



ΑΤΕΙ ΘΕΣΣΑΛΙΑΣ
ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ

Τμήμα Μηχανικών Πληροφορικής ΤΕ

Ανάπτυξη Γραφικού Περιβάλλοντος για Λογισμικό Προσομοίωσης Φαινομένων Μετάδοσης Θερμότητας

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Χαντζής Γιάννης (ΑΜ: Τ03043)

Επιβλέπων: Ιατρέλλης Όμηρος, Καθηγητής Εφαρμογών

ΛΑΡΙΣΑ 2014

«Εγώ ο Χαντζής Γιάννης, δηλώνω υπεύθυνα ότι η παρούσα Πτυχιακή Εργασία με τίτλο Ανάπτυξη Γραφικού Περιβάλλοντος για Λογισμικό Προσομοίωσης Φαινομένων Μεταβολής Θερμότητας είναι δική μου και βεβαιώνω ότι:

- Σε όλες περιπτώσεις έχω συμβουλευτεί δημοσιευμένη εργασία τρίτων, αυτό επισημαίνεται με σχετική αναφορά στα επίμαχα σημεία.*
- Σε όλες περιπτώσεις μεταφέρω λόγια τρίτων, αυτό επισημαίνεται με σχετική αναφορά στα επίμαχα σημεία. Με εξαίρεση τέτοιες περιπτώσεις, το υπόλοιπο κείμενο της πτυχιακής αποτελεί δική μου δουλειά.*
- Αναφέρω ρητά όλες τις πηγές βοήθειας που χρησιμοποίησα.*
- Σε περιπτώσεις που τμήματα της παρούσας πτυχιακής έγιναν από κοινού με τρίτους, αναφέρω ρητά ποια είναι η δική μου συνεισφορά και ποια των τρίτων.*
- Γνωρίζω πως η λογοκλοπή αποτελεί σοβαρότατο παράπτωμα και είμαι ενήμερος(-η) για την επέλευση των νομίμων συνεπειών»*

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή

Τόπος:

Ημερομηνία:

ΕΠΙΤΡΟΠΗ ΑΞΙΟΛΟΓΗΣΗΣ

1.
2.
3.

Περίληψη

Αντικείμενο της εργασίας είναι η ανάπτυξη ενός γραφικού περιβάλλοντος (GUI) κατάλληλου για ένα υπάρχον λογισμικό (bark) που δουλεύει σε γραμμή εντολών.

Το bark παρέχει την δυνατότητα μονοδιάστατης μεταβατικής προσομοίωσης φαινομένων μετάδοσης θερμότητας σε τοιχώματα πολλαπλών στρωμάτων και χρησιμοποιείται ήδη για διδακτικούς σκοπούς στο Τμήμα Μηχανολογίας. Δέχεται αρχεία εισόδου XML και εξάγει τα αποτελέσματα σε αρχεία ASCII.

Τελικός στόχος της εργασίας είναι ένα φιλικό περιβάλλον εργασίας το οποίο περιλαμβάνει την δημιουργία των αρχείων εισόδου, την εκτέλεση προσομοιώσεων και τέλος την απεικόνιση των αποτελεσμάτων. Η δημιουργία αρχείων εισόδου (XML) θα γίνεται σε εύχρηστο γραφικό περιβάλλον το οποίο θα δημιουργείται δυναμικά βάσει αρχείου XML schema για εύκολη επεκτασιμότητα. Η εκτέλεση των προσομοιώσεων και η απεικόνιση των αποτελεσμάτων θα έχουν κύριο στόχο την κατανόηση και εμπέδωση του μαθήματος.

Ευχαριστίες

Θα ήθελα να ευχαριστήσω τον καθηγητή Ονούφριο Χαραλάμπους, του τμήματος Μηχανολογίας του ΤΕΙ Λάρισας για την καθοδήγηση και έμπρακτη υποστήριξη, τον Ιωάννη Μπλουγουρά για την βοήθεια που μου προσέφερε και την κατανόηση που έδειξε, τους συμφοιτητές και φίλους μου για την ηθική υποστήριξή τους, καθώς και τον καθηγητή Όμηρο Ιατρέλλη για την συμμετοχή του στην παρούσα πτυχιακή εργασία.

Χαντζής Γιάννης

04/04/2014

Περιεχόμενα

ΠΕΡΙΛΗΨΗ	I
ΕΥΧΑΡΙΣΤΙΕΣ.....	III
ΠΕΡΙΕΧΟΜΕΝΑ.....	V
1 ΕΙΣΑΓΩΓΗ	1
2 ΛΟΓΙΣΜΙΚΟ ΜΕΤΑΔΟΣΗΣ ΘΕΡΜΟΤΗΤΑΣ BARK	3
2.1 ΓΕΝΙΚΑ.....	3
2.2 ΕΙΔΙΚΟΤΕΡΑ	4
2.3 ΜΑΘΗΜΑΤΙΚΟ ΜΟΝΤΕΛΟ	6
2.3.1 Ισοζύγιο ενέργειας.....	6
2.3.2 Ακτινοβολία σε αδιαφανή και διαφανή υλικά.....	6
2.3.3 Οριακές συνθήκες	7
2.3.4 Ηλιακή ακτινοβολία.....	7
3 ΓΡΑΦΙΚΟ ΠΕΡΙΒΑΛΛΟΝ BARK_GUI.....	11
3.1 ΑΝΑΛΥΤΙΚΗ ΠΕΡΙΓΡΑΦΗ ΘΕΜΑΤΟΣ	11
3.2 ΤΙ ΕΙΝΑΙ ΤΟ ΓΡΑΦΙΚΟ ΠΕΡΙΒΑΛΛΟΝ	12
3.3 ΠΕΡΙΓΡΑΦΗ ΓΛΩΣΣΩΝ ΚΑΙ ΕΡΓΑΛΕΙΩΝ	12
3.3.1 Ποιές γλώσσες χρησιμοποιήθηκαν και για ποιο λόγο.....	12
3.3.2 Εργαλεία – Λογισμικά	13
4 ΑΝΑΛΥΣΗ ΓΡΑΦΙΚΟΥ ΠΕΡΙΒΑΛΛΟΝΤΟΣ	19
4.1 ΔΙΑΓΡΑΜΜΑΤΑ UML	19
4.1.1 Τι είναι η UML	19
4.1.2 Περίπτωση χρήσης	19
4.1.3 Δομή παραθύρων και ροή εκτέλεσης	22
4.1.4 Προσαρμοσμένα στοιχεία ελέγχου.....	26
4.1.5 Δομή αντικειμένων-στοιχείων	32
4.1.6 Δομή και διαχείριση δεδομένων των αντικειμένων-στοιχείων.....	38

4.1.7	Διαχείριση αρχείων.....	39
4.1.8	Βοηθητικές κλάσεις.....	41
4.2	ΚΥΡΙΟ ΠΡΟΒΛΗΜΑ ΠΟΥ ΑΝΤΙΜΕΤΩΠΙΣΤΗΚΕ	42
4.3	ΒΗΜΑΤΑ ΕΠΙΛΥΣΗΣ ΠΡΟΒΛΗΜΑΤΟΣ	43
4.3.1	Δημιουργία σχήματος.....	43
4.3.2	Φόρτωση δεδομένων αρχείου XML.....	44
4.3.3	Επεξεργασία δεδομένων	46
4.3.4	Αποθήκευση δεδομένων σε αρχείο.....	47
4.3.5	Προσομοίωση.....	48
5	ΟΔΗΓΟΣ ΧΡΗΣΗΣ	51
5.1	ΕΠΕΞΕΡΓΑΣΙΑ ΔΕΔΟΜΕΝΩΝ ΕΙΣΟΔΟΥ	51
5.1.1	Άνοιγμα αρχείου δεδομένων XML (.brk)	51
5.1.2	Δημιουργία νέου αρχείου δεδομένων XML (.brk)	53
5.1.3	Ενέργειες στο δέντρο	53
5.1.4	Έλεγχος εγκυρότητας εισαγωγής και επεξεργασίας δεδομένων.....	58
5.1.5	Αποθήκευση αρχείου δεδομένων.....	59
5.1.6	Εκκίνηση προσομοίωσης.....	59
5.2	ΓΡΑΦΙΚΗ ΑΠΕΙΚΟΝΙΣΗ ΑΠΟΤΕΛΕΣΜΑΤΩΝ	60
5.2.1	Επιλογή αξόνων διαγράμματος	61
5.2.2	Εμφάνιση δεδομένων εξόδου bark σε διάγραμμα.....	61
6	ΕΠΙΛΟΓΟΣ.....	65
6.1	ΣΥΜΠΕΡΑΣΜΑ	65
6.2	ΜΕΛΛΟΝΤΙΚΕΣ ΕΠΕΚΤΑΣΕΙΣ	65
6.2.1	Ενσωμάτωση γραφημάτων στο αρχείο δεδομένων XML	65
6.2.2	Υποστήριξη συναρτήσεων (functions).....	66
6.2.3	Έλεγχος εγκυρότητας αρχείου δεδομένων XML.....	67
6.2.4	Προϋπάρχουσα συλλογή υλικών	67
6.2.5	Βελτίωση γραφικής απεικόνισης αποτελεσμάτων.....	67
	ΒΙΒΛΙΟΓΡΑΦΙΑ.....	69
	ΠΑΡΑΡΤΗΜΑ Α – ΔΟΜΗ XML	71
	ΠΑΡΑΡΤΗΜΑ Β – ΕΙΚΟΝΕΣ	73

1 Εισαγωγή

Σκοπός της πτυχιακής είναι ο χρήστης να μπορεί να συγκεντρώνεται στο να βγάζει συμπεράσματα πάνω στο θέμα που τον ενδιαφέρει και όχι να μαθαίνει μια γλώσσα στην προσπάθεια να το κάνει.

Οι κύριες λειτουργίες που μπορεί να κάνει ένας χρήστης είναι να φορτώσει αρχείο δεδομένων τύπου XML, να επεξεργαστεί τα περιεχόμενά του εύκολα και γρήγορα χωρίς να χρειαστεί να μάθει την γλώσσα XML, να αποθηκεύσει τις αλλαγές του στο αρχείο, να δημιουργήσει νέο αρχείο καθώς και να εκτελέσει προσομοίωση με τα δεδομένα εισόδου να παράξει τα δεδομένα εξόδου με χρήση του λογισμικού bark και τελικά να δημιουργήσει γραφήματα τα οποία αποτελούν γραφική αναπαράσταση των δεδομένων εξόδου.

Το παρόν θεωρητικό μέρος της πτυχιακής στοχεύει στην κατανόηση της διαδικασίας ανάπτυξης του πρακτικού μέρους, καθώς και στα επιμέρους στοιχεία που το απαρτίζουν.

Στα επόμενα κεφάλαια θα αναλυθούν:

Φαινόμενα μετάδοσης θερμότητας εις βάθος το οποίο είναι το θέμα που ασχολείται το λογισμικό bark. Το bark είναι πρόγραμμα το οποίο εκτελείται σε γραμμή εντολών για το οποίο η παρούσα πτυχιακή έχει δημιουργήσει το γραφικό περιβάλλον.

Στην αρχή γενικά και αργότερα λεπτομερώς η σχεδίαση του γραφικού περιβάλλοντος. Τα προβλήματα που αντιμετωπίστηκαν και πώς, η δομή του γραφικού περιβάλλοντος αλλά και των κλάσεων στα παρασκήνια, η ροή εκτέλεσης του προγράμματος είναι μερικά από τα κεντρικά θέματα τα οποία περιγράφονται στα επόμενα κεφάλαια.

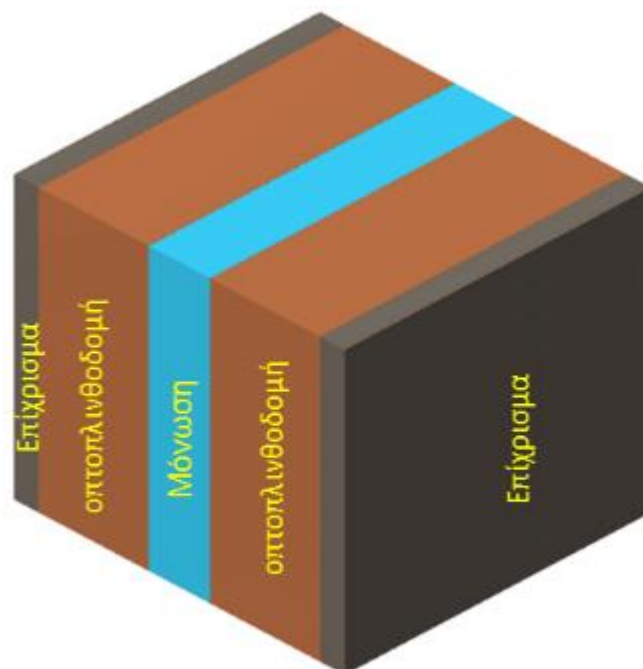
Ο τρόπος χρήσης του γραφικού περιβάλλοντος παραστατικά.

2 Λογισμικό Μετάδοσης Θερμότητας bark

2.1 Γενικά

Το πρόγραμμα bark χρησιμοποιείται για την ανάλυση μετάδοσης θερμότητας σε διάφορα τοιχώματα τα οποία αποτελούνται από στρώματα διαφορετικών υλικών στο εσωτερικό και εξωτερικό του κτηρίου, αλλά και στο εσωτερικό του τοιχώματος όπως φαίνεται στην Εικόνα 2.1. Για παράδειγμα μπορεί ένα τοίχωμα να αποτελείται από δύο στρώματα τούβλου στο εσωτερικό του κτηρίου (πλευρά δωματίου), στο εξωτερικό του κτηρίου από τσιμέντο και να υπάρχει άλλο ένα στρώμα αέρα ανάμεσα το οποίο αλλάζει την συμπεριφορά του τοιχώματος. Η πρακτική χρήση του λογισμικού είναι η διεξαγωγή συμπερασμάτων πάνω στην απόδοση ενός τοιχώματος υπό διάφορες συνθήκες.

Τα συμπεράσματα μπορούν να ωφελήσουν στην βελτίωση απόδοσης ενός κτηρίου με την επιλογή κατάλληλων υλικών για την κατασκευή του ανάλογα με τις συνθήκες στις οποίες θα βρίσκεται, αλλά μπορούν και να ελαττώσουν κατά πολύ την κατανάλωση ενέργειας για την διατήρηση μιας επιθυμητής κατάστασης του κτηρίου (π.χ. θερμοκρασία δωματίου).

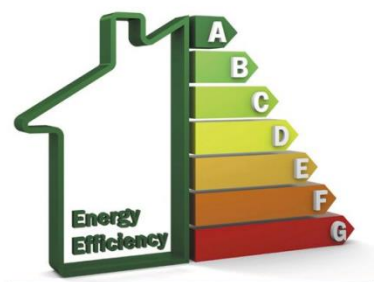


Εικόνα 2.1 – Διπλό δρομικό τοίχωμα

2.2 Ειδικότερα

Το πρόγραμμα bark ασχολείται με ένα τομέα που συναντάμε στην καθημερινότητα -μας και που η συμβολή του στο περιβάλλον είναι σημαντική. Η εξοικονόμηση ενέργειας μέσω των ενεργειακών κτιρίων προσφέρει μεγάλο κέρδος απόδοσης και χρημάτων.

Τελευταία όμως, βλέπουμε μια στροφή προς τον τόσο ευαίσθητο αυτό τομέα, αρχικά με την πριμοδότηση για αλλαγή παλιών ηλεκτρικών ειδών με νέα μεγαλύτερης ενεργειακής κλάσης, στη συνέχεια με μικρά βήματα όπως οι πράσινες στέγες και τελευταία με το μεγάλο βήμα που έγινε από το υπουργείο Περιβάλλοντος με τη νομοθεσία για τον Κανονισμό Ενεργειακής Απόδοσης Κτιρίων (Εικόνα 2.2).



Εικόνα 2.2 - KENAK (Κανονισμός Ενεργειακής Απόδοσης Κτηρίων)

Τα ενεργειακά κτίρια και η εξοικονόμηση ενέργειας σχετίζονται άμεσα με το αντικείμενο σπουδών των Μηχανολόγων - Μηχανικών που είναι η μετάδοση θερμότητας. Η ανάγκη για εξοικονόμηση ενέργειας είναι ιδιαίτερα εμφανής στον κτηριακό τομέα, ο οποίος καλύπτει το 36% περίπου της συνολικής τελικής ενεργειακής κατανάλωσης στην Ελλάδα, με μέσο ετήσιο ρυθμό αύξησης 7% και το 40% στην Ευρωπαϊκή Ένωση.

Στην Ελλάδα μέχρι πρότινος δεν είχε γίνει καμία αξιολογή προσπάθεια για να δοθούν κίνητρα στους πολίτες για την κατασκευή ενεργειακών κτιρίων, ούτε καν για μετατροπή συμβατικών κτιρίων σε ενεργειακά όπου αυτό ήταν δυνατό.

Η δε θερμική λειτουργία ενός κτιρίου αποτελεί μία δυναμική κατάσταση, η οποία: εξαρτάται από τις τοπικές κλιματικές και περιβαλλοντικές παραμέτρους (την ηλιοφάνεια, τη θερμοκρασία εξωτερικού αέρα, τη σχετική υγρασία, τον άνεμο, τη

βλάστηση, το σκιασμό από άλλα κτίρια), αλλά και τις συνθήκες χρήσης του κτιρίου (κατοικία, γραφεία, νοσοκομεία κλπ.). Βασίζεται στην αντίστοιχη ενεργειακή συμπεριφορά των δομικών του στοιχείων και (κατ' επέκταση) των ενσωματωμένων παθητικών ηλιακών συστημάτων, αλλά και το ενεργειακό προφίλ που προκύπτει από την λειτουργία του κτιρίου.

Με την βοήθεια του προγράμματος bark μπορούν να εκτιμηθούν οι θερμοφυσικές ιδιότητες ενός τοιχώματος και συγκεκριμένα της μόνωσης. Με αυτόν τον τρόπο μπορεί κάποιος να αναγνωρίσει το υλικό ενός συγκεκριμένου τοιχώματος χωρίς να το γνωρίζει εξ' αρχής. Στην εικόνα Εικόνα 2.3 απεικονίζεται ένα παράδειγμα χρήσης του bark.

time[s]	q1[W/m2]	q2[W/m2]	Tinf1[K]	Tsurf1[K]	Tsurf2[K]	Tinf2[K]
576000.00	-2.61	-535.30	293.15	293.71	320.84	282.79
590400.00	-9.41	-114.76	293.15	295.17	288.16	279.58
604800.00	-4.95	-58.80	293.15	294.22	279.66	275.18
619200.00	1.25	-28.39	293.15	292.88	276.18	274.01
633600.00	5.87	0.09	293.15	291.87	277.22	277.22
648000.00	6.76	-553.98	293.15	291.68	321.04	281.62
662400.00	-2.61	-535.30	293.15	293.71	320.84	282.79
676800.00	-9.41	-114.76	293.15	295.17	288.16	279.58
691200.00	-4.95	-58.80	293.15	294.22	279.66	275.18
705600.00	1.25	-28.39	293.15	292.88	276.18	274.01
720000.00	5.87	0.09	293.15	291.87	277.22	277.22
734400.00	6.76	-553.98	293.15	291.68	321.04	281.62
748800.00	-2.61	-535.30	293.15	293.71	320.84	282.79
763200.00	-9.41	-114.76	293.15	295.17	288.16	279.58
777600.00	-4.95	-58.80	293.15	294.22	279.66	275.18
792000.00	1.25	-28.39	293.15	292.88	276.18	274.01
806400.00	5.87	0.09	293.15	291.87	277.22	277.22
820800.00	6.76	-553.98	293.15	291.68	321.04	281.62
835200.00	-2.61	-535.30	293.15	293.71	320.84	282.79
849600.00	-9.41	-114.76	293.15	295.17	288.16	279.58
time[s]	q1[W/m2]	q2[W/m2]	Tinf1[K]	Tsurf1[K]	Tsurf2[K]	Tinf2[K]
864000.00	-4.95	-58.80	293.15	294.22	279.66	275.18
Nodes : 25						
Laboratory of Heat Transfer, TEILar						
Contact: onoufriou@teil.ar.gr						

Εικόνα 2.3 – Παράδειγμα προσομοίωσης με χρήση του λογισμικού bark

Διεξάγοντας διάφορες προσομοιώσεις που στόχο έχουν να συμπέσουν οι παρακάτω πειραματικές μετρήσεις:

1. την θερμορροή
2. θερμοκρασία τοιχώματος εσωτερικά
3. θερμοκρασία περιβάλλοντος
4. και θερμοκρασία χώρου

με τα αποτελέσματα του προγράμματος bark. Έτσι καταλήγουμε στο συμπέρασμα από τι υλικά αποτελείται το συγκεκριμένο τοίχωμα. Τα αποτελέσματα αυτά είναι απαραίτητα για την διεξαγωγή ενεργειακών επιθεωρήσεων σε κτήρια.

Επιπλέον στόχος της εργασίας είναι η χρήση των παραπάνω αποτελεσμάτων με τη βοήθεια του προγράμματος bark για την μελέτη προτάσεων εξοικονόμησης ενέργειας.

Μια τέτοια πρόταση είναι και η χρήση διάφανης μόνωσης. Στα πλαίσια μιας εργασίας μπορεί να υπολογίζεται επακριβώς το θερμικό κέρδος μιας τέτοιας διάταξης σε σύγκριση με μία συμβατική. Τα αποτελέσματα αυτά μπορούν να χρησιμοποιηθούν περαιτέρω για οικονομοτεχνικές μελέτες που αποδεικνύουν τη οικονομική και περιβαλλοντολογική αξία καινοτόμων λύσεων.

2.3 Μαθηματικό μοντέλο

Χρησιμοποιείται το παρακάτω μαθηματικό μοντέλο για τον υπολογισμό αποτελεσμάτων με χρήση των αρχείων εισόδου XML και παραγωγή των αρχείων εξόδου δεδομένων τύπου .dat στο λογισμικό bark.

2.3.1 Ισοζύγιο ενέργειας

Το μονοδιάστατο μεταβατικό ενεργειακό ισοζύγιο εκφράζεται από την εξίσωση [i]:

$$\frac{d}{dt}(\rho \cdot Cp \cdot T) = \frac{dT}{dx} \left(k \cdot \frac{dT}{dx} \right) + S$$

2.3.2 Ακτινοβολία σε αδιαφανή και διαφανή υλικά

Στην επιφάνεια των υλικών εφαρμόζεται το κλασσικό ισοζύγιο ακτινοβολίας που περιλαμβάνει συντελεστής απορρόφησης, διαπερατότητας και ανάκλασης:

$$\rho + \alpha + \tau = 1$$

Στην περίπτωση αδιαφανών υλικών, όλη η απορρόφηση ακτινοβολίας λαμβάνει χώρα στην επιφάνεια, ενώ για διαφανή υλικά χρησιμοποιείται ο νόμος εκθετικής μείωσης :

$$\alpha = 1 - e^{-kL}$$

Στην τελευταία περίπτωση η απορρόφηση ακτινοβολίας περιλαμβάνεται στον ισοζύγιο ενέργειας ως όρος πηγής.

2.3.3 Οριακές συνθήκες

Η παρακάτω εκφράσεις χρησιμοποιούνται για τις οριακές συνθήκες και στις δύο πλευρές:

$$\left[-k \frac{dT}{dx} \right]_{x=0} = h_{conv} (T_{\infty} - T_s) + \sigma \epsilon (T_{\infty}^4 - T_s^4) + \dot{q}''_{solar} + \dot{q}''$$

$$\left[-k \frac{dT}{dx} \right]_{x=L} = h_{conv} (T_s - T_{\infty}) + \sigma \epsilon (T_s^4 - T_{\infty}^4) - \dot{q}''_{solar} + \dot{q}''$$

όπου, οι τέσσερις όροι της δεξιάς πλευράς αντιστοιχούν στην συναγωγή, την ακτινοβολία περιβάλλοντος, την ηλιακή ακτινοβολία και οποιαδήποτε άλλη θερμοροή.

2.3.4 Ηλιακή ακτινοβολία

Για να εκτιμηθεί η ηλιακή ακτινοβολία, χρησιμοποιείται το "Μοντέλο Ηλιακής Ακτινοβολίας Καθαρού Ουρανού", όπως προτείνεται από την ASHRAE και περιγράφεται από τον Kuehn και συνεργάτες [ii]. Τέσσερις παράμετροι από πίνακες απαιτούνται για να υπολογιστεί η ηλιακή ακτινοβολία που φτάνει στην επιφάνεια της γης. Για ευκολία, χρησιμοποιούνται οι παρακάτω προσαρμοσμένες εξισώσεις:

$$A = 1161.97 + 76.565 \cos \left(\frac{month - \frac{11.6633}{30}}{6\pi} \right)$$

$$B = 0.171844 + 0.034158 \cos \left(\frac{month - \frac{17.3658}{30}}{6\pi} \right)$$

$$C = 0.0907911 + 0.0401521 \cos \left(\frac{month - \frac{15.041}{30}}{6\pi} \right)$$

Είναι επίσης απαραίτητη η απόκλιση της γης για όλη την περίοδο των δώδεκα μηνών:

$$d = 23.44 \cos 2\pi \left(\frac{month}{12} \right) \frac{\pi}{180}$$

Για τον υπολογισμό της ηλιακής ακτινοβολίας σε μια συγκεκριμένη επιφάνεια, πρέπει να είναι γνωστές μια σειρά από γωνίες. Η γωνία ηλιακού ύψους μεταξύ των ακτινών και του οριζόντιου επίπεδου μεταβάλλεται ημερήσια και εποχικά:

$$\sin \beta = \cos l \cos h \cos d + \sin l \sin d$$

Η γωνία ηλιακού αζιμούθιου μεταξύ της προβολής των ηλιακών ακτινών στο οριζόντιο επίπεδο και του άξονα Βορρά-Νότου μεταβάλλεται ομοίως:

$$\cos \varphi = \frac{1}{\cos \beta} (\cos d \sin l \cosh - \sin d \cos l)$$

Ο προσανατολισμός της επιφάνειας χρησιμοποιείται στον υπολογισμό της γωνίας ηλιακού αζιμούθιου-όψης

$$\gamma = |(\varphi - \psi)|$$

Τέλος όλα τα παραπάνω οδηγούν στον υπολογισμό της γωνίας πρόσπτωσης:

$$\cos \theta = \cos \beta \cos \gamma \sin \Sigma + \sin \beta \cos \Sigma$$

Η ένταση της άμεσης ηλιακής ακτινοβολίας στην επιφάνεια της γης δίνεται από την σχέση:

$$I_{DN} = A e^{-B/\sin \beta}$$

ενώ η ένταση της άμεσης ηλιακής ακτινοβολίας που προσπίπτει σε μια επιφάνεια:

$$I_D = I_{DN} \cos \theta$$

Η διάχυτη ηλιακή ακτινοβολία υπολογίζεται ως ποσοστό της άμεσης ηλιακής ακτινοβολίας:

$$I_d = C I_{DN} \cos \theta$$

όπου ο συντελεστής όψης ορίζεται ως:

$$F_s = \frac{1 + \cos \Sigma}{2}$$

Η ανακλώμενη ηλιακή ακτινοβολία περιλαμβάνει το μέγεθος της ακτινοβολίας που ανακλάται από όμορες επιφάνειες και υπολογίζεται ως:

$$I_R = \rho_g$$

$$I_R = \rho_g I_{DN} (C + \sin \beta) F_{Ag}$$

με τον κατάλληλο συντελεστή όψης:

$$F_{Ag} = \frac{1 - \cos \Sigma}{2}$$

όπου το Σ , υποδηλώνει την γωνία κλίσης της επιφάνειας.

Λαμβάνοντας υπόψη όλους τους παραπάνω όρους, η συνολική ηλιακή ακτινοβολία εκφράζεται ως:

$$I = I_D + I_d + I_R$$

3 Γραφικό Περιβάλλον bark_GUI

Για την διευκόλυνση της διαδικασίας επεξεργασίας αρχείων δεδομένων είναι απαραίτητο ένα γραφικό περιβάλλον το οποίο δεν πρέπει να περιορίζει τις δυνατότητες και λειτουργίες της διαδικασίας. Με αυτό το θέμα ασχολείται η παρούσα πτυχιακή, την ανάπτυξη ενός γραφικού περιβάλλοντος για το λογισμικό bark το οποίο έχει θέμα την προσομοίωση φαινομένων μετάδοσης θερμότητας κυρίως σε τοιχώματα κτηρίων. Το αρχείο εισόδου για το λογισμικό bark είναι αρχείο XML, επομένως το γραφικό περιβάλλον θα πρέπει να χειρίζεται τέτοια αρχεία και να παρέχει σε οποιονδήποτε χρήστη εύκολη επεξεργασία των δεδομένων του χωρίς να γνωρίζει την γλώσσα περιγραφής.

3.1 Αναλυτική Περιγραφή Θέματος

Το λογισμικό bark χρησιμοποιείται στην διδασκαλία του μαθήματος "Υπολογιστικές Μέθοδοι Φαινομένων Μεταφοράς" των Μηχανολόγων - Μηχανικών. Το πρόβλημα είναι ότι οι φοιτητές προβληματίζονται κατά την επεξεργασία των αρχείων XML και δυσκολεύονται να κατανοήσουν την ουσία της άσκησης. Αντιθέτως πρέπει να δίνεται έμφαση σε υπολογισμούς και στα αποτελέσματα και όχι στην εκμάθηση επεξεργασίας αρχείων XML.

Δεν αρκεί όμως να πετύχει μόνο αυτός ο σκοπός, διότι διαφορετικά τα αρχεία XML – ασκήσεις θα παρέμεναν ίδια με το πέρασμα το χρόνου με αποτέλεσμα να μην εξελίσσεται το μάθημα και η εκπαίδευση. Για αυτό το λόγο πρέπει να δημιουργηθεί ένα γραφικό περιβάλλον το οποίο δέχεται αρχεία XML με δεδομένα ασκήσεων ακόμη και αν αλλάξει η δομή των πληροφοριών, το λογισμικό να είναι σε θέση να τα χειριστεί κατάλληλα χωρίς περιορισμούς και προβλήματα.

Τα παραπάνω αποτελούν το πρώτο μέρος της παρούσας πτυχιακής, την εμφάνιση, επεξεργασία και αποθήκευση αρχείων XML με ένα γραφικό περιβάλλον φιλικό προς το χρήστη. Το δεύτερο μέρος αποτελεί την απεικόνιση των αποτελεσμάτων του λογισμικού bark με τέτοιο τρόπο ώστε ο χρήστης να είναι σε θέση να αποκομίσει γνώση και να βγάλει συμπεράσματα πάνω στο θέμα.

3.2 Τι είναι το Γραφικό Περιβάλλον

Γραφικό Περιβάλλον Χρήστη ή Γραφική Διασύνδεση Χρήστη (Graphical User Interface - GUI) στην πληροφορική ονομάζεται ένα σύνολο γραφικών στοιχείων, τα οποία εμφανίζονται στην οθόνη κάποιας ψηφιακής συσκευής (π.χ. Η/Υ) και χρησιμοποιούνται για την αλληλεπίδραση του χρήστη με τη συσκευή αυτή. Παρέχουν στον τελευταίο, μέσω γραφικών, ενδείξεις και εργαλεία προκειμένου αυτός να φέρει εις πέρας κάποιες επιθυμητές λειτουργίες. Για τον λόγο αυτό δέχονται και είσοδο από τον χρήστη και αντιδρούν ανάλογα στα συμβάντα που αυτός προκαλεί με τη βοήθεια κάποιας συσκευής εισόδου (π.χ. πληκτρολόγιο, ποντίκι).

3.3 Περιγραφή γλωσσών και εργαλείων

Αναφέρονται πλήρως τα εργαλεία που χρησιμοποιήθηκαν και για ποιο λόγο με συνοπτική περιγραφή για το καθένα. Επίσης αναφέρεται ποια εργαλεία είναι απαραίτητα για την αναπαράσταση της διαδικασίας ανάπτυξης της εφαρμογής και ποια είναι δευτερεύον/βοηθητικά.

3.3.1 Ποιές γλώσσες χρησιμοποιήθηκαν και για ποιο λόγο

Το θέμα της παρούσας πτυχιακής είναι η ανάπτυξη ενός γραφικού περιβάλλοντος, οπότε το πρώτο πράγμα που χρειάζεται είναι ένας τρόπος δημιουργίας του. Δεδομένου ότι η κεντρική λειτουργία του γραφικού περιβάλλοντος είναι η εύκολη ανάγνωση και επεξεργασία δεδομένων από φοιτητές του τμήματος της Μηχανολογίας, το λειτουργικό σύστημα θα είναι Microsoft Windows. Με τα προαναφερόμενα επιλέχθηκε η γλώσσα προγραμματισμού C# με πλαίσιο εργασίας (framework) Microsoft .NET.

Ο καλύτερος τρόπος χειρισμού πληροφορίας με χρήση κώδικα για τον συγκεκριμένο σκοπό είναι η χρήση αρχείων XML. Η **XML** (Extensible Markup Language) είναι μια περιγραφική γλώσσα και όχι γλώσσα προγραμματισμού, η οποία χρησιμοποιείται κυρίως για να περιγράψει τη δομή κάποιων πληροφοριών. Είναι μια απλή γλώσσα και έχει πολλά κοινά στοιχεία με την HTML (Hypertext Markup Language). Μια βασική δομή κάποιας πληροφορίας σε XML αποτελείται από στοιχεία (elements) και τις ιδιότητές τους (attributes). Οι πληροφορίες που εμπεριέχονται σε

αρχεία τύπου XML για την παρούσα πτυχιακή αφορούν περιπτώσεις προσομοίωσης μετάδοσης θερμότητας.

Δεν πρέπει όμως η λειτουργία του λογισμικού να περιορίζεται στην εμφάνιση και επεξεργασία συγκεκριμένων αρχικών παραδειγμάτων, για αυτό το λόγο υπήρξε η ανάγκη για την χρήση των αρχείων τύπου XSD. Η XML Schema W3C ή αλλιώς **XSD** (Extensible Schema Definition) είναι παρομοίως μια περιγραφική γλώσσα η οποία περιγράφει την δομή αρχείων XML. Βασίζεται κυρίως σε περιορισμούς - συσχετίσεις μεταξύ των στοιχείων και στο επιτρεπτό περιεχόμενο ενός αρχείου XML. Ένα αρχείο XSD μπορεί να χρησιμοποιηθεί και ως αρχείο XML αλλά δεν ισχύει το αντίστροφο.

3.3.2 Εργαλεία – Λογισμικά

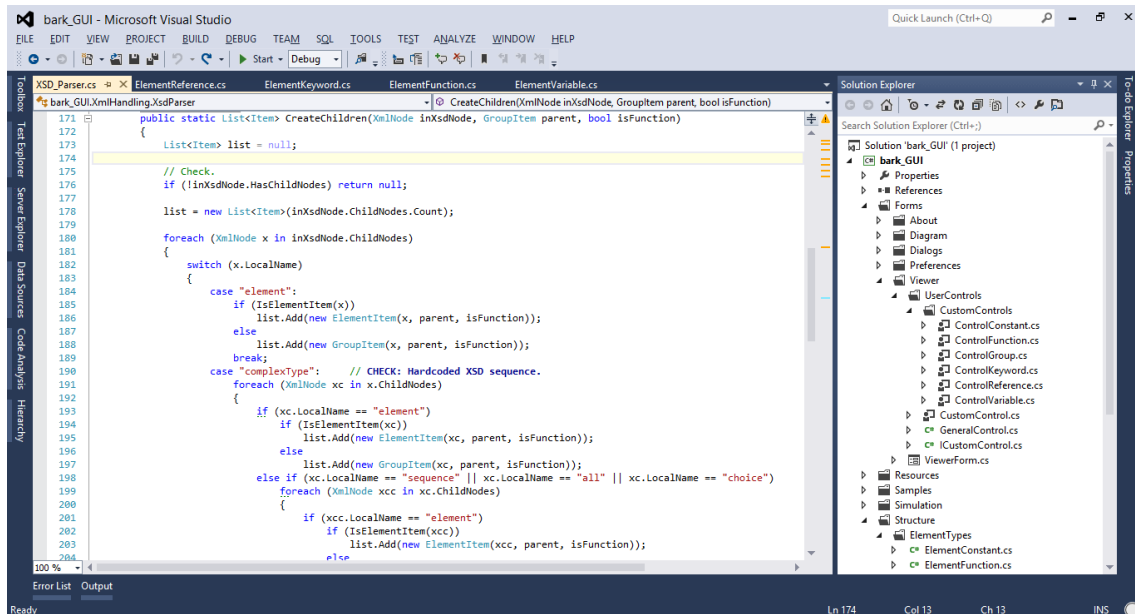
Τα παρακάτω λογισμικά ήταν απαραίτητα για την δημιουργία του γραφικού περιβάλλοντος, αλλά δεν είναι όλα αναγκαία για να επαναληφθεί η διαδικασία. Για παράδειγμα το λογισμικό ArgoUML χρησιμοποιήθηκε για την σχεδίαση διαγραμμάτων και μοντελοποίησης του λογισμικού. Τα διαγράμματα βοήθησαν στην κατανόηση της δομής του προγράμματος κατά τη διάρκεια της ανάπτυξης του γραφικού περιβάλλοντος, αλλά εφόσον έχει φτάσει σε αυτό το σημείο δεν είναι αναγκαία για την συνέχισή του.

Bark

Χρησιμοποιήθηκε για την εκτέλεση προσομοίωσης και για τον υπολογισμό των δεδομένων προσομοίωσης. Το λογισμικό Bark δημιουργήθηκε από τον καθηγητή Χαραλάμπους Ονούφριου του τμήματος Μηχανολογίας του ΤΕΙ Θεσσαλίας. Αναλυτικότερα η λειτουργία και χρησιμότητά του περιγράφεται στο δεύτερο κεφάλαιο.

Microsoft Visual Studio

Ευρέως διαδεδομένο περιβάλλον προγραμματισμού το οποίο χρησιμοποιήθηκε για την συγγραφή του κυρίως κώδικα του λογισμικού σε γλώσσα προγραμματισμού C#. Στην Εικόνα 3.1 φαίνεται ένα στιγμιότυπο του λογισμικού εν ώρα χρήσης.

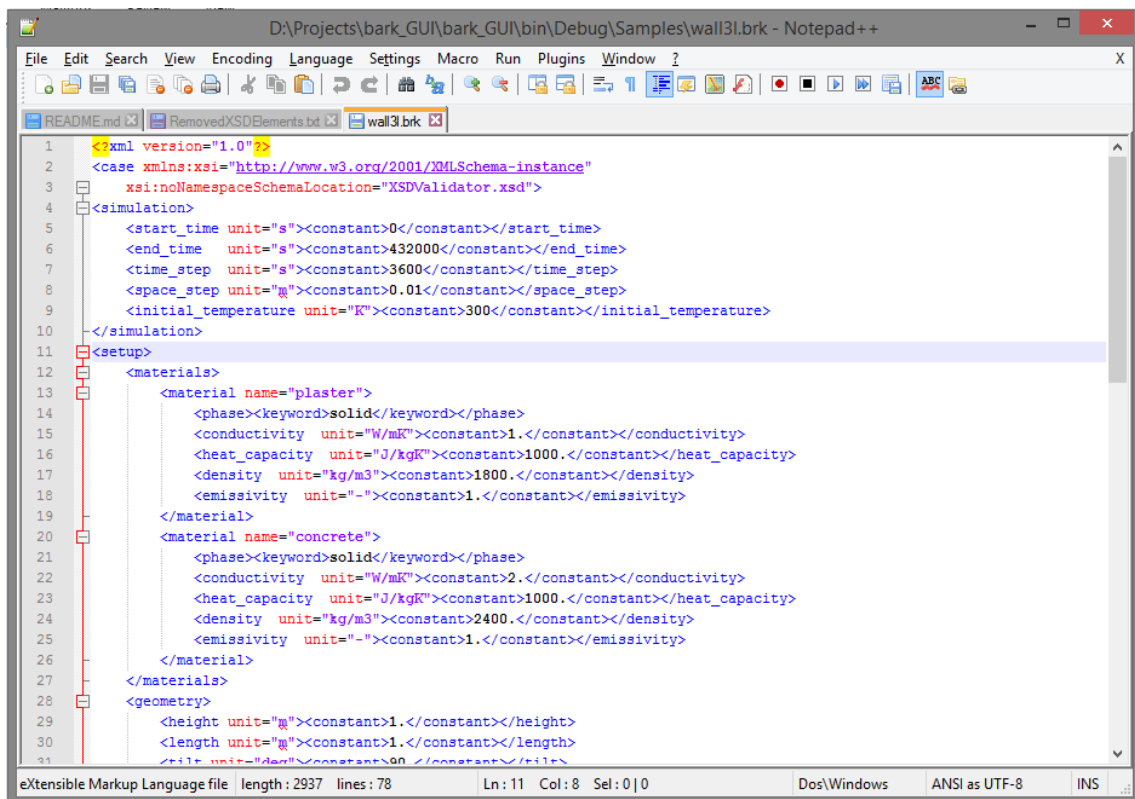


Εικόνα 3.1 – Περιβάλλον προγραμματισμού Visual Studio

Παρέχει χρωματοποίηση, αρίθμηση, στοίχιση και αυτόματη συμπλήρωση κώδικα, καθώς και εύρεση λαθών σε όσες γλώσσες υποστηρίζει. Επίσης παρέχει μεταγλώττιση του κώδικα και δημιουργία εκτελέσιμου αρχείου. Οι λειτουργίες του είναι υπερβολικά πολλές και δεν χρειάζεται να αναφερθούν όλες εδώ.

Notepad++ με την επέκταση XML Tools

Για την επεξεργασία των αρχείων XML όπως και για την συγγραφή των σχημάτων των τους (αρχεία XSD) χρησιμοποιήθηκε το λογισμικό ανοιχτού κώδικα Notepad++ και η επέκτασή του XML Tools. Στην Εικόνα 3.2 φαίνεται ένα στιγμιότυπο του λογισμικού εν ώρα χρήσης.

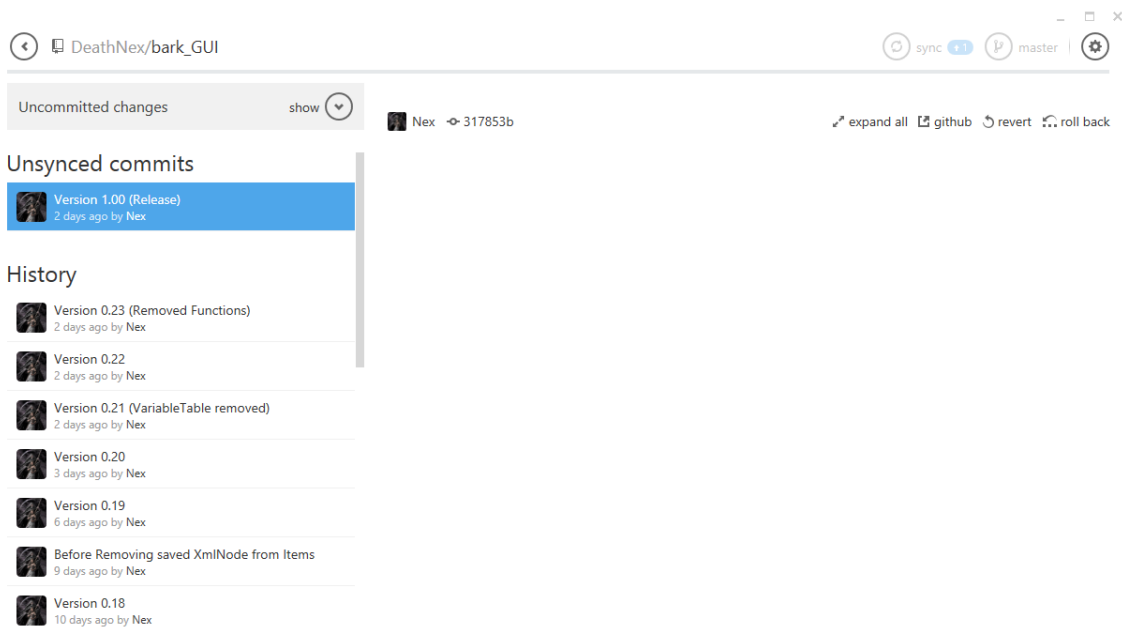


Εικόνα 3.2 – Λογισμικό Notepad++ με την επέκταση XML Tools

Παρέχει χρωματοποίηση, αρίθμηση και στοίχιση κώδικα, καθώς και εύρεση λαθών στα αρχεία τύπου XML. Επίσης επιτρέπει τον αυτόματο έλεγχο εγκυρότητας ενάντια ενός αρχείου σχήματος.

GitHub

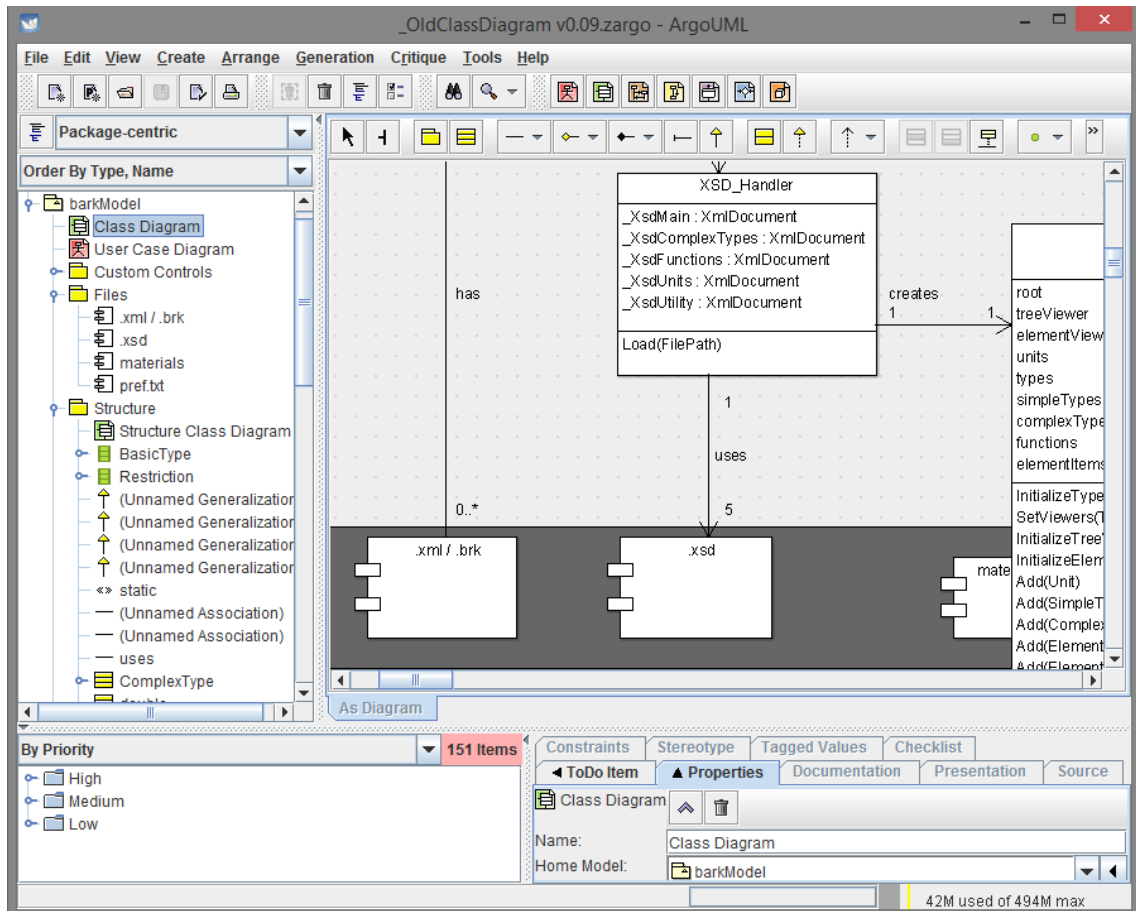
Για τον έλεγχο εκδόσεων κατά τη διαδικασία ανάπτυξης του γραφικού περιβάλλοντος χρησιμοποιήθηκε το GitHub. Δεν είναι απαραίτητο για την ανάπτυξη κώδικα, αλλά είναι πολύ χρήσιμο. Στην Εικόνα 3.3 φαίνεται ένα στιγμιότυπο του λογισμικού εν ώρα χρήσης.



Εικόνα 3.3 – Γραφικό περιβάλλον λογισμικού ελέγχου εκδόσεων GitHub

ArgoUML

Για την δημιουργία των διαγραμμάτων τύπου UML (Unified Modeling Language), τα οποία βοήθησαν στην κατανόηση της συνολικής δομής της εργασίας, χρησιμοποιήθηκε το ArgoUML. Παρομοίως δεν είναι απαραίτητο για την ανάπτυξη κώδικα, αλλά είναι πολύ χρήσιμο για την αναπαράσταση και κατανόηση της δομής. Στην Εικόνα 3.4 φαίνεται ένα στιγμιότυπο του λογισμικού εν ώρα χρήσης.



Εικόνα 3.4 – Λογισμικό διαγραμμάτων UML ArgoUML

ZedGraph

Για την σχηματική απεικόνιση των δεδομένων εξόδου του Bark χρησιμοποιήθηκε η βιβλιοθήκη ZedGraph η οποία είναι γραμμένη σε γλώσσα προγραμματισμού C#.

4 Ανάλυση Γραφικού Περιβάλλοντος

Παρά τα εμφανή στοιχεία του γραφικού περιβάλλοντος, η μεγαλύτερη δυσκολία υλοποίησής του ήταν η δημιουργία κατάλληλης δομής ώστε το πρόγραμμα να μην περιορίζει τη χρήση του. Αναλυτικότερα το προαναφερόμενο πρόβλημα περιγράφεται στο κεφάλαιο 4.2.

4.1 Διαγράμματα UML

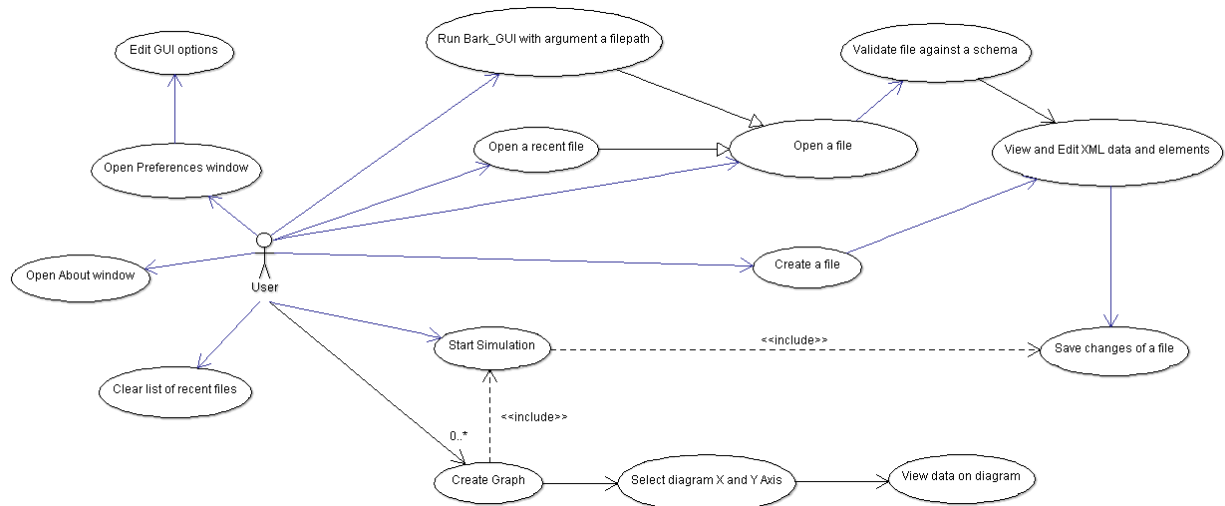
Για την κατανόηση, σχεδίαση και ανάπτυξη της δομής του γραφικού περιβάλλοντος ήταν απαραίτητα τα διαγράμματα UML. Είναι χρήσιμα διότι ο άνθρωπος έχει συνηθίσει να βλέπει σχήματα και όχι πληροφορίες σε δυσανάγνωστη μορφή. Παρακάτω περιέχονται τα διαγράμματα τα οποία περιγράφουν την δομή του γραφικού περιβάλλοντος σε απλή μορφή χωρίς λεπτομέρειες, ακολουθούμενες από μια συνοπτική περιγραφή του σχήματος.

4.1.1 Τι είναι η UML

Η Unified Modeling Language (UML) είναι μία γλώσσα που χρησιμοποιείται για προδιαγραφές, αναπαράσταση με οπτικό τρόπο, δημιουργία και τεκμηρίωση των τμημάτων των συστημάτων λογισμικού, καθώς και για μοντελοποίηση εταιρικών και άλλως συστημάτων που δεν αφορούν λογισμικό. Η UML αποτελεί ένα συνδυασμό των καλύτερων πρακτικών, οι οποίες έχουν αποδείξει την επιτυχία τους στη μοντελοποίηση μεγάλων και σύνθετων συστημάτων.

4.1.2 Περίπτωση χρήσης

Ένα διάγραμμα περίπτωσης χρήσης (use case) περιγράφει μια συνηθισμένη χρήση του προγράμματος από έναν απλό χρήστη και τη ροή εκτέλεσης με κάθε ενέργεια του χρήστη.



Εικόνα 4.1 Διάγραμμα UML – Σενάριο Χρήσης Bark_GUI

Αναλύοντας το παραπάνω διάγραμμα UML στην Εικόνα 4.1 με κέντρο αναφοράς τον χρήστη ξεκινώντας από πάνω με φορά ρολογιού διακρίνεται πως ο χρήστης (User) έχει τη δυνατότητα να κάνει τις παρακάτω ενέργειες:

- Εκτέλεση του γραφικού περιβάλλοντος με παράμετρο το μονοπάτι/τοποθεσία ενός αρχείου στον υπολογιστή.
- Επιλογή ενός πρόσφατα χρησιμοποιημένου αρχείου δεδομένων.
- Άνοιγμα ενός αρχείου δεδομένων από τον υπολογιστή.

Οι δυο προηγούμενες ενέργειες αποτελούν άλλον τρόπο για το άνοιγμα ενός αρχείου και υπάρχουν για λόγους ευχρηστίας του προγράμματος. Με το άνοιγμα του αρχείου ακολουθεί ο έλεγχος εγκυρότητάς του. Υπό κανονικές συνθήκες ο χρήστης δεν θα έχει γνώση αυτού του βήματος, αλλά χρησιμεύει σε περιπτώσεις λάθους επιλογής αρχείου και προβλήματος στη δομή των δεδομένων XML. Εφόσον το αρχείο είναι έγκυρο ο χρήστης μπορεί να δει και να επεξεργαστεί τα δεδομένα του αρχείου και τελικά να αποθηκεύσει τις αλλαγές του.

- Δημιουργία νέου αρχείου δεδομένων.

Επιτρέπει στο χρήστη να δημιουργήσει νέο αρχείο δεδομένων με την κατάλληλη δομή χωρίς να χρειαστεί να γνωρίζει την γλώσσα XML ή XSD και να το αποθηκεύσει για μελλοντική χρήση.

➤ Εκκίνηση προσομοίωσης.

Εκτελεί το λογισμικό bark με παράμετρο το μονοπάτι/τοποθεσία του ήδη φορτωμένου αρχείου δεδομένων στον υπολογιστή και δημιουργείται το αρχείο με κατάληξη .dat το οποίο περιέχει τα δεδομένα εξόδου – αποτελέσματα υπολογισμού. Απαιτείται φορτωμένο και αποθηκευμένο αρχείο XML με έναν από τους προαναφερόμενους τρόπους.

➤ Δημιουργία νέου γραφήματος.

Προτού δημιουργηθεί το γράφημα, ζητείται από το χρήστη να επιλέξει από τα δεδομένα εξόδου τους άξονες X και Y. Έστερα δημιουργείται το διάγραμμα για επισκόπηση από το χρήστη. Μπορεί να εκτελεστεί όσες φορές επιθυμεί ο χρήστης τοποθετώντας διαφορετικά δεδομένα στους άξονες κάθε φορά. Απαιτείται εκκίνηση προσομοίωσης στο φορτωμένο αρχείο XML προτού δημιουργηθεί το γράφημα.

➤ Εκκαθάριση λίστας πρόσφατα χρησιμοποιημένων αρχείων.

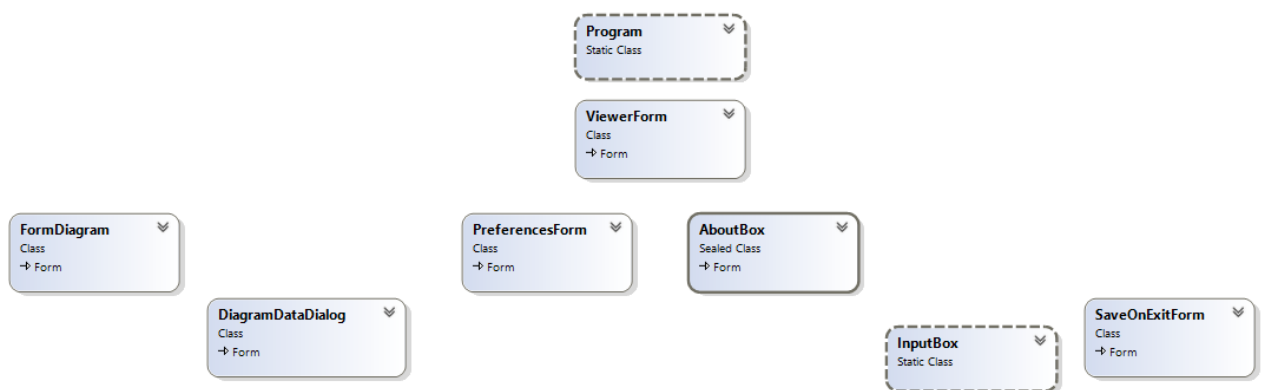
➤ Εμφάνιση παραθύρου πληροφοριών.

➤ Εμφάνιση παραθύρου προτιμήσεων.

Επεξεργασία και αποθήκευση επιλογών του γραφικού περιβάλλοντος στον υπολογιστή.

4.1.3 Δομή παραθύρων και ροή εκτέλεσης

Στο παρακάτω σχήμα στην Εικόνα 4.2 φαίνονται όλα τα παράθυρα που υπάρχουν στο γραφικό περιβάλλον και η ροή εκτέλεσης μεταξύ τους. Από πάνω προς τα κάτω είναι ιεραρχημένα με κριτήριο την σημαντικότητα και συχνότητα χρήσης, με αποτέλεσμα τα τελευταία παράθυρα να είναι διάλογοι με το χρήστη.



Εικόνα 4.2 Διάγραμμα UML – Δομή παραθύρων προγράμματος

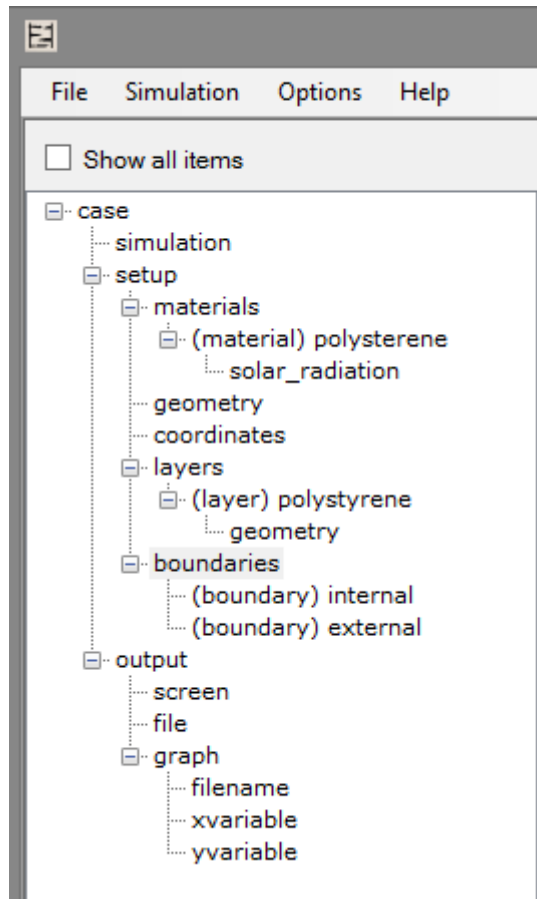
Program

Η εκτέλεση του προγράμματος ξεκινάει από την στατική κλάση Program. Εκεί δημιουργείται το κεντρικό παράθυρο ViewerForm το οποίο χρησιμοποιείται καθ' όλη τη διάρκεια χρήσης του προγράμματος. Μπορεί να χρησιμοποιηθεί και μια τοποθεσία αρχείου .brk ως παράμετρος του προγράμματος για να φορτωθεί αυτό το αρχείο αυτόματα κατά την έναρξη.

ViewerForm

Κεντρικό παράθυρο το οποίο επιτρέπει στο χρήστη να δημιουργήσει, φορτώσει, αποθηκεύσει αρχεία δεδομένων, να εκτελέσει προσομοίωση με χρήση του λογισμικού bark, να δημιουργήσει γραφήματα με τα δεδομένα εξόδου του bark, να αλλάξει τις ρυθμίσεις του γραφικού περιβάλλοντος μέσω των ενεργειών του μενού.

Το κεντρικό αυτό παράθυρο επίσης περιέχει και τα παραθυρικά στοιχεία ελέγχου (Window Controls). Ένα βασικό από αυτά είναι το TreeViewer, το οποίο είναι υπεύθυνο για την εμφάνιση της δομής του αρχείου δεδομένων σε δέντρο (TreeView Control) και βρίσκεται στα αριστερά του κεντρικού παραθύρου όπως φαίνεται στην Εικόνα 4.3.



Εικόνα 4.3 – Δομή αρχείου XML σε δέντρο (TreeView Control)

Το δεύτερο βασικό παραθυρικό στοιχείο ελέγχου είναι το ElementViewer και είναι υπεύθυνο για την σωστή εμφάνιση των υπόλοιπων προσαρμοσμένων χειριστηρίων (Custom Controls) τα οποία χρησιμοποιούνται για την επεξεργασία των δεδομένων ενός αρχείου XML.

Στο κεντρικό παράθυρο επίσης περιέχονται και τα στοιχεία ελέγχου τα οποία δείχνουν την κατάσταση του προγράμματος και το μονοπάτι του επιλεγμένου στοιχείου στο δέντρο.

FormDiagram

Το δεύτερο σημαντικότερο παράθυρο στο γραφικό περιβάλλον περιέχει τα διαγράμματα τα οποία παρέχει στο χρήστη. Μπορούν να γίνουν διάφορες ενέργειες για την διευκόλυνση της σωστής επισκόπησης δεδομένων. Το διάγραμμα δημιουργείται από το κεντρικό παράθυρο (ViewerForm) εφόσον ο χρήστης έχει επιλέξει από το μενού Simulation – Create Graph και επίσης έχει διαλέξει τους επιθυμητούς άξονες.

PreferencesForm

Δευτερεύον παράθυρο το οποίο δίνει στο χρήστη επιλογές που αφορούν το γραφικό περιβάλλον. Παράδειγμα επιλογής είναι οι τοποθεσίες αρχείων XML και μονοπάτια φακέλων στον υπολογιστή. Πολύ χρήσιμο παράθυρο για αντιμετώπιση προβλημάτων.

AboutBox

Παράθυρο πληροφοριών για το γραφικό περιβάλλον και για το λογισμικό bark. Οι σημαντικότερες πληροφορίες που περιέχει είναι ο αριθμός έκδοσης γραφικού περιβάλλοντος, πληροφορίες για τους δημιουργούς των λογισμικών και τρόπο επικοινωνίας.

DiagramDataDialog

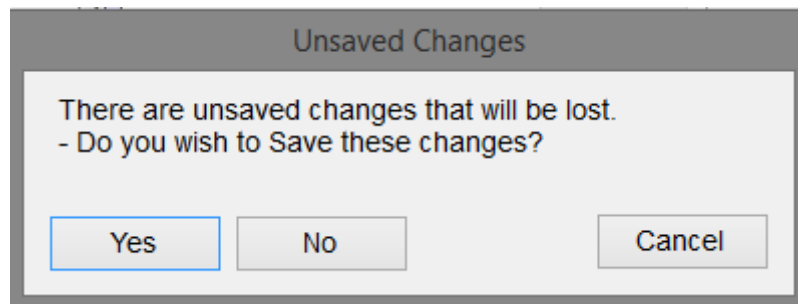
Διάλογος ο οποίος εμφανίζεται αμέσως προτού δημιουργηθεί το παράθυρο διαγράμματος. Επιτρέπει στο χρήστη να επιλέξει ποια δεδομένα να εμφανιστούν και σε ποιους άξονες. Μπορούν να επιλεγθούν πολλά στοιχεία για τον άξονα Y, αλλά μόνο ένα για τον άξονα X.

InputBox

Δυναμικός διάλογος ο οποίος δημιουργείται για να εμφανίσει στο χρήστη ένα φιλικό μήνυμα και ζητάει ένα αλφαριθμητικό στοιχείο το οποίο θα χρησιμοποιηθεί ανάλογα. Προς το παρόν έχει χρησιμοποιηθεί για την μετονομασία ενός στοιχείου XML (XML element) δια μέσω του δέντρου με χρήση δεξιού κλικ και επιλογή 'Rename'.

SaveOnExitForm

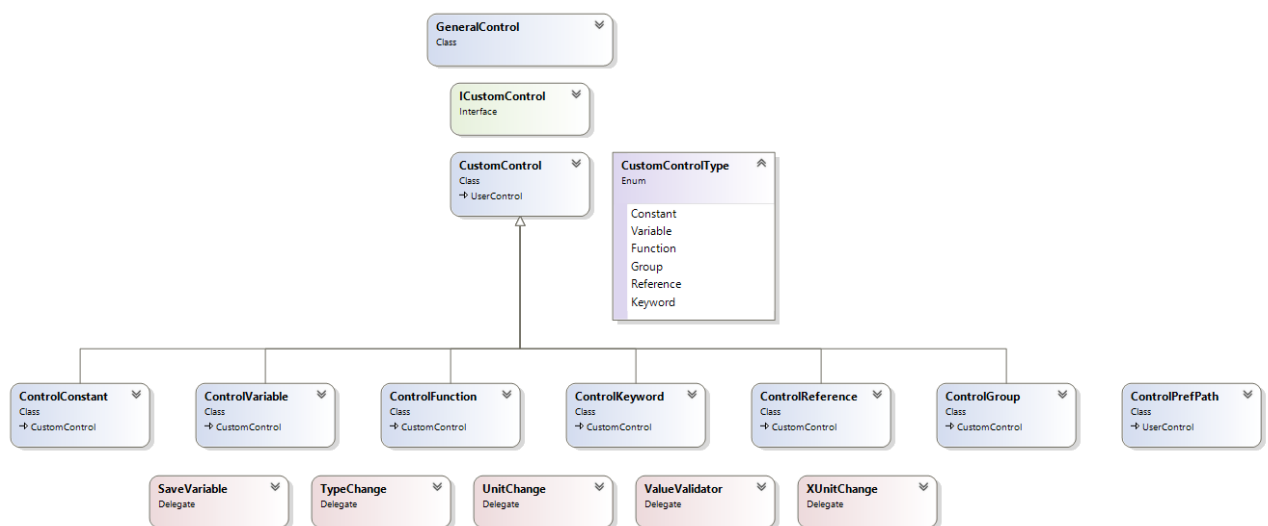
Απλός διάλογος ο οποίος εμφανίζεται όταν ο χρήστης κάνει κάποια ενέργεια να κλείσει το φορτωμένο αρχείο δεδομένων χωρίς να έχει αποθηκεύσει όλες τις αλλαγές που έχει κάνει. Ο χρήστης έχει τη επιλογή να αποθηκεύσει το αρχείο, να καταργήσει τις αλλαγές ή να ακυρώσει την ενέργεια που έκανε κρατώντας το φορτωμένο αρχείο ανοιχτό. Ο διάλογος έχει την μορφή που φαίνεται στην Εικόνα 4.4.



Εικόνα 4.4 – Διάλογος αποθήκευσης αλλαγών

4.1.4 Προσαρμοσμένα στοιχεία ελέγχου

Τα προσαρμοσμένα στοιχεία ελέγχου (Custom Controls) είναι το κύριο μέρος του γραφικού περιβάλλοντος το οποίο βλέπει ο χρήστης και χρησιμοποιεί εκτενώς καθ' όλη τη διάρκεια επεξεργασίας δεδομένων. Η δομή τους όπως φαίνεται στην Εικόνα 4.5 έχει σχεδιαστεί με τέτοιο τρόπο ώστε να είναι κοντά στη δομή των αντικειμένων του προγράμματος, αλλά και να επιτρέπουν στον χρήστη ευκολία και άνεση στην επεξεργασία δεδομένων. Για παράδειγμα εάν γίνει εναλλαγή του τύπου ενός αντικειμένου (π.χ. από σταθερό σε μεταβλητό), οι τιμές που έχει κάθε στοιχείο ελέγχου δεν χάνεται, με αποτέλεσμα ο χρήστης να μην φοβάται να κάνει εναλλαγή τύπου.



Εικόνα 4.5 Διάγραμμα UML – Προσαρμοσμένα στοιχεία ελέγχου

Υπάρχουν επτά είδη στοιχείων ελέγχου σε παράθυρα της εφαρμογής, από τα οποία τα πιο σημαντικά είναι τα έξι προσαρμοσμένα στοιχεία ελέγχου (CustomControls), το ControlConstant, ControlVariable, ControlFunction, ControlKeyword, ControlReference και ControlGroup. Το καθένα από αυτά βρίσκεται στον ElementViewer, οπότε φαίνονται συχνά στον χρήστη.

GeneralControl

Το GeneralControl είναι το γενικό στοιχείο ελέγχου το οποίο απλοποιεί τις ενέργειες που αλλιώς θα γίνονταν στα προσαρμοσμένα στοιχεία ελέγχου ένα-ένα ξεχωριστά. Για αυτό το λόγο περιέχει μια συλλογή από τα προαναφερόμενα προσαρμοσμένα στοιχεία ελέγχου. Το GeneralControl είναι το στοιχείο ελέγχου με το οποίο συσχετίζεται κάθε αντικείμενο-στοιχείο (Item) για να γίνει οποιαδήποτε ενέργεια στο γραφικό περιβάλλον που αφορά τα στοιχεία ελέγχου.

CustomControl

Είναι η γενίκευση των έξι παρακάτω προσαρμοσμένων στοιχείων ελέγχου και δεν υπάρχει αυτούσιο σε κάποιο παράθυρο. Περιέχει τις μεθόδους και μεταβλητές που είναι κοινές σε όλα τα αντικείμενα που το κληρονομούν. Μερικά από αυτά είναι το περιεχόμενο (τιμή) ενός στοιχείου ελέγχου, ο έλεγχος εγκυρότητας της τιμής κ.α. Υπάρχει για την διευκόλυνση στο προγραμματιστικό κομμάτι και για την αποφυγή επανάληψης κώδικα.

ControlConstant

Είναι μια υλοποίηση του CustomControl η οποία αντιπροσωπεύει ένα στοιχείο XML το οποίο έχει μια σταθερή τιμή με συγκεκριμένη μονάδα μέτρησης (π.χ. χρόνος σε δευτερόλεπτα). Επιπλέον περιέχει τρόπο αλλαγής τύπου, δηλαδή μπορεί από στοιχείο ελέγχου σταθερής τιμής να μετατραπεί σε μεταβλητής τιμής (ControlConstant → ControlVariable). Η μορφή του φαίνεται στην Εικόνα 4.6.



Εικόνα 4.6 – Προσαρμοσμένο στοιχείο ελέγχου σταθερής τιμής

ControlVariable

Είναι μια υλοποίηση του CustomControl η οποία αντιπροσωπεύει ένα στοιχείο XML το οποίο έχει πολλές τιμές οι οποίες αναπαρίστανται με έναν δισδιάστατο πίνακα. Η τιμή μεταβάλλεται κάθε φορά με κάποια δεύτερη παράμετρο (π.χ. με το χρόνο). Κάθε στήλη του πίνακα έχει συγκεκριμένη μονάδα μέτρησης (π.χ. θερμοκρασία σε βαθμούς Kelvin και χρόνος σε δευτερόλεπτα). Παρομοίως περιέχει τρόπο αλλαγής τύπου.

Επειδή η επεξεργασία δεδομένων τέτοιου είδους μπορεί φτάσει μεγάλη ποσότητα τιμών και να καταντήσει περίπλοκη και χρονοβόρα, δημιουργήθηκαν κάποιες επιπλέον διευκολύνσεις. Χρησιμοποιήθηκε ειδικό στοιχείο ελέγχου όπως φαίνεται στην Εικόνα 4.7 για γρήγορο χειρισμό μεγάλου όγκου δεδομένων χωρίς προβλήματα το οποίο είναι συμβατό με το Microsoft Excel. Επίσης ο χρήστης μπορεί να έχει τα δεδομένα σε κάποιο αρχείο τύπου Microsoft Excel και να τα αντιγράψει στο παρόν στοιχείο ελέγχου καθώς και το αντίστροφο. Ο χρήστης μπορεί να προσθέτει και να διαγράφει εύκολα επιπλέον σειρές στον πίνακα κατά βούληση χωρίς καμία επιπλοκή.

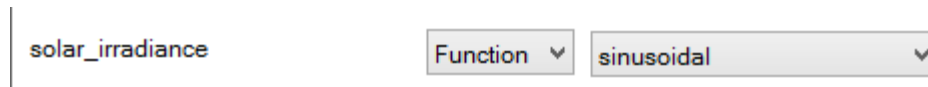
0	30.7
0.001388...	30.9
0.002777...	30.9
0.004166...	30
0.005555...	30.5
0.006944...	30.7
0.008333...	30.3
0.009722...	30.4
0.011111...	30.4
0.0125	30.3

Εικόνα 4.7 – Προσαρμοσμένο στοιχείο ελέγχου μεταβλητής τιμής

ControlFunction

Είναι μια ελλιπής υλοποίηση του CustomControl η οποία αντιπροσωπεύει ένα στοιχείο XML το οποίο έχει μεταβλητή τιμή με συγκεκριμένη συνάρτηση. Οι

συναρτήσεις ορίζονται στο αρχείο XSD και ο χρήστης επιλέγει μια από αυτές. Παρομοίως περιέχει τρόπο αλλαγής τύπου όπως φαίνεται στην Εικόνα 4.8.

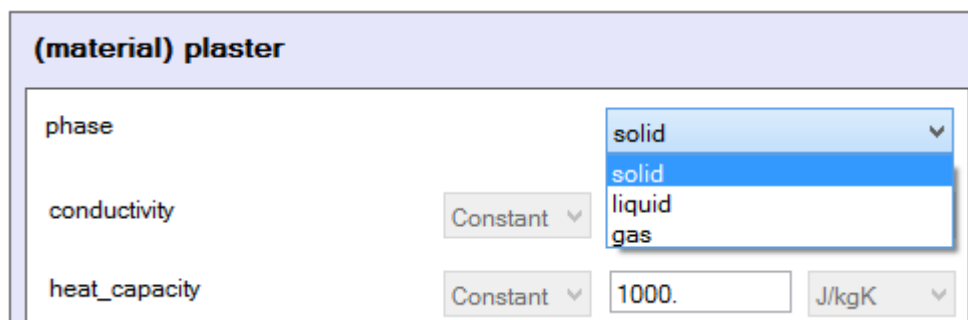


The image shows a user interface element for a parameter named 'solar_irradiance'. To the right of the parameter name is a dropdown menu labeled 'Function'. The selected option in the dropdown is 'sinusoidal'.

Εικόνα 4.8 – Προσαρμοσμένο στοιχείο ελέγχου συνάρτησης

ControlKeyword

Είναι μια υλοποίηση του CustomControl η οποία αντιπροσωπεύει ένα στοιχείο XML το οποίο έχει μια σταθερή τιμή η οποία είναι προκαθορισμένη από το αρχείο σχήματος τύπου XSD (XSDValidatorSimpleTypes.xsd). Ο χρήστης δεν μπορεί να πληκτρολογήσει κάποια τιμή, αντ' αυτού επιλέγει μια από τις ήδη προκαθορισμένες. Η μορφή του στοιχείου φαίνεται στην Εικόνα 4.9.



The image shows a user interface for a material property control. The title is '(material) plaster'. There are three rows of controls: 1. 'phase' with a dropdown menu showing 'solid', 'liquid', and 'gas' options. 2. 'conductivity' with a 'Constant' dropdown. 3. 'heat_capacity' with a 'Constant' dropdown, a text input field containing '1000.', and a unit dropdown showing 'J/kgK'.

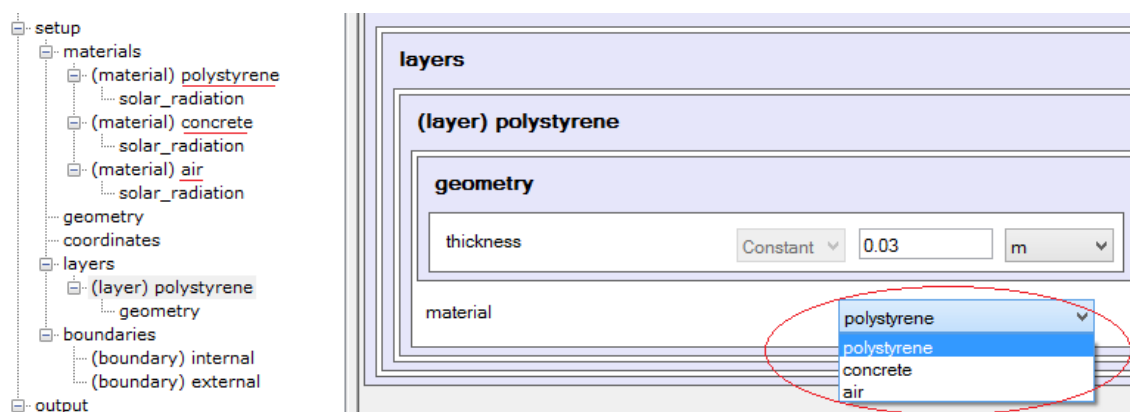
Εικόνα 4.9 – Προσαρμοσμένο στοιχείο ελέγχου προκαθορισμένης τιμής

ControlReference

Είναι μια υλοποίηση του CustomControl η οποία αντιπροσωπεύει ένα στοιχείο XML το οποίο φαινομενικά είναι ίδιο με το προαναφερόμενο στοιχείο ελέγχου ControlKeyword. Η διαφορά είναι ότι το παρόν στοιχείο ελέγχου δεν είναι αυτόνομο, αλλά η τιμή που δέχεται είναι μια αναφορά σε άλλο αντικείμενο-στοιχείο. Η τιμή που μπορεί να πάρει αλλάζει ανάλογα με τα αντικείμενα-στοιχεία που υπάρχουν σε ένα αρχείο δεδομένων.

Για την καλύτερη κατανόηση του στοιχείου ελέγχου χρειάζεται ένα πρακτικό παράδειγμα. Έστω ότι υπάρχει ένας τοίχος ο οποίος αποτελείται από τρία στρώματα τούβλου. Το αντίστοιχο αρχείο δεδομένων θα περιείχε ένα στοιχείο Material και τρία στοιχεία Layer. Η κάθε στρώση δεν έχει διαφορετικό υλικό, γιαυτό το λόγο αρκεί μια αναφορά στο υλικό αυτό (τούβλο) και αποτελεί την μοναδική επιλογή στο στοιχείο ελέγχου ControlReference. Εάν υπήρχε και άλλο υλικό, θα υπήρχαν και άλλες επιλογές στο στοιχείο ελέγχου.

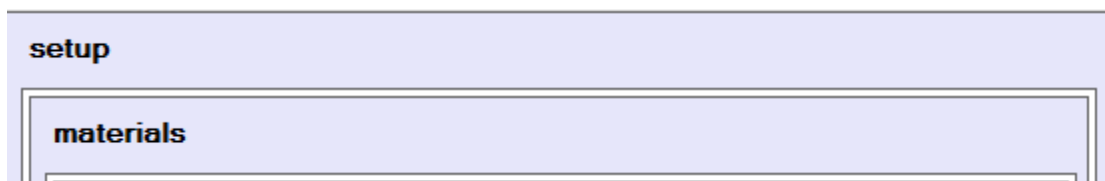
Με λίγα λόγια το παρόν στοιχείο ελέγχου δεν έχει προκαθορισμένες τιμές στο αρχείο σχήματος, αλλά έχει ορισμένο έναν περιορισμό. Ο περιορισμός αυτός αφορά άλλα στοιχεία XML και πιο συγκεκριμένα ορίζει το κανόνα με τον οποίο θα επιλέγει τα στοιχεία XML τα οποία αποτελούν πιθανή τιμή για αναφορά. Στην Εικόνα 4.10 φαίνεται η μορφή του.



Εικόνα 4.10 – Προσαρμοσμένο στοιχείο ελέγχου αναφοράς σε άλλο στοιχείο

ControlGroup

Είναι μια υλοποίηση του CustomControl η οποία αντιπροσωπεύει ένα στοιχείο XML το οποίο δεν έχει καμία τιμή, αλλά περιέχει άλλα στοιχεία XML. Σε αυτό το στοιχείο περιέχονται οι περισσότερες αναδρομικές μέθοδοι, καθώς και η λογική τους. Ένα παράδειγμα είναι ποια προσαρμοσμένα στοιχεία ελέγχου πρέπει να εμφανίζονται στο χρήστη και πότε. Εφόσον δεν έχει τιμή, διαφέρει από τα υπόλοιπα προσαρμοσμένα στοιχεία ελέγχου, ωστόσο διαφέρει περισσότερο στην λογική και στον τρόπο υλοποίησης παρά στη δομή του. Η μορφή του φαίνεται στην Εικόνα 4.11.



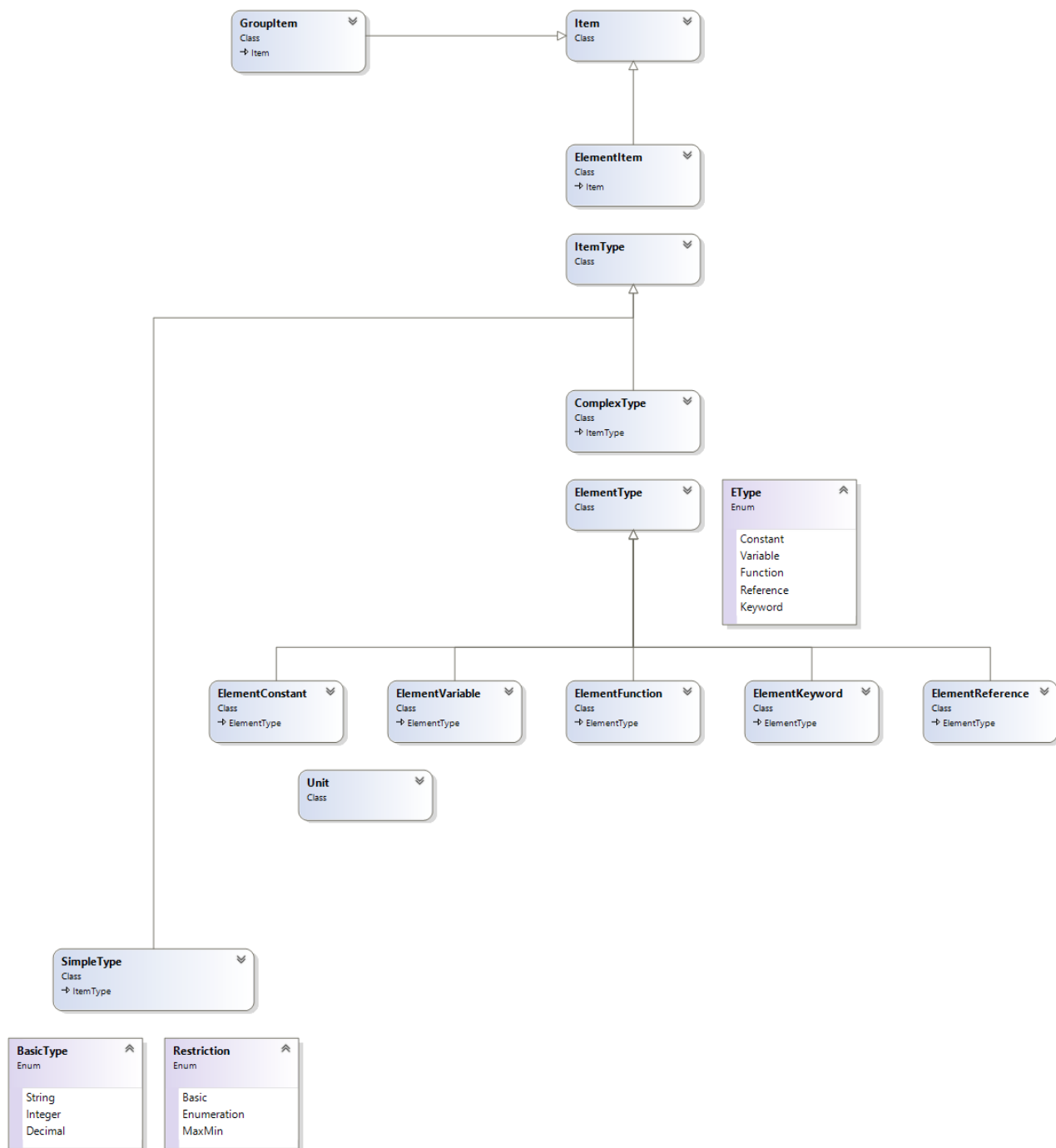
Εικόνα 4.11 – Προσαρμοσμένο στοιχείο ελέγχου ομαδοποίησης

ControlPrefPath

Το ControlPrefPath είναι δευτερεύον στοιχείο ελέγχου το οποίο χρησιμοποιείται στο παράθυρο προτιμήσεων για εισαγωγή και επεξεργασία του μονοπατιού/τοποθεσίας ενός αρχείου στον υπολογιστή.

4.1.5 Δομή αντικειμένων-στοιχείων

Τα αντικείμενα-στοιχεία (Items) αντιπροσωπεύουν ένα στοιχείο XML (element). Χρησιμοποιούνται στην εφαρμογή για να χειρίζονται τις πληροφορίες με μεγαλύτερη ευκολία απ' ό,τι θα γινόταν απ' ευθείας στα στοιχεία XML. Αποτελούν την κύρια δομή του προγράμματος στα παρασκήνια. Στην Εικόνα 4.12 φαίνεται η βασική δομή και κυρίως η ιεραρχία των κλάσεων.



Εικόνα 4.12 Διάγραμμα UML – Δομή αντικειμένων-στοιχείων [Σχέδιο Α]

Item

Ένα αντικείμενο-στοιχείο περιέχει όλες τις μεταβλητές και μεθόδους που χρειάζονται για να περιγράψουν ένα στοιχείο τύπου XML. Κάποια από αυτά είναι το όνομα του στοιχείου, ένα πιο φιλικό προς το χρήστη όνομα και το αντίστοιχο γενικό στοιχείο ελέγχου (GeneralControl). Υπάρχουν δύο τύποι αντικειμένων-στοιχείων, αυτά που περιέχουν άλλα αντικείμενα στοιχεία (GroupItem) και αυτά που έχουν κάποια τιμή (ElementItem).

GroupItem

Χρησιμοποιείται για την ομαδοποίηση αντικειμένων-στοιχείων και υλοποίηση ιεραρχίας. Οι μεταβλητές και οι μέθοδοι που περιέχει αφορούν τον μαζικό αλλά συμπυκνόμενο χειρισμό των αντικειμένων-στοιχείων. Το αντίστοιχο προσαρμοσμένο στοιχείο ελέγχου του μπορεί να είναι μόνο το ControlGroup.

ElementItem

Χρησιμοποιείται εκτενώς κατά την διαδικασία επεξεργασίας δεδομένων από το χρήστη. Είναι υπεύθυνο για την εισαγωγή, επεξεργασία, διαγραφή και έλεγχο εγκυρότητας κάθε δεδομένου εισόδου. Για να πετύχει όμως αυτό χρειάζεται να υπάρχει μια πιο περίπλοκη δομή στην κλάση αυτή.

Ένα τέτοιο αντικείμενο μπορεί να είναι διαφόρων ειδών και όχι μόνο αυτό, μπορεί να εναλλάσσεται μεταξύ αυτών κατά τη διαδικασία της επεξεργασίας δεδομένων. Περιέχει τον παρών επιλεγμένο τύπο αντικειμένου καθώς και την λογική για επαλήθευση και αποθήκευση των κατάλληλων δεδομένων. Η προαναφερθείσα λογική βρίσκεται στην κλάση ComplexType.

ItemType

Καθορίζεται ως η γενίκευση των SimpleType και ComplexType και χρησιμοποιείται ως στοιχείο δομής.

SimpleType

Αποτελεί την βάση όλων των ελέγχων εγκυρότητας τιμής ενός αντικειμένου στοιχείου. Περιέχει όλους τους περιορισμούς που φορτώθηκαν από τα αρχεία σχήματος και κάθε υλοποίησή του επιλέγει έναν ή περισσότερους από αυτούς τους περιορισμούς.

Προέρχεται από το ομώνυμο γνώρισμα της γλώσσας XSD το οποίο ομοίως περιγράφει περιορισμούς τιμών. Έχει δημιουργηθεί με τέτοιο τρόπο ώστε να συμβαδίζει πλήρως με το ομώνυμο γνώρισμά του.

Αναλυτικότερα μια τιμή μπορεί να υπόκειται σε δυο ειδών *περιορισμούς*:

Restriction:

- Βασικός Τύπος

Δεν ισχύει επιπλέον περιορισμός, εκτός από αυτόν που επιλέχθηκε στο *BasicType*. Εάν και εκεί έχει επιλεγθεί το αλφαριθμητικό, τότε δεν ισχύει κανένας περιορισμός σε αυτή την τιμή.

- Απαρίθμηση Επιλογών

Συνήθως εφαρμόζεται σε αλφαριθμητικούς και επιτρέπει τιμές από μια λίστα η οποία έχει οριστεί στα αρχεία σχήματος (XSD). Οποιαδήποτε άλλη τιμή δεν είναι αποδεκτή.

- Μέγιστο – Ελάχιστο

Μπορεί να εφαρμοστεί μόνο σε αριθμούς, είτε ακέραιους είτε δεκαδικούς. Θέτει ένα ή δυο όρια για την τιμή τα οποία αν δεν τηρούνται, η τιμή καταλήγει άκυρη και δεν είναι αποδεκτή. Συνήθως χρησιμοποιείται για να οριστεί ένας αριθμός μόνο θετικός ή μόνο αρνητικός, αλλά δεν περιορίζεται μόνο σε αυτή τη χρήση.

Αξίζει να σημειωθεί ότι τα όρια που ορίζονται περιέχουν τον αριθμό που τέθηκε. Για παράδειγμα εάν οριστεί ελάχιστο όριο το μηδέν, τότε το μηδέν είναι αποδεκτός αριθμός αλλά το -1 δεν είναι.

BasicType:

Αποτελείται από τους βασικούς τύπους μιας μεταβλητής – τιμής:

- Αλφαριθμητικό

Στην ουσία αυτός ο τύπος δεν έχει κανέναν περιορισμό, επιτρέπει αριθμούς, γράμματα και οποιονδήποτε χαρακτήρα. Ωστόσο μπορεί να υπάρξει επιπλέον περιορισμός από το Restriction.

- Ακέραιος αριθμός

Δέχεται μόνο ακέραιους θετικούς και αρνητικούς αριθμούς (π.χ. -2, -1, 0, 1, 2, ...) εκτός και αν ισχύει επιπλέον περιορισμός από το Restriction.

- Δεκαδικός αριθμός

Δέχεται οποιονδήποτε πραγματικό αριθμό με δεκαδικά στοιχεία ή χωρίς, θετικούς ή αρνητικούς εκτός και αν ισχύει επιπλέον περιορισμός από το Restriction.

ComplexType

Αποτελεί μια από τις βασικότερες κλάσεις η οποία είναι υπεύθυνη για ένα αντικείμενο-στοιχείο τιμής (ElementItem), για τον τύπο που είναι αυτή τη στιγμή επιλεγμένος, για τους τύπους στους οποίους μπορεί να μετατραπεί, τις τιμές τους καθώς και τον έλεγχο εγκυρότητας των τιμών τους. Είναι μια αντιπροσώπευση από ένα ομώνυμο γνώρισμα της γλώσσας XSD το οποίο περιγράφει έναν περίπλοκο τύπο από στοιχεία (elements).

ElementType

Γενίκευση των κλάσεων που το κληρονομούν (ElementConstant, ElementVariable, ...) η οποία περιέχει τα κοινά στοιχεία τους. Κάποια από αυτά είναι η τιμή του καθενός αντικειμένου, η προεπιλεγμένη τιμή η οποία ορίζεται στα αντίστοιχα αρχεία σχήματος, τον έλεγχο εγκυρότητας μιας τιμής με βάση συγκεκριμένους περιορισμούς οι οποίοι βρίσκονται στην κλάση SimpleType.

ElementConstant

Είναι η αντιπροσώπευση ενός στοιχείου σταθερής τιμής με τα αντίστοιχα δεδομένα. Περιέχει επιπλέον μια μονάδα μέτρησης (Unit) μαζί με την προεπιλεγμένη τιμή της και μεθόδους διαχείρισής της.

ElementVariable

Είναι η αντιπροσώπευση ενός στοιχείου μεταβλητής τιμής με τα αντίστοιχα δεδομένα. Περιέχει επιπλέον δυο μονάδες μέτρησης (Units) μαζί με τις προεπιλεγμένες τιμές τους και μεθόδους διαχείρισής τους.

ElementFunction

Είναι η αντιπροσώπευση ενός στοιχείου συνάρτησης και περιέχει συλλογή με τα ονόματα των επιτρεπτών ως επιλογή συναρτήσεων.

ElementKeyword

Είναι η αντιπροσώπευση ενός στοιχείου προκαθορισμένης τιμής και περιέχει συλλογή με τις επιτρεπτές τιμές.

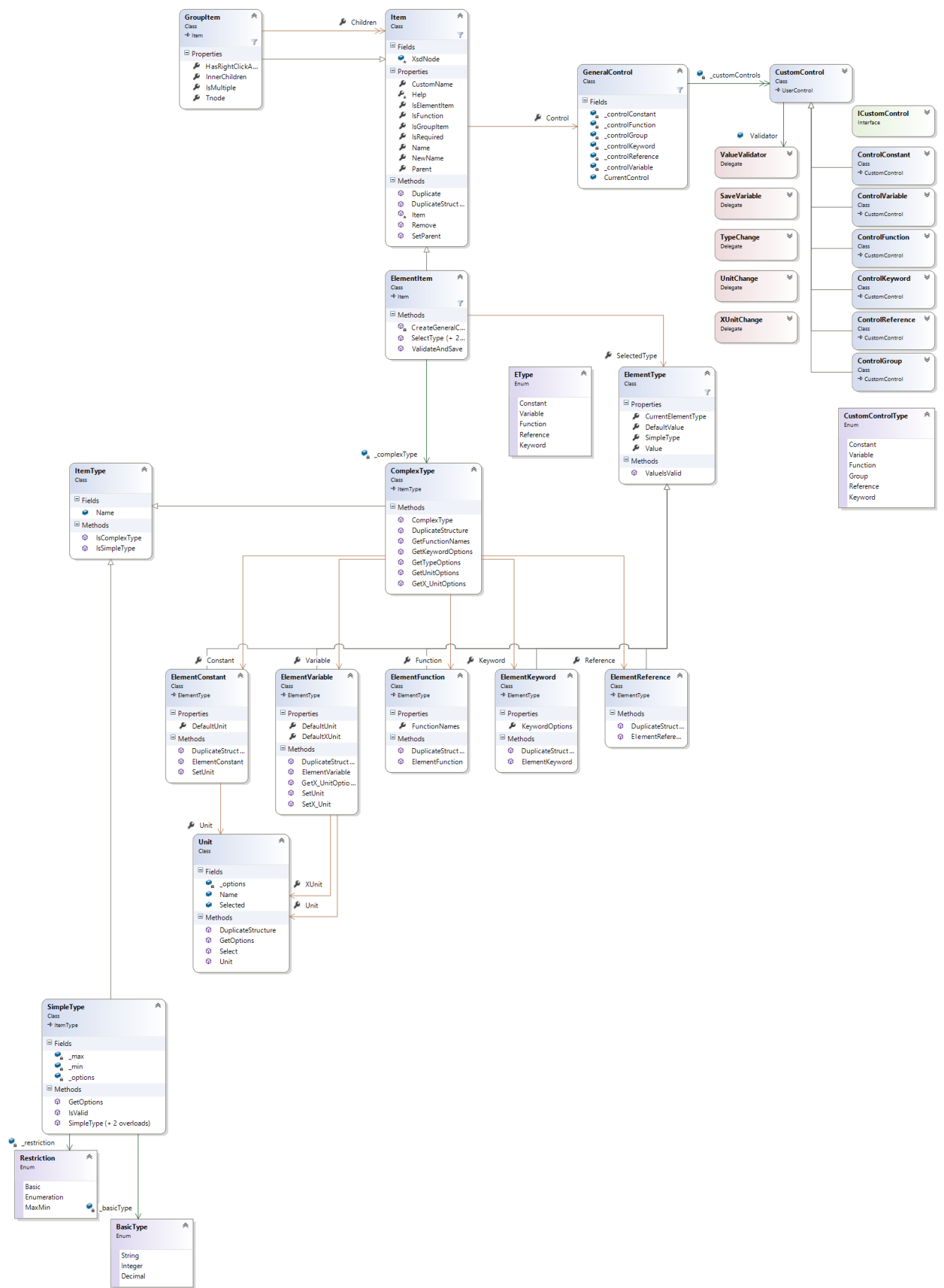
ElementReference

Είναι η αντιπροσώπευση ενός στοιχείου αναφοράς με τα αντίστοιχα δεδομένα.

Unit

Αντιπροσωπεύει μια μονάδα μέτρησης με τις προεπιλεγμένες πιθανές τιμές της και την παρούσα επιλεγμένη τιμή. Χρησιμοποιείται στα αντικείμενα-στοιχεία σταθερής και μεταβλητής τιμής.

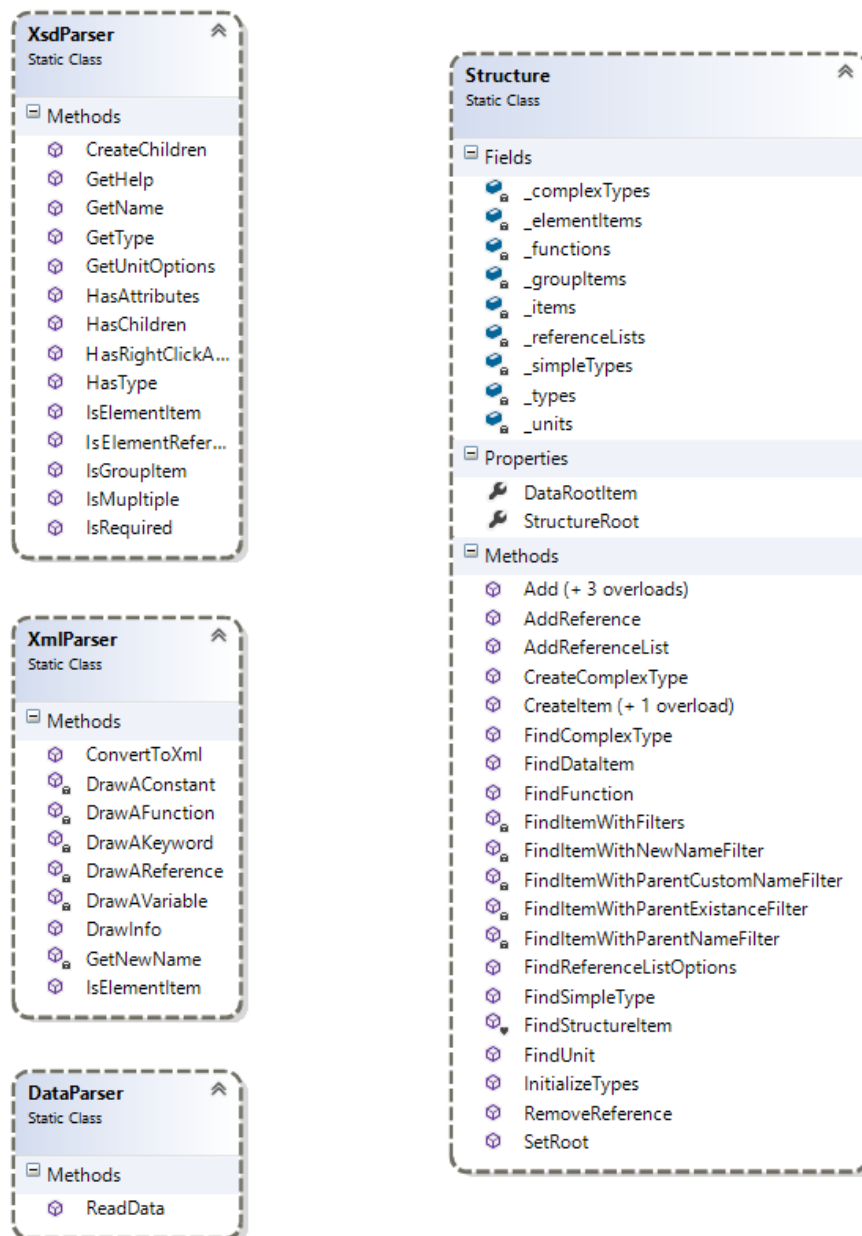
Παρακάτω στην Εικόνα 4.13 απεικονίζονται πιο αναλυτικά τα μέλη της κάθε κλάσης καθώς και τις συσχετίσεις μεταξύ τους. Επίσης φαίνεται και ο τρόπος σύνδεσης των στοιχείων ελέγχους με τα αντικείμενα-στοιχεία. Καθώς το σχέδιο είναι αρκετά πιο πολύπλοκο, δεν θα αναλυθεί εδώ, ωστόσο είναι αρκετά σημαντικό και χρήσιμο να υπάρχει για όποιον θέλει να μπει εις βάθος στην δομή του γραφικού περιβάλλοντος.



Εικόνα 4.13 Διάγραμμα UML – Δομή αντικειμένων-στοιχείων [Σχέδιο B]

4.1.6 Δομή και διαχείριση δεδομένων των αντικειμένων-στοιχείων

Μέχρι τώρα παρουσιάστηκε η δομή των αντικειμένων-στοιχείων ως επιμέρους κλάσεις, αλλά όχι το πώς συνδέονται. Όπως δείχνει η παρακάτω Εικόνα 4.14 με στατικές κλάσεις.



Εικόνα 4.14 Διάγραμμα UML – Δομή δεδομένων των αντικειμένων-στοιχείων

Structure

Αποτελεί τον σκελετό όλων των στοιχείων-αντικειμένων. Δημιουργείται όταν φορτώνεται το αρχείο XSD και περιέχει τη συνολική δομή των αντικειμένων και της

εφαρμογής. Συνήθως για το υπόλοιπο πρόγραμμα χρησιμοποιούνται μόνο τα αντικείμενα-στοιχεία και καθόλου τα μέρη που το αποτελούν, οπότε οι συλλογές αντικειμένων που εκτίθενται στο υπόλοιπο πρόγραμμα δεν είναι πολλές.

XsdParser

Βοηθητική κλάση η οποία δημιουργήθηκε με σκοπό να περιέχει ο,τι έχει σχέση με την γλώσσα XSD και να μην μπερδεύει/περιπλέκει την ροή εκτέλεσης του προγράμματος. Στην ουσία είναι μεταφραστής δεδομένων από γλώσσα XSD σε μια μορφή που βολεύει η διαχείρισή τους.

XmlParser

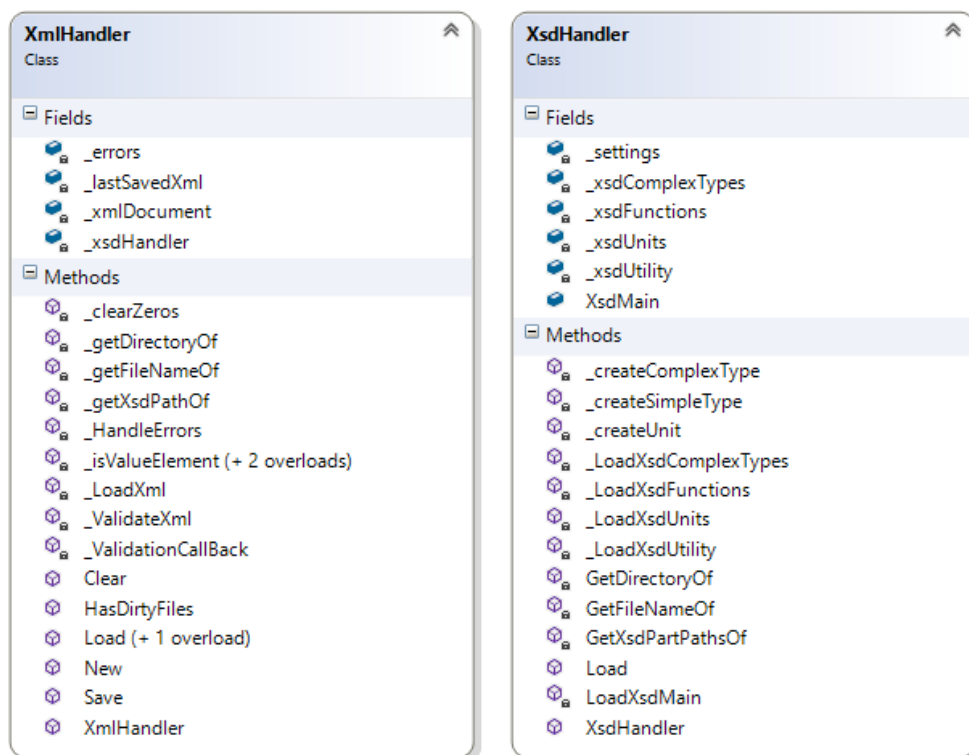
Παρόμοια με τον XsdParser, είναι βοηθητική κλάση η οποία δημιουργήθηκε με σκοπό να περιέχει ο,τι έχει σχέση με την γλώσσα XML και να μην μπερδεύει/περιπλέκει την ροή εκτέλεσης του προγράμματος. Στην ουσία είναι μεταφραστής δεδομένων από γλώσσα XML σε μια μορφή που βολεύει η διαχείρισή τους. Αντιθέτως με τον XsdParser εδώ δεν είναι πολλές μικρές μέθοδοι, αλλά λίγες και αναδρομικές οι οποίες περικλείουν την λογική εκτέλεσης.

DataParser

Απλή βοηθητική κλάση η οποία μεταφράζει τα αρχεία εξόδου του λογισμικού bark με κατάληξη .dat σε επεξεργάσιμα δεδομένα από το γραφικό περιβάλλον για εισαγωγή στα γραφήματα.

4.1.7 Διαχείριση αρχείων

Η διαδικασία η οποία βρίσκεται στο ενδιάμεσο των καταστάσεων, τα δεδομένα βρίσκονται στο αρχείο XML και τα δεδομένα βρίσκονται στην εφαρμογή. Ασχολείται κυρίως με τεχνικά θέματα όπως άνοιγμα αρχείου, ανάγνωση με συγκεκριμένη σειρά και συγκεκριμένο τρόπο καθώς και για πιθανά προβλήματα στη διαδικασία φόρτωσης αρχείου. Επίσης είναι υπεύθυνο για τον έλεγχο εγκυρότητας κάθε αρχείου XML ενάντια στο αρχείο σχήματος XSD. Στην Εικόνα 4.15 παρουσιάζονται οι προαναφερόμενες κλάσεις.



Εικόνα 4.15 Διάγραμμα UML – Διαχείριση αρχείων

XmlHandler

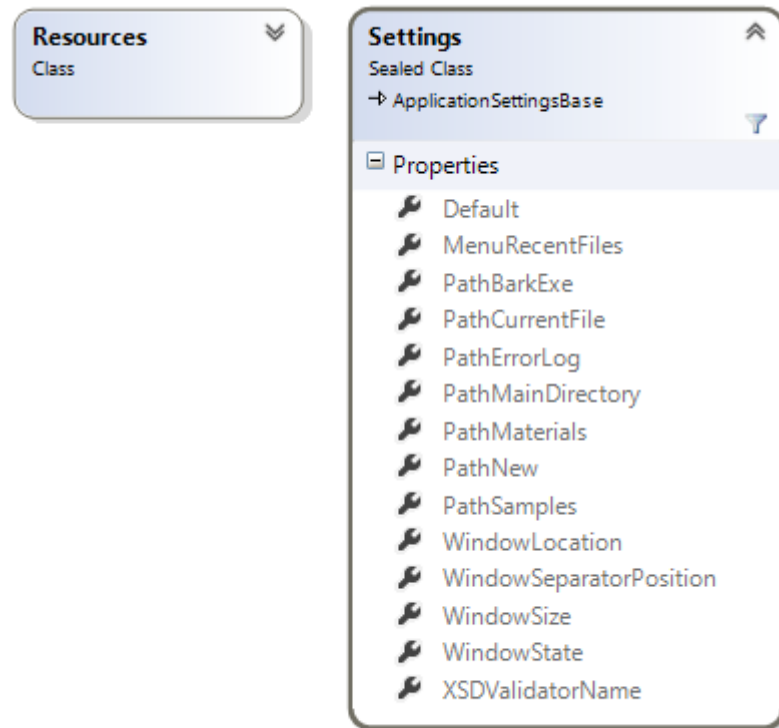
Διαχειρίζεται τις τεχνικές λεπτομέρειες που αφορούν αρχεία, καθώς και τον έλεγχο εγκυρότητας ενός αρχείου XML προτού φορτωθεί αλλά και προτού αποθηκευτεί. Επίσης μια δευτερεύον διαδικασία για την οποία είναι υπεύθυνο είναι ο έλεγχος αν υπάρχουν αλλαγές στο παρόν αρχείο από την τελευταία φορά που αποθηκεύτηκε.

XsdHandler

Ανήκει και χρησιμοποιείται μόνο στον XmlHandler διότι χωρίς το XML δεν υπάρχει λόγος να χειριστείς το XSD. Φορτώνει τα αρχεία σχήματος από την τοποθεσία που βρίσκεται στο κάθε αρχείο XML με συγκεκριμένη σειρά για να μην υπάρξουν προβλήματα.

4.1.8 Βοηθητικές κλάσεις

Στην Εικόνα 4.16 εμφανίζονται οι βοηθητικές κλάσεις.



Εικόνα 4.16 Διάγραμμα UML – Βοηθητικές κλάσεις

Resources

Αυτόματα παραγόμενη κλάση η οποία περιέχει τα αρχεία τα οποία χρησιμοποιούνται από το γραφικό περιβάλλον. Παράδειγμα είναι το εικονίδιο του Bark_GUI το οποίο βρίσκεται πάνω αριστερά γωνία στο κεντρικό παράθυρο.

Settings

Αυτόματα παραγόμενη κλάση η οποία περιέχει κάποιες προτιμήσεις/επιλογές οι οποίες είναι πιθανό να αλλάζουν εύκολα. Δεν χρειάστηκε να τοποθετηθούν όλες στο παράθυρο επιλογών. Επίσης περιέχει την τοποθεσία του παρόντος φορτωμένου αρχείου στον υπολογιστή για χρήση από οποιαδήποτε κλάση του προγράμματος.

Επιπλέον περιέχει κάποιες παραμέτρους του κεντρικού παραθύρου τις οποίες αποθηκεύει για την επόμενη φορά που θα χρησιμοποιήσει κάποιος την εφαρμογή.

4.2 Κύριο πρόβλημα που αντιμετωπίστηκε

Το κυριότερο πρόβλημα που αντιμετωπίστηκε ήταν το επίπεδο δυναμικότητας από άποψη κώδικα και ευελιξίας από άποψη λογισμικού που ήταν απαραίτητο για να παραμείνει το πρόγραμμα διαχρονικό. Κατά την ανάλυση των απαιτήσεων και τη σχεδίαση του πλάνου δημιουργίας του λογισμικού έγινε έρευνα για πιθανές λύσεις που αφορούσαν την αρχιτεκτονική των λειτουργιών πίσω από το γραφικό περιβάλλον για να εξασφαλισθεί η απαιτούμενη δυναμικότητα.

Υπήρξε η επιλογή αυτοματοποιημένης δημιουργίας κλάσεων – αντικειμένων από τα αρχεία XML. Αυτό σήμαινε πως θα δημιουργόντουσαν περιπτώσεις στο πρόγραμμα για καθένα από τα αρχεία XML, το οποίο είναι πολύ περιοριστικό. Επίσης, εφόσον γινόταν αυτό, οποιαδήποτε αλλαγή σε κάποιο αρχείο XML απαιτούσε την δημιουργία καινούργια προγράμματος με χρήση του ίδιου εργαλείου. Οπότε η συγκεκριμένη λύση κατέληξε να είναι ανεπαρκής και καθόλου εφαρμόσιμη στο πρόβλημα που προαναφέρθηκε.

Ύστερα ανακαλύφθηκε ένα άλλο εργαλείο το οποίο μετέτρεπε αυτοματοποιημένα τα αρχεία δομής XSD, τα σχήματα, σε κλάσεις. Αυτήν η λύση ήταν εφαρμόσιμη πάνω στο πρόβλημα, αλλά δυστυχώς συμπεριλάμβανε και αυτή επιπλοκές. Η δημιουργία τέτοιου προγράμματος θα είχε ως επόμενο τον περιορισμό του μόνιμο και απαράλλαχτου αρχείου XSD, πράγμα που καθιστά αδύνατη οποιαδήποτε εξέλιξη του μαθήματος. Σε περίπτωση αλλαγής του XSD θα έπρεπε να δημιουργηθεί νέο πρόγραμμα το οποίο θα περιέχει τις αντίστοιχες τροποποιήσεις σε επίπεδο κλάσεων – αντικειμένων. Το κύριο πρόβλημα που προκαλούσε τις προαναφερόμενες επιπλοκές ήταν ότι το εργαλείο δεν δημιουργούσε τις κλάσεις κατά τη διαδικασία εκτέλεσης του προγράμματος, αλλά πριν την μεταγλώτισή του σε εκτελέσιμο αρχείο. Επομένως και αυτή η λύση απορρίφθηκε.

Ήταν δυνατόν να χρησιμοποιηθεί η προηγούμενη λύση (αυτόματη παραγωγή κλάσεων από αρχείο σχήματος) σε συνδυασμό με μεταγλώτιση προγράμματος κατά τη διαδικασία εκτέλεσής του. Αποφεύχθηκε αυτή η λύση για δυο κύριους λόγους. Πρώτον θα ανέβαζε πολύ το επίπεδο πολυπλοκότητας του κώδικα του λογισμικού και δεύτερον σε σύνδεση με το πρώτο, θα καθιστούσε την αποσφαλμάτωση κώδικα πολύ δύσκολη, ίσως και ακατόρθωτη σε κάποιες περιπτώσεις.

Η τελική λύση που δόθηκε είναι η δημιουργία της δομής του XSD και των περιορισμών του σε κλάσεις στο πρόγραμμα. Με αυτόν τον τρόπο και να αλλάξει το

αρχείο XML δεν επηρεάζεται καθόλου η λειτουργία του γραφικού περιβάλλοντος. Εάν τροποποιηθεί το XSD με κατάλληλο τρόπο, μπορεί να αλλάξει μέχρι και η δομή των αρχείων XML με το γραφικό περιβάλλον άθικτο και λειτουργικό. Αυτή η λύση είναι η πιο ευέλικτη, διότι για να καταστεί μη-λειτουργικό πρέπει να αλλάξει η ίδια η γλώσσα XSD, το οποίο πάλι δεν είναι απόλυτο. Εάν υπάρχει κάποια λειτουργία στο XSD η οποία δεν εμπεριέχεται στο γραφικό περιβάλλον διότι θεωρήθηκε ανούσιο ή παραλήφθηκε, μπορεί απλά κάποια στιγμή να προστεθεί στην παρούσα πτυχιακή. Βέβαια για να επιτευχθεί αυτή η λύση, υπήρξε πολύ περισσότερο βάρος στην εργασία του προγραμματιστή.

Η πολυπλοκότητα της αρχιτεκτονικής του σχεδίου πίσω από το γραφικό περιβάλλον προκάλεσε τρεις αποτυχημένες προσπάθειες δημιουργίας λογισμικού το οποίο είτε δεν ήταν δυνατό να πληρεί τις προϋποθέσεις που τέθηκαν είτε η αδυναμία επιτυχίας της διαδικασίας αποσφαλμάτωσης είχε φτάσει το λογισμικό σε αδιέξοδο.

4.3 Βήματα επίλυσης προβλήματος

4.3.1 Δημιουργία σχήματος

Η πρώτη και πλήρως απαραίτητη ενέργεια που έγινε ήταν η δημιουργία του σχήματος των αρχείων XML (των αρχείων που υπήρχαν αλλά και αυτών που ίσως δημιουργόντουσαν κάποτε). Μετά από έρευνα πάνω στο θέμα και εκμάθηση της γλώσσας DTD έγινε η πρώτη πρακτική επαφή με τα σχήματα των XML. Σε συνεργασία με τον καθηγητή Ονούφριο Χαραλάμπους σε πρώτη φάση δημιουργήθηκε μια απλή δομή. Ύστερα με έρευνα και ανάλυση του προγράμματος, ανακαλύφθηκε ότι παρότι το DTD ήταν πιο απλό στη χρήση και ευανάγνωστο, το XSD ήταν πιο κατάλληλο για τα εργαλεία που χρησιμοποιούντουσαν αλλά και για τον στόχο που είχε τεθεί. Στην επόμενη φάση δημιουργήθηκε ένα πλήρες σχήμα το οποίο κάλυπτε τις ανάγκες εξέλιξης των πιθανών δεδομένων και πληροφοριών. Επίσης δημιουργήθηκε απλό, σύντομο και περιεκτικό βοηθητικό αρχείο με χρήσιμες εντολές του XSD.

Το σχήμα αποτελείται από πέντε σε αριθμό αρχεία, μόνο για λόγους απλοποίησης. Το κεντρικό αρχείο XSDValidator.xsd αποτελεί την κύρια περιγραφή δομής των αρχείων XML σε επίπεδο στοιχείων. Τα αρχεία XSDValidatorComplexTypes.xsd και XSDValidatorSimpleTypes.xsd περιέχουν τους βασικούς τύπους των στοιχείων μαζί με τους περιορισμούς – συσχετίσεις αυτών των στοιχείων. Τέλος τα αρχεία

XSDValidatorUnits.xsd και XSDValidatorFunctions.xsd περιέχουν τύπους συγκεκριμένους για το λογισμικό bark. Κατά τη διάρκεια της ανάπτυξης του γραφικού περιβάλλοντος γινόταν τροποποίηση στα αρχεία σχήματος ώστε να συμφωνούν περισσότερο με τη δομή του προγράμματος και να είναι πιο ευανάγνωστα. Στην Εικόνα 4.17 φαίνεται ένα δείγμα του αρχείου XSDValidator.xsd.

```
1 <?xml version="1.0"?>
2 <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
3   <xs:include schemaLocation="XSDValidatorUnits.xsd"/>
4   <xs:include schemaLocation="XSDValidatorComplexTypes.xsd"/>
5   <xs:include schemaLocation="XSDValidatorFunctions.xsd"/>
6   <xs:include schemaLocation="XSDValidatorSimpleTypes.xsd"/>
7   <xs:element name="case">
8     <xs:complexType>
9       <xs:sequence>
10        <xs:element name="simulation">
11          <xs:annotation>
12            <xs:documentation>This is where the help text can exist.</xs:documentation>
13          </xs:annotation>
14          <xs:complexType>
15            <xs:all>
16              <xs:element name="start_time" type="time">
17                <xs:annotation>
18                  <xs:documentation>This is also where the help text can exist.</xs:documentation>
19                </xs:annotation>
20              </xs:element>
21              <xs:element name="end_time" type="time"/>
22              <xs:element name="time_step" type="time_positive"/>
23              <xs:element name="space_step" type="space"/>
24              <xs:element name="initial_temperature" type="initial_temperature"/>
25            </xs:all>
26          </xs:complexType>
27        </xs:element>
28        ...
```

Εικόνα 4.17 – Δείγμα αρχείου σχήματος XSDValidator.xsd

4.3.2 Φόρτωση δεδομένων αρχείου XML

Το επόμενο βήμα είναι να φορτώνεται τα δεδομένα του αρχείου XML σε κατάλληλη μορφή στο γραφικό περιβάλλον. Γιαυτό το λόγο υπήρξε η ανάγκη δημιουργίας μιας δομής για να υφίσταται χειρισμός των δεδομένων χωρίς επιπλοκές. Μετά από δυο αποτυχημένες προσπάθειες και πολλές προσαρμοές της δομής κατέληξε η δομή να αποτελεί αναπαράσταση των στοιχείων της γλώσσας XSD με επιπλέον γνωρίσματα αποκλειστικά των δεδομένων (π.χ. μονάδα μέτρησης).

Η ροή εκτέλεσης ακολουθεί ως εξής:

Έλεγχος εγκυρότητας του αρχείου XML

Γίνεται έλεγχος εγκυρότητας του επιθυμητού αρχείου XML ενάντια στο αρχείο σχήματος XSD. Η τοποθεσία του σχήματος στον υπολογιστή συμπεριλαμβάνεται στο αρχείο XML. Εάν υπάρξουν προβλήματα κατά τη διαδικασία ελέγχου εγκυρότητας, ενημερώνεται ο χρήστης με ανάλογο μήνυμα και πιο λεπτομερές με τη φύση του προβλήματος για να λυθεί χειροκίνητα. Κάθε αρχείο που παράγεται από το γραφικό περιβάλλον είναι αυτόματα έγκυρο.

Φόρτωση του αρχείου XML

Με τη φόρτωση του αρχείου XML συμπεριλαμβάνονται τεχνικές λεπτομέρειες που αφορούν αρχεία, την ύπαρξή τους και τον τύπο τους. Μετά από αυτό το στάδιο είναι γνωστό ότι το αρχείο υπάρχει και δεν είναι κατεστραμμένο. Επίσης αποκτούνται τα μονοπάτια στον υπολογιστή των επιμέρους αρχείων σχήματος.

Φόρτωση και ανάλυση-μετάφραση του αρχείου σχήματος XSD

Φορτώνεται το αρχείο σχήματος και διαβάζεται από το πρόγραμμα με συγκεκριμένη σειρά για να αποκτηθούν τα δεδομένα που χειρίζονται από το γραφικό περιβάλλον. Σε αυτό το βήμα γίνεται η ανάλυση-μετάφραση των δεδομένων από γλώσσα XSD στην επιθυμητή μορφή. Παράλληλα με την ανάγνωση του αρχείου σχήματος εκτελείται και το επόμενο βήμα.

Δημιουργία της δομής

Δημιουργείται η δομή των αντικειμένων-στοιχείων (Items) στο γραφικό περιβάλλον. Σε αυτό το βήμα δημιουργούνται για πρώτη φορά τα στοιχεία ελέγχου του κάθε αντικειμένου αποκτούν τις προεπιλεγμένες τιμές τους καθώς και τις μεθόδους για έλεγχο εγκυρότητας τιμής.

με τα δεδομένα από το αρχείο σχήματος XSD. Μαζί με τα στοιχεία ελέγχου, τα αντικείμενα-στοιχεία και κάθε πιθανή-επιτρεπτή τους τιμή.

Φόρτωση των δεδομένων του αρχείου XML

Τελευταίο βήμα είναι η απόκτηση των δεδομένων του χρήστη από το αρχείο που επέλεξε και συμπλήρωση αυτών των δεδομένων στα κατάλληλα αντικείμενα-στοιχεία.

Κατά τη διάρκεια αυτής της διαδικασίας δημιουργούνται νέα αντικείμενα-στοιχεία για καθένα από τα φορτωμένα στοιχεία που προκύπτει, διότι διαφορετικά δεν θα ήταν δυνατόν να υπάρχουν πολλά αντικείμενα-στοιχεία τα οποία αντιπροσωπεύουν ένα στοιχείο. Με αυτό το τρόπο διαχωρίζεται η δομή των αντικειμένων του αρχείου σχήματος από τη δομή των αντικειμένων του αρχείου δεδομένων.

4.3.3 Επεξεργασία δεδομένων

Κατά την επεξεργασία δεδομένων από το χρήστη οι κύριες διαδικασίες που πραγματοποιούνται είναι οι εξής:

Έλεγχος εγκυρότητας τιμής/δεδομένου

Ο έλεγχος εγκυρότητας εκτελείται κάθε φορά που ο χρήστης αλλάζει τιμή σε οποιοδήποτε στοιχείου ελέγχου. Ο έλεγχος εγκυρότητας προέρχεται από την αντίστοιχη μέθοδο η οποία βρίσκεται σε κάθε αντικείμενο-στοιχείο και αλυσιδωτά σε κάθε τύπο στοιχείου τον οποίο μπορεί να έχει. Ωστόσο η λογική πηγάζει αρχικά από το αρχείο σχήματος, δηλαδή μπορεί οποιαδήποτε στιγμή να αλλάξει εάν τροποποιηθεί το XSD κατάλληλα.

Μια τιμή είναι έγκυρη εάν τηρεί τις προϋποθέσεις και τους περιορισμούς που τέθηκαν σε αυτόν τον τύπο (όπως προαναφέρθηκε). Ένα στοιχείο είναι έγκυρο όταν έχει μια έγκυρη τιμή ή αν είναι προαιρετικό στοιχείο. Ένα προαιρετικό στοιχείο ορίζεται στο XSD με την ιδιότητα *minOccurs* όπως φαίνεται στην παρακάτω γραμμή κώδικα.

```
<xs:element name="emissivity" type="emissivity" minOccurs="0" />
```

Ένα στοιχείο ομάδας για να είναι έγκυρο πρέπει όλα τα στοιχεία που περιέχει να είναι έγκυρα. Εξαίρεση αποτελεί εάν όλα τα παιδιά δεν έχουν τιμή ή έχουν την προεπιλεγμένη τιμή και ταυτόχρονα το στοιχείο-ομάδα είναι προαιρετικό.

Λογική εμφάνισης στοιχείων ελέγχου

Επειδή τα στοιχεία ελέγχου είναι πολλά, δεν φαίνονται όλα για να μην μπερδεύεται και χάνεται ο χρήστης. Τα στοιχεία που δεν εμφανίζονται είναι τα εξής:

- Τα προαιρετικά στοιχεία χωρίς τιμή ή με την προεπιλεγμένη τιμή τους.

- Τα στοιχεία τα οποία δεν έχουν σχέση με το επιλεγμένο στο δέντρο στοιχείο, δηλαδή δεν ανήκουν στα παιδιά του επιλεγμένου στοιχείου, ούτε είναι πρόγονοί του.

Η μέθοδος η οποία είναι υπεύθυνη για αυτή την διαδικασία βρίσκεται στην κλάση ViewerForm.

Ενέργειες δεξιού κλικ του ποντικιού στο δέντρο

Οι ενέργειες στο δέντρο είναι σημαντικές και ευαίσθητες διότι αποτελούν ενέργειες στην δομή των αντικειμένων-στοιχείων. Η δημιουργία νέου στοιχείου-ομάδας σημαίνει δημιουργία όλων των στοιχείων που το απαρτίζουν χωρίς να επηρεαστούν τα ήδη υπάρχοντα στοιχεία τέτοιου είδους. Η διαγραφή του, σημαίνει διαγραφή από τη δομή που δημιουργήθηκε κατά τη διαδικασία φόρτωσης του XML και όχι του XSD. Η μετονομασία ενός στοιχείου είναι επίσης σημαντική, διότι στη δομή υπάρχουν πολλά στοιχεία με το ίδιο όνομα (π.χ. υλικό) και ο μόνος τρόπος να διαχωριστούν είναι το νέο όνομα (NewName) το οποίο στο XML εκφράζεται ως ιδιότητα με το τίτλο «name».

Αλλαγή του τύπου ενός στοιχείου

Όταν ο χρήστης εναλλάσσει ένα στοιχείο ελέγχου για παράδειγμα από σταθερής τιμής σε μεταβλητής, το πρώτο στοιχείο ελέγχου αφαιρείται από τον ElementViewer και στη θέση του τοποθετείται το νέο στοιχείο ελέγχου. Φαινομενικά δεν αλλάζει τίποτα, παρά μόνο τα πεδία εισαγωγής τιμής.

4.3.4 Αποθήκευση δεδομένων σε αρχείο

Κατά τη διαδικασία αποθήκευσης πρώτα ελέγχεται εάν όλα τα στοιχεία ελέγχου είναι έγκυρα. Εάν δεν είναι, δεν εκτελείται η διαδικασία αποθήκευσης, ενώ αν είναι έγκυρα τότε συνεχίζεται η διαδικασία κανονικά. Εκτός από τη χειροκίνητη αποθήκευση του αρχείου μπορεί να εκτελεστεί και αυτόματα κατά το κλείσιμο του παρόντος αρχείου, μετά από ανάλογο διάλογο με τον χρήστη. Για να πετύχει αυτό, πρώτα ελέγχεται εάν υπάρχουν αλλαγές από την τελευταία φορά που αποθηκεύτηκε το αρχείο με αυτό που θα παραχθεί τώρα. Εάν δεν υπάρχουν αλλαγές, τότε η διαδικασία αποθήκευσης προσπερνάται.

4.3.5 Προσομοίωση

Εκτέλεση προσομοίωσης

Εκτελείται το λογισμικό bark.exe με παράμετρο το παρόν φορτωμένο αρχείο για να μπορέσει να μεταβεί σε υπολογισμούς και να βγάλει το αρχείο εξόδου με τα δεδομένα αποτελέσματος.

Φόρτωση δεδομένων εξόδου

Τα δεδομένα εξόδου έχουν την παρακάτω μορφή όπως φαίνονται στην Εικόνα 4.18:

1	time[s]	Ciy.time[h]	Sol.time[h]	qconv1[W/m2]	qrad1[W/m2]	qsol1[W/m2]	qconv2[W/m2]
2	0.00	0.00	0.00	0.00	0.00	0.00	0.00
3	14400.00	4.00	0.00	-2.58	-0.59	0.00	-30.17
4	28800.00	8.00	0.00	-7.06	-1.61	0.00	-30.36
5	43200.00	12.00	0.00	-10.97	-2.51	0.00	-31.48
6	57600.00	16.00	0.00	-14.32	-3.28	0.00	-32.52
7	72000.00	20.00	0.00	-17.20	-3.95	0.00	-33.42
8	86400.00	0.00	0.00	-19.68	-4.52	0.00	-34.20
9	100800.00	4.00	0.00	-21.81	-5.02	0.00	-34.87
10	115200.00	8.00	0.00	-23.65	-5.45	0.00	-35.44
11	129600.00	12.00	0.00	-25.23	-5.82	0.00	-35.94
12	144000.00	16.00	0.00	-26.60	-6.14	0.00	-36.37
13	158400.00	20.00	0.00	-27.77	-6.42	0.00	-36.74
14	172800.00	0.00	0.00	-28.78	-6.65	0.00	-37.05
15	187200.00	4.00	0.00	-29.65	-6.86	0.00	-37.32
16	201600.00	8.00	0.00	-30.40	-7.03	0.00	-37.56
17	216000.00	12.00	0.00	-31.04	-7.19	0.00	-37.76
18	230400.00	16.00	0.00	-31.59	-7.32	0.00	-37.93
19	244800.00	20.00	0.00	-32.07	-7.43	0.00	-38.08
20	259200.00	0.00	0.00	-32.48	-7.53	0.00	-38.21

Εικόνα 4.18 – Δείγμα αρχείου εξόδου τύπου .dat

Πρώτη σειρά αποτελείται από τους τίτλους της κάθε στήλης. Τα δεδομένα αναπαρίστανται ανά στήλες. Σκοπός είναι να ληφθούν αυτά τα δεδομένα και να δοθεί η επιλογή στο χρήστη με βάση τους τίτλους τον επιθυμητό τρόπο αναπαράστασης δεδομένων.

Για να πετύχει αυτό στην αρχή διαβάζεται το αρχείο με χρήση της στατικής κλάσης DataParser ώστε να παρθούν οι τίτλοι των δεδομένων και να εμφανιστούν στον χρήστη. Αφού επιλέξει ο χρήστης, αποστέλλονται τα δεδομένα μαζί με τις επιλογές τους χρήστη για τη δημιουργία του διαγράμματος.

Δημιουργία γραφήματος

Σε αυτό το βήμα δημιουργείται ένας πίνακας με τα δεδομένα σε δεκαδικούς αριθμούς και επιλέγονται οι στήλες που διάλεξε ο χρήστης για να δημιουργηθούν οι αντίστοιχοι άξονες. Χρησιμοποιώντας τη βιβλιοθήκη ZedGraph αρχικοποιείται η γραφική αναπαράσταση των δεδομένων ως σημεία στο γράφημα. Στην αρχή επιλέγονται διακριτά προεπιλεγμένα χρώματα για την αναπαράσταση των γραμμών στο γράφημα και για επιπλέον χρώματα επιλέγονται από μια λίστα χρωμάτων τυχαία εξαιρώντας τα ήδη επιλεγμένα.

5 Οδηγός Χρήσης

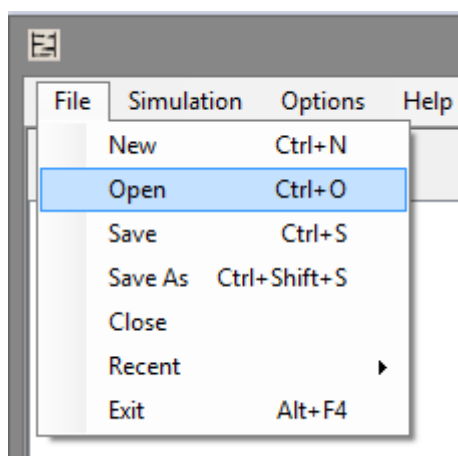
Σε αυτό το κεφάλαιο περιγράφεται μια συνηθισμένη διαδικασία χρήσης με παραστατικές εικόνες για ευκολία ανάγνωσης. Ο οδηγός απευθύνεται στον χρήστη και όχι σε προγραμματιστή, οπότε αποφεύχθηκε η ανάλυση τεχνικών λεπτομερειών. Καλύπτονται οι ενέργειες από την επεξεργασία δεδομένων μέχρι και την γραφική απεικόνιση των αποτελεσμάτων.

5.1 Επεξεργασία δεδομένων εισόδου

Η επεξεργασία των δεδομένων του αρχείου XML γίνεται με τέτοιο τρόπο ώστε να μην χρειαστεί κανείς να επεξεργαστεί το ίδιο το αρχείο, οπότε δεν χρειάζεται και να γνωρίζει τι είναι το XML. Τα αρχεία δεδομένων τα οποία είναι αρχεία εισόδου για το λογισμικό bark και για το αντίστοιχο γραφικό περιβάλλον έχουν κατάληξη .brk.

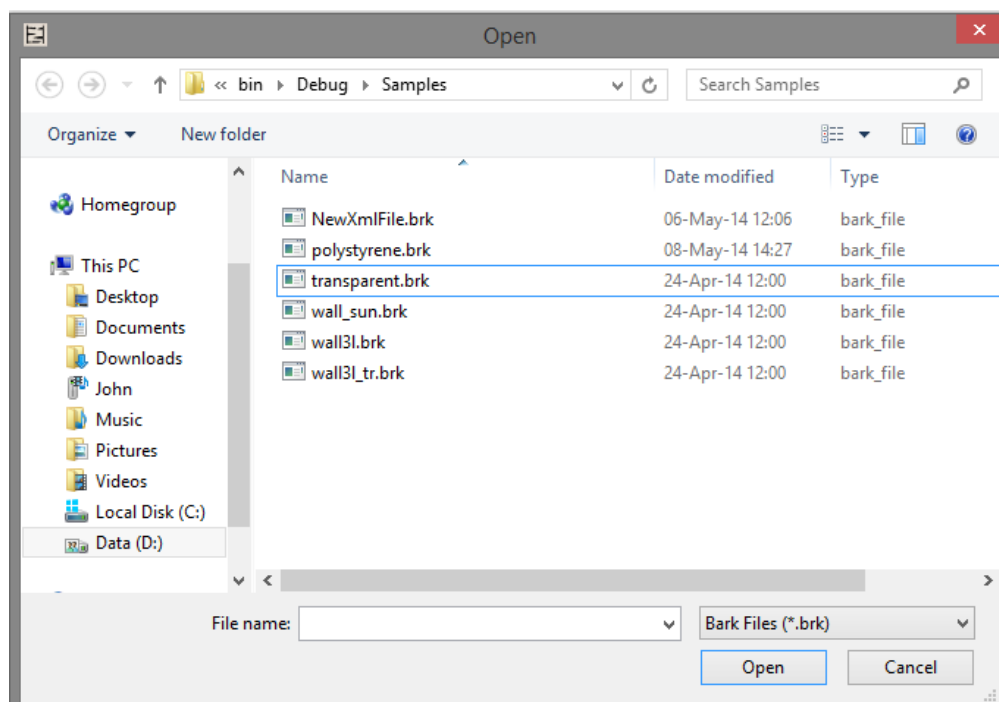
5.1.1 Άνοιγμα αρχείου δεδομένων XML (.brk)

Για να γίνει η επεξεργασία των πληροφοριών που εμπεριέχονται σε ένα αρχείο τύπου XML, θα πρέπει πρώτα να φορτωθεί αυτό το αρχείο στην εφαρμογή (γραφικό περιβάλλον). Αυτό γίνεται μέσω του μενού στο πάνω μέρος του γραφικού περιβάλλοντος επιλέγοντας την ενέργεια Open στην κατηγορία File όπως φαίνεται στην Εικόνα 5.1.



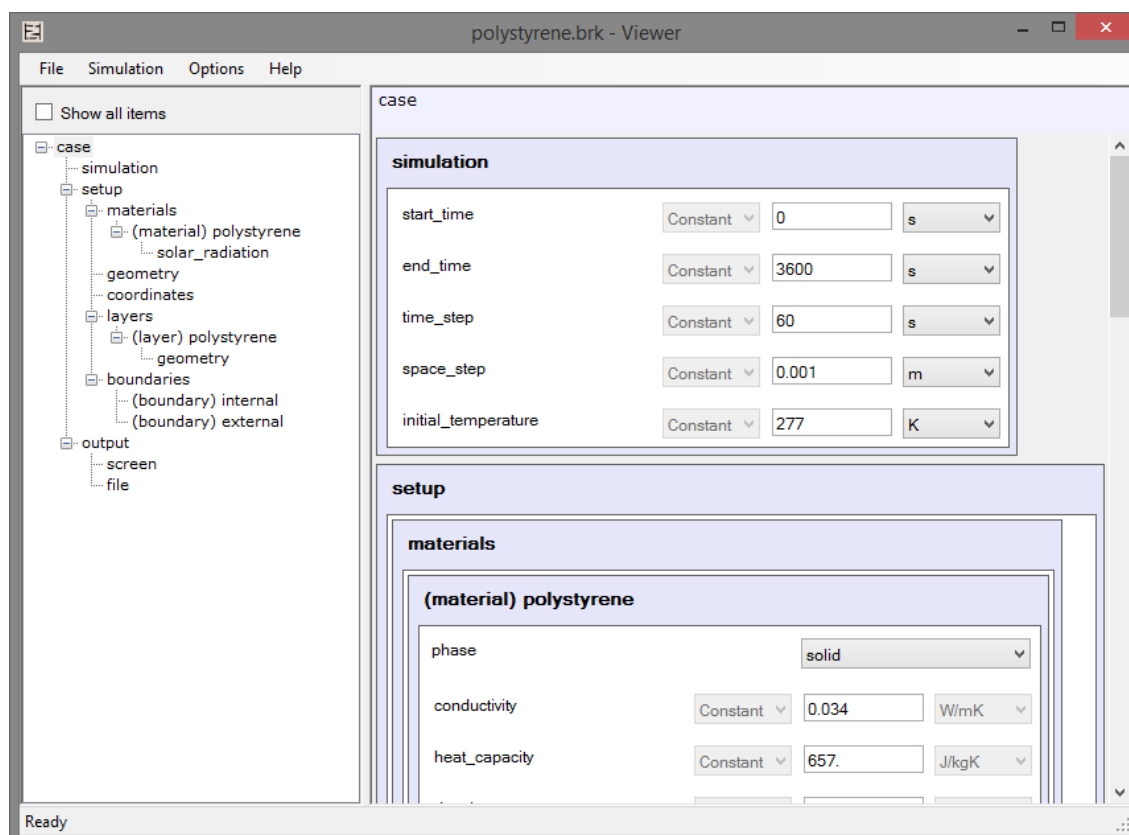
Εικόνα 5.1 – Άνοιγμα αρχείου δεδομένων XML από το μενού

Εμφανίζεται ένας διάλογος όπως παρουσιάζει η Εικόνα 5.2 για την επιλογή του αρχείου μέσα στον υπολογιστή από τον χρήστη και εφόσον επιλεγθεί το αρχείο φορτώνεται στο πρόγραμμα για εμφάνιση και επεξεργασία.



Εικόνα 5.2 – Διάλογος για το άνοιγμα αρχείου δεδομένων XML

Μετά την επιλογή κατάλληλου αρχείου (.brk/.xml) φορτώνονται τα δεδομένα του στο γραφικό περιβάλλον όπως φαίνεται στην Εικόνα 5.3.



Εικόνα 5.3 – Παράδειγμα φόρτωσης αρχείου δεδομένων

5.1.2 Δημιουργία νέου αρχείου δεδομένων XML (.brk)

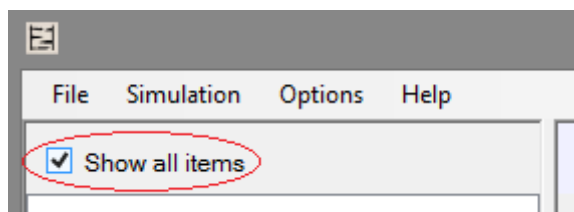
Εναλλακτικά ο χρήστης αντί να φορτώσει ένα ήδη υπάρχον αρχείο από το σύστημα, μπορεί να δημιουργήσει νέο. Για την δημιουργία νέου αρχείου ο χρήστης επιλέγει από το μενού *File* την επιλογή *New* και φορτώνονται τα άδεια στοιχεία ελέγχου για εισαγωγή δεδομένων. Όταν θα πρέπει να αποθηκεύσει το νέο αρχείο, θα εμφανιστεί διάλογος για την τοποθεσία του αρχείου αντίθετα με την διαδικασία ανοίγματος αρχείου.

5.1.3 Ενέργειες στο δέντρο

Καθώς οι ενέργειες που μπορεί να κάνει ένας χρήστης στα στοιχεία ελέγχου του δεξιού μέρους του κεντρικού παραθύρου είναι ξεκάθαρες (να επεξεργαστεί δεδομένα), το δέντρο είναι λίγο πιο δύσκολο διότι χειρίζεται την δομή των δεδομένων (σειρά, όνομα στοιχείο, πλήθος στοιχείων). Γι'αυτό το λόγο σε αυτό το κομμάτι θα γίνει μια περιγραφή των δυνατοτήτων του δέντρου στο αριστερό μέρος του κεντρικού παραθύρου.

Η επιλογή **Show all items**

Όταν το αρχείο δεδομένων είναι μεγάλο αλλά τα στοιχεία τα οποία έχουν σημασία είναι λίγα, η επεξεργασία δεδομένων μπορεί να καταντήσει κουραστική με το ψάξιμο και την προσπάθεια εύρεσης των στοιχείων που είναι ενδιαφέροντα. Γιαυτό το λόγο υπάρχει η επιλογή πάνω από το δέντρο με την επιγραφή “Show all items” όπως φαίνεται στην Εικόνα 5.4, η οποία από προεπιλογή δεν είναι ενεργή.

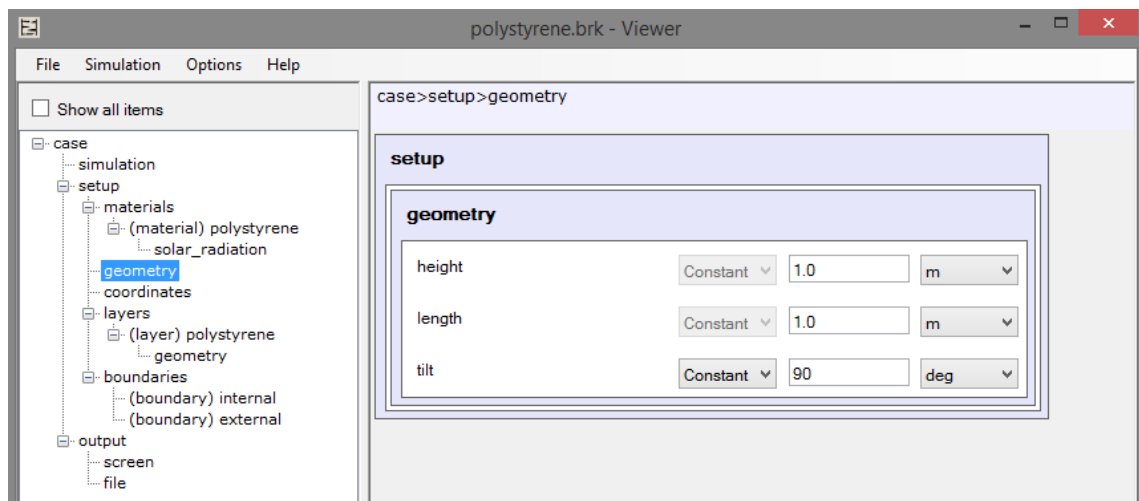


Εικόνα 5.4 – Επιλογή εμφάνισης όλων ή μερικών στοιχείων ελέγχου

Τα στοιχεία τα οποία δεν εμφανίζονται είναι τα προαιρετικά και χωρίς τιμή διαφορετική από την προεπιλεγμένη. Παρομοίως, τα στοιχεία ομαδοποίησης (Group Items) δεν φαίνονται εάν, με βάση τα παραπάνω κριτήρια, όλα τα στοιχεία που περιέχει δεν εμφανίζονται. Επίσης τα στοιχεία ομαδοποίησης τα οποία δεν περιέχουν στοιχεία δεν φαίνονται.

Πλοήγηση στοιχείων μέσω του δέντρου

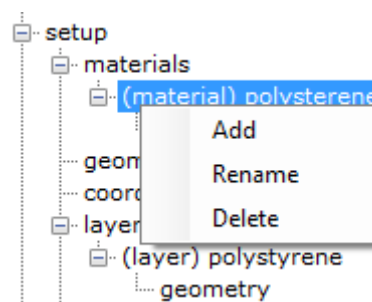
Όταν ο χρήστης θέλει να δει και να επεξεργαστεί μόνο συγκεκριμένα στοιχεία τα οποία βρίσκονται σε μια ομάδα, δεν χρειάζεται να τα ψάχνει στο πλήθος των στοιχείων ελέγχου, αρκεί μόνο να επιλέξει το στοιχείο προς ενδιαφέρον στο δέντρο και απ' ευθείας θα εμφανιστούν μόνο τα στοιχεία τα οποία εμπεριέχονται στην επιλεγμένη ομάδα. Στην παρακάτω Εικόνα 5.5 η ομάδα με όνομα “geometry” έχει επιλεγθεί, επομένως στα δεξιά βλέπουμε μόνο τις τιμές που αφορούν την επιλεγμένη ομάδα.



Εικόνα 5.5 – Παράδειγμα επιλογής ομάδας από το δέντρο

Επιλογές δεξιού κλικ του ποντικιού

Κάποιες προσομοιώσεις απαιτούν περισσότερα ή και λιγότερα υλικά ή στρώματα. Για την πρόσθεση ή αφαίρεση στοιχείων στο δέντρο χρησιμοποιείται το δεξί κλικ του ποντικιού πάνω στο επιθυμητό στοιχείο (π.χ. υλικό). Στην Εικόνα 5.6 φαίνονται οι ενέργειες που μπορούν να πραγματοποιηθούν σε ένα τέτοιο στοιχείο.



Εικόνα 5.6 – Ενέργειες δεξιού κλικ του ποντικιού σε στοιχείο δέντρου

Add

Δημιουργεί νέο στοιχείο-αντικείμενο το οποίο περιέχει τα ίδια στοιχεία με τα υπόλοιπα, αλλά μπορεί να πάρει διαφορετικές τιμές.

Αυτή η ενέργεια μπορεί να γίνει και από το στοιχείο το οποίο τα περιέχει. Για παράδειγμα η δημιουργία νέου υλικού-material μπορεί να γίνει με δεξί κλικ στην ομάδα υλικών-materials)

Rename

Μετονομάζει το επιθυμητό στοιχείο σε ο,τι όνομα δώσει ο χρήστης. Προσοχή όμως, δεν επιτρέπεται να υπάρχουν δυο στοιχεία ίδιου τύπου με το ίδιο όνομα (π.χ. δυο υλικά με όνομα 'στερεό'). Εάν υπάρχει αναφορά σε αυτό το στοιχείο, η αναφορά ανανεώνεται με το καινούργιο όνομα αυτόματα.

Delete

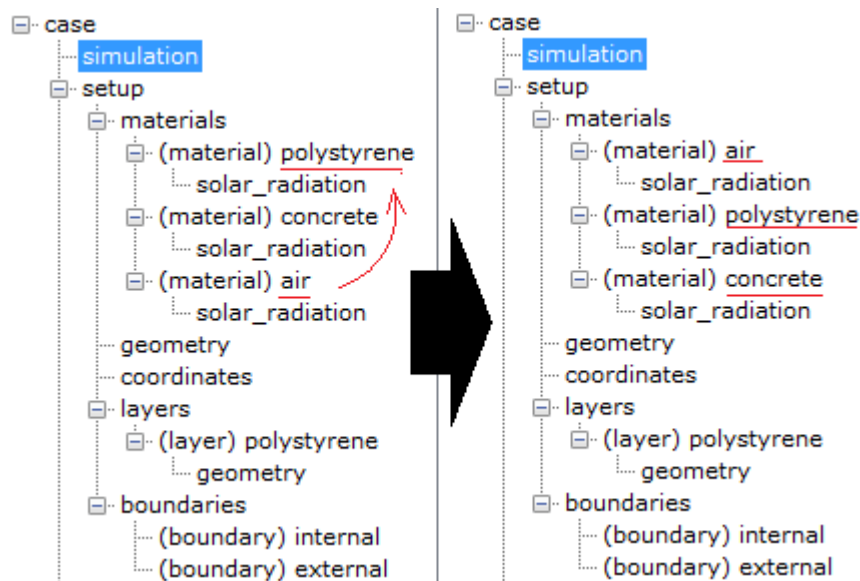
Διαγράφει το επιθυμητό στοιχείο από την ομάδα μαζί με τα δεδομένα του. Δεν μπορεί η ομάδα να μην έχει κανένα στοιχείο, οπότε δεν μπορεί να διαγραφεί το τελευταίο στοιχείο της ομάδας, θα πρέπει πάντα να υπάρχει τουλάχιστον ένα.

Αξίζει να σημειωθεί πως δεν είναι επιτρεπτές οι παραπάνω ενέργειες σε οποιοδήποτε στοιχείο του δέντρου, παρά μόνο στα στοιχεία που μπορούν να υπάρχουν από μια έως άπειρες φορές (ορίζεται στο XSD).

Μετακίνηση θέσης στοιχείου

Στα στοιχεία τα οποία έχουν ενέργειες με δεξί κλικ του ποντικιού, μπορεί επίσης να υπάρχει μετακίνηση θέσης ανάμεσα στα στοιχεία. Είναι σημαντικό ο χρήστης να βλέπει τη σωστή σειρά των στρώσεων ενός τοίχου για να μην μπερδεύεται, αλλά και το αρχείο δεδομένων εισόδου να περιέχει τα στοιχεία με σωστή σειρά.

Για να γίνει η μετακίνηση ενός στοιχείου χρησιμοποιείται το πάτημα του αριστερού κλικ του ποντικιού (όπως γίνεται η επιλογή ενός στοιχείου) αλλά χωρίς να αφεθεί. Ο χρήστης με κρατημένο το αριστερό κλικ μετατοπίζει τον δείκτη του ποντικιού (Drag 'n Drop) στο επιθυμητό στοιχείο-στόχο το οποίο θέλει να πάρει τη θέση του το συρόμενο στοιχείο και αφήνει το κλικ. Υπάρχει ένδειξη για σωστό στοιχείο στόχο ή λάθος με αντίστοιχο εικονίδιο κατά την διαδικασία συρσίματος. Στην Εικόνα 5.7 φαίνεται η μετακίνηση του υλικού air στη θέση του υλικού polysterene και το αποτέλεσμα μετά τη μετακίνηση.



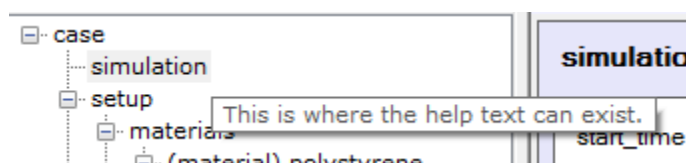
Εικόνα 5.7 – Παράδειγμα μετακίνησης θέσης ενός υλικού στο δέντρο

Δεν μπορεί ένα στοιχείο να μετακινηθεί έξω από την ομάδα στην οποία βρίσκεται και να μεταφερθεί σε άλλη ομάδα (π.χ. ένα υλικό να πάει στα στρώματα). Όταν το επιθυμητό στοιχείο τοποθετηθεί στη θέση που επιλέχθηκε, τα κατώτερα στοιχεία μετατοπίζονται μια θέση κάτω, ενώ τα ανώτερα παραμένουν ως έχουν.

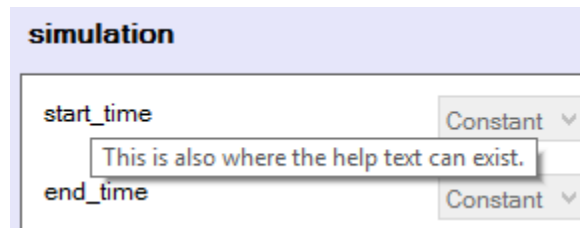
Βοηθητικό κείμενο

Εάν υπάρχει κάποια απορία για ένα συγκεκριμένο στοιχείο είτε στο δέντρο είτε στα στοιχεία ελέγχου στα δεξιά, μπορεί ο χρήστης να μετακινήσει το δείκτη του ποντικιού πάνω από το επιθυμητό στοιχείο και να δει ένα κείμενο το οποίο είναι βοηθητικό.

Στις Εικόνα 5.8 και Εικόνα 5.9 φαίνεται ένα δείγμα κειμένου το οποίο εμφανίζεται στα στοιχεία με όνομα simulation και start time, στο δέντρο και στο στοιχείο ελέγχου αντίστοιχα.



Εικόνα 5.8 – Δείγμα βοηθητικού κειμένου στο στοιχείο simulation του δέντρου



Εικόνα 5.9 – Δείγμα βοηθητικού κειμένου στο στοιχείο ελέγχου start_time

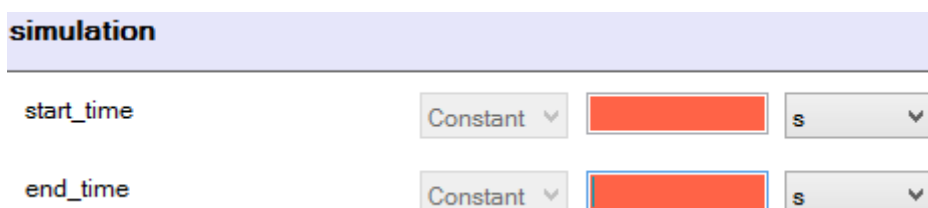
Το περιεχόμενο του βοηθητικού κειμένου ορίζεται στο αρχείο σχήματος XSD.

5.1.4 Έλεγχος εγκυρότητας εισαγωγής και επεξεργασίας δεδομένων

Κατά τη διαδικασία επεξεργασίας δεδομένων μπορεί ο χρήστης να συναντήσει πρόβλημα με τα δεδομένα που εισάγει ή επεξεργάζεται. Αυτό φαίνεται με κόκκινο χρώμα και σημαίνει ότι ο χρήστης έκανε λάθος σε μια από τις δυο παρακάτω περιπτώσεις:

Απαραίτητο Στοιχείο

Εάν το στοιχείο θεωρείται απαραίτητο για την προσομοίωση θα πρέπει να υπάρχει και να έχει συμπληρωθεί κάποια τιμή, διαφορετικά το πλαίσιο εισαγωγής αποκτά κόκκινο χρώμα ως ένδειξη λάθους. Τότε ο χρήστης θα πρέπει οπωσδήποτε να συμπληρώσει μια τιμή για να μπορέσει να αποθηκεύσει το αρχείο του, καθώς και για να εκτελέσει προσομοίωση. Για να οριστεί ένα στοιχείο απαραίτητο θα πρέπει να τροποποιηθεί ο ορισμός του στο αντίστοιχο αρχείο XSD. Στην παρακάτω Εικόνα 5.10 φαίνεται ένα παράδειγμα απαραίτητου στοιχείου χωρίς τιμή.



Εικόνα 5.10 – Παράδειγμα ελέγχου εγκυρότητας απαραίτητου στοιχείου

Έγκυρη Τιμή

Εάν το στοιχείο λάβει τιμή η οποία δεν πληρεί κάποιες προϋποθέσεις, τότε το χρώμα της τιμής θα γίνει κόκκινο ως ένδειξη λάθους του χρήστη. Για να μπορέσει ο χρήστης να αποθηκεύσει τα δεδομένα στο αρχείο ή να εκτελέσει προσομοίωση, θα πρέπει πρώτα να διορθώσει το λάθος και να εισάγει μια έγκυρη τιμή. Οι προϋποθέσεις που πρέπει να ισχύουν διαφέρουν από στοιχείο σε στοιχείο και ορίζονται στα αντίστοιχα αρχεία XSD. Στην παρακάτω Εικόνα 5.11 φαίνεται ένα παράδειγμα εισαγωγής άκυρης αλφαριθμητικής τιμής “0.03a” σε ένα στοιχείο το οποίο δέχεται μόνο δεκαδικούς αριθμούς.



Εικόνα 5.11 – Παράδειγμα ελέγχου εγκυρότητας τιμής

5.1.5 Αποθήκευση αρχείου δεδομένων

Για την αποθήκευση του αρχείου XML (εφόσον φυσικά έχει ήδη φορτωθεί) χρησιμοποιείται η επιλογή Save από την κατηγορία File του μενού στο κεντρικό παράθυρο (ακριβώς κάτω από την επιλογή ανοίγματος αρχείου). Υπάρχει και η επιλογή αποθήκευσης του παρόντος αρχείου ως νέο αρχείο για διατήρηση των προηγούμενων δεδομένων (Save As).

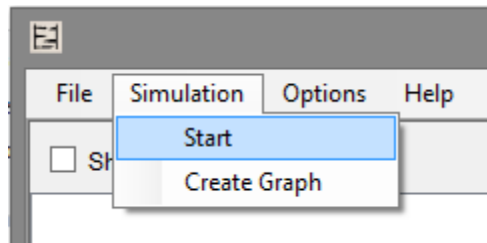
Η αποθήκευση των δεδομένων είναι απαραίτητο κομμάτι των ενεργειών του γραφικού περιβάλλοντος και εκτελείται με σχεδόν κάθε ενέργεια, είτε από τον χρήστη είτε αυτόματα.

5.1.6 Εκκίνηση προσομοίωσης

Το τελικό βήμα της επεξεργασίας δεδομένων είναι η εκτέλεση προσομοίωσης, καθώς και ο σκοπός της διαδικασίας επεξεργασίας δεδομένων. Από τη στιγμή που εκτελεστεί η προσομοίωση μπορεί ο χρήστης να μεταβεί στο δεύτερο μέρος του γραφικού

περιβάλλοντος και να δει τα αποτελέσματα της προσομοίωσης σε γράφημα. Χωρίς αυτό το βήμα δεν μπορεί να δημιουργηθεί γράφημα (με τα παρών δεδομένα).

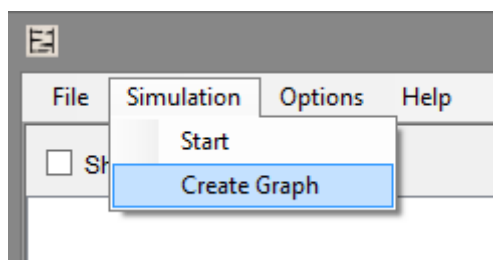
Είναι ένα εύκολο βήμα το οποίο αρκεί μόνο ο χρήστης να επιλέξει στο κεντρικό παράθυρο απ' το μενού την κατηγορία Simulation και την επιλογή Start (όπως φαίνεται παρακάτω στην Εικόνα 5.12) και τα υπόλοιπα γίνονται αυτόματα.



Εικόνα 5.12 – Εκκίνηση προσομοίωσης από το μενού του κεντρικού παραθύρου

5.2 Γραφική απεικόνιση αποτελεσμάτων

Για να μπορέσει ο χρήστης να δει τα αποτελέσματα σε γράφημα αρκεί μόνο να επιλέξει από την κατηγορία Simulation το Create Graph, όπως φαίνεται στην Εικόνα 5.13.

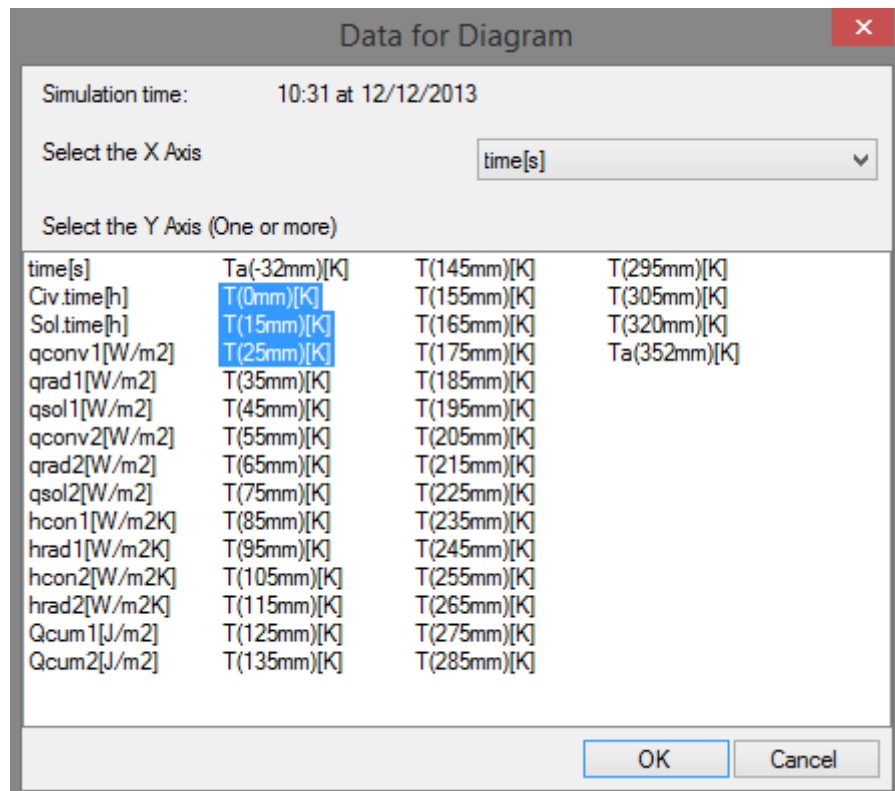


Εικόνα 5.13 – Δημιουργία νέου γραφήματος από το μενού

Αξίζει να σημειωθεί ότι προτού δημιουργηθεί το γράφημα πρέπει να έχουν παραχθεί τα αποτελέσματα από το λογισμικό bark, δηλαδή ο χρήστης να έχει ήδη εκτελέσει την προσομοίωση όπως αναφέρεται στο προηγούμενο βήμα.

5.2.1 Επιλογή αξόνων διαγράμματος

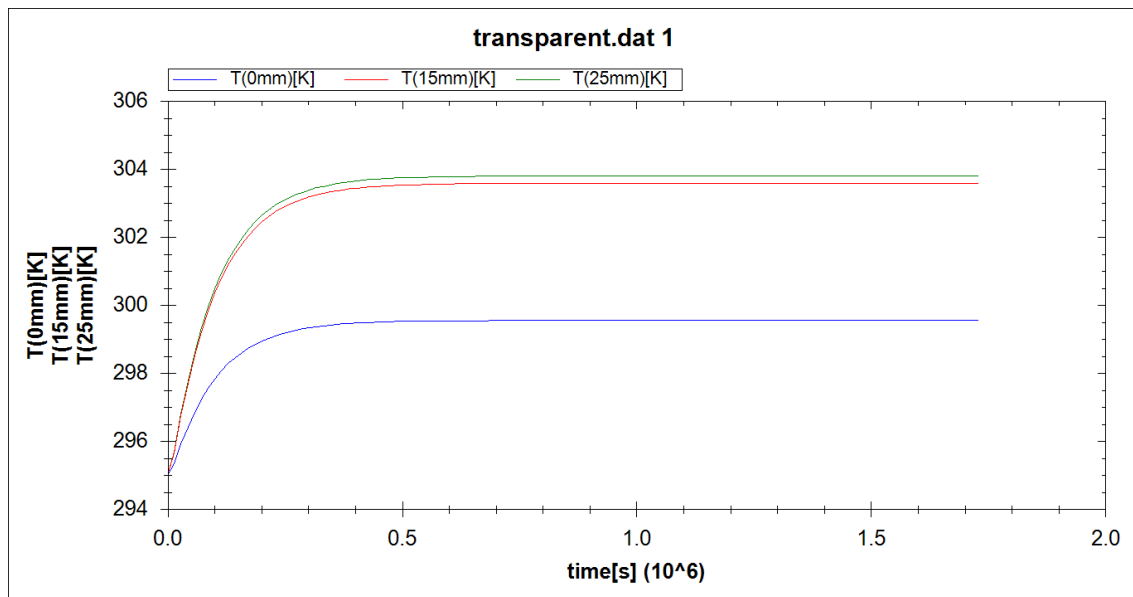
Εφόσον ο χρήστης έχει επιλέξει να δημιουργήσει νέο γράφημα από το μενού, εμφανίζεται ένας διάλογος για να επιλέξει ποια δεδομένα θέλει να τοποθετήσει σε ποιους άξονες. Στην Εικόνα 5.14 φαίνεται ένα παράδειγμα επιλογής αξόνων.



Εικόνα 5.14 – Διάλογος επιλογής αξόνων γραφήματος

5.2.2 Εμφάνιση δεδομένων εξόδου bark σε διάγραμμα

Εφόσον έχουν επιλεγθεί οι κατάλληλοι άξονες, μπορεί ο χρήστης να δει καθαρά τα δεδομένα εξόδου του bark και να βγάλει αντίστοιχα συμπεράσματα. Παρακάτω αναφέρονται κάποιες ενέργειες που μπορεί να κάνει ο χρήστης για τον έλεγχο απεικόνισης των αποτελεσμάτων ενός γραφήματος όπως στην Εικόνα 5.15.



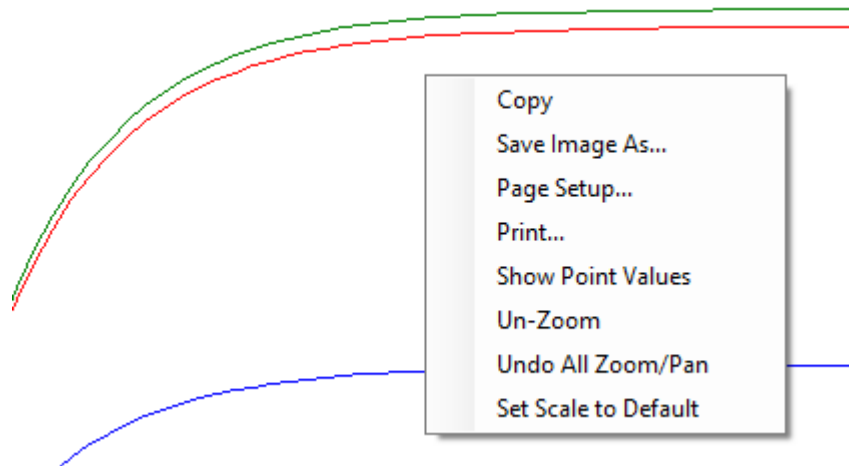
Εικόνα 5.15 – Παράδειγμα γραφήματος

Ενέργειες – Έλεγχος απεικόνισης

- Αριστερό κουμπί ποντικιού και κράτημα: Επιλογή περιοχής για μεγέθυνση.
- Πλήκτρο **Ctrl** + Αριστερό κλικ ποντικιού και κράτημα: Μεταβολή της θέσης στο διάγραμμα.
- Μεσαίο κλικ και κράτημα: Μεταβολή θέσης στο διάγραμμα.
- Δεξί κλικ: Εμφάνιση επιλογών

Επιλογές δεξιού κλικ του ποντικιού στο διάγραμμα:

Οι πιθανές επιλογές και οι ενέργειες που εκτελούν όπως απεικονίζονται στην Εικόνα 5.16.



Εικόνα 5.16 – Επιλογές δεξιού κλικ του ποντικιού στο διάγραμμα

- Copy: Αντιγραφή του διαγράμματος ως εικόνα για επικόλληση σε κατάλληλο λογισμικό (π.χ. Ζωγραφική των Windows).
- Save Image As: Αποθήκευση του διαγράμματος ως εικόνα στον επιλεγμένο από το χρήστη φάκελο.
- Page Setup: Επιλογές σελίδας για εκτύπωση.
- Print: Επιλογές εκτύπωσης.
- Show Point Values: Επιλογή εμφάνισης ή όχι των σημείων στο διάγραμμα όταν περνάει ο δείκτης του ποντικιού από πάνω.
- Un-Zoom: Αναίρεση της τελευταίας ενέργειας μεγέθυνσης ή σμίκρυνσης.
- Undo All Zoom/Pan: Αναίρεση όλων των ενεργειών μεγέθυνσης ή σμίκρυνσης και επιστροφή στην αρχική θέση στο διάγραμμα.
- Set Scale to Default: Αυτόματη μεγέθυνση και μεταβολή θέσης ώστε να φαίνονται όλες οι γραμμές στο διάγραμμα. *(Προτείνεται προς συχνή χρήση)*

6 Επίλογος

6.1 Συμπέρασμα

Το γραφικό περιβάλλον δεν είναι απαραίτητο όταν η χρήση του προγράμματος γίνεται από προγραμματιστές ή άτομα ειδικευμένα στην πληροφορική. Ωστόσο τις περισσότερες φορές απαιτείται ένα καλό γραφικό περιβάλλον για την εύκολη χρήση του προγράμματος, ειδικά όταν χρησιμοποιείται για εκπαιδευτικούς σκοπούς.

Εκτός από την ανάγκη για ύπαρξη γραφικού περιβάλλοντος, μπορεί να υπάρχει για ταχύτερη και βέλτιστη χρήση ενός λογισμικού.

Η παρούσα πτυχιακή έλυσε αυτό το πρόβλημα, αλλά όχι μόνο. Έδωσε την επιλογή για επέκταση και εξέλιξη του προγράμματος χωρίς το περιορίζει το γραφικό περιβάλλον.

6.2 Μελλοντικές επεκτάσεις

Για να προστεθούν ή επεξεργαστούν στοιχεία και δεδομένα στο XML τα οποία πρέπει να χειρίζεται το γραφικό περιβάλλον, θα πρέπει να τροποποιηθεί η κλάση XmlParser και πιο συγκεκριμένα η μέθοδος ConvertToXml(Item) με τον επιθυμητό τρόπο. Αυτήν η μέθοδος είναι υπεύθυνη για την δημιουργία και αποθήκευση της δομής και των δεδομένων στο αρχείο.

6.2.1 Ενσωμάτωση γραφημάτων στο αρχείο δεδομένων XML

Διαχείριση των γραφημάτων μέσω του γραφικού περιβάλλοντος και αποθήκευση στο αρχείο XML ως στοιχείο 'graph' με τη δομή που φαίνεται στην Εικόνα 6.1.

```

1  <!--InnerElement D1 <graph>-->
2  <xs:element name="graph" minOccurs="0">
3  <xs:complexType>
4  <xs:sequence>
5  .....
6  <!--AttributeElement <filename>-->
7  <xs:element name="filename">
8  <xs:complexType>
9  <xs:attribute name="name" type="xs:string"/>
10 <xs:attribute name="filename" type="xs:string"/>
11 </xs:complexType>
12 </xs:element>
13 <!--AttributeElement <xvariable>-->
14 <xs:element name="xvariable">
15 <xs:complexType>
16 <xs:attribute name="title" type="xs:string"/>
17 </xs:complexType>
18 </xs:element>
19 <!--AttributeElement <yvariable>-->
20 <xs:element name="yvariable">
21 <xs:complexType>
22 <xs:attribute name="title" type="xs:string"/>
23 </xs:complexType>
24 </xs:element>
25 .....
26 </xs:sequence>
27 </xs:complexType>
28 </xs:element>
29 <!--END InnerElement D1 <graph>-->

```

Εικόνα 6.1 – Στοιχείο XML «Graph»

6.2.2 Υποστήριξη συναρτήσεων (functions)

Το αρχείο σχήματος XSDValidatorFunctions.xsd περιέχει συναρτήσεις οι οποίες μπορούν να ενσωματωθούν στο γραφικό περιβάλλον για ευκολία χρήσης και μείωση του όγκου δεδομένων προς επεξεργασία. Ένα παράδειγμα συνάρτησης είναι η ημιτονοειδής συνάρτηση (sinusoidal). Κάθε συνάρτηση στη δομή του γραφικού περιβάλλοντος θα πρέπει να εκφράζεται από ένα GroupItem ή μια παρόμοια υλοποίηση. Θα περιέχεται σε όποια ElementItems ορίζει το αντίστοιχο αρχείο XSD και θα πρέπει να υλοποιηθεί η μετατροπή του στοιχείου ελέγχου από οποιοδήποτε άλλο σε ControlFunction το οποίο θα πρέπει να περιέχει τις επιλογές πιθανών συναρτήσεων (έχει ήδη υλοποιηθεί) μαζί με τα στοιχεία που περιέχονται στην συνάρτηση για εισαγωγή δεδομένων. Επίσης θα πρέπει να γίνει αντίστοιχη υλοποίηση μετατροπής του Item σε μορφή XML το οποίο όπως προαναφέρθηκε θα γίνει στην στατική κλάση XmlParser και συγκεκριμένα στην μέθοδο ConvertToXml. Προσοχή στις ήδη υλοποιημένες λειτουργίες οι οποίες εφαρμόζονται σε όλα τα ElementItem και

CustomControls, καθώς και στα GroupItems, να εφαρμόζονται σωστά και στις συναρτήσεις. Παράδειγμα των προαναφερόμενων λειτουργιών είναι οι μεταβλητές IsRequired, Value καθώς και οι μέθοδοι ShowOrHide, ValueValidator delegate.

6.2.3 Έλεγχος εγκυρότητας αρχείου δεδομένων XML

Εφόσον υπήρξε ανάγκη για την δημιουργία σχήματος των XML συνεπάγεται ότι είναι εύκολο να δημιουργηθούν αρχεία XML τα οποία δεν ακολουθούν τους αυστηρούς κανόνες του σχήματος. Επομένως μπορεί να προστεθεί η αυτόματη διόρθωση μη-έγκυρων αρχείων XML με βάση τα αρχεία XSD, όσο αυτό είναι εφικτό.

6.2.4 Προϋπάρχουσα συλλογή υλικών

Μπορεί να προστεθεί συλλογή υλικών (material) για να εισάγονται και να χρησιμοποιούνται πιο εύκολα από το να εμπεριέχονται σε κάθε αρχείο XML. Τα υλικά θα βρίσκονται σε έναν φάκελο από τον οποίο θα φορτώνονται μέσα στο πρόγραμμα όταν φορτώνεται ένα αρχείο XML και θα δημιουργούνται επιτόπου.

6.2.5 Βελτίωση γραφικής απεικόνισης αποτελεσμάτων

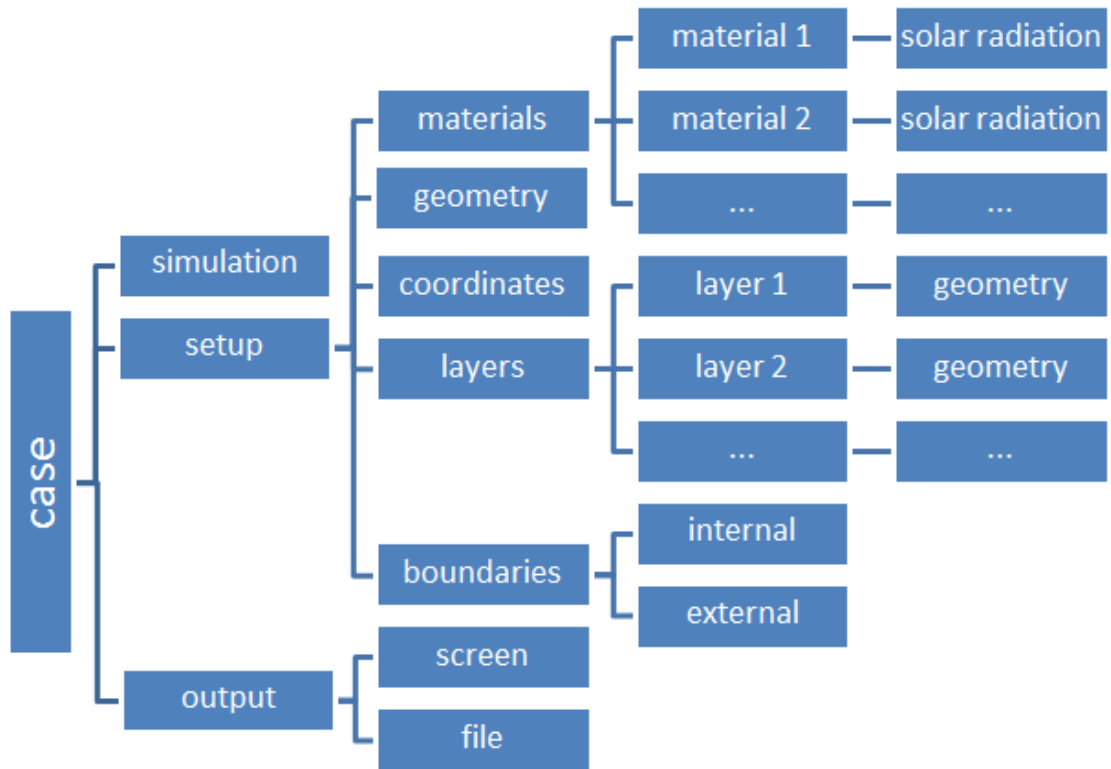
Μπορούν να βελτιωθούν οι λειτουργίες κατά την απεικόνιση των αποτελεσμάτων ανάλογα με τις ανάγκες του μαθήματος. Για να γίνει όμως αυτό πρέπει πρώτα να χρησιμοποιηθεί το γραφικό περιβάλλον στην πράξη και να γίνει έρευνα πάνω στην συμπεριφορά των φοιτητών και τις δυσκολίες που αντιμετωπίζουν.

Αναβάθμιση του δεύτερου μέρους του γραφικού περιβάλλοντος, δηλαδή της απεικόνισης των αποτελεσμάτων με χρήση γραφικών παραστάσεων. Να δημιουργηθεί βιβλιοθήκη ή να χρησιμοποιηθεί ήδη υπάρχουσα με περισσότερες λειτουργίες για τους χρήστες ανάλογα με τις ανάγκες που θα εμφανιστούν με την χρήση του γραφικού περιβάλλοντος. Τελικός σκοπός είναι οι φοιτητές να βγάζουν συμπεράσματα χωρίς να ασχολούνται με τεχνικές λεπτομέρειες κάποιου προγράμματος ή του υπολογιστή. Εάν προκύψει κάποια ανάγκη με την πάροδο του χρόνου μπορεί να δώσει βάση για επέκταση της παρούσας πτυχιακής.

Βιβλιογραφία

- [1] Πτυχιακή εργασία με τίτλο «Ανάπτυξη μη καταστροφικής μεθοδολογίας για την εκτίμηση της θερμοπερατότητας άγνωστου τοιχώματος» από τους Κωστόπουλος Ψαριανός Δημήτριος και Βαρβαδούκας Δημήτριος του τμήματος Μηχανολόγων – Μηχανικών του ΤΕΙ Θεσσαλίας.
- [2] <http://www.mech.teilar.gr/download/bark> - Εκτελέσιμο αρχείο bark.
- [3] <http://www.visualstudio.com> – Σύνολο εργαλείων της Microsoft ανάπτυξης λογισμικού. [04/04/2014]
- [4] <http://notepad-plus-plus.org> – Λογισμικό επεξεργασίας κειμένου και κώδικα. [04/04/2014]
- [5] <https://github.com> – Λογισμικό ελέγχου εκδόσεων. [04/04/2014]
- [6] <http://argouml.tigris.org> – Λογισμικό μοντελοποίησης UML. [04/04/2014]
- [7] <http://zedgraph.sourceforge.net/index.html> - Βιβλιοθήκη γραφημάτων.
- [8] <http://stackoverflow.com> – Ιστοσελίδα για βοήθεια στον προγραμματισμό. [04/04/2014]
- [9] <http://www.w3schools.org> – Ιστοσελίδα εκπαίδευσης θεμάτων προγραμματισμού (π.χ. γλώσσες).

Παράρτημα Α – Δομή XML



Εικόνα 6.2 – Δομή συνόλου στοιχείων αρχείου XML

Τύπος	Σύνταξη	Περιγραφή
ομάδα	<pre><tag> <child_tag1>... </child_tag1> <child_tag2>... </child_tag2> </tag></pre>	Μια ομάδα παραμέτρων.
σταθερά	<pre><tag> <constant unit="..."> 1.0 </constant> </tag></pre>	Ένα μέγεθος που παραμένει σταθερό καθόλη την προσομοίωση. Δηλώνεται μόνο μία αριθμητική τιμή μαζί με μονάδα.
μεταβλητή	<pre><tag> <variable x_unit="..." unit="..."> 0.0 1.0 1.0 4.0 </variable> </tag></pre>	Ένα μέγεθος που μεταβάλλεται συναρτήσει ενός άλλου μεγέθους. Δηλώνεται ένας πίνακας δύο στηλών και απεριόριστων γραμμών. Δηλώνονται μονάδες και για τις δύο στήλες.
συνάρτηση	<pre><tag> <function> <function1> </function1> </function> </tag></pre>	Ένα μέγεθος που μεταβάλλεται συναρτήσει ενός άλλου μεγέθους. Επιλέγεται μία εσωτερική συνάρτηση και δηλώνονται οι απαραίτητες αριθμητικές παράμετροι.
αναφορά	<pre><tag reference="target_tag"/></pre>	Μια αναφορά σε άλλη παράμετρο δηλωμένη ονομαστικά.
λέξη-κλειδί	<pre><tag> <keyword> keyword1 </keyword> </tag></pre>	Μια παράμετρος που ορίζεται με λέξεις-κλειδιά.

Εικόνα 6.3 – Ανάλυση στοιχείων XML

Παράρτημα Β – Εικόνες

Εικόνα 2.1 – Διπλό δρομικό τοίχωμα.....	3
Εικόνα 2.2 - ΚΕΝΑΚ (Κανονισμός Ενεργειακής Απόδοσης Κτηρίων)	4
Εικόνα 2.3 – Παράδειγμα προσομοίωσης με χρήση του λογισμικού bark.....	5
Εικόνα 3.1 – Περιβάλλον προγραμματισμού Visual Studio.....	14
Εικόνα 3.2 – Λογισμικό Notepad++ με την επέκταση XML Tools	15
Εικόνα 3.3 – Γραφικό περιβάλλον λογισμικού ελέγχου εκδόσεων GitHub	16
Εικόνα 3.4 – Λογισμικό διαγραμμάτων UML ArgoUML	17
Εικόνα 4.1 Διάγραμμα UML – Σενάριο Χρήσης Bark_GUI.....	20
Εικόνα 4.2 Διάγραμμα UML – Δομή παραθύρων προγράμματος.....	22
Εικόνα 4.3 – Δομή αρχείου XML σε δέντρο (TreeView Control)	23
Εικόνα 4.4 – Διάλογος αποθήκευσης αλλαγών.....	25
Εικόνα 4.5 Διάγραμμα UML – Προσαρμοσμένα στοιχεία ελέγχου.....	26
Εικόνα 4.6 – Προσαρμοσμένο στοιχείο ελέγχου σταθερής τιμής	27
Εικόνα 4.7 – Προσαρμοσμένο στοιχείο ελέγχου μεταβλητής τιμής.....	28
Εικόνα 4.8 – Προσαρμοσμένο στοιχείο ελέγχου συνάρτησης	29
Εικόνα 4.9 – Προσαρμοσμένο στοιχείο ελέγχου προκαθορισμένης τιμής	29
Εικόνα 4.10 – Προσαρμοσμένο στοιχείο ελέγχου αναφοράς σε άλλο στοιχείο .	30
Εικόνα 4.11 – Προσαρμοσμένο στοιχείο ελέγχου ομαδοποίησης.....	31
Εικόνα 4.12 Διάγραμμα UML – Δομή αντικειμένων-στοιχείων [Σχέδιο Α].....	32
Εικόνα 4.13 Διάγραμμα UML – Δομή αντικειμένων-στοιχείων [Σχέδιο Β].....	37
Εικόνα 4.14 Διάγραμμα UML – Δομή δεδομένων των αντικειμένων-στοιχείων.	38
Εικόνα 4.15 Διάγραμμα UML – Διαχείριση αρχείων	40
Εικόνα 4.16 Διάγραμμα UML – Βοηθητικές κλάσεις.....	41
Εικόνα 4.17 – Δείγμα αρχείου σχήματος XSDValidator.xsd	44
Εικόνα 4.18 – Δείγμα αρχείου εξόδου τύπου .dat.....	48
Εικόνα 5.1 – Άνοιγμα αρχείου δεδομένων XML από το μενού	51
Εικόνα 5.2 – Διάλογος για το άνοιγμα αρχείου δεδομένων XML.....	52
Εικόνα 5.3 – Παράδειγμα φόρτωσης αρχείου δεδομένων.....	53
Εικόνα 5.4 – Επιλογή εμφάνισης όλων ή μερικών στοιχείων ελέγχου	54

Εικόνα 5.5 – Παράδειγμα επιλογής ομάδας από το δέντρο	55
Εικόνα 5.6 – Ενέργειες δεξιού κλικ του ποντικιού σε στοιχείο δέντρου	55
Εικόνα 5.7 – Παράδειγμα μετακίνησης θέσης ενός υλικού στο δέντρο.....	57
Εικόνα 5.8 – Δείγμα βοηθητικού κειμένου στο στοιχείο simulation του δέντρου	57
Εικόνα 5.9 – Δείγμα βοηθητικού κειμένου στο στοιχείο ελέγχου start_time.....	58
Εικόνα 5.10 – Παράδειγμα ελέγχου εγκυρότητας απαραίτητου στοιχείου.....	58
Εικόνα 5.11 – Παράδειγμα ελέγχου εγκυρότητας τιμής.....	59
Εικόνα 5.12 – Εκκίνηση προσομοίωσης από το μενού του κεντρικού παραθύρου	60
Εικόνα 5.13 – Δημιουργία νέου γραφήματος από το μενού	60
Εικόνα 5.14 – Διάλογος επιλογής αξόνων γραφήματος.....	61
Εικόνα 5.15 – Παράδειγμα γραφήματος	62
Εικόνα 5.16 – Επιλογές δεξιού κλικ του ποντικιού στο διάγραμμα.....	63
Εικόνα 6.1 – Στοιχείο XML «Graph»	66
Εικόνα 6.2 – Δομή συνόλου στοιχείων αρχείου XML	71
Εικόνα 6.3 – Ανάλυση στοιχείων XML	72
