 8 -Traffic : esriFieldTypeInteger = 9 -UrbanTraffic : esriFieldTypeInteger = 10

«CodedValueDomain» MeteoMeasurementPrinciple
+FieldType : esriFieldType = esriFieldTypeInteger +MergePolicy : esriMergePolicyType = esriMPTDefaultValue +SplitPolicy : esriSplitPolicyType = esriSPTDefaultValue +vane_cup_anemometer : esriFieldTypeInteger = 1 +platinum_resistance : esriFieldTypeInteger = 2 +hair_hydrometer : esriFieldTypeInteger = 3 -aneroid_barometer : esriFieldTypeInteger = 4 -thermotension : esriFieldTypeInteger = 5

«CodedValueDomain» PolutantMeasurementPrinciple
+FieldType : esriFieldType<unspecified> = esriFieldTypeInteger +MergePolicy : esriMergePolicyType<unspecified> = esriMPTDefaultValue +SplitPolicy : esriSplitPolicyType<unspecified> = esriSPTDefaultValue +UV_fluorocence : <unspecified> = 6 +IR_absortion : <unspecified> = 7 +chemolominescence : <unspecified> = 8 -UV_absortion : <unspecified> = 9 -FID_with_separation_columnn : <unspecified> = 10 -thermodesorption : <unspecified> = 11 -beta_absortion : <unspecified> = 12

Σχήμα Α.3: Τύποι κωδικοποίησης Β'

Α'.2.3 Τύποι Περιοχής Τιμών (Range Value Domains)

Στην συνέχεια περιγράφονται οι τύποι περιοχής τιμών που χρησιμοποιούνται για την οριοθέτηση των τιμών των μετρήσεων ανάλογα με τον τύπο τους. Τα πεδία MinValue και MaxValue αντιστοιχούν στην ελάχιστη και μέγιστη επιτρεπτά μετρούμενη τιμή. Τιμές μέτρησης έξω από τα όρια αυτά υποδηλώνουν εσφαλμένα δεδομένα.

Στα σχήματα Α'.4, Α'.5, Α'.6, Α'.7 ορίζονται οι περιοχές τιμών για τις μετρήσεις:

- RangeDomainBenzeneOxylene
- RangeDomainCO
- RangeDomainNMHC
- RangeDomainNO
- RangeDomainNO2
- RangeDomainO3_PM10_Airborne_Particles
- RangeDomainSO2
- RangeDomainToluene
- RangeDomainHumidity
- RangeDomainpressure
- RangeDomainRadiation
- RangeDomainradiation
- RangeDomaintemperature
- RangeDomainUnknown
- RangeDomainwinddirection
- RangeDomainwindvelocity

ΠΑΡΑΡΤΗΜΑ Α'. ΤΟ ΣΧΗΜΑ ΤΗΣ ΓΕΩΓΡΑΦΙΚΗΣ ΒΑΣΗΣ ΔΕΔΟΜΕΝΩΝ

- RangeDomainCH4
- RangeDoaminTHC
- RangeDomainDust
- RangeDomainMixingHeight

ΠΑΡΑΡΤΗΜΑ Α'. ΤΟ ΣΧΗΜΑ ΤΗΣ ΓΕΩΓΡΑΦΙΚΗΣ ΒΑΣΗΣ ΔΕΔΟΜΕΝΩΝ

«RangeDomain» RangeDomainBenzene_Oxylene
+FieldType : esriFieldType<unspecified> = esriFieldTypeDouble +MergePolicy : esriMergePolicyType<unspecified> = esriMPTDefaultValue +SplitPolicy : esriSplitPolicyType<unspecified> = esriSPTDefaultValue +MinValue : esriFieldTypeDouble<unspecified> = 0 +MaxValue : esriFieldTypeDouble<unspecified> = 0,1

«RangeDomain» RangeDomainCO
+FieldType : esriFieldType<unspecified> = esriFieldTypeDouble +MergePolicy : esriMergePolicyType<unspecified> = esriMPTDefaultValue +SplitPolicy : esriSplitPolicyType<unspecified> = esriSPTDefaultValue +MinValue : esriFieldTypeDouble<unspecified> = 0 +MaxValue : esriFieldTypeDouble<unspecified> = 120

«RangeDomain» RangeDomainHMHC
+FieldType : esriFieldType<unspecified> = esriFieldTypeDouble +MergePolicy : esriMergePolicyType<unspecified> = esriMPTDefaultValue +SplitPolicy : esriSplitPolicyType<unspecified> = esriSPTDefaultValue +MinValue : esriFieldTypeDouble<unspecified> = 0 +MaxValue : esriFieldTypeDouble<unspecified> = 5,35

«RangeDomain» RangeDomainHIO
+FieldType : esriFieldType<unspecified> = esriFieldTypeDouble +MergePolicy : esriMergePolicyType<unspecified> = esriMPTDefaultValue +SplitPolicy : esriSplitPolicyType<unspecified> = esriSPTDefaultValue +MinValue : esriFieldTypeDouble<unspecified> = 0 +MaxValue : esriFieldTypeDouble<unspecified> = 1,35

«RangeDomain» RangeDomainH02
+FieldType : esriFieldType<unspecified> = esriFieldTypeDouble +MergePolicy : esriMergePolicyType<unspecified> = esriMPTDefaultValue +SplitPolicy : esriSplitPolicyType<unspecified> = esriSPTDefaultValue +MinValue : esriFieldTypeDouble<unspecified> = 0 +MaxValue : esriFieldTypeDouble<unspecified> = 2

Σχήμα Α.4: Τύποι περιοχής τιμών Α'

<p>«RangeDomain» RangeDomainO3_PM10_Airborne_Particles</p> <p>+FieldType : esriFieldType<unspecified> = esriFieldTypeDouble +MergePolicy : esriMergePolicyType<unspecified> = esriMPTDefaultValue +SplitPolicy : esriSplitPolicyType<unspecified> = esriSPTDefaultValue +MinValue : esriFieldTypeDouble<unspecified> = 0 +MaxValue : esriFieldTypeDouble<unspecified> = 1</p>
<p>«RangeDomain» RangeDomainSO2</p> <p>+FieldType : esriFieldType<unspecified> = esriFieldTypeDouble +MergePolicy : esriMergePolicyType<unspecified> = esriMPTDefaultValue +SplitPolicy : esriSplitPolicyType<unspecified> = esriSPTDefaultValue +MinValue : esriFieldTypeDouble<unspecified> = 0 +MaxValue : esriFieldTypeDouble<unspecified> = 1,4</p>
<p>«RangeDomain» RangeDomainToluene</p> <p>+FieldType : esriFieldType<unspecified> = esriFieldTypeDouble +MergePolicy : esriMergePolicyType<unspecified> = esriMPTDefaultValue +SplitPolicy : esriSplitPolicyType<unspecified> = esriSPTDefaultValue +MinValue : esriFieldTypeDouble<unspecified> = 0 +MaxValue : esriFieldTypeDouble<unspecified> = 0,3</p>
<p>«RangeDomain» RangeDomainhumidity</p> <p>+FieldType : esriFieldType<unspecified> = esriFieldTypeDouble +MergePolicy : esriMergePolicyType<unspecified> = esriMPTDefaultValue +SplitPolicy : esriSplitPolicyType<unspecified> = esriSPTDefaultValue +MinValue : esriFieldTypeDouble<unspecified> = 10 +MaxValue : esriFieldTypeDouble<unspecified> = 100</p>
<p>«RangeDomain» RangeDomainpressure</p> <p>+FieldType : esriFieldType<unspecified> = esriFieldTypeDouble +MergePolicy : esriMergePolicyType<unspecified> = esriMPTDefaultValue +SplitPolicy : esriSplitPolicyType<unspecified> = esriSPTDefaultValue +MinValue : esriFieldTypeDouble<unspecified> = 950 +MaxValue : esriFieldTypeDouble<unspecified> = 1050</p>

Σχήμα Α.5: Τύποι περιοχής τιμών Β'

<p>«RangeDomain» RangeDomainradiation</p> <p>+FieldType : esriFieldType<unspecified> = esriFieldTypeDouble +MergePolicy : esriMergePolicyType<unspecified> = esriMPTDefaultValue +SplitPolicy : esriSplitPolicyType<unspecified> = esriSPTDefaultValue +MinValue : esriFieldTypeDouble<unspecified> = 0 +MaxValue : esriFieldTypeDouble<unspecified> = 0,2</p>
<p>«RangeDomain» RangeDomaintemperature</p> <p>+FieldType : esriFieldType<unspecified> = esriFieldTypeDouble +MergePolicy : esriMergePolicyType<unspecified> = esriMPTDefaultValue +SplitPolicy : esriSplitPolicyType<unspecified> = esriSPTDefaultValue +MinValue : esriFieldTypeDouble<unspecified> = -30 +MaxValue : esriFieldTypeDouble<unspecified> = 50</p>
<p>«RangeDomain» RangeDomainUnknown</p> <p>+FieldType : esriFieldType<unspecified> = esriFieldTypeDouble +MergePolicy : esriMergePolicyType<unspecified> = esriMPTDefaultValue +SplitPolicy : esriSplitPolicyType<unspecified> = esriSPTDefaultValue +MinValue : esriFieldTypeDouble<unspecified> = 0 +MaxValue : esriFieldTypeDouble<unspecified> = 100</p>
<p>«RangeDomain» RangeDomainwinddirection</p> <p>+FieldType : esriFieldType<unspecified> = esriFieldTypeDouble +MergePolicy : esriMergePolicyType<unspecified> = esriMPTDefaultValue +SplitPolicy : esriSplitPolicyType<unspecified> = esriSPTDefaultValue +MinValue : esriFieldTypeDouble<unspecified> = 0 +MaxValue : esriFieldTypeDouble<unspecified> = 360</p>
<p>«RangeDomain» RangeDomainwindvelocity</p> <p>+FieldType : esriFieldTypeDouble<unspecified> = esriFieldTypeDouble +MergePolicy : esriMergePolicyType<unspecified> = esriMPTDefaultValue +SplitPolicy : esriSplitPolicyType<unspecified> = esriSPTDefaultValue +MinValue : esriFieldTypeDouble<unspecified> = 0 +MaxValue : esriFieldTypeDouble<unspecified> = 35</p>

Σχήμα Α'.6: Τύποι περιοχής τιμών Γ'

«RangeDomain» RangeDomainCH4
+FieldType : esriFieldType<unspecified> = esriFieldTypeDouble +MergePolicy : esriMergePolicyType<unspecified> = esriMPTDefaultValue +SplitPolicy : esriSplitPolicyType<unspecified> = esriSPTDefaultValue +MinValue : esriFieldTypeDouble<unspecified> = 0 +MaxValue : esriFieldTypeDouble<unspecified> = 120

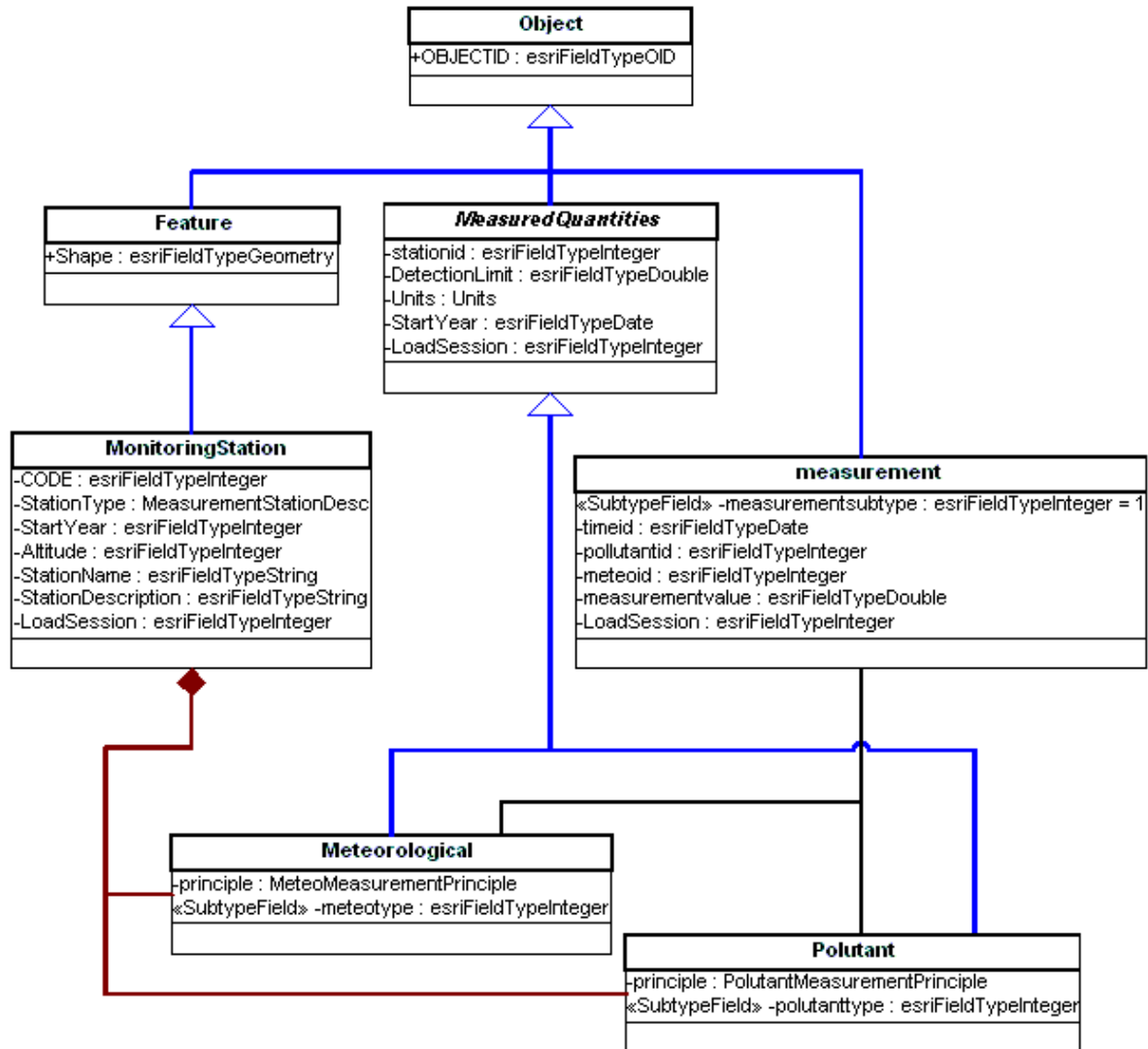
«RangeDomain» RangeDomainTHC
+FieldType : esriFieldType<unspecified> = esriFieldTypeDouble +MergePolicy : esriMergePolicyType<unspecified> = esriMPTDefaultValue +SplitPolicy : esriSplitPolicyType<unspecified> = esriSPTDefaultValue +MinValue : esriFieldTypeDouble<unspecified> = 0 +MaxValue : esriFieldTypeDouble<unspecified> = 120

«RangeDomain» RangeDomainDust
+FieldType : esriFieldType<unspecified> = esriFieldTypeDouble +MergePolicy : esriMergePolicyType<unspecified> = esriMPTDefaultValue +SplitPolicy : esriSplitPolicyType<unspecified> = esriSPTDefaultValue +MinValue : esriFieldTypeDouble<unspecified> = 0 +MaxValue : esriFieldTypeDouble<unspecified> = 120

«RangeDomain» RangeDomainMixingHeight
+FieldType : esriFieldType<unspecified> = esriFieldTypeDouble +MergePolicy : esriMergePolicyType<unspecified> = esriMPTDefaultValue +SplitPolicy : esriSplitPolicyType<unspecified> = esriSPTDefaultValue +MinValue : esriFieldTypeDouble<unspecified> = 0 +MaxValue : esriFieldTypeDouble<unspecified> = 9999

Σχήμα Α.7: Τύποι περιοχής τιμών Δ'

Α'.2.4 Περιγραφή Επίγειων Σταθμών Μέτρησης



Σχήμα Α'.8: Το μοντέλο για τους επίγειους σταθμούς μέτρησης

Στο σχήμα Α'.8 αποδίδεται το κεντρικό μοντέλο για τα δεδομένα που αφορούν το δίκτυο των επίγειων σταθμών μέτρησης και οι κλάσεις που περιλαμβάνει περιγράφονται στην συνέχεια :

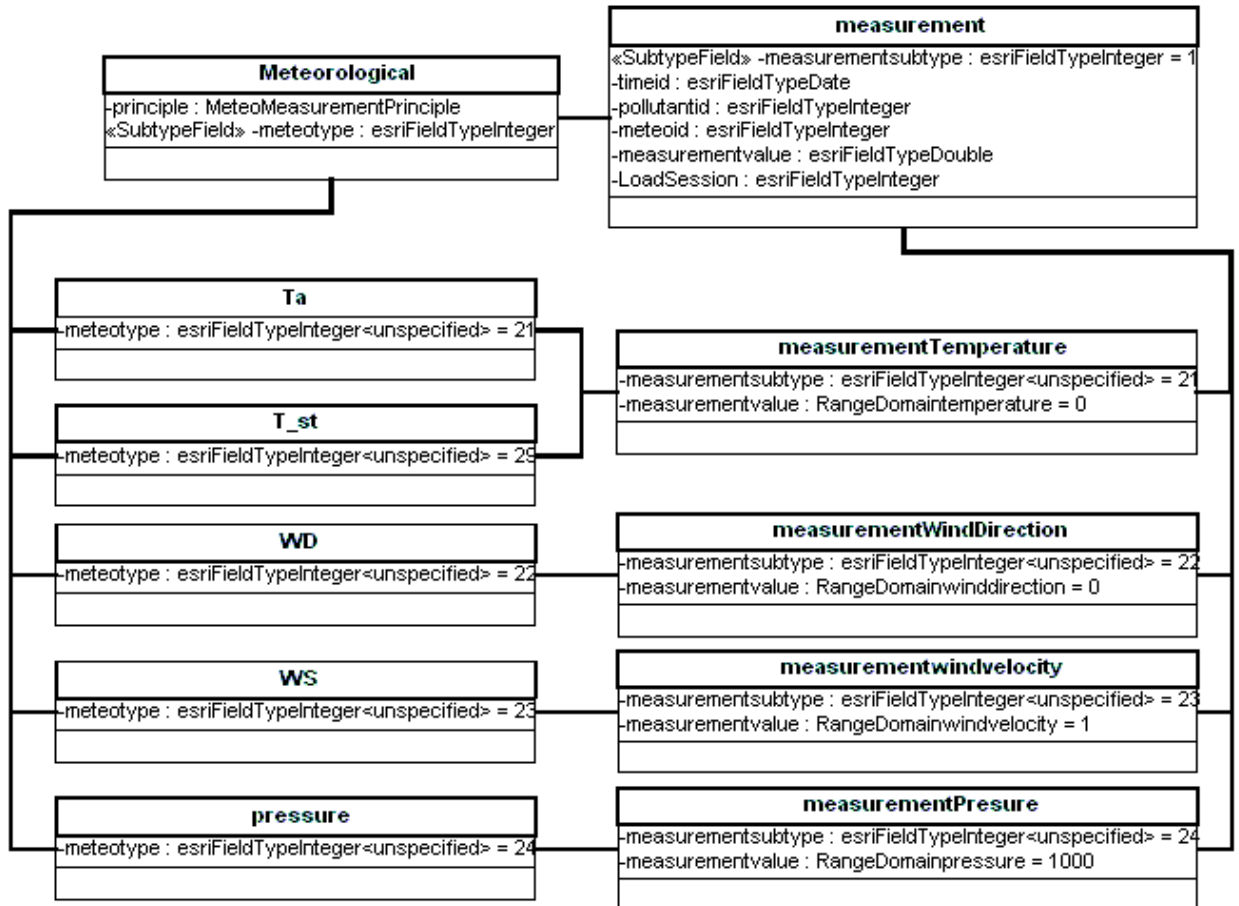
- Object: Η κεντρική κλάση από όπου κληρονομούν όλα τα αντικείμενα της γεωγραφικής βάσης δεδομένων.

- OBJECTID: Μοναδικό αναγνωριστικό για κάθε αντικείμενο της κλάσης (Πρωτεύον κλειδί)
- Feature: Η κεντρική κλάση από όπου κληρονομούν όλα τα αντικείμενα της γεωγραφικής βάσης δεδομένων τα οποία περιλαμβάνουν και γεωγραφική πληροφορία.
 - SHAPE: Αποθηκεύει όλη τη γεωγραφική πληροφορία για ένα γεωγραφικό αντικείμενο της βάσης (συντεταγμένες και γεωγραφική αναφορά). Ο χειρισμός των πεδίων αυτών γίνεται εσωτερικά από το ΣΔΒΔ
- MonitoringStation: Η κλάση που περιλαμβάνει αντικείμενα του τύπου «Σταθμός Μέτρησης». Περιγράφεται από σημειακή γεωγραφική πληροφορία. Τα επιμέρους χαρακτηριστικά περιγράφονται στην συνέχεια:
 - CODE: Ένας κωδικός αριθμός που χαρακτηρίζει το σταθμό μέτρησης και αποδίδεται από την υπηρεσία στην οποία ανήκει ο σταθμός μέτρησης.
 - StationType:MeasurementStationDesc: Ο τύπος του σταθμού μέτρησης ανάλογα με τη θέση του στο αστικό πλέγμα.
 - StartYear: Η χρονολογία έναρξης της λειτουργίας του σταθμού.
 - Altitude: Το υψόμετρο στο οποίο είναι εγκατεστημένος και λειτουργεί ο σταθμός.
 - StationName: Το όνομα του σταθμού μέτρησης όπως αποδίδεται από την υπηρεσία στην οποία ανήκει ο σταθμός.
 - StationDescription: Μία περιγραφή του σταθμού μέτρησης ή μία αναλυτικότερη ονομασία
 - LoadSession: Κωδικός αριθμός που αποδίδεται εσωτερικά από το ΣΔΒΔ και αντιστοιχεί σε ξένο κλειδί στην κλάση LoadHistory (βλέπε σχήμα Α'.1).
- measurement: Πρόκειται για την κλάση που περιλαμβάνει αντικείμενα του τύπου «Μετρήσεις» είτε ρύπων, είτε μετεωρολογικών μεγεθών. Περιλαμβάνει τις ακόλουθες ιδιότητες:

- «SubtypeField» measurementsubtype: Υπότυπος τύπου μέτρησης.
- timeid: Χρονική στιγμή που φορά τη μέτρηση (ημερομηνία και ώρα).
- pollutantid: Ξένο κλειδί στην κλάση Pollutant.
- meteoid: Ξένο κλειδί στην κλάση Meteorological.
- measurementvalue: Η αριθμητική τιμή της μέτρησης.
- LoadSession: όπως και στη προηγούμενη κλάση.
- MeasuredQuantities: Αφηρημένη κλάση (*Abstract*) από την οποία κληρονομούν οι κλάσεις για τις μετρούμενες ποσότητες ρύπων και μετεωρολογικών μεγεθών.
 - stationid: Ξένο κλειδί στην κλάση MonitoringStation.
 - DetectionLimit: Μέγιστη επιτρεπτή τιμή μέτρησης.
 - Units: Μονάδα μέτρησης ανάλογα με την μετρούμενη ποσότητα.
 - StartYear: Χρονολογία έναρξης λειτουργίας του οργάνου για την συγκεκριμένη μέτρηση.
 - LoadSession: όπως και στην προηγούμενη κλάση.
- Meteorological: Μετρήσεις μετεωρολογικών μεγεθών.
 - principle:MeteoMeasurementPrinciple: Αρχή λειτουργίας του οργάνου μέτρησης, για μετεωρολογικές μετρήσεις.
 - «SubtypeField» meteotype: Υπότυπος του μετρούμενου μεγέθους.
- Polutant: Μετρήσεις ρύπων.
 - principle:PolutantMeasurementPrinciple: Αρχή λειτουργίας του οργάνου μέτρησης, για μετρήσεις ρύπων.
 - «SubtypeField» polutantype: Υπότυπος του μετρούμενου ρύπου.

Οι υπότυποι των μετεωρολογικών μετρήσεων αντιστοιχούν στα διάφορα μετεωρολογικά μεγέθη που μπορούν να αποθηκευτούν στη γεωγραφική βάση δεδομένων. Κάθε τέτοιο μέγεθος αντιστοιχίζεται σε ένα υπότυπο μίας μέτρησης. Έτσι για κάθε μετρούμενη ποσότητα μπορούμε να αντιστοιχίσουμε μετρήσεις με τιμές που αντιστοιχούν σε ένα συγκεκριμένο πεδίο τιμών (RangeDomain).

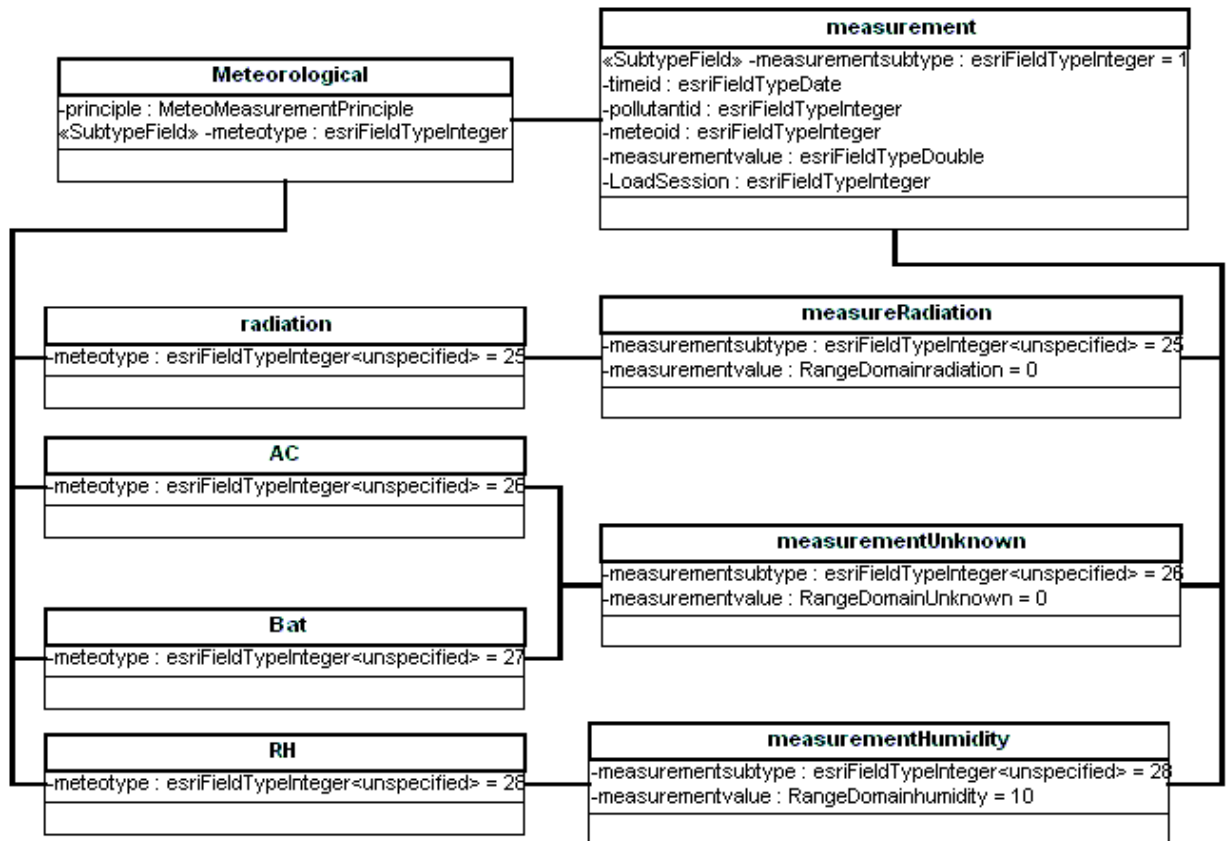
Περιγραφή Υπότυπων Μετεωρολογικών Μετρήσεων



Σχήμα Α'.9: Υπότυποι μετεωρολογικών μετρήσεων Α'

Στα σχήματα Α'.9 και Α'.10 βλέπουμε του υπότυπους που σχετίζονται με τα μετεωρολογικά δεδομένα και τις αντιστοιχίσεις με τα πεδία τιμών για τις επιμέρους μετρήσεις. Από το όνομα του υπότυπου μπορεί να ξεχωρίσει κανείς για τι τύπου μέτρηση πρόκειται.

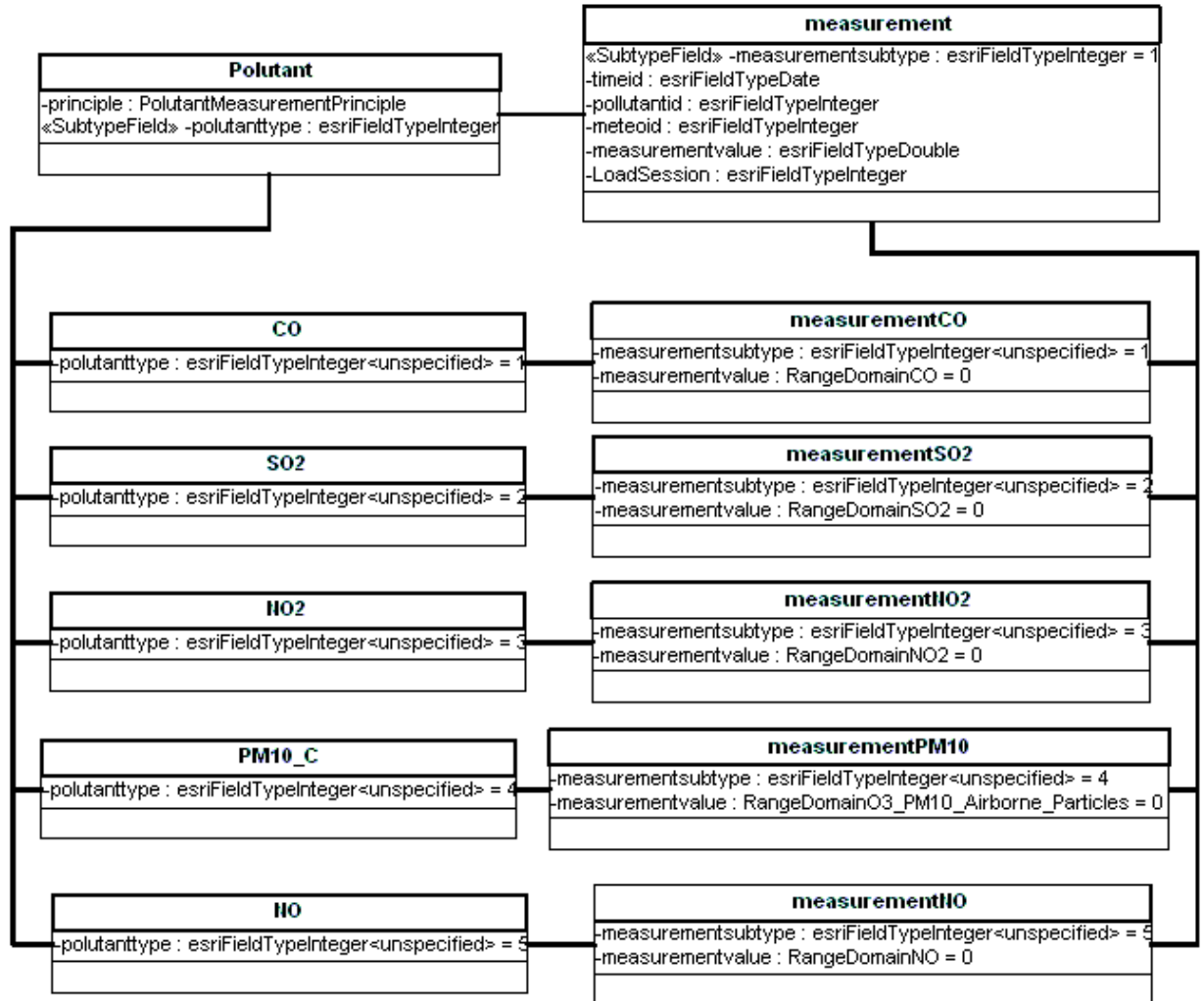
- Ta, T_st – measurementTemperature
- WD – measurementWindDirection
- WS – measurementWindVelocity



Σχήμα Α'.10: Υπότυποι μετεωρολογικών μετρήσεων Β'

- pressure – measurementPressure
- radiation – measurRadiation
- AC, Bat – measurementUnknown
- RH – measurementHumidity

Περιγραφή Υπότυπων Μετρήσεων Ρύπων

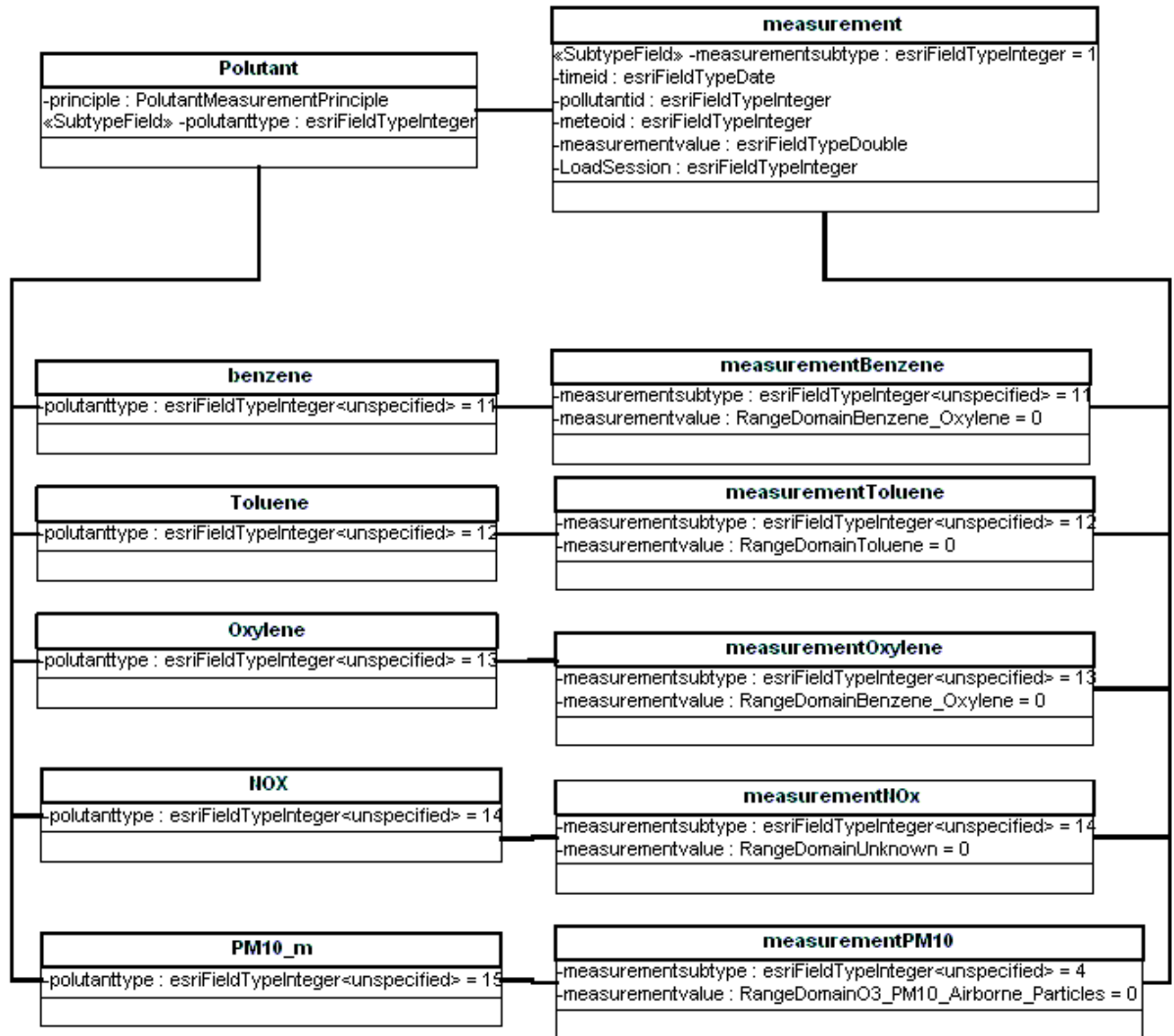


Σχήμα Α.11: Υπότυποι μετρήσεων ρύπων Α'

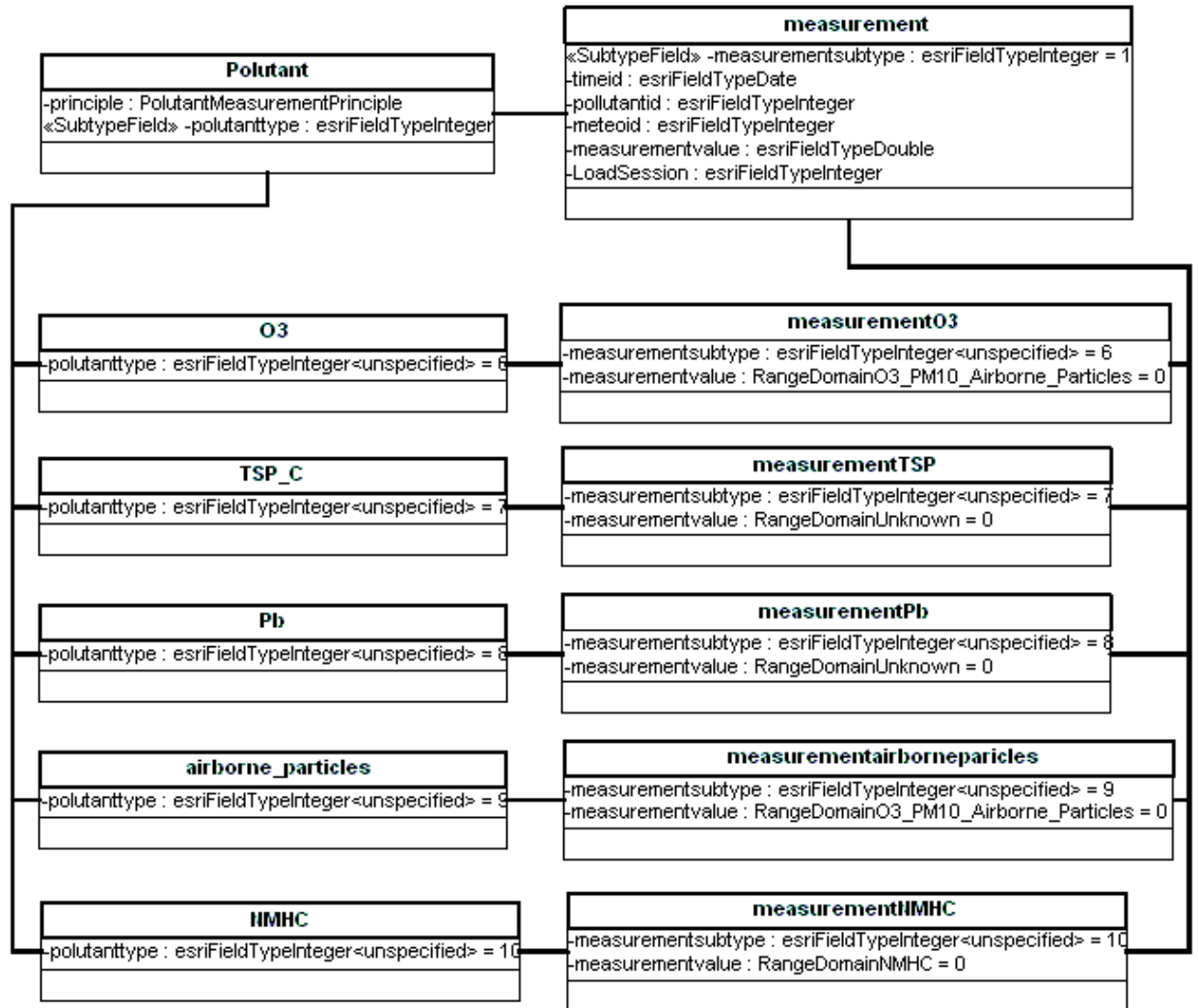
Στα σχήματα Α.11, Α.12, Α.13 και Α.14 βλέπουμε του υπότυπους που σχετίζονται με τους ρύπους και τις αντιστοιχίσεις με τα πεδία τιμών για τις επιμέρους μετρήσεις. Από το όνομα του υπότυπου μπορεί να ξεχωρίσει κανείς για τι τύπου μέτρηση πρόκειται.

- CO – measurementCO
- SO2 – measurmeentSO2

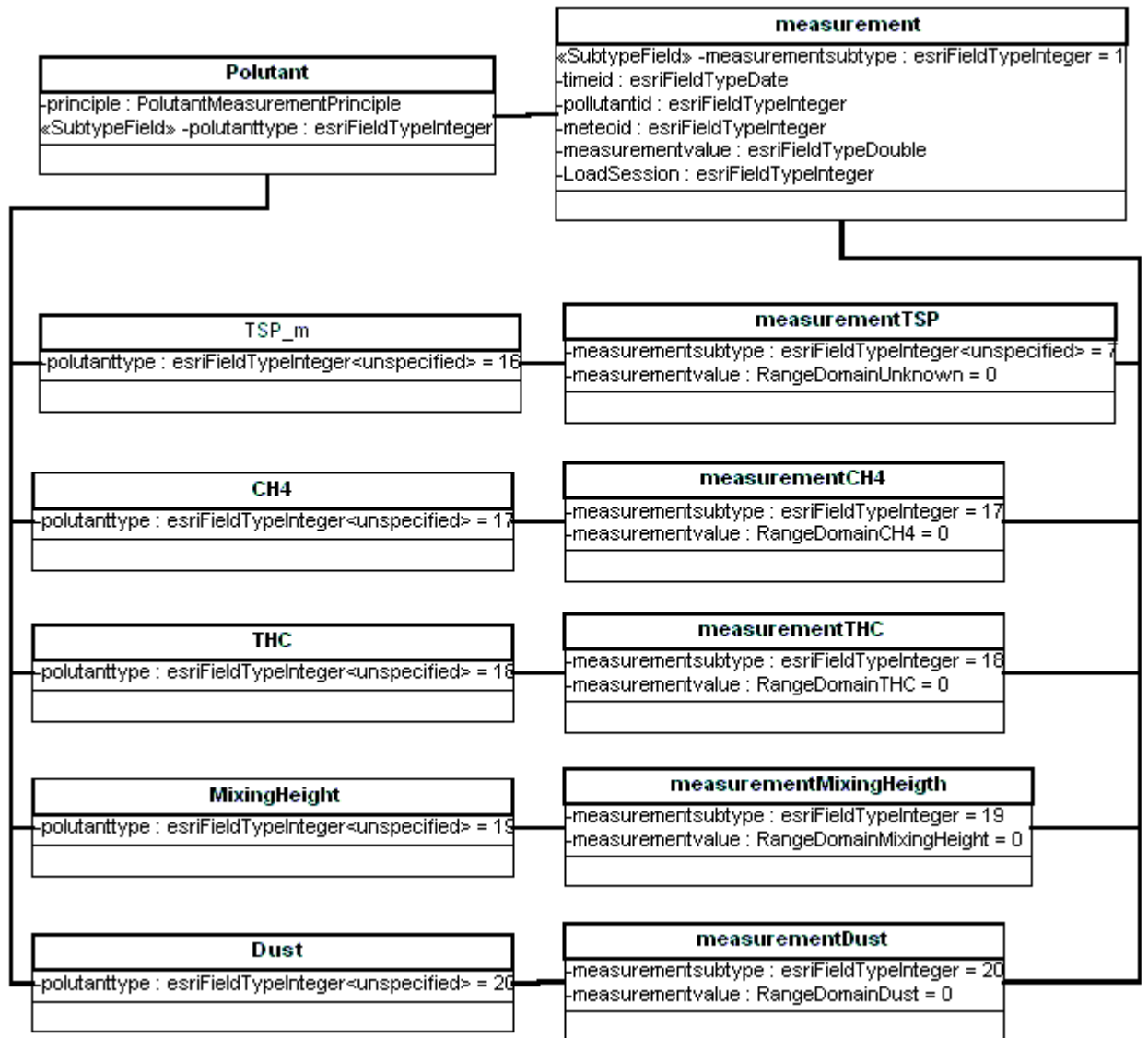
- NO2 - measurementNO2
- PM10_C - measurementPM10
- NO - measurementNO
- benzene - measurementBenzene
- Toluene - measurementToluene
- Oxylene - measurementOxylene
- NOX - measurementNOx
- PM10_m - measurementPM10
- O3 - measurmentO3
- TSP_C - measurementTSP
- Pb - measurementPb
- airborne_particles - measurementairborneparticles
- NMHC - measurementNMHC
- TSP_m - measurementTSP
- CH4 - measurementCH4
- THC - measurementTHC
- MixingHeight - measurementMixingHeight
- Dust - measurementDust



Σχήμα Α'.12: Υπότυποι μετρήσεων ρύπων Β'

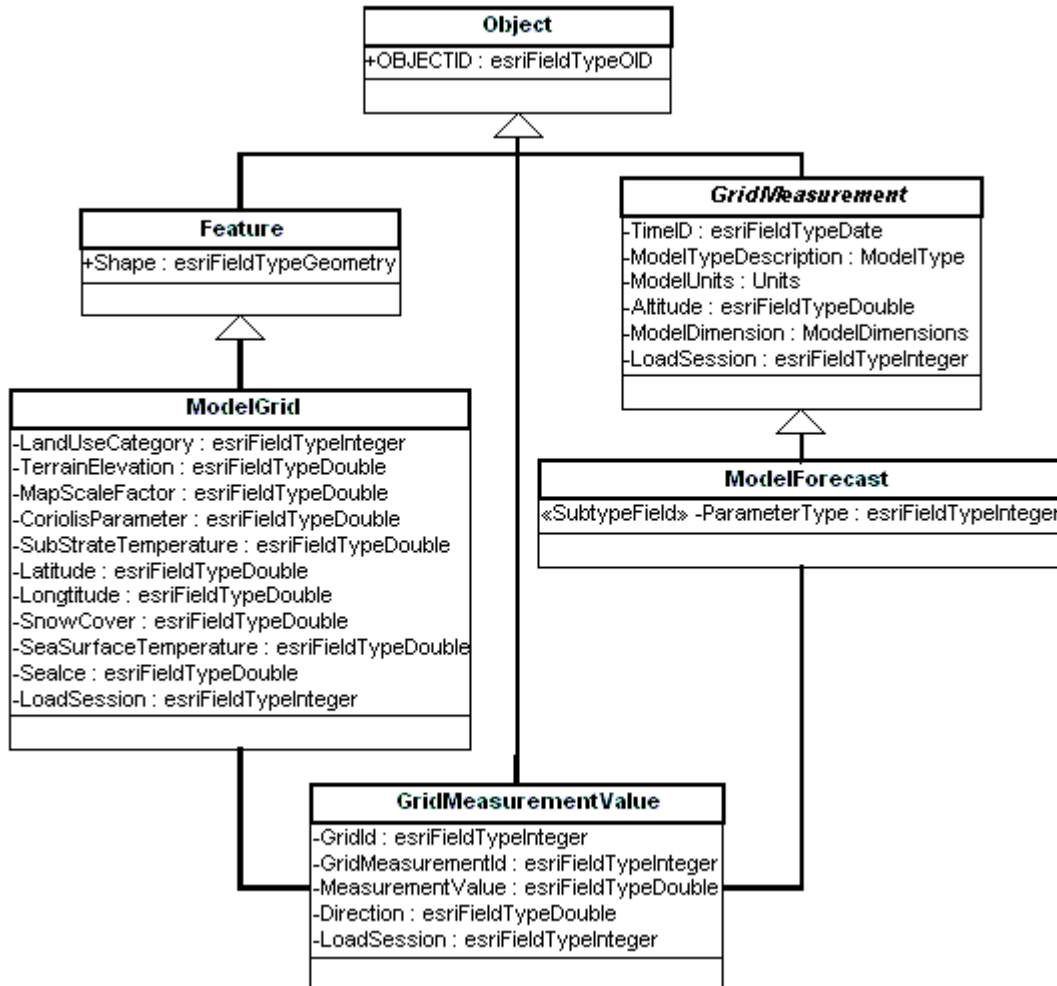


Σχήμα Α.13: Υπότυποι μετρήσεων ρύπων Γ'



Σχήμα Α'.14: Υπότυποι μετρήσεων ρύπων Δ'

Α'.2.5 Περιγραφή Ατμοσφαιρικών Μοντέλων



Σχήμα Α'.15: Μοντέλο για την περιγραφή των ατμοσφαιρικών μοντέλων

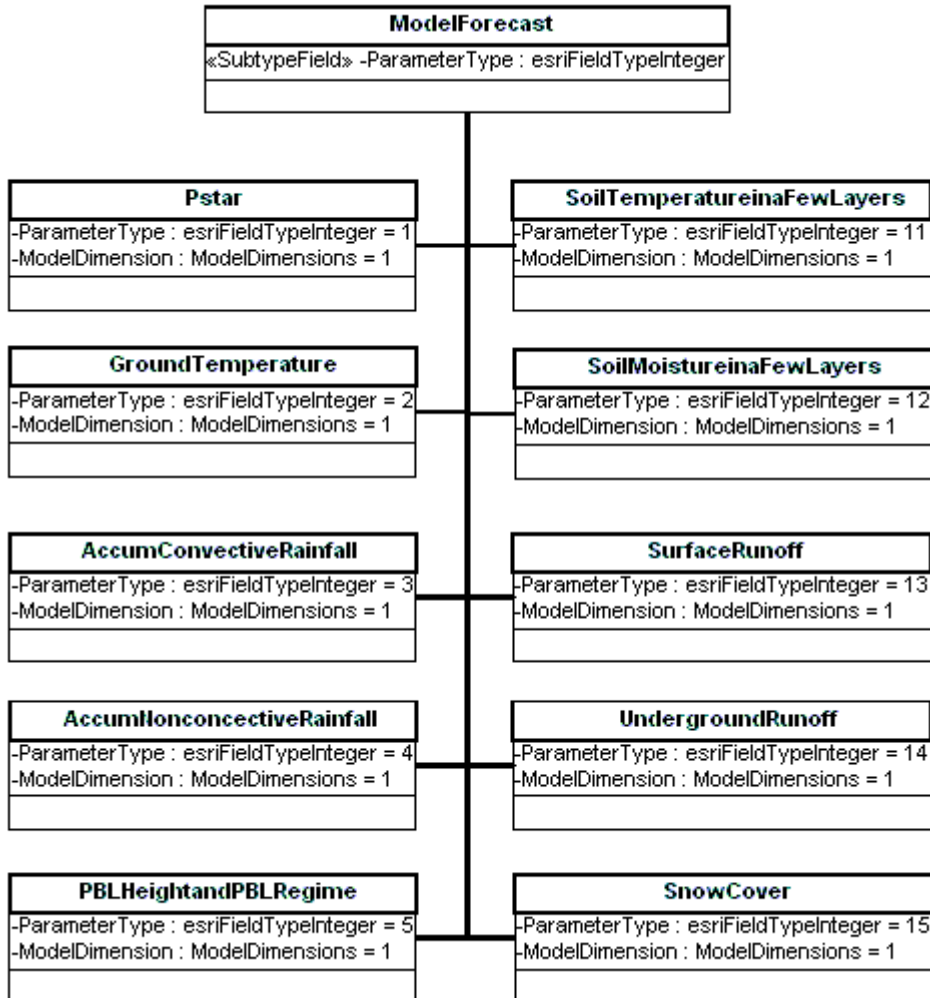
Στο σχήμα Α'.15 αποδίδεται το κεντρικό μοντέλο για τα δεδομένα που αφορούν τα αποτελέσματα από τα μοντέλα εξομοίωσης της συμπεριφοράς της ατμόσφαιρας και οι κλάσεις που περιλαμβάνει περιγράφονται στη συνέχεια :

- Object: Η κεντρική κλάση από όπου κληρονομούν όλα τα αντικείμενα της γεωγραφικής βάσης δεδομένων.
 - OBJECTID: Μοναδικό αναγνωριστικό για κάθε αντικείμενο της κλάσης (Πρωτεύον κλειδί)

- Feature: Η κεντρική κλάση από όπου κληρονομούν όλα τα αντικείμενα της γεωγραφικής βάσης δεδομένων τα οποία περιλαμβάνουν και γεωγραφική πληροφορία.
 - SHAPE: Αποθηκεύει όλη τη γεωγραφική πληροφορία για ένα γεωγραφικό αντικείμενο της βάσης (συντεταγμένες και γεωγραφική αναφορά). Ο χειρισμός των πεδίων αυτών γίνεται εσωτερικά από το ΣΔΒΔ.
- ModelGrid: Γεωγραφική κλάση που περιλαμβάνει όλα τα σημεία ενός πλέγματος πάνω στο οποίο δίνουν αποτελέσματα τα μοντέλα συμπεριφοράς της ατμόσφαιρας. Στις ιδιότητες του κάθε αντικειμένου-σημείου συμπεριλαμβάνονται και κάποιες χαρακτηριστικές σταθερές για την καλύτερη απεικόνιση της κατάστασής του, όπως φαίνεται και στη συνέχεια.
 - LandUseCategory
 - TerrainElevation
 - MapScaleFactor
 - CoriolisParameter
 - SubStrateTemperature
 - Latitude
 - Longitude
 - SnowCover
 - SeaSurface
 - SeaIce
 - LoadSession: όπως και στην κλάση MonitoringStation.
- GridMeasurementValue: Αποθηκεύει την τιμή για το αποτέλεσμα του ατμοσφαιρικού μοντέλου για ένα σημείο του πλέγματος. Οι ιδιότητες της κλάσης φαίνονται στην συνέχεια:
 - GridId: Ξένο κλειδί στην κλάση ModelGrid.

- GridMeasurmentId: Ξένο κλειδί στην κλάση GridMeasurement.
- MeasurementValue: Η αριθμητική τιμή από το αποτέλεσμα του μοντέλου για το συγκεκριμένο σημείο.
- Direction: Η διεύθυνση από το αποτέλεσμα του μοντέλου για το συγκεκριμένο σημείο στην περίπτωση που πρόκειται για διανυσματικό μέγεθος.
- LoadSession: όπως και στην προηγούμενη κλάση.
- GridMeasurement: Αφηρημένη κλάση (*Abstract*).
 - TimeID: Η χρονική στιγμή στην οποία αναφέρονται οι μετρήσεις για όλο το πλέγμα των αποτελεσμάτων.
 - ModelTypeDescription:ModelType: Ο τύπος του μοντέλου που χρησιμοποιήθηκε για τους υπολογισμούς (βλέπε και σχήμα Α'.2).
 - ModelUnits:Units: Οι μονάδες μέτρησης στις οποίες αναφέρονται τα αριθμητικά αποτελέσματα από το μοντέλο.
 - Altitude: Το υψόμετρο στο οποίο αναφέρονται τα αποτελέσματα του μοντέλου.
 - ModelDimension: Οι διαστάσεις του μοντέλου - δισδιάστατο ή τρισδιάστατο.
 - LoadSession: όπως και στην προηγούμενη κλάση.
- ModelForecast: Περιλαμβάνει στοιχεία για το ατμοσφαιρικό μοντέλο από το οποίο προέρχονται τα αποτελέσματα.
 - «SubtypeField» ParameterType: Καθορίζει τον υπότυπο του αποτελέσματος του ατμοσφαιρικού μοντέλου.

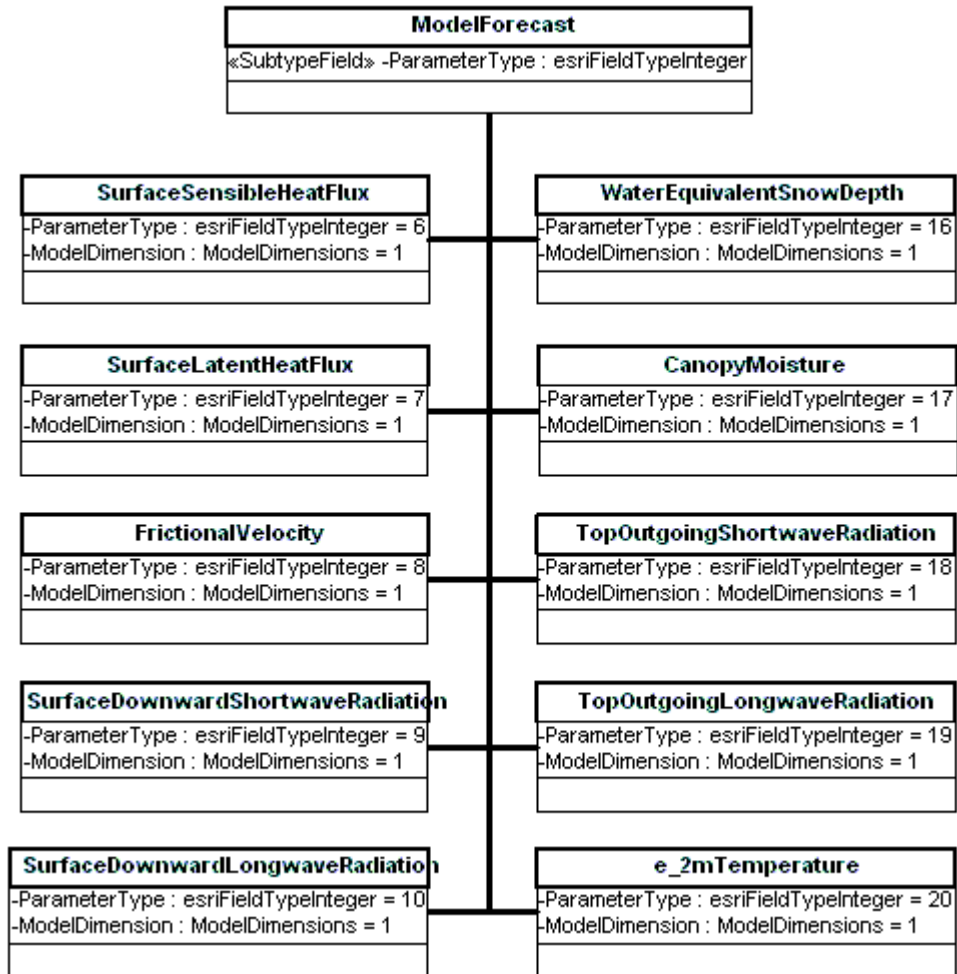
Υπότυποι Ατμοσφαιρικών Μοντέλων



Σχήμα Α'.16: Υπότυποι ατμοσφαιρικών μοντέλων Α'

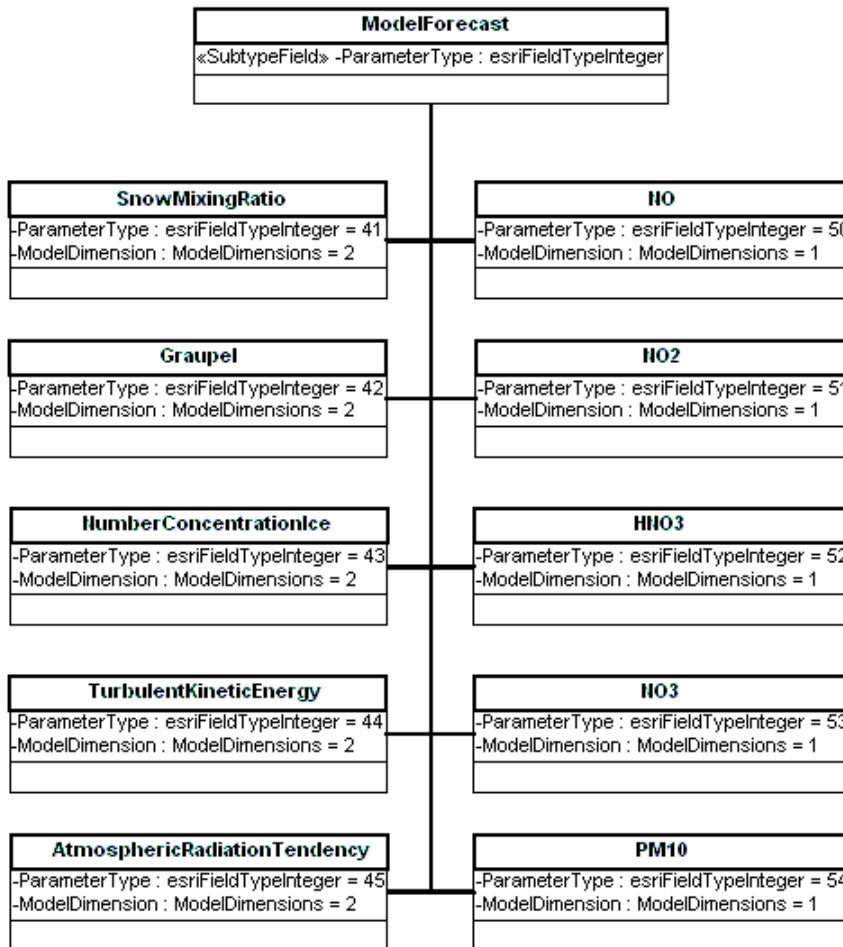
Στα σχήματα Α'.16, Α'.17, Α'.18, Α'.19, Α'.20 και Α'.21 δίνονται οι υπότυποι για όλα τα δυνατά αποτελέσματα των ατμοσφαιρικών μοντέλων, τα οποία έχουν ληφθεί υπ' όψη για την ανάπτυξη της εφαρμογής. Η απαρίθμηση των δυνατών τύπων αποτελεσμάτων δίνεται στη συνέχεια :

- Pstar
- GroundTemperature



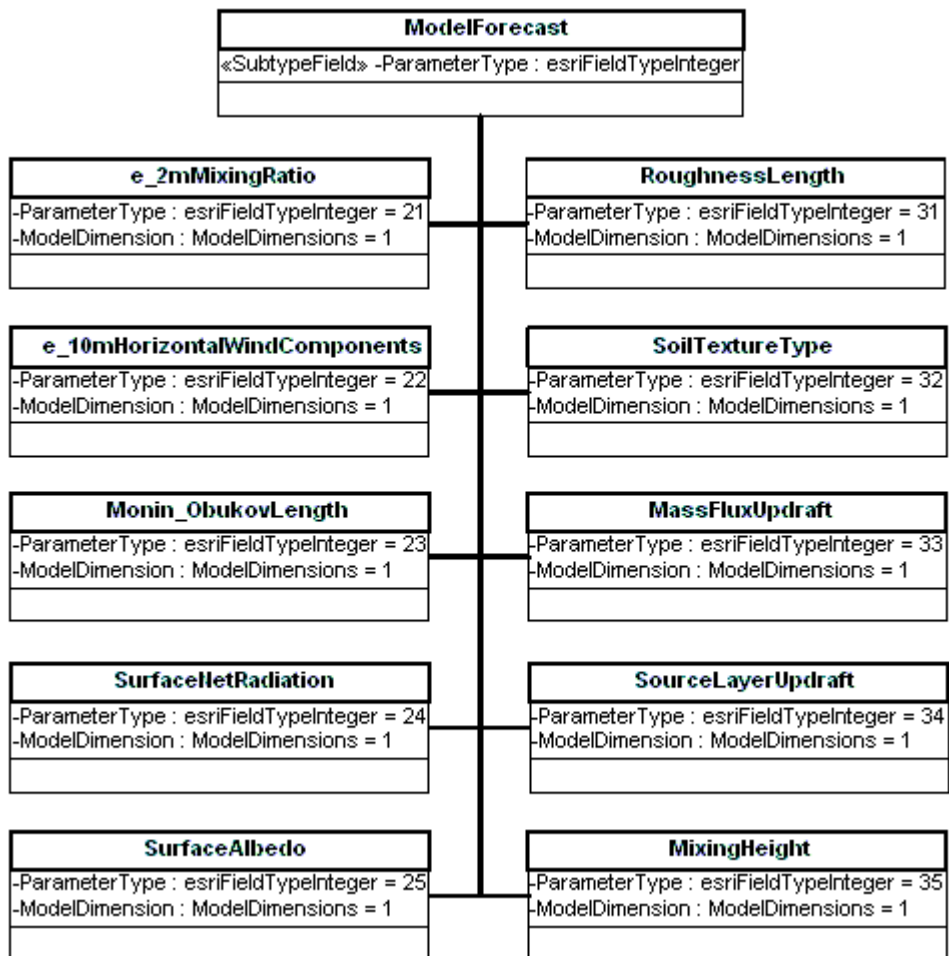
Σχήμα Α'.17: Υπότυποι ατμοσφαιρικών μοντέλων Β'

- AccumConvectiveRainfall
- AccumNonconvectiveRainfall
- PBLHeightandPBLRegime
- SoilTemperatureinaFewLayers
- SoilMoistureinaFewLayers
- SurfaceRunoff
- UndergroundRunoff



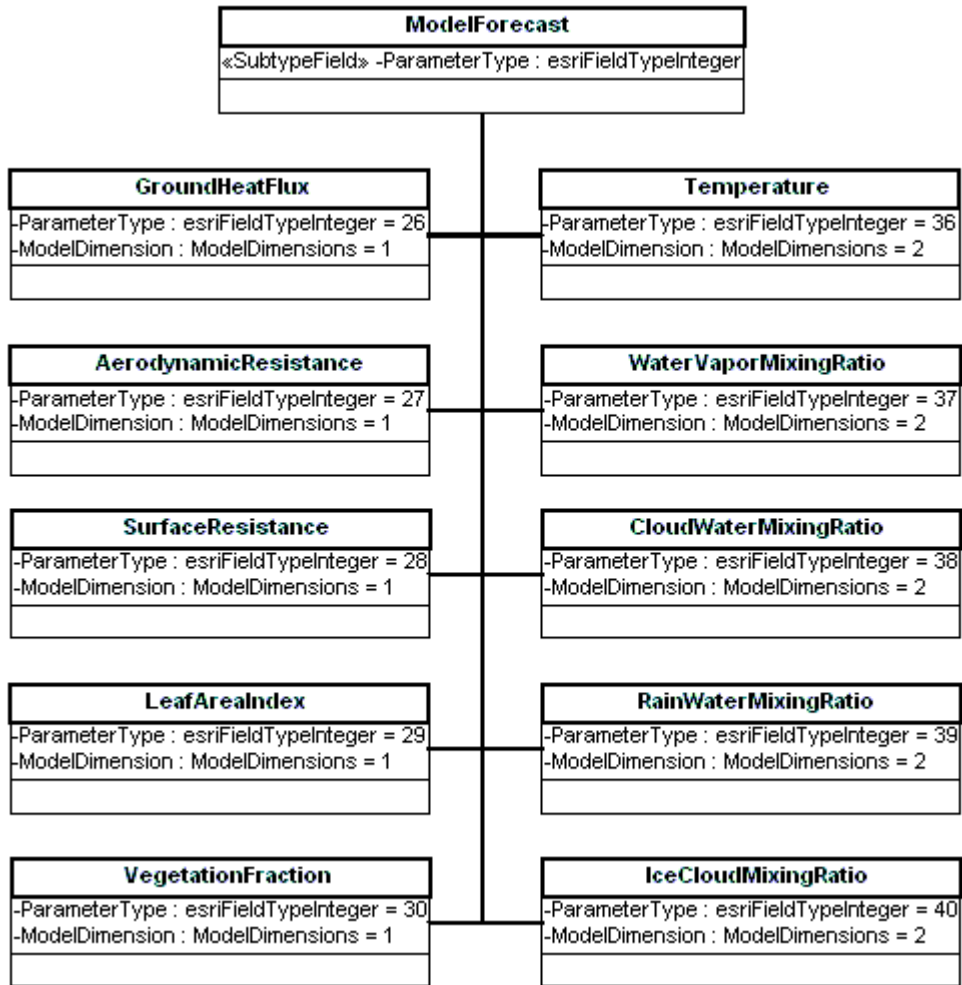
Σχήμα Α'.18: Υπότυποι ατμοσφαιρικών μοντέλων Γ

- SnowCover
- SurfaceSensibleHeatFlux
- SurfaceLatentHeatFlux
- FrictionalVelocity
- SurfaceDownwardShortwaveRadiation
- SurfaceDownwardLongwaveRadiation
- WaterEquivalentSnowDepth



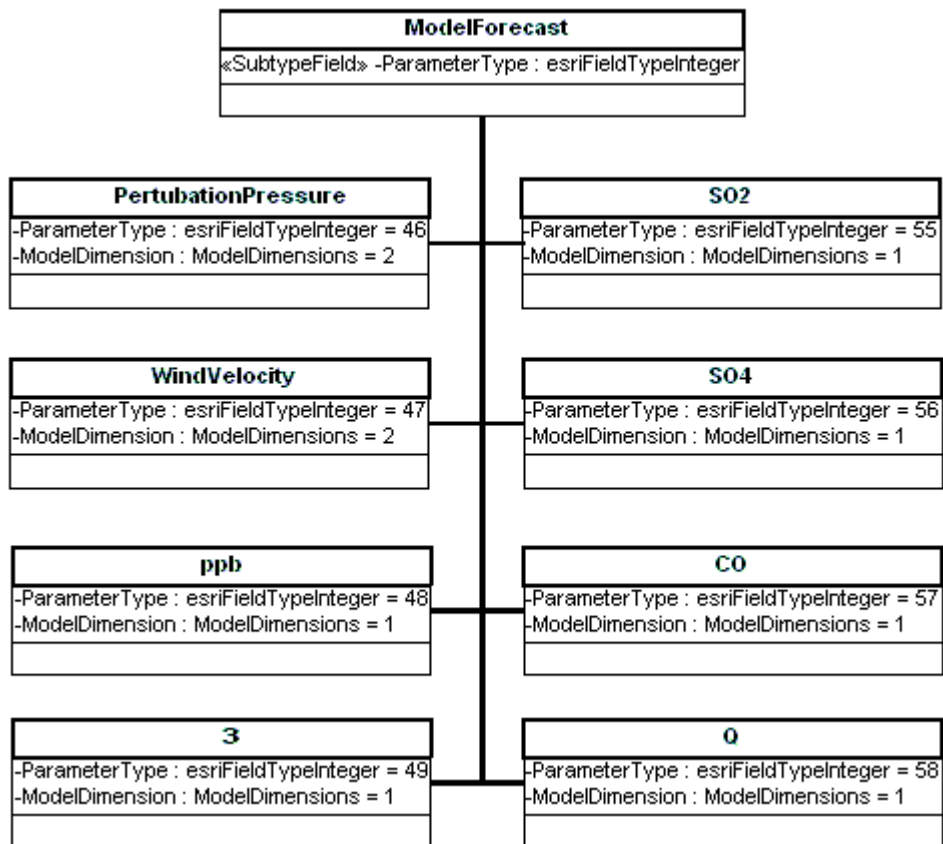
Σχήμα Α'.19: Υπότυποι ατμοσφαιρικών μοντέλων Δ

- CanopMoisture
- TopOutgoingShortwaveRadiation
- TopOutgoingLongwaveRadiation
- e_2mTemperature
- SnowMixingRatio
- Graupel
- NumberConcentrationIce



Σχήμα Α'.20: Υπότυποι ατμοσφαιρικών μοντέλων Ε'

- TurbulentKineticEnergy
- AtmosphericRadiationTendency
- NO
- NO2
- HNO3
- NO3



Σχήμα Α'.21: Υπότυποι ατμοσφαιρικών μοντέλων ΣΤ

- PM10
- e_2mMixingRatio
- e_10mHorizntalWindComponents
- Monin_ObukovLength
- SurfaceNetRadiation
- SurfaceAlbedo
- RoughnessLength
- SoilTextureType

- MassFluxUpdraft
- SourceLayerUpdraft
- MixingHeight
- GroundHeatFlux
- AerodynamicResistance
- SurfaceResistance
- LeafAreaIndex
- VegetationFraction
- Temperature
- WaterVaporMixingRatio
- CloudWaterMixingRatio
- RainWaterMixingRatio
- IceCloudMixingRatio
- PerturbationPressure
- WindVelocity
- ppb
- O3
- SO2
- SO4
- CO
- Q

Παράρτημα Β΄

Το σχήμα ορισμού των αρχείων της XML

Β΄.1 Εισαγωγή

Στο παράρτημα αυτό δίνεται το σχήμα ορισμού του αρχείου XML το οποίο χρησιμοποιείται για την περιγραφή και την διακίνηση αρχείων δεδομένων τα οποία αφορούν το σύστημα *Icaros.NET*. Αρχεία που συμμορφώνονται με το σχήμα αυτό μπορούν άμεσα να εισαχθούν στην πλατφόρμα.

Β΄.2 Παρουσίαση

Η δημιουργία του αρχείου ορισμού της XML - .xsd ακολουθεί πιστά τις προδιαγραφές που δόθηκαν για το σχήμα της βάσης δεδομένων στο παράρτημα Α΄. Οποιοσδήποτε αλλαγές γίνονται εκεί, για παράδειγμα εισαγωγή νέων τύπων μετρήσεων, πρέπει να συνοδεύονται με αντίστοιχες αλλαγές στο αρχείο αυτό. Το αρχείο αυτό συμπεριλαμβάνεται στο συνοδευτικό οπτικό δίσκο όπως περιγράφεται το παράρτημα Δ΄.

Παράρτημα Γ΄

Η υλοποίηση της βιβλιοθήκης

Γ΄.1 Εισαγωγή

Στο παράρτημα αυτό, παρουσιάζεται το συνοπτικό μοντέλο UML που ακολουθεί ο κώδικας Visual Basic (βλέπε παράρτημα Δ΄) για τις ανάγκες του συστήματος *Icaros.NET*. Περιγράφονται τα αντικείμενα που υλοποιούν όλη την απαραίτητη λειτουργικότητα για τη διαχείριση (επεξεργασία και δημιουργία) αρχείων XML τα οποία ακολουθούν το σχήμα XML όπως αυτό δόθηκε στο παράρτημα Β΄.

Γ΄.2 Παρουσίαση

Στην συνέχεια παρουσιάζονται μία, μία, όλες οι κλάσεις που υλοποιήθηκαν και περιγράφονται συνοπτικά οι ιδιότητες και οι μέθοδοι που συνιστούν την κάθε μία. Επισημαίνεται ότι ιδιότητες ή μέθοδοι οι οποίες εμφανίζουν το σύμβολο (-) μπροστά από την ονομασία τους, στα σχήματα, είναι ιδιωτικές, (Private), για την κλάση και δεν είναι ορατές στο προγραμματιστή που απλώς χρησιμοποιεί τις κλάσεις αυτές για να επεξεργαστεί τα αρχεία XML του συστήματος *Icaros.NET*. Περιγράφονται εδώ για λόγους πληρότητας.

IcarosXML::CicarosData
<pre> -NODE_NAME : Variant = "IcarosData" -m_collItems : Collection -aLoadType(2) : String -aLoadResult(2) : String +version : String -xmlschema : XMLSchemaCache40 -xmlDom : DOMDocument40 -ATTRIBUTE_version : Variant = "version" +fLoadType(in i : eLoadType) : String +fLoadResult(in i : eLoadResult) : Variant +add(in IcarosDataSource : CicarosDataSource) «accessor» +get_item(in index : Long) : CicarosDataSource «accessor» +set_item(in index : Long, in objIcarosDataSource : CicarosDataSource) «accessor» +get_count() : Long +remove(in index : Long) «accessor» +get_NewEnum() : IUnknown «accessor» +get_NodeName() : String «accessor» +get_XML() : String «accessor» +set_XML(in strNewValue : String) +CreateNode(in parent : IXMLDOMNode = Nothing) : IXMLDOMNode +LoadNode(in node : IXMLDOMNode) -Class_Initialize() -Class_Terminate() -NodeGetText(in node : IXMLDOMNode, in strNodeName : String) : String -NodeAppendAttribute(in dom : DOMDocument, in node : IXMLDOMNode, in strName : String, in varValue : Variant) -NodeAppendChildElement(in dom : DOMDocument, in node : IXMLDOMNode, in Name : String, in Value : Variant) +initIcarosData(in urn : String, in xsdfile : String) : Boolean +LoadIcarosData(in xmlfile : String) : IXMLDOMElement </pre>

Σχήμα Γ.1: CicarosData

Γ.2.1 Η κλάση CicarosData

Η κλάση αυτή παρουσιάζεται στο σχήμα Γ.1. Τα αντικείμενά της μπορούν να αποθηκεύσουν το σύνολο της πληροφορίας ενός αρχείου XML του συστήματος *Icaros.NET* και μπορεί να αξιοποιηθεί είτε για το διάβασμα ενός υπάρχοντος XML αρχείου, είτε για την δημιουργία ενός νέου. Έχει τα ακόλουθα χαρακτηριστικά:

- Ιδιότητες
 - NODE_NAME: Το όνομα του στοιχείου XML (Element)
 - m_collItems: Συλλογή αντικειμένων για την υλοποίηση της σχέσης περιεχομένου.
 - aLoadType: Πίνακας που περιέχει την κωδικοποίηση για τον τύπο δεδομένων που πρόκειται να εισαχθούν στην πλατφόρμα.

- `aLoadResult`: Πίνακας που περιέχει την κωδικοποίηση για το αποτέλεσμα εισαγωγής δεδομένων.
- `version`: Έκδοση για το σχήμα της XML που αφορά την πλατφόρμα *Icaros.NET*.
- `xmlschema`: Περιέχει την δομή που προκύπτει από το σχήμα της XML. Είναι αντικείμενο του MSXML parser.
- `xmlldom`: Περιέχει την δομή του αρχείου XML που εισάγεται ή δημιουργείται. Είναι αντικείμενο του MSXML parser.
- `ATTRIBUTE_version`: Ονομασία της ιδιότητας XML για την έκδοση του αρχείου.

- Μέθοδοι

- `fLoadType(eLoadType)`: Επιστρέφει τον τύπο των δεδομένων όπως είναι αποθηκευμένος στο πίνακα `aLoadType` σύμφωνα με την απαρίθμηση `eLoadType`.
- `fLoadResult(eLoadResult)`: Επιστρέφει το αποτέλεσμα της εισαγωγής των δεδομένων όπως είναι αποθηκευμένο στο πίνακα `aLoadResult` σύμφωνα με την απαρίθμηση `eLoadResult`.
- `add(CIcarosDataSource)`: Προσθέτει ακόμα ένα αντικείμενο τύπου `CIcarosDataSource` στην συλλογή ενός αντικειμένου της παρούσας κλάσης.
- `get_item(Long) : CIcarosDataSource`: Ανακτά ένα αντικείμενο `CIcarosDataSource` από την συλλογή ενός αντικειμένου της παρούσας κλάσης.
- `set_item(Long, CIcarosDataSource)`: Εισάγει ένα αντικείμενο `CIcarosDataSource` στη συλλογή ενός αντικειμένου της παρούσας κλάσης.
- `get_count() : Long`: Επιστρέφει το πλήθος των αντικειμένων που περιλαμβάνει η συλλογή ενός αντικειμένου της παρούσας κλάσης.
- `remove(Long)`: Αφαιρεί ένα αντικείμενο από την συλλογή των αντικειμένων ενός αντικειμένου της παρούσας κλάσης.

- `get_NewEnum()`: Απαιτείται για την υλοποίηση δομών For Each στην Visual Basic.
- `getNodeName()`: Επιστρέφει το όνομα του τρέχοντος κόμβου XML.
- `get_XML()`: Επιστρέφει τον κώδικα XML ο οποίος αντιστοιχεί στα αντικείμενα που βρίσκονται ήδη φορτωμένα στην μνήμη.
- `let_XML(String)`: Ορίζει τον κώδικα XML.
- `CreateNode()`: Δημιουργεί ένα νέο κόμβο XML.
- `LoadNode()`: Φορτώνει ένα κόμβο XML στη μνήμη.
- `Class_Initialize()`: Λειτουργίες αρχικοποίησης της κλάσης.
- `Class_Terminate()`: Λειτουργίες τερματισμού της κλάσης.
- `NodeGetText()`: Επιστρέφει την τιμή του κόμβου της XML.
- `NodeAppendAttribute()`: Εισάγει την τιμή μίας ιδιότητας στον κόμβο της XML.
- `NodeAppendChildElement()`: Δημιουργεί ένα στοιχείο παιδί για τον κόμβο της XML.
- `initIcarosData()`: Αρχικοποιεί τα αντικείμενα της κλάσης προσδιορίζοντας το αρχείο ορισμού της XML. Πρέπει να καλείται σαν πρώτη λειτουργία πριν από οτιδήποτε άλλο.
- `LoadIcarosData()`: Διαβάζει ένα αρχείο XML που συμφωνεί με το αρχείο ορισμού, και το φορτώνει στην μνήμη.

Γ'.2.2 Η κλάση **CIcarosDataSource**

Η κλάση αυτή παρουσιάζεται στο σχήμα Γ.2 και μπορεί να αποθηκεύσει τα στοιχεία ενός αρχείου με δεδομένα που αφορούν το σύστημα *Icaros.NET*, όπως για παράδειγμα ένα αρχείο ASCII με δεδομένα από ένα σταθμό μέτρησης ή με αποτελέσματα από ένα ατμοσφαιρικό μοντέλο. Έχει τα ακόλουθα χαρακτηριστικά:

- Ιδιότητες

IcarosXML::CicarosDataSource
<pre> -NODE_NAME : Variant = "IcarosDataSource" -m_collItems : Collection -m_timezone : String -aSpatialReference(7) : String -aUnits(13) : String +FileName : String -ATTRIBUTE_FileName : Variant = "FileName" +FileType : String -ATTRIBUTE_FileType : Variant = "FileType" +SpatialReference : String -ATTRIBUTE_SpatialReference : Variant = "SpatialReference" +DateConverted : Date -ATTRIBUTE_DateConverted : Variant = "DateConverted" +fSpatialReference(in i : eSpatialReference) : String +fUnits(in i : eUnits) : String «accessor» +get_timezone() : String «accessor» +let_timezone(in tz : String) +add(in obj : Object) «accessor» +get_item(in index : Long) : Object «accessor» +set_item(in index : Long, in obj : Object) «accessor» +get_count() : Long +remove(in index : Long) «accessor» +get_NewEnum() : IUnknown «accessor» +get_NodeName() : String «accessor» +get_XML() : String «accessor» +let_XML(in strNewValue : String) +CreateNode(in parent : IXMLDOMNode = Nothing) : IXMLDOMNode +LoadNode(in node : IXMLDOMNode) -Class_Initialize() -Class_Terminate() -NodeGetText(in node : IXMLDOMNode, in strNodeName : String) : String -NodeAppendAttribute(in dom : DOMDocument, in node : IXMLDOMNode, in strName : String, in varValue : Variant) -NodeAppendChildElement(in dom : DOMDocument, in node : IXMLDOMNode, in Name : String, in Value : Variant) </pre>

Σχήμα Γ.2: CicarosDataSource

- NODE_NAME: Το όνομα του στοιχείου XML (Element)
- m_collItems: Συλλογή αντικειμένων για την υλοποίηση της σχέσης περιεχομένου.
- m_timezone: Η ζώνη ώρας στην οποία αναφέρονται τα δεδομένα.
- aSpatialReference: Πίνακας που περιλαμβάνει περιγραφές για την γεωαναφορά των δεδομένων.
- aUnits: Πίνακας που περιλαμβάνει κωδικοποίηση για τις μονάδες μέτρησης των δεδομένων.

- `Filename`: Το όνομα του αρχείου από το οποίο αντλήθηκαν δεδομένα για να μετατραπούν στην μορφή XML.
- `ATTRIBUTE_FileName`: Το όνομα της ιδιότητας για το αρχείο XML.
- `FileType`: Ο τύπος του αρχείου από το οποίο αντλήθηκαν δεδομένα.
- `ATTRIBUTE_FileType`: Το όνομα της ιδιότητας για το αρχείο XML.
- `SpatialReference`: Η γεωαναφορά στην οποία αναφέρονται τα δεδομένα.
- `ATTRIBUTE_SpatialReference`: Το όνομα της ιδιότητας για το αρχείο XML.
- `DateConverted`: Η ημερομηνία κατά την οποία έγινε η μετατροπή από το αρχείο δεδομένων στην μορφή που απαιτεί το αρχείο XML.
- `ATTRIBUTE_DateConverted`: Το όνομα της ιδιότητας για το αρχείο XML.

- Μέθοδοι

- `fSpatialReference(eSpatialReference)`: Επιστρέφει την περιγραφή της γεωαναφοράς των δεδομένων σύμφωνα με την κωδικοποίηση στον πίνακα `aSpatialReference`, με βάση την απαρίθμηση `eSpatialReference`.
- `fUnits(eUnits)`: Επιστρέφει τις μονάδες μέτρησης των δεδομένων που αποθηκεύονται στα αντικείμενα της κλάσης, σύμφωνα με την κωδικοποίηση στον πίνακα `aUnits`, με βάση την απαρίθμηση `eUnits`.
- `get_timezone()`: Επιστρέφει την ζώνη ώρας στην οποία αναφέρονται τα δεδομένα των αντικειμένων της κλάσης.
- `let_timezone()`: Θέτει την ζώνη ώρας στην οποία αναφέρονται τα δεδομένα ενός αντικειμένου της κλάσης.
- `add(Object)`: Προσθέτει ακόμα ένα αντικείμενο τύπου `Object` στην συλλογή ενός αντικειμένου της παρούσας κλάσης.

- `get_item(Long)` : Object: Ανακτά ένα αντικείμενο Object από την συλλογή ενός αντικειμένου της παρούσας κλάσης.
- `set_item(Long, Object)`: Εισάγει ένα αντικείμενο Object στη συλλογή ενός αντικειμένου της παρούσας κλάσης.
- `get_count()` : Long: Επιστρέφει το πλήθος των αντικειμένων που περιλαμβάνει η συλλογή ενός αντικειμένου της παρούσας κλάσης.
- `remove(Long)`: Αφαιρεί ένα αντικείμενο από την συλλογή των αντικειμένων ενός αντικειμένου της παρούσας κλάσης.
- `get_NewEnum()`: Απαιτείται για την υλοποίηση δομών For Each στην Visual Basic.
- `get_NodeName()`: Επιστρέφει το όνομα του τρέχοντος κόμβου XML.
- `get_XML()`: Επιστρέφει τον κώδικα XML ο οποίος αντιστοιχεί στα αντικείμενα που βρίσκονται ήδη φορτωμένα στην μνήμη.
- `let_XML(String)`: Ορίζει τον κώδικα XML.
- `CreateNode()`: Δημιουργεί ένα νέο κόμβο XML.
- `LoadNode()`: Φορτώνει ένα κόμβο XML στη μνήμη.
- `Class_Initialize()`: Λειτουργίες αρχικοποίησης της κλάσης
- `Class_Terminate()`: Λειτουργίες τερματισμού της κλάσης
- `NodeGetText()`: Επιστρέφει την τιμή του κόμβου της XML.
- `NodeAppendAttribute()`: Εισάγει την τιμή μίας ιδιότητας στον κόμβο της XML.
- `NodeAppendChildElement()`: Δημιουργεί ένα στοιχείο ένα στοιχείο-ο παιδί για τον κόμβο της XML.

Γ'.2.3 Η κλάση CMonitoringStation

IcarosXML::CMonitoringStation
<pre> -NODE_NAME : Variant = "MonitoringStation" -m_collItems : Collection -aStationType(10) : String +code : String -ATTRIBUTE_CODE : Variant = "CODE" +StationDescription : Variant -ATTRIBUTE_StationDescription : Variant = "StationDescription" +StationName : Variant -ATTRIBUTE_StationName : Variant = "StationName" +StationType : Variant -ATTRIBUTE_StationType : Variant = "StationType" +Altitude : Variant -ATTRIBUTE_Altitude : Variant = "Altitude" +LATITUDE : Variant -ATTRIBUTE_LATITUDE : Variant = "LATITUDE" +LONGTITUDE : Variant -ATTRIBUTE_LONGTITUDE : Variant = "LONGTITUDE" +StartYear : Variant -ATTRIBUTE_StartYear : Variant = "StartYear" +fStationType(in i : eStationType) : String +add(in measuredquantity : CMeasuredQuantities) «accessor» +get_item(in index : Long) : CMeasuredQuantities «accessor» +set_item(in index : Long, in objmeasuredquantity : CMeasuredQuantities) «accessor» +get_count() : Long +remove(in index : Long) «accessor» +get_NewEnum() : IUnknown «accessor» +get_NodeName() : String «accessor» +get_XML() : String «accessor» +set_XML(in strNewValue : String) +CreateNode(in parent : IXMLDOMNode = Nothing) : IXMLDOMNode +LoadNode(in node : IXMLDOMNode) -Class_Initialize() -Class_Terminate() -NodeGetText(in node : IXMLDOMNode, in strNodeName : String) : String -NodeAppendAttribute(in dom : DOMDocument, in node : IXMLDOMNode, in strName : String, in varValue : Variant) -NodeAppendChildElement(in dom : DOMDocument, in node : IXMLDOMNode, in Name : String, in Value : Variant) </pre>

Σχήμα Γ.3: CMonitoringStation

Η κλάση αυτή παρουσιάζεται στο σχήμα Γ.3 και κάθε αντικείμενό της αντιστοιχεί σε ένα επίγειο σταθμό μέτρησης, από όπου προέρχονται δεδομένα για το σύστημα *Icaros.NET*. Έχει τα ακόλουθα χαρακτηριστικά:

- Ιδιότητες
 - NODE_NAME: Το όνομα του στοιχείου XML (Element)

- `m_collItems`: Συλλογή αντικειμένων για την υλοποίηση της σχέσης περιεχομένου.
- `aStationType`: Πίνακας με την κωδικοποίηση των σταθμών μέτρησης σε σχέση με τον αστικό ιστό.
- `code`: Ο κωδικός του σταθμού μέτρησης
- `ATTRIBUTE_CODE`: Το όνομα της ιδιότητας για το αρχείο XML.
- `StationDescription`: Μία σύντομη περιγραφή ή εκτεταμένη ονομασία του σταθμού μέτρησης.
- `ATTRIBUTE_StationDescription`: Το όνομα της ιδιότητας για το αρχείο XML.
- `StationName`: Το όνομα του σταθμού μέτρησης συνοπτικά.
- `ATTRIBUTE_StationName`: Το όνομα της ιδιότητας για το αρχείο XML.
- `StationType`: Ο τύπος του σταθμού μέτρησης σε σχέση με την θέση του στον αστικό ιστό.
- `ATTRIBUTE_StationType`: Το όνομα της ιδιότητας για το αρχείο XML.
- `Altitude`: Το υψόμετρο λειτουργίας του σταθμού μέτρησης.
- `ATTRIBUTE_Altitude`: Το όνομα της ιδιότητας για το αρχείο XML.
- `Latitude`: Το γεωγραφικό πλάτος της θέσης του σταθμού μέτρησης.
- `ATTRIBUTE_Latitude`: Το όνομα της ιδιότητας για το αρχείο XML.
- `Longitude`: Το γεωγραφικό μήκος της θέσης του σταθμού μέτρησης.
- `ATTRIBUTE_Longitude`: Το όνομα της ιδιότητας για το αρχείο XML.
- `StartYear`: Η χρονολογία έναρξης λειτουργίας του σταθμού.
- `ATTRIBUTE_StartYear`: Το όνομα της ιδιότητας για το αρχείο XML.

- Μέθοδοι
 - `fStationType(eStationType)`: Επιστρέφει τον τύπο του σταθμού μέτρησης σε σχέση με την θέση του στον αστικό ιστό, σύμφωνα με την κωδικοποίηση του πίνακα `aStationType`, όπως περιγράφεται από την απαρίθμηση `eStationType`.
 - `add(CMeasuredQuantities)`: Προσθέτει ακόμα ένα αντικείμενο τύπου `CMeasuredQuantities` στην συλλογή ενός αντικειμένου της παρούσας κλάσης.
 - `get_item(Long) : CMeasuredQuantities`: Ανακτά ένα αντικείμενο `CMeasuredQuantities` από την συλλογή ενός αντικειμένου της παρούσας κλάσης.
 - `set_item(Long, CMeasuredQuantities)`: Εισάγει ένα αντικείμενο `CMeasuredQuantities` στη συλλογή ενός αντικειμένου της παρούσας κλάσης.
 - `get_count() : Long`: Επιστρέφει το πλήθος των αντικειμένων που περιλαμβάνει η συλλογή ενός αντικειμένου της παρούσας κλάσης.
 - `remove(Long)`: Αφαιρεί ένα αντικείμενο από την συλλογή των αντικειμένων ενός αντικειμένου της παρούσας κλάσης.
 - `get_NewEnum()`: Απαιτείται για την υλοποίηση δομών `For Each` στην `Visual Basic`.
 - `get_NodeName()`: Επιστρέφει το όνομα του τρέχοντος κόμβου XML.
 - `get_XML()`: Επιστρέφει τον κώδικα XML ο οποίος αντιστοιχεί στα αντικείμενα που βρίσκονται ήδη φορτωμένα στην μνήμη.
 - `let_XML(String)`: Ορίζει τον κώδικα XML.
 - `CreateNode()`: Δημιουργεί ένα νέο κόμβο XML.
 - `LoadNode()`: Φορτώνει ένα κόμβο XML στη μνήμη.
 - `Class_Initialize()`: Λειτουργίες αρχικοποίησης της κλάσης
 - `Class_Terminate()`: Λειτουργίες τερματισμού της κλάσης
 - `NodeGetText()`: Επιστρέφει την τιμή του κόμβου της XML.

- NodeAppendAttribute(): Εισάγει την τιμή μίας ιδιότητας στον κόμβο της XML.
- NodeAppendChildElement(): Δημιουργεί ένα στοιχείο ένα στοιχείο-ο παιδί για τον κόμβο της XML.

Γ'.2.4 Η κλάση CicarosHelper

IcarosXML::CicarosHelper
<pre> +julian2date(in julian : Integer, in year : String) : Date +iso8601tz2dt(in isotime : String, inout timezone : String) : Date +dt2iso8601tz(in dtime : Date, in timezone : String = "") : String +daylight(in dd : Date) : Date </pre>

Σχήμα Γ.4: CicarosHelper

Η κλάση αυτή παρουσιάζεται στο σχήμα Γ.4. Αποτελεί μία βοηθητική κλάση η οποία παρέχει μόνο κάποιες συναρτήσεις μετατροπής. Έχει τα ακόλουθα χαρακτηριστικά:

- Ιδιότητες
 - Καμία
- Μέθοδοι
 - julian2date(): Μετατροπή Ιουλιανής ημερομηνία σε ημερομηνία της μορφής Ημέρα/Μήνας/Έτος.
 - iso8601tz2dt(): Μετατροπή ημερομηνίας-ώρας από την μορφή του προτύπου ISO8601tz σε ημερομηνία της μορφής Ημέρα/Μήνας/Έτος, με συνυπολογισμό της παγκόσμιας διαφοράς ώρας.
 - dt2iso8601tz(): Μετατροπή ημερομηνίας-ώρας από την μορφή Ημέρα/Μήνας/Έτος σε ημερομηνία της μορφής του προτύπου ISO8601tz, με συνυπολογισμό της παγκόσμιας διαφοράς ώρας.
 - daylight(): Μετατροπή ημερομηνίας-ώρας με βάση το συνυπολογισμό θερινής-χειμερινής ώρας για την κανονικοποίηση των μεγεθών σε σχέση με την παγκόσμια ώρα GMT

Γ'.2.5 Η κλάση CMeasuredQuantities

Η κλάση αυτή παρουσιάζεται στο σχήμα Γ.5 και περιλαμβάνει σαν αντικείμενα τους διαφορετικούς τύπους μετρήσεων που μπορεί να πραγματοποιήσει ή πραγματοποιεί ένας σταθμός. Έχει τα ακόλουθα χαρακτηριστικά:

- Ιδιότητες
 - NODE_NAME: Το όνομα του στοιχείου XML (Element)
 - m_collItems: Συλλογή αντικειμένων για την υλοποίηση της σχέσης περιεχομένου.
 - aMeasurementType: Πίνακας με την κωδικοποίηση για την κατηγορία της μέτρησης.
 - aMeasurementSubType: Πίνακας με την κωδικοποίηση για τον υπότυπο της μέτρησης.
 - aType: Πίνακας με την κωδικοποίηση για τον τύπο της μέτρησης.
 - aPrinciple: Πίνακας με την κωδικοποίηση για την αρχή λειτουργίας του οργάνου που πραγματοποιεί την μέτρηση.
 - MeasurementType: Η κατηγορία στην οποία ανήκει η μέτρηση (Ρύποι ή Μετεωρολογικά)
 - ATTRIBUTE_MeasurementType: Το όνομα της ιδιότητας για το αρχείο XML.
 - MeasurementSubType: Ο υπότυπος της μέτρησης, δηλαδή η μετρούμενη ποσότητα.
 - ATTRIBUTE_MeasurementSubType: Το όνομα της ιδιότητας για το αρχείο XML.
 - Units: Οι μονάδες μέτρησης των μετρούμενων μεγεθών.
 - ATTRIBUTE_Units: Το όνομα της ιδιότητας για το αρχείο XML.
 - DetectionLimitLow: Το κάτω όριο μέτρησης του οργάνου.
 - ATTRIBUTE_DetectionLimitLow: Το όνομα της ιδιότητας για το αρχείο XML.

- `DetectionLimitHigh`: Το πάνω όριο μέτρησης του οργάνου.
- `ATTRIBUTE_DetectionLimitHigh`: Το όνομα της ιδιότητας για το αρχείο XML.
- `StartYear`: Η χρονολογία έναρξης λειτουργίας του οργάνου.
- `ATTRIBUTE_StartYear`: Το όνομα της ιδιότητας για το αρχείο XML.
- `Principle`: Η αρχή λειτουργίας του οργάνου.
- `ATTRIBUTE_Principle`: Το όνομα της ιδιότητας για το αρχείο XML.
- `measurementValueType`: Ο τύπος των μετρούμενων ποσοτήτων.
- `ATTRIBUTE_MeasurementValueType`: Το όνομα της ιδιότητας για το αρχείο XML.

- Μέθοδοι

- `fMeasurementType(eMeasurementType)`: Επιστρέφει την κατηγορία της μέτρησης όπως κωδικοποιείται στο πίνακα `aMeasurementType` με βάση την απαρίθμηση `eMeasurementType`.
- `fMeasurementSubType(eMeasurementSubType)`: Επιστρέφει τον τύπο της μέτρησης σύμφωνα με το πίνακα `aMeasurementSubType` με βάση την απαρίθμηση `eMeasurementSubType`.
- `fType(eType)`: Επιστρέφει τον τύπο των μετρούμενων ποσοτήτων όπως κωδικοποιείται στον πίνακα `aType` με βάση την απαρίθμηση `eType`.
- `fPrinciple(ePrinciple)`: Επιστρέφει την περιγραφή της αρχής λειτουργίας του οργάνου μέτρησης, όπως κωδικοποιείται στον πίνακα `aPrinciple` με βάση την απαρίθμηση `ePrinciple`.
- `add(CMeasurement)`: Προσθέτει ακόμα ένα αντικείμενο τύπου `CMeasurement` στην συλλογή ενός αντικειμένου της παρούσας κλάσης.
- `get_item(Long) : CMeasurement`: Ανακτά ένα αντικείμενο `CMeasurement` από την συλλογή ενός αντικειμένου της παρούσας κλάσης.

- `set_item(Long, CMeasurement)`: Εισάγει ένα αντικείμενο `CMeasurement` στη συλλογή ενός αντικειμένου της παρούσας κλάσης.
- `get_count()` : `Long`: Επιστρέφει το πλήθος των αντικειμένων που περιλαμβάνει η συλλογή ενός αντικειμένου της παρούσας κλάσης.
- `remove(Long)`: Αφαιρεί ένα αντικείμενο από την συλλογή των αντικειμένων ενός αντικειμένου της παρούσας κλάσης.
- `get_NewEnum()`: Απαιτείται για την υλοποίηση δομών `For Each` στην `Visual Basic`.
- `get_NodeName()`: Επιστρέφει το όνομα του τρέχοντος κόμβου XML.
- `get_XML()`: Επιστρέφει τον κώδικα XML ο οποίος αντιστοιχεί στα αντικείμενα που βρίσκονται ήδη φορτωμένα στην μνήμη.
- `let_XML(String)`: Ορίζει τον κώδικα XML.
- `CreateNode()`: Δημιουργεί ένα νέο κόμβο XML.
- `LoadNode()`: Φορτώνει ένα κόμβο XML στη μνήμη.
- `Class_Initialize()`: Λειτουργίες αρχικοποίησης της κλάσης
- `Class_Terminate()`: Λειτουργίες τερματισμού της κλάσης
- `NodeGetText()`: Επιστρέφει την τιμή του κόμβου της XML.
- `NodeAppendAttribute()`: Εισάγει την τιμή μίας ιδιότητας στον κόμβο της XML.
- `NodeAppendChildElement()`: Δημιουργεί ένα στοιχείο ένα στοιχείο-ο παιδί για τον κόμβο της XML.

IcarosXML::CMeasuredQuantities
<pre> -NODE_NAME : Variant = "MeasuredQuantities" -m_collItems : Collection -aMeasurementType(2) : String -aMeasurementSubType(28) : String -aType(29) : String -aPrinciple(12) : String +MeasurementType : String -ATTRIBUTE_measurementType : Variant = "measurementType" +MeasurementSubType : String -ATTRIBUTE_measurementSubType : Variant = "measurementSubType" +Units : Variant -ATTRIBUTE_Units : Variant = "Units" +DetectionLimitLow : Variant -ATTRIBUTE_DetectionLimitLow : Variant = "DetectionLimitLOW" +DetectionLimitHigh : Variant -ATTRIBUTE_DetectionLimitHigh : Variant = "DetectionLimitHIGH" +StartYear : Variant -ATTRIBUTE_StartYear : Variant = "StartYear" +Principle : Variant -ATTRIBUTE_Principle : Variant = "Principle" +measurementValueType : Variant -ATTRIBUTE_measurementValueType : Variant = "MeasurementValueType" +fMeasurementType(in i : eMeasurementType) : Variant +fMeasurementSubType(in i : eMeasurementSubType) : Variant +fType(in i : eType) : String +fPrinciple(in i : ePrinciple) : String +add(in measurement : Cmeasurement) «accessor» +get_item(in index : Long) : Cmeasurement «accessor» +set_item(in index : Long, in objmeasurement : Cmeasurement) «accessor» +get_count() : Long +remove(in index : Long) «accessor» +get_NewEnum() : IUnknown «accessor» +get_NodeName() : String «accessor» +get_XML() : String «accessor» +set_XML(in strNewValue : String) +CreateNode(in parent : IXMLDOMNode = Nothing) : IXMLDOMNode +LoadNode(in node : IXMLDOMNode) -Class_Initialize() -Class_Terminate() -NodeGetText(in node : IXMLDOMNode, in strNodeName : String) : String -NodeAppendAttribute(in dom : DOMDocument, in node : IXMLDOMNode, in strName : String, in varValue : Variant) -NodeAppendChildElement(in dom : DOMDocument, in node : IXMLDOMNode, in Name : String, in Value : Variant) </pre>

Σχήμα Γ.5: CMeasuredQuantities

Γ'.2.6 Η κλάση CModelGrid

IcarosXML: CModelGrid
<pre> -NODE_NAME : Variant = "ModelGrid" +LONGITUDE : Double -ATTRIBUTE_LONGITUDE : Variant = "Longitude" +LATITUDE : Double -ATTRIBUTE_LATITUDE : Variant = "Latitude" +LandUseCategory : Variant -ATTRIBUTE_LandUseCategory : Variant = "LandUseCategory" +TerrainElevation : Variant -ATTRIBUTE_TerrainElevation : Variant = "TerrainElevation" +MapScaleFactor : Variant -ATTRIBUTE_MapScaleFactor : Variant = "MapScaleFactor" +CoriolisParameter : Variant -ATTRIBUTE_CoriolisParameter : Variant = "CoriolisParameter" +SubstrateTemperature : Variant -ATTRIBUTE_SubstrateTemperature : Variant = "SubstrateTemperature" +SnowCover : Variant -ATTRIBUTE_SnowCover : Variant = "SnowCover" +SeaSurfaceTemperature : Variant -ATTRIBUTE_SeaSurfaceTemperature : Variant = "SeaSurfaceTemperature" +Sealce : Variant -ATTRIBUTE_Sealce : Variant = "Sealce" «accessor» +get_NodeName() : String «accessor» +get_XML() : String «accessor» +let_XML(in strNewValue : String) +CreateNode(in parent : IXMLDOMNode = Nothing) : IXMLDOMNode +LoadNode(in node : IXMLDOMNode) -NodeGetText(in node : IXMLDOMNode, in strNodeName : String) : String -NodeAppendAttribute(in dom : DOMDocument, in node : IXMLDOMNode, in strName : String, in varValue : Variant) -NodeAppendChildElement(in dom : DOMDocument, in node : IXMLDOMNode, in Name : String, in Value : Variant) </pre>

Σχήμα Γ.6: CModelGrid

Η κλάση αυτή παρουσιάζεται στο σχήμα Γ.6 και περιλαμβάνει τα αντικείμενα που ανήκουν στο πλέγμα των σημείων για τα οποία δίνει αποτελέσματα ένα μοντέλο της ατμοσφαιρικής συμπεριφοράς. Έχει τα ακόλουθα χαρακτηριστικά:

- Ιδιότητες
 - NODE_NAME: Το όνομα του στοιχείου XML (Element)
 - LONGITUDE: Το γεωγραφικό μήκος στο οποίο βρίσκεται το σημείο του πλέγματος.

- ATTRIBUTE_LONGITUDE: Το όνομα της ιδιότητας για το αρχείο XML.
- LATITUDE: Το γεωγραφικό πλάτος στο οποίο βρίσκεται το σημείο του πλέγματος.
- ATTRIBUTE_LATITUDE: Το όνομα της ιδιότητας για το αρχείο XML.
- LandUseCategory: Η χρήση γης για το συγκεκριμένο σημείο.
- ATTRIBUTE_LandUseCategory: Το όνομα της ιδιότητας για το αρχείο XML.
- TerrainElevation: Η ανύψωση του επιπέδου.
- ATTRIBUTE_TerrainElevation: Το όνομα της ιδιότητας για το αρχείο XML.
- MapScaleFactor: Συντελεστής κλίμακας χάρτη.
- ATTRIBUTE_MapScaleFactor: Το όνομα της ιδιότητας για το αρχείο XML.
- CoriolisParameter: Παράμετρος Coriolis.
- ATTRIBUTE_CoriolisParameter: Το όνομα της ιδιότητας για το αρχείο XML.
- SubStrateTemperature: Θερμοκρασία υποστρώματος.
- ATTRIBUTE_SubStrateTemperature: Το όνομα της ιδιότητας για το αρχείο XML.
- SnowCover: Κάλυψη Χιονιού.
- ATTRIBUTE_SnowCover: Το όνομα της ιδιότητας για το αρχείο XML.
- SeaSurfaceTemperature: Θερμοκρασία επιφανείας της θάλασσας.
- ATTRIBUTE_SeaSurfaceTemperature: Το όνομα της ιδιότητας για το αρχείο XML.
- SeaIce: Θαλάσσια παγοκάλυψη.

– ATTRIBUTE_SeaIce: Το όνομα της ιδιότητας για το αρχείο XML.

- Μέθοδοι

- get_NodeName(): Επιστρέφει το όνομα του τρέχοντος κόμβου XML.

- get_XML(): Επιστρέφει τον κώδικα XML ο οποίος αντιστοιχεί στα αντικείμενα που βρίσκονται ήδη φορτωμένα στην μνήμη.

- let_XML(String): Ορίζει τον κώδικα XML.

- CreateNode(): Δημιουργεί ένα νέο κόμβο XML.

- LoadNode(): Φορτώνει ένα κόμβο XML στη μνήμη.

- NodeGetText(): Επιστρέφει την τιμή του κόμβου της XML.

- NodeAppendAttribute(): Εισάγει την τιμή μίας ιδιότητας στον κόμβο της XML.

- NodeAppendChildElement(): Δημιουργεί ένα στοιχείο ένα στοιχείο-ο παιδί για τον κόμβο της XML.

Γ.2.7 Η κλάση CModelResult

IcarosXML::CModelResult
-NODE_NAME : Variant = "ModelResult" +Value : Double +LONGTITUDE : Double -ATTRIBUTE_LONGTITUDE : Variant = "LONGTITUDE" +LATITUDE : Double -ATTRIBUTE_LATITUDE : Variant = "LATITUDE" +Direction : Variant -ATTRIBUTE_Direction : Variant = "Direction"
«accessor» +get_NodeName() : String «accessor» +get_XML() : String «accessor» +set_XML(in strNewValue : String) +CreateNode(in parent : IXMLDOMNode = Nothing) : IXMLDOMNode +LoadNode(in node : IXMLDOMNode) -NodeGetText(in node : IXMLDOMNode, in strNodeName : String) : String -NodeAppendAttribute(in dom : DOMDocument, in node : IXMLDOMNode, in strName : String, in varValue : Variant) -NodeAppendChildElement(in dom : DOMDocument, in node : IXMLDOMNode, in Name : String, in Value : Variant)

Σχήμα Γ.7: CModelResult

Η κλάση αυτή παρουσιάζεται στο σχήμα Γ.7 και αποθηκεύει την τιμή που έχει υπολογιστεί από το μοντέλο για ένα μέγεθος και ένα σημείο. Έχει τα ακόλουθα χαρακτηριστικά:

- Ιδιότητες
 - NODE_NAME: Το όνομα του στοιχείου XML (Element)
 - Value: Η υπολογιζόμενη τιμή από το μοντέλο.
 - LONGTITUDE: Το γεωγραφικό μήκος του σημείου στο οποίο αφορά η υπολογιζόμενη τιμή.
 - ATTRIBUTE_LONGTITUDE: Το όνομα της ιδιότητας για το αρχείο XML.
 - LATITUDE: Το γεωγραφικό πλάτος του σημείου στο οποίο αφορά η υπολογιζόμενη τιμή.
 - ATTRIBUTE_LATITUDE: Το όνομα της ιδιότητας για το αρχείο XML.
 - Direction: Η διεύθυνση του μετρούμενου μεγέθους, αν πρόκειται για διανυσματικό.

– ATTRIBUTE_Direction: Το όνομα της ιδιότητας για το αρχείο XML.

- Μέθοδοι

- get_NodeName(): Επιστρέφει το όνομα του τρέχοντος κόμβου XML.

- get_XML(): Επιστρέφει τον κώδικα XML ο οποίος αντιστοιχεί στα αντικείμενα που βρίσκονται ήδη φορτωμένα στην μνήμη.

- let_XML(String): Ορίζει τον κώδικα XML.

- CreateNode(): Δημιουργεί ένα νέο κόμβο XML.

- LoadNode(): Φορτώνει ένα κόμβο XML στη μνήμη.

- NodeGetText(): Επιστρέφει την τιμή του κόμβου της XML.

- NodeAppendAttribute(): Εισάγει την τιμή μίας ιδιότητας στον κόμβο της XML.

- NodeAppendChildElement(): Δημιουργεί ένα στοιχείο ένα στοιχείο-ο παιδί για τον κόμβο της XML.

Γ'.2.8 Η κλάση CModelResults

Η κλάση αυτή παρουσιάζεται στο σχήμα Γ.8 αποθηκεύει τις παραμέτρους που αφορούν το σύνολο των σημείων για τα οποία έχουν υπολογιστεί τιμές από ένα μοντέλο. Έχει τα ακόλουθα χαρακτηριστικά:

- Ιδιότητες
 - NODE_NAME: Το όνομα του στοιχείου XML (Element)
 - m_collItems: Συλλογή αντικειμένων για την υλοποίηση της σχέσης περιεχομένου.
 - aUnits: Πίνακας που περιέχει τις κωδικοποιήσεις για τις μονάδες μέτρησης των μεγεθών.
 - aParameterType: Πίνακας με την κωδικοποίηση για τον τύπο της μετρούμενης ποσότητας.
 - aModelDimension: Πίνακας με την κωδικοποίηση για τις διαστάσεις του μοντέλου.
 - aModelTypeDescription: Πίνακας με την κωδικοποίηση για τον τύπο του μοντέλου.
 - m_timezone: Η ζώνη ώρας στην οποία αναφέρονται τα δεδομένα.
 - TimePoint: Η χρονική στιγμή για την οποία έχουν υπολογιστεί τα αποτελέσματα του μοντέλου.
 - ATTRIBUTE_TimePoint: Το όνομα της ιδιότητας για το αρχείο XML.
 - ModelTypeDescription: Ο τύπος του μοντέλου από το οποίο προήλθαν τα αποτελέσματα.
 - ATTRIBUTE_ModelTypeDescription: Το όνομα της ιδιότητας για το αρχείο XML.
 - ModelDimension: Οι διαστάσεις του μοντέλου.
 - ATTRIBUTE_ModelDimension: Το όνομα της ιδιότητας για το αρχείο XML.

- ModelUnits: Οι μονάδες μέτρησης των ποσοτήτων που υπολόγισε το μοντέλο.
 - ATTRIBUTE_ModelUnits: Το όνομα της ιδιότητας για το αρχείο XML.
 - ParameterType: Ο τύπος των μετρούμενων ποσοτήτων.
 - ATTRIBUTE_ParameterType: Το όνομα της ιδιότητας για το αρχείο XML.
 - Altitude: Το υψόμετρο στο οποίο αναφέρονται οι υπολογιζόμενες ποσότητες.
 - ATTRIBUTE_Altitude: Το όνομα της ιδιότητας για το αρχείο XML.
- Μέθοδοι
 - fUnits(eUnits): Επιστρέφει τις μονάδες μέτρησης των ποσοτήτων σύμφωνα με την κωδικοποίηση στο πίνακα aUnits με βάση την απαρίθμηση eUnits.
 - fParameterType(eParameterType): Επιστρέφει τον τύπο των υπολογιζόμενων ποσοτήτων σύμφωνα με την κωδικοποίηση στον πίνακα aParameterType με βάση την απαρίθμηση eParameterType.
 - fModelDimension(eModelDimension): Επιστρέφει τις διαστάσεις του μοντέλου σύμφωνα με την κωδικοποίηση στο πίνακα aModelDimension με βάση την απαρίθμηση eModelDimension.
 - fModelTypeDescription(eModelTypeDescription): Επιστρέφει τον τύπο του μοντέλου σύμφωνα με την κωδικοποίηση στο πίνακα aModelTypeDescription με βάση την απαρίθμηση eModelTypeDescription.
 - get_timezone(): Επιστρέφει την ζώνη ώρας στην οποία αναφέρονται τα δεδομένα των αντικειμένων της κλάσης.
 - let_timezone(): Θέτει την ζώνη ώρας στην οποία αναφέρονται τα δεδομένα ενός αντικειμένου της κλάσης.

- `add(CModelResult)`: Προσθέτει ακόμα ένα αντικείμενο τύπου `CModelResult` στην συλλογή ενός αντικειμένου της παρούσας κλάσης.
- `get_item(Long) : CModelResult`: Ανακτά ένα αντικείμενο `CModelResult` από την συλλογή ενός αντικειμένου της παρούσας κλάσης.
- `set_item(Long, CModelResult)`: Εισάγει ένα αντικείμενο `CModelResult` στη συλλογή ενός αντικειμένου της παρούσας κλάσης.
- `get_count() : Long`: Επιστρέφει το πλήθος των αντικειμένων που περιλαμβάνει η συλλογή ενός αντικειμένου της παρούσας κλάσης.
- `remove(Long)`: Αφαιρεί ένα αντικείμενο από την συλλογή των αντικειμένων ενός αντικειμένου της παρούσας κλάσης.
- `get_NewEnum()`: Απαιτείται για την υλοποίηση δομών `For Each` στην `Visual Basic`.
- `get_NodeName()`: Επιστρέφει το όνομα του τρέχοντος κόμβου XML.
- `get_XML()`: Επιστρέφει τον κώδικα XML ο οποίος αντιστοιχεί στα αντικείμενα που βρίσκονται ήδη φορτωμένα στην μνήμη.
- `let_XML(String)`: Ορίζει τον κώδικα XML.
- `CreateNode()`: Δημιουργεί ένα νέο κόμβο XML.
- `LoadNode()`: Φορτώνει ένα κόμβο XML στη μνήμη.
- `Class_Initialize()`: Λειτουργίες αρχικοποίησης της κλάσης
- `Class_Terminate()`: Λειτουργίες τερματισμού της κλάσης
- `NodeGetText()`: Επιστρέφει την τιμή του κόμβου της XML.
- `NodeAppendAttribute()`: Εισάγει την τιμή μίας ιδιότητας στον κόμβο της XML.
- `NodeAppendChildElement()`: Δημιουργεί ένα στοιχείο ένα στοιχείο-ο παιδί για τον κόμβο της XML.

IcarosXML::CModelResults
-NODE_NAME : Variant = "ModelResults" -m_collItems : Collection -aUnits(13) : String -aParameterType(58) : String -aModelDimension(2) : String -aModelTypeDescription(4) : String -m_timezone : String +TimePoint : Date -ATTRIBUTE_TimePoint : Variant = "TimePoint" +ModelTypeDescription : Variant -ATTRIBUTE_ModelTypeDescription : Variant = "ModelTypeDescription" +ModelDimension : Variant -ATTRIBUTE_ModelDimension : Variant = "ModelDimension" +ModelUnits : Variant -ATTRIBUTE_ModelUnits : Variant = "ModelUnits" +ParameterType : Variant -ATTRIBUTE_ParameterType : Variant = "ParameterType" +Altitude : Variant -ATTRIBUTE_Altitude : Variant = "Altitude"
+fUnits(in i : eUnits) : String +fParameterType(in i : eParameterType) : String +fModelDimension(in i : eModelDimension) : String +fModelTypeDescription(in i : eModelTypeDescription) : Variant «accessor» +get_timezone() : String «accessor» +set_timezone(in tz : String) +add(in ModelResult : CModelResult) «accessor» +get_item(in index : Long) : CModelResult «accessor» +set_item(in index : Long, in objModelResult : CModelResult) «accessor» +get_count() : Long +remove(in index : Long) «accessor» +get_NewEnum() : IUnknown «accessor» +get_NodeName() : String «accessor» +get_XML() : String «accessor» +set_XML(in strNewValue : String) +CreateNode(in parent : IXMLDOMNode = Nothing) : IXMLDOMNode +LoadNode(in node : IXMLDOMNode) -Class_Initialize() -Class_Terminate() -NodeGetText(in node : IXMLDOMNode, in strNodeName : String) : String -NodeAppendAttribute(in dom : DOMDocument, in node : IXMLDOMNode, in strName : String, in varValue : Variant) -NodeAppendChildElement(in dom : DOMDocument, in node : IXMLDOMNode, in Name : String, in Value : Variant)

Σχήμα Γ.8: CModelResults

Γ'.2.9 Η κλάση Cmeasurement

IcarosXML::Cmeasurement
-NODE_NAME : Variant = "measurement" -m_timezone : String +Value : Double +TimePoint : Date -ATTRIBUTE_TimePoint : Variant = "timepoint"
«accessor» +get_timezone() : String «accessor» +set_timezone(in tz : String) «accessor» +get_NodeName() : String «accessor» +get_XML() : String «accessor» +set_XML(in strNewValue : String) +CreateNode(in parent : IXMLDOMNode = Nothing) : IXMLDOMNode +LoadNode(in node : IXMLDOMNode) -Class_Initialize() -NodeGetText(in node : IXMLDOMNode, in strNodeName : String) : String -NodeAppendAttribute(in dom : DOMDocument, in node : IXMLDOMNode, in strName : String, in varValue : Variant) -NodeAppendChildElement(in dom : DOMDocument, in node : IXMLDOMNode, in Name : String, in Value : Variant)

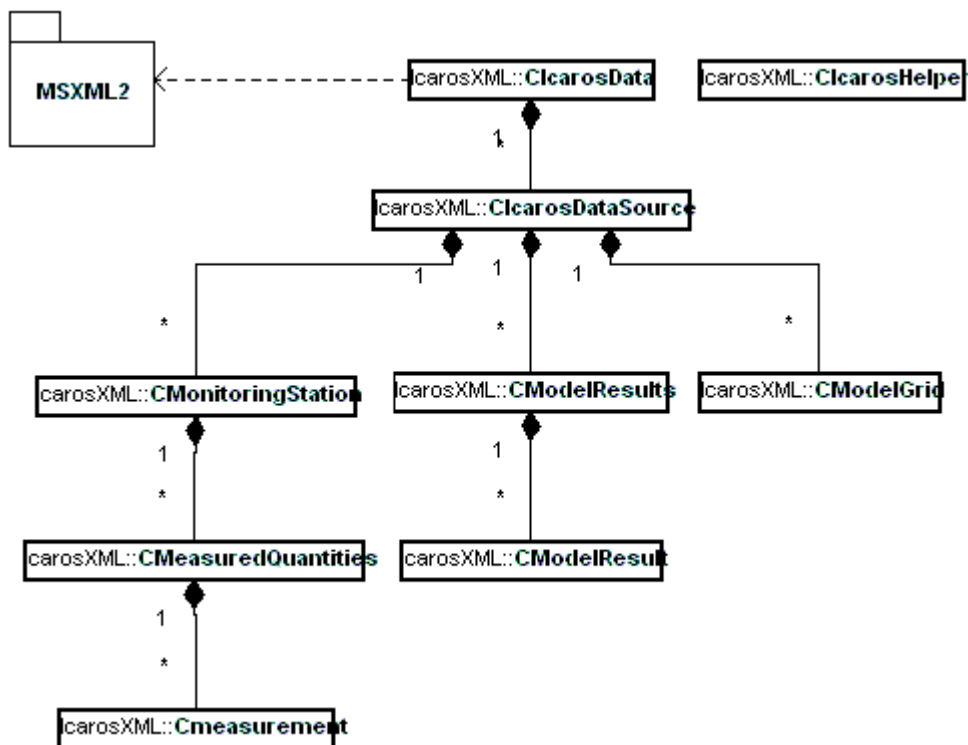
Σχήμα Γ.9: Η κλάση Cmeasurement που αποθηκεύει δεδομένα μετρήσεων

Η κλάση αυτή παρουσιάζεται στο σχήμα Γ.9 και αποθηκεύει τα αποτελέσματα των μετρήσεων από τους επίγειους σταθμούς μέτρησης. Έχει τα ακόλουθα χαρακτηριστικά:

- Ιδιότητες
 - NODE_NAME: Το όνομα του στοιχείου XML (Element)
 - m_timezone: Η ζώνη ώρας στην οποία αναφέρεται ο χρονικός προσδιορισμός της μέτρησης.
 - Value: Η τιμή της μέτρησης.
 - TimePoint: Ο χρονικός προσδιορισμός της μέτρησης.
 - ATTRIBUTE_TimePoint: Το όνομα της ιδιότητας για το αρχείο XML.
- Μέθοδοι
 - get_timezone(): Επιστρέφει την ζώνη ώρας στην οποία αναφέρονται τα δεδομένα των αντικειμένων της κλάσης.

- `let_timezone()`: Θέτει την ζώνη ώρας στην οποία αναφέρονται τα δεδομένα ενός αντικειμένου της κλάσης.
- `get_NodeName()`: Επιστρέφει το όνομα του τρέχοντος κόμβου XML.
- `get_XML()`: Επιστρέφει τον κώδικα XML ο οποίος αντιστοιχεί στα αντικείμενα που βρίσκονται ήδη φορτωμένα στην μνήμη.
- `let_XML(String)`: Ορίζει τον κώδικα XML.
- `CreateNode()`: Δημιουργεί ένα νέο κόμβο XML.
- `LoadNode()`: Φορτώνει ένα κόμβο XML στη μνήμη.
- `Class_Initialize()`: Λειτουργίες αρχικοποίησης της κλάσης
- `Class_Terminate()`: Λειτουργίες τερματισμού της κλάσης
- `NodeGetText()`: Επιστρέφει την τιμή του κόμβου της XML.
- `NodeAppendAttribute()`: Εισάγει την τιμή μίας ιδιότητας στον κόμβο της XML.
- `NodeAppendChildElement()`: Δημιουργεί ένα στοιχείο ένα στοιχείο-ο παιδί για τον κόμβο της XML.

Γ'.2.10 Συνοπτικό μοντέλο Visual Basic – UML



Σχήμα Γ.10: Συνοπτικό μοντέλο Visual Basic – UML για τις επιμέρους σχέσεις των διαφόρων αντικειμένων

Στο σχήμα Γ.10 παρουσιάζονται συνοπτικά οι σχέσεις που διέπουν τα διάφορα αντικείμενα μεταξύ τους. Είναι εμφανής η εξάρτησή τους από τον MSXML parser, που μπορεί όμως να αντικατασταθεί από οποιοδήποτε άλλο πακέτο αντίστοιχης λειτουργικότητας. Διακρίνονται επίσης οι σχέσεις περιεχομένου (*Aggregation*) μεταξύ των διαφόρων αντικειμένων. Η κλάση *CClearosHelper* η οποία είναι επικουρική, για λόγους απλούστευσης δεν παρουσιάζεται με κάποια σύνδεση με τις υπόλοιπες κλάσεις παρ' όλο που η λειτουργικότητά της αξιοποιείται από αυτές.

Παράρτημα Δ΄

Ο συνοδευτικός οπτικός δίσκος

Δ΄.1 Εισαγωγή

Κρίθηκε σκόπιμο η εργασία να συνοδευτεί από έναν οπτικό δίσκο με συνοδευτικό υλικό που αφορά τα διάφορα τμήματα που υλοποιήθηκαν και την ηλεκτρονική της έκδοση σε μορφή PDF στο αρχείο `diploma.pdf`.

Δ΄.2 UML και XML

Το UML μοντέλο της γεωγραφικής βάσης δεδομένων βρίσκεται στο αρχείο `Icaros.vsd` και απαιτεί το Visio 2000 Professional Edition. Το XML Schema που αναπτύχθηκε για τις ανάγκες της περιγραφής των δεδομένων του συστήματος *Icaros.NET* βρίσκεται στο `.τεξτλατινισαρος.ξσδ` Και απαιτεί ένα απλό text editor για την επεξεργασία του.

Δ΄.3 Ο πηγαίος Κώδικας

Ο πηγαίος κώδικας σε Visual Basic με βάση τον οποίο έγινε η υλοποίηση των κλάσεων που περιγράφονται στο παράρτημα Γ΄ βρίσκεται στο φάκελο `xmldll`. Ένα συνοπτικό, στατικό διάγραμμα κλάσεων UML που αφορά την δομή της βιβλιοθήκης βρίσκεται σε μορφή HTML στο φάκελο `dlloverview\IcarosXML.htm`

Βιβλιογραφία

- [1] CASE Tools Tutorial: Creating custom features and geodatabase schemas. ESRI.
- [2] Daylight Saving Time. <http://webexhibits.org/daylightsaving/b.html>.
- [3] Environmental Systems Research Institute. <http://www.esri.com>.
- [4] The Extensible Markup Language – XML. <http://www.w3c.org/XML>.
- [5] The Extensible Markup Language Schema Definition– XML–Schema. <http://www.w3c.org/XML/Schema>.
- [6] International Organization for Standardization. <http://www.iso.org>.
- [7] Microsoft web page for xml. <http://www.microsoft.com/xml>.
- [8] Numeric representation of Dates and Time: The ISO solution to a long-standing source of confusion. <http://www.iso.ch/iso/en/prods-services/popstds/datesandtime.html>.
- [9] The Object Management Group. <http://www.omg.org>.
- [10] The OpenGIS Consortium. <http://www.opengis.org>.
- [11] The Unified Modeling Language. <http://www.uml.org>.
- [12] The World Wide Web Consortium. <http://www.w3c.org>.
- [13] *OMG Unified Modeling Language Specification*. OMG, June 1999. Ver. 1.3.

- [14] ArcGIS 8.1: An ESRI White Paper, January 2001.
- [15] *Getting Started with the Map Control Using Visual Basic*. ESRI, 2001.
- [16] *What is ArcGIS*. ESRI, 2001.
- [17] *ArcXML Programmer's Reference Guide*. ESRI, 2002.
- [18] *Using ArcIMS*. ESRI, 2002.
- [19] Alia Abdelmoty, I. και Chris Jones, B. Towards Maintaining Consistency of Spatial Databases. Στο *CIKM '97*, σελίδες 293 – 300. ACM, 1997.
- [20] Andrea Aime, Flavio Bonfatti και Paola Daniela Monari. Making GIS closer to end users of urban environmental data. Στο *ACM GIS '99*, σελίδες 122 – 127. ACM, 1999.
- [21] Tom Barclay, Jim Gray και Don Slutz. Microsoft TerraServer: A Spatial Data Warehouse. Τεχνική Αναφορά υπ. αριθμ. MS-TR-99-29, Microsoft Research – Advanced Technology Division, June 1999.
- [22] Yvan Bédard. Visual Modelling of Spatial Databases: Towards Spatial PVL and UML. *Geoinformatica*, 53(2):169 – 186, 1999.
- [23] Yvan Bédard και Marie Josée Proulx. *Perceptory 2000 - User Guide*. University Laval, June 2000.
- [24] Claudio Bettini, Sushil Jojodia και Sean Wang, X. *Time Granularities in Databases, Data Mining, and Temporal Reasoning*. Springer, 2000.
- [25] Andreas Blaser, D. και Max Egenhofer, J. A Visual Tool for Querying Geographic Databases. Στο *AVI 2000*, σελίδες 211 – 216. ACM, 2000.
- [26] Christine Bonhomme, Claude Trépiéd, Marie-Aude-Aufaure και Robert Laurini. A Visual Language for Querying Spatio-Temporal Databases. Στο *GIS '99*, σελίδες 34 – 39. ACM, 1999.

- [27] Grady Booch. *Object-Oriented Analysis and Design with Applications*. Addison-Wesley, 2η έκδοση, 1994.
- [28] Grady Booch, James Rumbaugh και Ivar Jacobson. *The Unified Modeling Language User Guide*. Addison-Wesley, 1998.
- [29] Bob Booth και Andy Mitchell. *Getting Started with ArcGIS*. ESRI, 2001.
- [30] Karla Borges, A.V., Davis Jr. Clodoveu, A. και Alberto Laender, H.F. OMT-G:An Object-Oriented Data Model for Geographic Applications. *GeoInformatica*, 5(3):221 – 260, 2001.
- [31] Karla Borges, A.V., Alberto Laender, H.F. και Clodoveu Jr. Davis, A. Spatial Data Integrity Constraints in Object Oriented Geograpic Data Modeling. Στο *ACM GIS '99*. ACM, 1999.
- [32] Jean Brodeur, Yvan Bédard και Marie Josée Proulx. Modelling Geospatial Application Databases using UML-based Repositories Aligned with International Standards in Geomatics. Στο *8th ACM Symposium on GIS*, σελίδες 39 – 46. ACM, 2000.
- [33] Peter Burrough, A. και Rachael McDonnell, A. *Principles of Geographical Information Systems*. Oxford University Press, 1998.
- [34] Edward Chan, P.F. και Jonathan Wong, M.T. Querying and Visualization of Geometric Data. Στο *GIS '96*. ACM, 1996.
- [35] Shu Ching Chen, Xinran Wang, Naphtali Rishe και Mark Allen Weiss. A High-Performance Web-Based System Design for Spatial Data Accesses. Στο *8th ACM Symposium on GIS*, σελίδες 33 – 38. ACM, 2000.
- [36] Sophie Cockcroft. A Taxonomy of Spatial Data Integrity Constraints. *GeoInformatica*, 1(4):327 – 343, 1997.
- [37] J. Córcoles, E. και P. González. Analysis of Different Approaches for Storing GML Documents. Στο *GIS '02*, σελίδες 11 – 16. ACM, 2002.

- [38] Simon Cox, Paul Daisey, Ron Lake, Clemens Portele και Arliss Whiteide. OpenGIS® Geography Markup Language (GML) Implementation Specification. Τεχνική Αναφορά υπ. αριθμ. OCG 02-023r4, Open GIS Consortium, Inc., 2003.
- [39] Martin Ester, Alexander Frommelt, Hans Peter Kriegel και Jörg Sander. Spatial Data Mining: Database Primitives, Algorithms and Efficient DBMS Support. *Data Mining and Knowledge Discovery*, 4:193 – 216, 2000.
- [40] Jugurta Filho, Lisboa και Cirano Iochpe. Specifying Analysis Patterns for Geographic Databases on the basis of a Conceptual Framework. Στο *ACM GIS '99*, σελίδες 7 – 13. ACM, 1999.
- [41] Jere Folgert, P. An Investigation of Dynamic Simulation: The Integration of GIS with Air Dispersion Modelling to Facilitate Data Handling and Visualisation. Διπλωματική εργασία Master, University of Edinburgh, Department of Geography, September 1997.
- [42] Anders Friis-Christensen, Nectaria Tryfona και Christian Jensen, S. Requirements and Research Issues in Geographic Data Modeling. Στο *GIS '01*. ACM, November 2001.
- [43] S. Gordillo, F. Balaguer και F. Das Neves. Generating the Architecture of GIS Applications with Design Patterns. Στο *GIS '97*, σελίδες 30 – 34. ACM, 1997.
- [44] Silvia Gordillo και Federico Balaguer. Refining an object-oriented GIS design model: Topologies and Field Data. Στο *GIS '98*. ACM, November 1998.
- [45] Silvia Gordillo, Federico Balaguer, Catalina Mostaccio και Fernando Neves, Das. Developing GIS Applications with Objects: A Design Patterns Approach. *GeoInformatica*, 3(1):7 – 32, 1999.
- [46] Ian Graham. *Object-Oriented Methods: Principles & Practice*. Addison-Wesley, 3η έκδοση, 2001.

- [47] Dave Gundgeiger και Patrick Escarcega. XML Wrapper Template: Transform XML Documents into Visual Basic Classes. *MSDN*, January 2001.
- [48] Ralf Güting, Hartmut. An Introduction to Spatial Database Systems. Στο *VLDB Journal, Invited contribution*, επιμελητής: Hans J Schek, τόμος 3, σελίδες 357 - 399. 1994.
- [49] Jiawei Han και Micheline Kamber. *Data Mining: Concepts and Techniques*. Morgan Kaufmann, 2001.
- [50] Clem Henriksen, J. LAuzon, P. και Scott Morehouse. Open Geodata Access Through Standards. *StandardView*, 2(3):169 - 174, September 1994.
- [51] Richard Holowczak, D., Soon Chun, Ae, Francisco Artigas, J. και Vijayalakshmi Atluri. Customized Geospatial workflows for E-Government Services. Στο *GIS '01*, σελίδες 64 - 69. ACM, November 2001.
- [52] Steven Holzner. *XML Complete*. Mc Graw Hill, 1998.
- [53] Jung Rae Hwang, Ji Hyeon Oh και Ki Joune Li. Query Transformation Method by Delaunay Triangulation for Multi-Source Distributed Spatial Database Systems. Στο *GIS '01*, σελίδες 41 - 46. ACM, November 2001.
- [54] M. Indulska και M. E. Orłowska. On Aggregation Issues in Spatial Data Management. Στο *Research and Practice in Information Technology*, επιμελητής: Xiaofang Zhou, τόμος 5, σελίδες 75 - 84. Australian Computer Society, Inc., 2002. Thirteenth Australian Database Conference.
- [55] Myoung Ah Kang και Sylvie Servign. Animated Cartography for Urban Soundscape Information. Στο *ACM GIS '99*. ACM, 1999.

- [56] Vijay Khatri, Sudha Ram, Richard Snodgrass, T. και Grady O'Brien, M. Supporting user-defined granularities in a spatiotemporal conceptual model. *Annals of Mathematics and Artificial Intelligence*, 36:195 - 232, 2002.
- [57] Markus Kuhn. A Summary on the International Standard Date and Time Notation. <http://www.cl.cam.ac.uk/mgk25/iso-time.html>.
- [58] Vished Kumar, Alejandro Bugacov, Murilo Coutinho και Robert Neches. Integrating Geographic Information Systems, Spatial Digital Libraries and Information Spaces for conducting Humanitarian Assistance and disaster Relief Operations in Urban Environments. Στο *ACM GIS '99*, σελίδες 146 - 151. ACM, 1999.
- [59] Gabriel Kuper, Sridhar Ramaswamy, Kyuseok Shim και Jianwen Su. A Constraint-based Spatial Extension to SQL. Στο *GIS '98*. ACM, 1998.
- [60] Yee Leung. *Intelligent Spatial Decision Support Systems*. Springer, 1997.
- [61] Andrew MacDonald. *Building a Geodatabase*. ESRI, 2001.
- [62] D. Scott Mackay. Semantic Integration of Environmental Models for Application to Global Information Systems and Decision-Making. Στο *SIGMOD*, τόμος 28, σελίδες 13 - 19. ACM, March 1999.
- [63] Kevin McCurley, S. Geospatial Mapping and Navigation of the Web. Στο *WWW10*, σελίδες 221 - 229. ACM, 2001.
- [64] Petr Mikšovský και Zdeněk Kouba. GOLAP - Geographical Online Analytical Processing. Στο *DEXA 2001*, επιμελητής: H.C. Mayr et al., σελίδες 442 - 449, 2001.
- [65] Michael Minami. *Using ArcMap*. ESRI, 2000.
- [66] George Nagy και Wagle Sharad. Geographic Data Processing. *ACM Computing Surveys*, 11(2):139 - 181, June 1979.

- [67] Eric Naiburg, J. και Robert Maksimchuk, A. *UML for Database Design*. Addison Wesley, July 2001.
- [68] Marie Neal και Mark Neal. A Software Tool and Techniques for Converting Map Data into an Object. Στο *ACM GIS '98*. ACM, 1998.
- [69] The CHOROCHRONOS Participants. CHOROCHRONOS A RESEARCH NETWORK FOR SPATIOTEMPORAL DATABASE SYSTEMS. *SIGMOD*, 1999.
- [70] Cédric du Mouza και Philippe Rigaux. Web Architectures for Scalable Moving Object Servers. Στο *GIS '02*, σελίδες 17 – 22. ACM, 2002.
- [71] Alberto d'Onofrio και Elaheh Pourabbas. Formalization of Temporal Thematic Map Contents. Στο *GIS '01*, σελίδες 15 – 20. ACM, November 2001.
- [72] Luciana Vargas da Rocha, Nina Edelweiss και Cirano Iochpe. GeoFrame-T: A Temporal Conceptual Framework for Data Modeling. Στο *GIS '01*, σελίδες 124 – 129. ACM, 2001.
- [73] Christine Parent, Stefano Spaccapietra και Esteban Zimányi. Spatio-Temporal Conceptual Models: Data Structures + Space + Time. Στο *ACM GIS '99*, σελίδες 26 – 33. ACM, 1999.
- [74] Donna Peuquet, J. Making Space for Time: Issues in Space-Time Data Representation. *GeoInformatica*, 5(1):11 – 32, 2001.
- [75] Niki Pissinou, Kia Makki και E. Park, K. Towards the Design and Development of New Architecture for Geographic Information Systems. Στο *CIKM '93*, σελίδες 565 – 573. ACM, 1993.
- [76] L. Polasek, A. Zambrano, S. Gordillo, J. Bazzocco, G. Trilla, D. Cano και A. Llitas. Taking advantage of Design Patterns in GIS domain. Στο *OOPSLA 2000 Companion*, σελίδες 67 – 68. ACM, 2000.
- [77] R. Price, K. Ramamohanarao και Srinivasan B. Spatiotemporal Extensions to Unified Modeling Language. Στο *Asia Technology of Object Oriented Languages and Systems*, σελίδες 163 – 174, 1999.

-
- [78] Rosanne Price, Nectaria Tryfona και Christian Jensen, S. Extended Spatiotemporal UML: motivations, requirement, and constructs. *Journal of Database Management*, 1(4), 2000. Special Issue on UML.
- [79] Terry Quatrani. *Visual Modeling with Rational Rose 2000 and UML*. Addison-Wesley, 2000.
- [80] A. Raffaetà, C. Renso και F. Turini. Enhancing GISs for Spatio-Temporal Reasoning. Στο *GIS '02*, σελίδες 42 – 48. ACM, November 2002.
- [81] Peter Revesz και Mengchu Cai. Efficient querying and animation of periodic spatio-temporal databases. *Annals of Mathematics and Artificial Intelligence*, 36:437 – 457, 2002.
- [82] Philippe Rigaux, Michel Scholl και Agnès Voisard. *Spatial Databases with application to GIS*. Morgan Kaufmann, 2002.
- [83] Denis Sarigiannis, A. και Nikolaos Soulakellis, A. *Integrated Computational Assessment of Urban Air Quality via Remote Observation System Network - ICAROS-NET SPECIFICATIONS*. ICAROS-NET Consortium, December 2002. IST-2000-29264.
- [84] Markus Schneider. Metric Operations on Fuzzy Spatial Objects in Databases. Στο *8th ACM Symposium on GIS*, σελίδες 21 – 26. ACM, 2000.
- [85] Markus Schneider. Fuzzy Topological Predicates, Their Properties, and Their Integration into Query Languages. Στο *GIS '01*. ACM, November 2001.
- [86] Cyrus Shahabi, Mohammad Kolahdouzan, R., Snedal Thakkar, Jose Luis Ambite και Craig Knoblock, A. Efficiently Querying Moving Objects with Pre-defined Paths in a Distributed Environment. Στο *GIS '01*, σελίδες 34 – 40. ACM, 2001.
- [87] Jeff Shaner και Jennifer Wrightsell. *Editing in ArcMap*. ESRI, 2000.

- [88] Shashi Shekhar, Mark Coyle, Brajesh Goyal, Duen Ren Liu και Shyamsundar Sarkar. Data Models in Geographic Information Systems. *Communications of the ACM*, 40(4):103 - 111, April 1997.
- [89] Shashi Shekhar, Ranga Vatsavai, Raju, Namita Sehay, Thomas Burk, E. και Stephen Lime. WMS and GML based Interoperable Web Mapping System. Στο *GIS '01*, σελίδες 106 - 111. ACM, November 2001.
- [90] Abraham Silberschatz, Henry Korth, F. και S. Sudarshan. *Database System Concepts*. McGraw Hill, 3η έκδοση, 1996.
- [91] Aaron Skonnard. The XML files: What's New in MSXML 4.0. *MSDN*, December 2001.
- [92] Alexandre Sorokine και Kurt Ackermann. Scripting in GIS Applications: Experimental Standards-based Framework for Perl. Στο *8th ACM Symposium on GIS*, σελίδες 102 - 107. ACM, 2000.
- [93] Alexandre Sorokine και Irina Merzliakova. Interactive Map Applet for Illustrative Purposes. Στο *ACM GIS '98*. ACM.
- [94] Ken Spencer. Serving the Web: XML Data Manipulation with ADO 2.5. *MSDN Magazine*, August 2000.
- [95] Emmanuel Stefanakis. Modelling the history of semi-structured geographical entities. *International Journal of Geographical Information Science*, 17(6):517 - 546, 2003.
- [96] Eric Strand, J., Rajiv Mehta, P. και Raju Jairam. Applications Thrive on Open Systems Standards. *StandardView*, 1994.
- [97] Egemen Tanin, František Brabec και Hanan Samet. Remote Access to Large Spatial Databases. Στο *GIS '02*, σελίδες 5 - 10. ACM, 2002.
- [98] Carol Traynor και Marian Williams, G. A Study of End-User Programming for Geographic Information Systems.

- [99] Nectaria Tryfona και Thanasis Hadzilacos. Geographic Applications Development: Models and Tools for the Conceptual Level. Στο 3rd *ACM Workshop in Geographic Information Systems*. ACM, 1995.
- [100] Nectaria Tryfona και Christian Jensen, S. Conceptual Data Modeling for Spatiotemporal Applications. *GeoInformatica*, 3(3):245 – 268, 1999.
- [101] I. Tsanis, K. και S. Boyle. A 2D hydrodynamic/pollutant transport GIS model. *Advances in Engineering Software*, 32:353 – 361, 2001. Elsevier.
- [102] Corey Tucker. *Using ArcToolbox*. ESRI, 2000.
- [103] Ranga Vatsavai, Raju, Thomas Burk, E., Tyler Wilson, B. και Shashi Shekhar. A Web-based browsing and spatial analysis system for regional natural resource analysis and mapping. Στο 8th *ACM Symposium on GIS*, σελίδες 95 – 101. ACM, 2000.
- [104] Aleta Vienneau. *Using ArcCatalog*. ESRI, 2001.
- [105] Agnès Voisard. Geologic Hypermaps are more than Clickable Maps! Στο *ACM GIS '98*. ACM, 1998.
- [106] Fangju Wang. A Distributed Geographic Information System on the Common Object Request Broker Architecture – CORBA. *GeoInformatica*, 4(1):89 – 115, 2000.
- [107] Feng Wang, Jichang Sha, Huowang Chen και Shuqiang Yang. Geo-SQL: A Spatial Query Language of Object-oriented GIS. Στο *Proceeding of the 2nd International Workshop on Computer Science and Information Technologies*. CSIT '2000, 2000.
- [108] Norbert Widmann και Peter Baumann. Towards Comprehensive Database Support for Geoscientific Raster Data. Στο *GIS '97*, σελίδες 54 – 57. ACM, 1997.

- [109] Michael Worboys, F. *GIS: A Computing Perspective*. Taylor & Francis, 1995.
- [110] Michael Zeiler. *Modeling Our World: The ESRI Guide to Geodatabase Design*. ESRI Press, 1999.
- [111] Michael Zeiler. *Exploring ArcObjects™ - Applications and Cartography*, τόμος 1. ESRI, 2001.
- [112] Michael Zeiler. *Exploring ArcObjects™ - Geographic Data Management*, τόμος 2. ESRI, 2001.
- [113] Alexander Zipf και Sven Krüger. *TGML - Extending GML by Temporal Constructs - A Proposal for a Spatiotemporal Framework in XML*. ACM, 2001.