



*Πολυτεχνείο
Κρήτης*

Σύστημα μετάδοσης μετεωρολογικών δεδομένων σε πραγματικό χρόνο μέσω του Internet

Βλάχος Βασίλης



Εξεταστική Επιτροπή

Αν. Καθηγητής κ. Κώστας Καλαϊτζάκης
Καθηγητής κ. Νικόλαος Βούλγαρης
Καθηγητής κ. Μιχάλης Πατεράκης

...στους γονείς μου και σε
όσους με ανέχθηκαν τα
τελευταία χρόνια

Πρόλογος

Ξεκινώντας, θα ήθελα να ευχαριστήσω ιδιαίτερα τους ανθρώπους με τους οποίους είχα τιμή να συνεργαστώ στο πλαίσιο αυτής της διπλωματικής εργασίας. Θα πρέπει να αναφερθώ ιδιαίτερα στον κ. Καλαϊτζάκη για την πολύτιμη και ουσιαστική βοήθεια που μου πρόσφερε και κυρίως για όλα όσα αποκόμισα από την συνεργασία μας. Θεωρώ επίσης υποχρέωση μου να ευχαριστήσω και τον κ. Βούλγαρη για την τιμή που μου έκανε προτείνοντας μου να αναλάβω αυτή την εργασία. Επιπλέον θα ήθελα να αναφερθώ και στην βοήθεια που μου πρόσφερε ο κ. Πατεράκης όπου χρειάστηκε. Σημαντικότερη και ουσιαστικότερη ήταν και η συνεργασία που είχα με τον μεταπτυχιακό φοιτητή Λάζαρο Ακίτογλου. Η συμμετοχή και η συμβολή του στην αντιμετώπιση κάποιων δυσκολιών που προέκυψαν ήταν καθοριστική. Εξίσου σημαντική ήταν και η βοήθεια που μου πρόσφερε ο διαχειριστής συστημάτων του εργαστηρίου Σπύρος Κοψίδας.

ΠΕΡΙΕΧΟΜΕΝΑ

<i>Κεφάλαιο 1 Γενικά Χαρακτηριστικά του Συστήματος RES-ADMIN.....</i>	<i>13</i>
1.Γενικές πληροφορίες για το σύστημα	13
2.Περιγραφή των ανανεώσιμων πηγών ενέργειας	16
2.1 Η φωτοβολταϊκή διάταξη	16
2.2 Η ανεμογεννήτρια.....	18
3 Περιγραφή της συνδεσμολογίας με το σύστημα διαχείρισης ενέργειας (Energy Management System-EMS)	20
4 Περιγραφή της συνδεσμολογίας χωρίς το σύστημα διαχείρισης ενέργειας	22
5 Το σύστημα παρακολούθησης (monitoring)	23
<i>Κεφάλαιο 2 Εργαλεία Ανάπτυξης Συστήματος</i>	<i>27</i>
2.1 Επιλογή γλώσσας προγραμματισμού.....	27
2.2 Επιλογή εργαλείων ανάπτυξης της εφαρμογής.....	32
2.3 Πλατφόρμα ανάπτυξης της εφαρμογής.....	33
<i>Κεφάλαιο 3 Πρόγραμμα Συλλογής και Μετάδοσης Δεδομένων</i>	<i>34</i>
3.1 Σύντομη περιγραφή.....	34
3.2 Γενικά χαρακτηριστικά	35
3.3 Τεχνικά χαρακτηριστικά του υποσυστήματος	36

3.4 Τεχνική ανάλυση	42
3.4.1 Η μέθοδος <i>main()</i>	43
3.4.2 Ο <i>Constructor</i> για τον <i>Collector</i> που διαχειρίζεται την σειριακή θύρα	47
3.4.3 Η Μέθοδος <i>serialEvent</i>	49
3.4.4 Ο <i>Constructor</i> για τον <i>Collector</i> που διαχειρίζεται την ροή των δεδομένων προς των <i>Server</i>	54
3.4.5 Η Μέθοδος <i>run</i>	57
Κεφάλαιο 4 Πρόγραμμα Λήψης, Διαχείρισης και Δρομολόγησης Δεδομένων	60
4.1 Σύντομη περιγραφή.....	60
4.2 Γενικά χαρακτηριστικά	61
4.3 Τεχνικά χαρακτηριστικά του υποσυστήματος	61
4.4 Τεχνική ανάλυση	64
4.4.1 Η συνάρτηση <i>main</i>	65
4.4.2 Ο <i>Constructor</i> του <i>CollerThread</i>	67
4.4.3 Η συνάρτηση <i>run</i>	68
4.4.4 Η μέθοδος <i>addBuffer</i>	68
4.4.5 Η συνάρτηση <i>fileData</i>	70
4.4.6 Ο <i>Constructor</i> του <i>AppletThread</i>	72
4.4.7 Η μέθοδος <i>run</i> του <i>AppletThread</i>	73
Κεφάλαιο 5 Πρόγραμμα παρουσίασης δεδομένων.....	74
5.1 Σύντομη περιγραφή.....	74
5.2 Γενικά χαρακτηριστικά	75
5.3 Τεχνικά χαρακτηριστικά του υποσυστήματος	75

5.4 Τεχνική ανάλυση	77
5.4.1 Το Interface του RES-ADMIN Applet	78
5.4.2 Η συνάρτηση JAP2	80
5.4.3 Η συνάρτηση init	80
5.4.4 Η συνάρτηση start	85
5.4.5 Η συνάρτηση run	85
5.4.6 Η συνάρτηση stop	85
5.4.7 Η συνάρτηση getData	86
5.4.8 Η συνάρτηση paint	86
Κεφάλαιο 6 Βοηθητικά Προγράμματα	87
6.1 Γενικά	87
6.2 Πρόγραμμα ειδοποίησης	89
6.3 Προγράμματα επανεκκίνησης του συστήματος	90
6.3.1 Επανεκκίνηση του προγράμματος Collector	90
6.3.2 Επανεκκίνηση του προγράμματος Server	91
6.4 Πρόγραμμα διαχείρισης αρχείων	91
6.5 Πρόγραμμα εκκίνησης εφαρμογής RES-ADMIN	93
6.6 Προγράμματα ελέγχου της σειριακής θύρας	96
6.6.1 Το πρόγραμμα BlackBox	96
6.6.2 Το πρόγραμμα SimpleRead	96
6.6.3 Το πρόγραμμα SimpleWrite	97
6.7 Το Δοκιμαστικό πρόγραμμα Light	98
6.7 Το Πρόγραμμα ελέγχου Applets	99
Κεφάλαιο 7 Εγχειρίδιο Χρήσης	101

7.1 Γενικά	101
7.2 Εκτέλεση του προγράμματος Collector.....	105
7.3 Εκτέλεση του προγράμματος RES-ADMIN Server	107
7.4 Εκτέλεση του RES-ADMIN Applet	108
<i>Κεφάλαιο 8 Συμπεράσματα</i>	<i>110</i>
<i>Κεφάλαιο 9 Μελλοντικές Επεκτάσεις</i>	<i>112</i>

Εισαγωγή

Τα τελευταία πενήντα χρόνια έχουν γίνει τεράστιες ανακαλύψεις στον χώρο της μικροηλεκτρονικής, των ημιαγωγών αλλά και όσων επιστημών εξαρτώνται άμεσα από αυτές με κυριότερη ίσως την πληροφορική. Η ανακάλυψη του τρανζίστορ το 1948 από τους Bratten, Britten και Shockley θα μπορούσε αδιαμφισβήτητα να θεωρηθεί ένα από τα μεγαλύτερα επιτεύγματα του αιώνα, αφού μετέβαλλε ριζικά των τρόπο με τον οποίο εργάζεται ο άνθρωπος. Η εφεύρεση αυτή επέτρεψε τη δημιουργία πανίσχυρων για την εποχή υπολογιστικών μηχανών και δημιούργησε μια εντελώς νέα βιομηχανία, τη βιομηχανία των ημιαγωγών και των επεξεργαστών, η οποία εξακολουθεί να παρουσιάζει ραγδαίες εξελίξεις μέχρι και σήμερα.

Ο σύγχρονος μηχανικός πλέον, αλλά και ο απλός άνθρωπος, έχει τη δυνατότητα να αποκτήσει υπολογιστές με τεράστια επεξεργαστική ισχύ, αλλά και ειδικούς αισθητήρες καθώς και διαφόρους μικροελεγκτές έτοιμους να καλύψουν την κάθε ανάγκη που μπορεί να παρουσιαστεί όσο επεξεργαστικά απαιτητική ή εξειδικευμένη να είναι.

Ειδικότερα στο χώρο της επιστήμης η συνδρομή των εργαλείων αυτών είναι ανεκτίμητη, αφού παρέχει τεράστιες δυνατότητες στον επιστήμονα ειδικότερα για πραγματοποίηση πειραμάτων που απαιτούν τη μέγιστη δυνατή ακρίβεια και αξιοπιστία. Ακόμα, πολλές δύσκολες και επικίνδυνες εργασίες εκτελούνται από ειδικά σχεδιασμένες για το σκοπό αυτό μηχανές, ενώ έχει καταστεί εφικτό πολύπλοκα συστήματα να λειτουργούν εντελώς αυτόνομα σε απομακρυσμένα και επικίνδυνα περιβάλλοντα. Κοινό χαρακτηριστικό όλων των παραπάνω είναι η ύπαρξη τεχνολογικά άρτιων συστημάτων, επεξεργαστών, αισθητήρων και μικροελεγκτών τα οποία αναλαμβάνουν να φέρουν σε πέρας την αποστολή τους. Σε κάθε περίπτωση θα μπορούσε να υποστηριχτεί ότι η σημερινή τεχνολογία στα συστήματα αυτά έχει σε πολύ μεγάλο βαθμό υπερκαλύψει τις ανάγκες του συγχρόνου μηχανικού.

Αντίθετα ορισμένες ατέλειες και κενά παρουσιάζονται στη μετάδοση και διάδοση των πληροφοριών αυτών. Τόσο στην περίπτωση του επιστήμονα που πραγματοποιεί ένα πείραμα και απαιτεί την άμεση λήψη των αποτελεσμάτων του, την επεξεργασία τους καθώς και την περαιτέρω μετάδοση του σε άλλα μέλη της κοινότητας, όσο και στην περίπτωση της συσκευής που λειτουργεί αυτόνομα σε κάποιο απομακρυσμένο ή μη περιβάλλον, υπάρχει η ανάγκη για άμεση πρόσβαση στις πληροφορίες που λαμβάνονται αλλά και η δυνατότητα για συνεχή παρακολούθησή τους. Δεν θα ήταν σε καμία περίπτωση υπερβολικό εάν υποστηριχτεί ότι με τις τελευταίες τεχνολογικές εξελίξεις η παρακολούθηση της πληροφορίας σε πραγματικό χρόνο είναι εξίσου σημαντική με την ίδια την παραγωγή της πληροφορίας.

Εάν για την ηλεκτρονική αποτέλεσε σταθμό η ανακάλυψη του τρανζίστορ, η οποία επέφερε τη ραγδαία πρόοδο που επακολούθησε, ανάλογης σημασίας ήταν και είναι για το χώρο των τηλεπικοινωνιών η δημιουργία και διάδοση του Internet. Το διαδίκτυο δημιουργήθηκε στα μέσα της δεκαετίας του εβδομήντα και αποτέλεσε την προσπάθεια του αμερικανικού

Υπουργείου Αμύνης (U.S. Defense Advanced Research Projects Agency-DARPA) για τη δημιουργία ενός ευρέως κατανεμημένου δικτύου χωρίς καμία ιεραρχική δομή ικανό να λειτουργήσει ακόμα και αν πληγεί μεγάλος αριθμός των κόμβων που το αποτελούν. Έκτοτε λίγα πράγματα έχουν παραμείνει τα ίδια αφού πλέον το Internet είναι ευρύτατα διαδεδομένο παρέχοντας πλήθος δυνατοτήτων και υπηρεσιών. Η εξάπλωσή του είναι τόσο εκτεταμένη που το καθιστά σχεδόν απαραίτητο σε πολλές δραστηριότητες της ανθρώπινης ζωής. Η επιστημονική και εμπορική χρήση του διαδικτύου έχει επιβάλλει νέες τάσεις και απαιτήσεις στην κατασκευή συστημάτων λογισμικού, όπου η ενσωμάτωση του δικτύου θεωρείται πλέον εκ των ουκ άνευ. Τα περισσότερα συστήματα λογισμικού που χρησιμοποιούνται στα διάφορα υπολογιστικά συστήματα πλέον έχουν τη δυνατότητα αυτή.

Θα πρέπει ωστόσο να ληφθεί σοβαρά υπόψη ότι παρόμοιες δυνατότητες δεν έχουν αναπτύξει πλήρως τα συστήματα λογισμικού για μονάδες μικρότερες από τους επεξεργαστές, όπως οι μικροελεγκτές και οι αισθητήρες. Ενώ δηλαδή οι συσκευές αυτές παρέχουν εξαιρετικές δυνατότητες στο να λαμβάνουν ιδιαίτερα ακριβείς μετρήσεις ή να πραγματοποιούν πολύ αποτελεσματικά συγκεκριμένες λειτουργίες δεν είναι ακόμα σε θέση να τις παρουσιάζουν με ιδιαίτερα αποτελεσματικό τρόπο. Προφανώς για να θεωρηθεί αποτελεσματικός ο τρόπος μετάδοσης θα πρέπει να συγκεντρώνει ορισμένα χαρακτηριστικά. Αναμφίβολα θεωρείται βασικό να υπάρχει η δυνατότητα λήψης της πληροφορίας σε πραγματικό χρόνο και η παρακολούθηση της να είναι εφικτή απ' όσο το δυνατόν περισσότερα σημεία και με όσο το δυνατόν περισσότερους τρόπους. Όπως γίνεται αντιληπτό το μέσο το οποίο προσφέρει σήμερα τη μεγαλύτερη δυνατή κάλυψη σε παγκόσμιο επίπεδο καθώς και τους περισσότερους τρόπους σύνδεσης σε αυτό είναι το Internet. Είναι γνωστό ότι το Internet είναι προσβάσιμο από το μεγαλύτερο κομμάτι του πλανήτη, καθώς και ότι η σύνδεση αυτή μπορεί να πραγματοποιηθεί με πλήθος διαφορετικών τρόπων που περιλαμβάνουν από το απλό αναλογικό ή ψηφιακό τηλεφωνικό δίκτυο και τις γραμμές ISDN έως δορυφορικές ζεύξεις, κινητή τηλεφωνία και οπτικές ίνες. Επιπλέον το εύρος ζώνης διαρκώς αυξάνει και νέες τεχνολογίες, όπως η αξιοποίηση της υποδομής της καλωδιακής τηλεόρασης ή η τεχνολογία DSL συμβάλλουν προς

αυτό. Επιπρόσθετα, η πρόσβαση στο διαδίκτυο μπορεί να επιτευχθεί χρησιμοποιώντας αρκετές διαφορετικές συσκευές πέραν του ηλεκτρονικού υπολογιστή. Συγκεκριμένα, κινητά τηλέφωνα, φορητοί και υποφορητοί υπολογιστές καθώς και ηλεκτρονικά σημειωματάρια παρέχουν πλέον πρόσβαση στο Internet. Είναι λοιπόν προφανές ότι η λειτουργικότητα των μικροελεγκτών, μικροεπεξεργαστών και αισθητήρων θα αύξανε αλματωδώς αν υπήρχε το κατάλληλο λογισμικό που θα τους παρείχε τη δυνατότητα αξιοποίησης τους με τη βοήθεια Internet.

Στόχος της διπλωματικής εργασίας αυτής είναι η κατασκευή ενός συστήματος λογισμικού (RES-ADMIN) το οποίο λαμβάνει δεδομένα σε πραγματικό χρόνο από μια ή περισσότερες πηγές καθιστώντας τα διαθέσιμα μέσω του Internet.

- Τα γενικά χαρακτηριστικά του συστήματος RES-ADMIN αναλύονται στο Κεφάλαιο 1.
- Στο Κεφάλαιο 2 περιγράφονται τα εργαλεία τα οποία χρησιμοποιήθηκαν για την ανάπτυξη της εφαρμογής, καθώς και τα υπολογιστικά συστήματα που υποστηρίζουν το σύστημα RES-ADMIN.
- Το Κεφάλαιο 3 καλύπτει όλα τα στοιχεία που αφορούν το υποπρόγραμμα Collector.
- Στο Κεφάλαιο 4 υπάρχουν όλες οι πληροφορίες για το RES-ADMIN Server.
- Στο Κεφάλαιο 5 αναλύεται ο τρόπος κατασκευής και η λειτουργία του RES-ADMIN Applet.
- Το Κεφάλαιο 6 αναφέρεται αναλυτικά σε όλα τα βοηθητικά προγράμματα που υποστηρίζουν το σύστημα.
- Το Κεφάλαιο 7 είναι το εγχειρίδιο χρήσης το οποίο επεξηγεί πλήρως τις απαιτούμενες ενέργειες για την ορθή λειτουργία του συστήματος.
- Στο Κεφάλαιο 8 καλύπτονται τα συμπεράσματα τα οποία αποκομίστηκαν από τη δίμηνη λειτουργία του συστήματος.
- Τέλος στο Κεφάλαιο 9 υπάρχουν οι προτεινόμενες μελλοντικές επεκτάσεις.

Κεφάλαιο 1

Γενικά Χαρακτηριστικά του Συστήματος RES-ADMIN

1. Γενικές πληροφορίες για το σύστημα

Το σύστημα RES-ADMIN βασίσθηκε στο ερευνητικό έργο «Ανάπτυξη Αυτόνομου Υβριδικού Συστήματος για την Ενεργειακή Υποστήριξη Τηλεοπτικού Αναμεταδότη Ισχύος 1,5KW», το οποίο χρηματοδοτήθηκε από το δεύτερο κοινοτικό πλαίσιο στήριξης. Το έργο αυτό εντάχθηκε στο Περιφερειακό Επιχειρησιακό Πρόγραμμα Κρήτης (1994-1999) ως πρόγραμμα

εφαρμοσμένης έρευνας. Ανάδοχος φορέας υπήρξε το Πολυτεχνείο Κρήτης, ενώ Επιστημονικός υπεύθυνος του έργου ήταν ο Καθηγητής κ. Ν. Βούλγαρης. Συνεργαζόμενος φορέας ήταν το παράρτημα Χανίων του Τ.Ε.Ι. Κρήτης, υπεύθυνος υπήρξε ο κ. Ν. Φραγκιαδάκης. Φορέας χρήστης ήταν οι Ραδιοτηλεοπτικές επιχειρήσεις «ΙΚΑΡΟΣ» (Κρήτη TV) με υπεύθυνο τον κ. Δημητρίου.

Στη συνέχεια συμμετείχε το εργαστήριο Ηλεκτρικών Κυκλωμάτων και Ανανεώσιμων Πηγών Ενέργειας, επικεφαλής του οποίου είναι ο Αναπληρωτής Καθηγητής κ. Καλαϊτζάκης. Η τελική μορφή του έργου περιλαμβάνει τον οικίσκο (Σχήμα 1) στον οποίο συλλέγονται τα μετεωρολογικά δεδομένα και παράγεται ηλεκτρική ενέργεια με φιλικούς προς το περιβάλλον τρόπους (ανεμογεννήτρια και φωτοβολταϊκές συστοιχίες), την ασύρματη ζεύξη με το εργαστήριο Microcontrol καθώς και το λογισμικό για μετάδοση και παρακολούθηση των δεδομένων σε πραγματικό χρόνο μέσω του Internet.

Αναλυτικά ο εξοπλισμός του εργαστηρίου περιλαμβάνει

- Μια ονομαστική ανεμογεννήτρια ισχύος 2 KW με δρομέα τριών πτερυγίων άμεσα συνδεδεμένο σε μια σύγχρονη πολύ-πολική γεννήτρια.
- Φωτοβολταϊκές (ηλιακές) διατάξεις ισχύος 900 W.
- Έναν DC/DC μετατροπέα συνδεδεμένο στην έξοδο της ανεμογεννήτριας ώστε να ανιχνεύεται το σημείο μέγιστης ισχύος.
- Έναν DC/DC μετατροπέα συνδεδεμένο στην έξοδο της φωτοβολταϊκής διάταξης ώστε να ανιχνεύεται το σημείο μέγιστης ισχύος
- Ένα σύστημα μπαταριών
- Έναν αναστροφέα διασυνδεδεμένο στο ηλεκτρικό δίκτυο.
- Έναν αυτόνομο αναστροφέα που τροφοδοτεί το τοπικό φορτίο καταναλωτή.
- Μια σειρά αισθητήρων για την παρακολούθηση και την καταγραφή της ταχύτητας και της κατεύθυνσης του ανέμου, της ολικής ηλιακής ακτινοβολίας, της θερμοκρασίας περιβάλλοντος, της υγρασίας

περιβάλλοντος, της ατμοσφαιρικής πίεσης, του παραγόμενου ρεύματος και της τάσης από την ανεμογεννήτρια καθώς και του παραγόμενου ρεύματος και τάσης των φωτοβολταϊκών πανέλων.

- Ένα σύστημα συλλογής δεδομένων το οποίο περιλαμβάνει και έναν Ηλεκτρονικό Υπολογιστή εξοπλισμένο με μια A/D-D/A κάρτα ο οποίος εκτελεί το πρόγραμμα LabView.
- Μια σειριακή ασύρματη τηλεπικοινωνιακή ζεύξη για την μετάδοση δεδομένων από και προς τον Server του εργαστηρίου.
- Το λογισμικό το οποίο πραγματοποιεί την συγκέντρωση και τη μετάδοση των δεδομένων στο Internet σε πραγματικό χρόνο.



Σχήμα 1 Ο οικίσκος RES

Στο σύστημα αυτό χρησιμοποιούνται για πειραματικούς σκοπούς δυο διαφορετικές συνδεσμολογίες. Μια με διαχείριση ενέργειας (συνδεσμολογία Α) και μια χωρίς διαχείριση ενέργειας (συνδεσμολογία Β). Οι συνδεσμολογίες αυτές θα περιγραφούν αναλυτικά παρακάτω.

2. Περιγραφή των ανανεώσιμων πηγών ενέργειας

Οι ανανεώσιμες πηγές ενέργειας του εργαστηρίου αποτελούνται από μια ανεμογεννήτρια καθώς και μία φωτοβολταϊκή διάταξη. Τα χαρακτηριστικά των πηγών ακολουθούν.

2.1 Η φωτοβολταϊκή διάταξη

Η φωτοβολταϊκή διάταξη (Σχήμα 2) αποτελείται από 12 πάνελα κατασκευασμένα από τη SIEMENS. Τα πάνελα είναι στραμμένα νότια και η κλίση τους σε σχέση με τον οριζόντιο άξονα είναι 30° . Τα χαρακτηριστικά ακολουθούν στον Πίνακα 1.



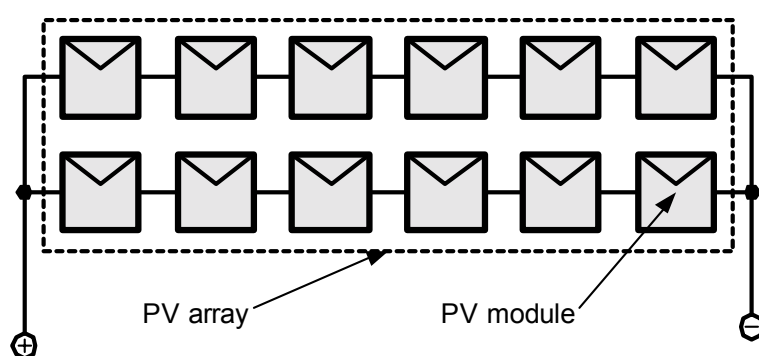
Σχήμα 2 Η φωτοβολταϊκή διάταξη

Πίνακας 1 Χαρακτηριστικά λειτουργίας των ηλιακών πανέλων		
Χαρακτηριστικό	Περιγραφή	Μονάδα
Μοντέλο	SIEMENS SP75	
Ασφάλεια	Non-incendive, Class 1, Division 2, Group A,B,C,D, Fire Class C, Fuse 8A	
Μέγιστη Ισχύς (1 kW/m^2 , 25°C)	75	W
Τάση ανοικτοκυκλώματος	21.7	V
Τάση βραχυκυκλώματος	4.8	A
Ονομαστικό Ρεύμα	4.4	A
Ονομαστική Τάση	17.0	V
Μέγιστη τάση ανοικτοκυκλώματος	600	V

Για πειραματικούς σκοπούς χρησιμοποιήθηκαν, όπως περιγράφηκε παραπάνω, δύο διαφορετικές συνδεσμολογίες στις συνδέσεις των πανέλων.

2.1.1 Συνδεσμολογία A

Δύο γραμμές από 6 πανέλα σε σειρά οι οποίες είναι συνδεδεμένες παράλληλα, όπως φαίνεται στο Σχήμα 3, παρέχοντας συνολική τάση εξόδου στο σύστημα $V_{oc \text{ tot}} = 130.2 \text{ V}$, συνολικό ρεύμα βραχυκυκλώματος $I_{sc \text{ tot}} = 9.6 \text{ A}$, καθώς και μέγιστη ισχύ $P_{\text{max}} = 900 \text{ W}$.



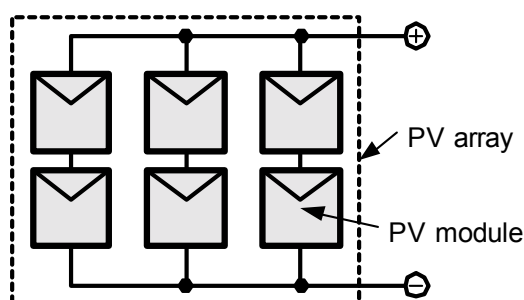
Σχήμα 3 Η PV διάταξη που χρησιμοποιείται για τη συνδεσμολογία A

Η συνδεσμολογία αυτή προσφέρει τη μέγιστη τάση που μπορεί να ανεχτεί ο μετατροπέας ο οποίος είναι συνδεδεμένος με την έξοδο να ανεχτεί. Με αυτό τον τρόπο το ρεύμα διατηρείται στο ελάχιστο, ελαχιστοποιώντας τις

απώλειες ισχύος που παρατηρούνται στα μεγάλου μήκους καλώδια που συνδέουν την PV διάταξη με τον μετατροπέα.

2.1.2 Συνδεσμολογία B

Τρεις γραμμές από 2 πάνελα σε σειρά συνδέονται παράλληλα, όπως φαίνεται στο Σχήμα 4, παρέχοντας συνολική τάση εξόδου $V_{oc\ tot} = 43.4\ V$, συνολικό ρεύμα βραχυκυκλώματος $I_{sc\ tot} = 14.4\ A$ και συνολική μέγιστη ισχύ $P_{max} = 450\ W$. Τα υπόλοιπα 6 πάνελα δεν χρησιμοποιούνται σε αυτήν την συνδεσμολογία.



Σχήμα 4 Η PV διάταξη που χρησιμοποιείται για τη συνδεσμολογία B

2.2 Η ανεμογεννήτρια

Τα χαρακτηριστικά της ανεμογεννήτριας (Σχήμα 5) παραθέτονται στον Πίνακα 2. Η ανεμογεννήτρια είναι συνδεδεμένη σε ένα πυργίσκο ύψους 2 μέτρων. Η τάση της γεννήτριας ανορθώνεται από ένα τριφασικό ανορθωτή με διόδους. Επιπλέον η έξοδος του ανορθωτή είναι συνδεδεμένη μέσω διακόπτη σε dummy load προστατεύοντας την γεννήτρια από υπερβολική ταχύτητα περιστροφής, στην περίπτωση κατά την οποία η παραγόμενη ισχύς δεν μπορεί να απορροφηθεί από το υπόλοιπο κύκλωμα.

Πίνακας 2 Χαρακτηριστικά της ανεμογεννήτριας		
Χαρακτηριστικά	Περιγραφή	Μονάδα
Δρομέας	3 ελίκων, 3.6 m διάμετρος, direct coupled	
Γεννήτρια	Σύγχρονη, πολύ-πολική, μόνιμοι μαγνήτες	
Ταχύτητα εκκινήσεως	3	m/s
Ταχύτητα αποκοπής	25	m/s
Ονομαστική Ισχύς (12 m/s)	2000	W
Ονομαστική Τάση (12 m/s, RMS)	100	V
Μηχανική αποτελεσματικότητα	80	%
Ηλεκτρική αποτελεσματικότητα	80	%
Ονομαστική Γεννήτρια RPM	390	rpm



Σχήμα 5 Η ανεμογεννήτρια

3 Περιγραφή της συνδεσμολογίας με το σύστημα διαχείρισης ενέργειας (Energy Management System-EMS)

Η συνδεσμολογία Α χρησιμοποιείται για την σύνδεση της PV διάταξης. Οι έξοδοι των ανανεώσιμων πηγών ενέργειας (Renewable Energy Sources-RES) τροφοδοτούν δύο DC/DC μετατροπείς τύπου buck, αντίστοιχα. Ο κάθε μετατροπέας παίζει διπλό ρόλο:

- Μετατροπή της υψηλής τάσης εξόδου, τόσο της PV διάταξης όσο και της ανεμογεννήτριας, σε χαμηλότερη τάση της τάξης των 24 V η οποία είναι κατάλληλη για την φόρτιση των μπαταριών.
- Παρακολούθηση του Μέγιστου Σημείου Ισχύος τόσο της PV διάταξης όσο και της ανεμογεννήτριας.

Οι buck converters έχουν σχεδιαστεί και κατασκευαστεί στο Εργαστήριο. Βασίζονται στην Pulse Width Modulation (PWM) technique. Η τάση εισόδου τους επιλέγεται λαμβάνοντας υπόψη τις απώλειες ισχύος κατά μήκος των καλωδίων από τις RES, οι οποίες μειώνονται όταν η τάση αυξάνεται (χαμηλότερο ρεύμα για την ίδια ισχύ) και του κόστους των ημιαγωγών, το οποίο αυξάνεται όταν η τάση λειτουργίας τους αυξάνεται.

Η ενέργεια που παράγεται αποθηκεύεται σε ένα σύστημα μπαταριών αποτελούμενο από δύο μπαταρίες μολύβδου οξέος (two lead-acid batteries) με ονομαστική τάση 12 V και ονομαστική χωρητικότητα 225 Ah η κάθε μία. Τα χαρακτηριστικά κάθε μπαταρίας παραθέτονται στον Πίνακα 3. Οι δύο μπαταρίες είναι συνδεδεμένες σε σειρά και παρέχουν ρεύμα στον τοπικό καταναλωτή στην περίπτωση που δεν παράγεται ρεύμα από το σύστημα RES και ταυτόχρονα το ηλεκτρικό δίκτυο δεν είναι διαθέσιμο. Οι μετατροπείς είναι συνδεδεμένοι στις μπαταρίες μέσω διόδων προκειμένου να προφυλαχτούν από την ροή ανάστροφου ρεύματος προερχόμενου από τις μπαταρίες.

Πίνακας 3 Χαρακτηριστικά λειτουργίας των μπαταριών	
Off-Load Τάση	Συνθήκη φόρτισης
>12.80V	100%
12.55V	75%
12.32V	50%
12.18	25%
<12V	0%

Ο τοπικός καταναλωτής τροφοδοτείται από έναν DC/AC αναστροφέα με ονομαστική ισχύ 1.5 kW. Ένας κατάλληλος πίνακας διακόπτων (switch array) αυτόματα συνδέει το καταναλωτή στο δίκτυο παροχής ισχύος (power grid), όταν η RES ενέργεια δεν επαρκεί και οι μπαταρίες είναι αφόρτιστες εξασφαλίζοντας την αδιάκοπη παροχή στον καταναλωτή.

Το παραπάνω περιγραφόμενο σύστημα ελέγχεται από μια μονάδα ελέγχου βασιζόμενη στον μικροελεγκτή (microcontroller μC) 80C196KC της INTEL. Ο μικροελεγκτής εκτελεί τον αλγόριθμο για την διαχείριση της ενέργειας του συστήματος (EMS). Στόχος του EMS είναι η συνεχής ικανοποίηση των ακόλουθων συνθηκών, οι οποίες παρατίθενται κατά φθίνουσα προτεραιότητα:

- Παραγωγή μέγιστης δυνατής ισχύος από το RES.
- Αδιάκοπη παροχή στον καταναλωτή.
- Παρακολούθηση της κατάστασης φόρτισης (charge state) στις μπαταρίες.

Ταυτόχρονα πρέπει να εκτελούνται διαγνωστικές ρουτίνες δυσλειτουργιών προκειμένου να προστατευτούν οι συσκευές του συστήματος, όπως:

Προστασία των μπαταριών από υπερφόρτιση (over-charging) ή από υπερεκφόρτιση (over-discharging).

Σύμφωνα με το διάγραμμα του Σχήματος 4, για την ικανοποίηση των συνθηκών λειτουργίας καθώς και των διαγνωστικών ελέγχων για δυσλειτουργίες, οι διαδικασίες που πρέπει να πραγματοποιεί ο EMS αλγόριθμος είναι:

- Έλεγχος των παραμέτρων των PWM DC/DC μετατροπών για MPT παρακολούθηση τόσο για τον PV διάταξης όσο και για την ανεμογεννήτρια.
- ON/OFF έλεγχος του dummy load της ανεμογεννήτριας
- Έλεγχος της κατάστασης φόρτισης της μπαταρίας

4 Περιγραφή της συνδεσμολογίας χωρίς το σύστημα διαχείρισης ενέργειας

Η συνδεσμολογία B χρησιμοποιείται για την σύνδεση της διάταξης PV. Διαφορετική χρήση κάθε RES εφαρμόζεται σε αυτήν την συνδεσμολογία. Η PV διάταξη τροφοδοτεί τον τοπικό καταναλωτή, ενώ η ενέργεια που παράγεται από την ανεμογεννήτρια εισέρχεται στο ηλεκτρικό δίκτυο.

Η έξοδος της PV διάταξης καταλήγει στον φορτιστή (charger) SIEMENS SR20M, ο οποίος δέχεται 24 V τάση εισόδου και παρέχει μέγιστο ρεύμα εξόδου 20 A. Ο καταναλωτής τροφοδοτείται μέσω του SIEMENS SP1500/24 αναστροφέα, παρέχοντας τα παρακάτω χαρακτηριστικά (Πίνακας 4):

Πίνακας 4 Χαρακτηριστικά λειτουργίας του SP1500/24 αναστροφέα.	
Τάση εισόδου	24 V
Τάση εξόδου	230 V
Συχνότητα εξόδου	50 Hz
Ονομαστική ισχύς εξόδου	1500 W
Απόδοση	> 91 %

Η ανεμογεννήτρια είναι διασυνδεδεμένη στο ενεργειακό δίκτυο μέσω ενός DC/AC αναστροφέα κατασκευασμένου από την FRONIUS, μοντέλο Sunrise-Mini. Η συνολική ενέργεια η οποία παράγεται από την ανεμογεννήτρια εισάγεται αποκλειστικά στο ενεργειακό δίκτυο. Τα χαρακτηριστικά του αναστροφέα παραθέτονται στον Πίνακα 5.

Πίνακας 5 Χαρακτηριστικά λειτουργίας του Sunrise-Mini αναστροφέα	
Τάση εισόδου	120-300 V
Μέγιστη τάση εισόδου	350 V
Τάση δικτύου	230 V
Συχνότητα δικτύου	50 Hz
Παραμόρφωση αρμονικής εξόδου	< 3 %
Ονομαστική ισχύς εξόδου	1000 W
Αποτελεσματικότητα	93 %

5 Το σύστημα παρακολούθησης (monitoring)

Οι περιβαλλοντικές παράμετροι καθώς και οι ποσότητες που αφορούν την παραγόμενη από τις RES ενέργεια παρακολουθούνται σε μόνιμη βάση. Ένα κατάλληλο σύστημα παρακολούθησης έχει αναπτυχθεί για αυτόν τον σκοπό αποτελούμενο από τους απαραίτητους αισθητήρες και τα interface κυκλώματα, ένα σύστημα μετάδοσης δεδομένων καθώς και ειδικό σύστημα μετάδοσης των μετρήσεων μέσω του Internet. Το σύστημα παρακολούθησης μαζί με τα ενεργειακά συστήματα που περιγράφονται παραπάνω είναι τοποθετημένα σε ένα μικρό οικίσκο κοντά στις RES.



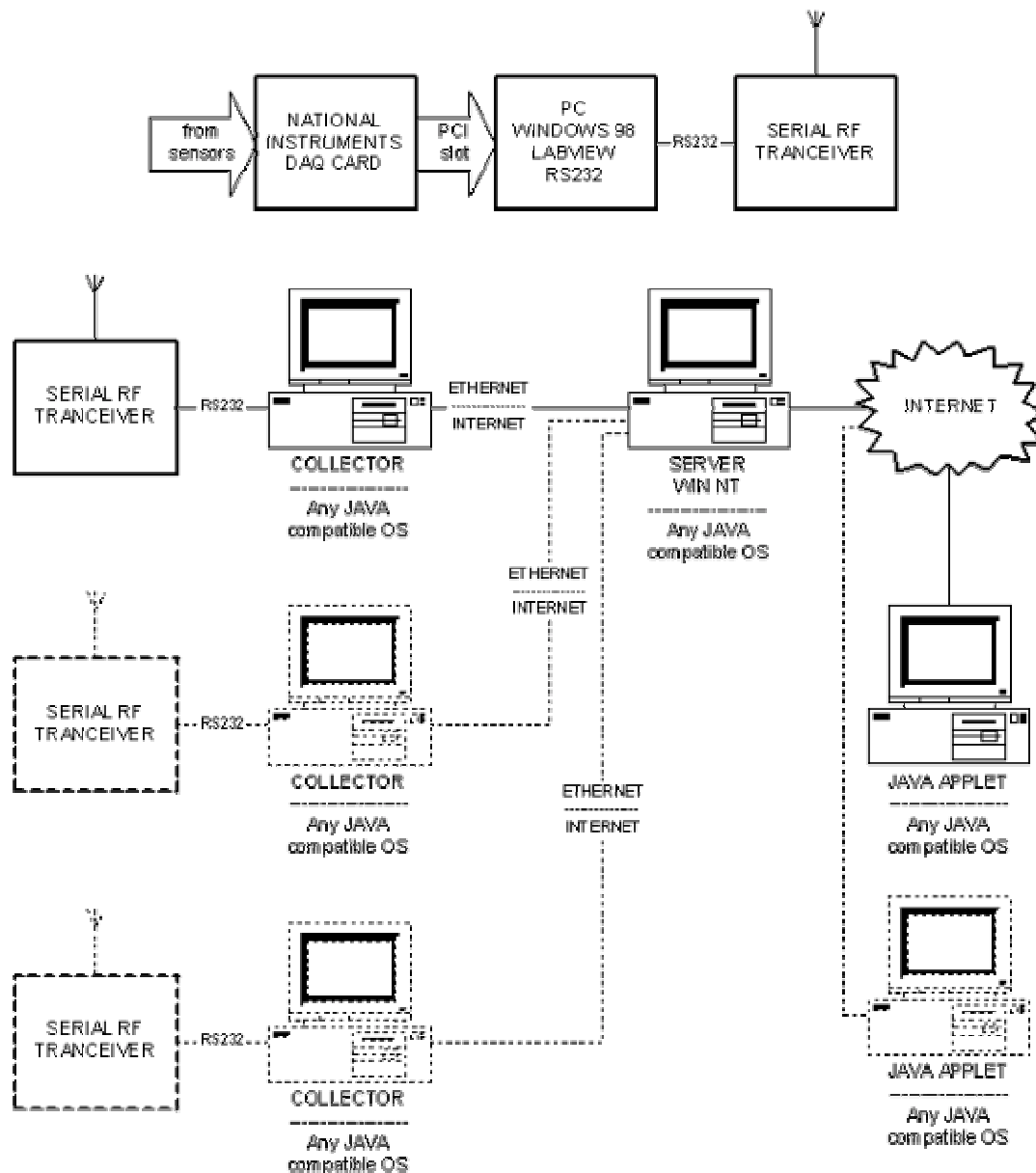
Σχήμα 6 Το σύστημα δειγματοληψίας

Τα interface κυκλώματα και ένα μέρος από τους αισθητήρες έχουν αναπτυχθεί στο εργαστήριο. Το σύστημα μετάδοσης δεδομένων χρησιμοποιεί

την NATIONAL INSTRUMENTS PCI-6024E DAQ κάρτα με το CB-68LP R686 1m connector block (Σχήμα 6). Η κάρτα οδηγείται από το λογισμικό LabView το οποίο είναι κατάλληλα ρυθμισμένο για τους υπάρχοντες αισθητήρες και εκτελείται σε Η/Υ ο οποίος βρίσκεται στον οικίσκο που είναι οι RES. Τα δεδομένα που συλλέγονται αποθηκεύονται καθημερινά σε αρχεία. Τα δεδομένα αυτά μεταδίδονται σε πραγματικό χρόνο ασύρματα στον RES-ADMIN Server ο οποίος βρίσκεται στο εργαστήριο σε απόσταση περίπου 100 μέτρων από τον RES οικίσκο. Το λογισμικό το οποίο έχει αναπτυχθεί στα πλαίσια αυτής της διπλωματικής εργασίας, αναλαμβάνει την συλλογή και την μετάδοση των δεδομένων αυτών μέσω του Internet, καθιστώντας την μελέτη της λειτουργίας του συστήματος δυνατή από απόσταση απαιτώντας την ύπαρξη μόνο ενός συνδεδεμένου στο Internet υπολογιστή. Η διάταξη του συστήματος παρακολούθησης και μετάδοσης δεδομένων φαίνεται στο Σχήμα 7.

Οι αισθητήρες που χρησιμοποιήθηκαν για τις μετρήσεις αναφέρονται παρακάτω:

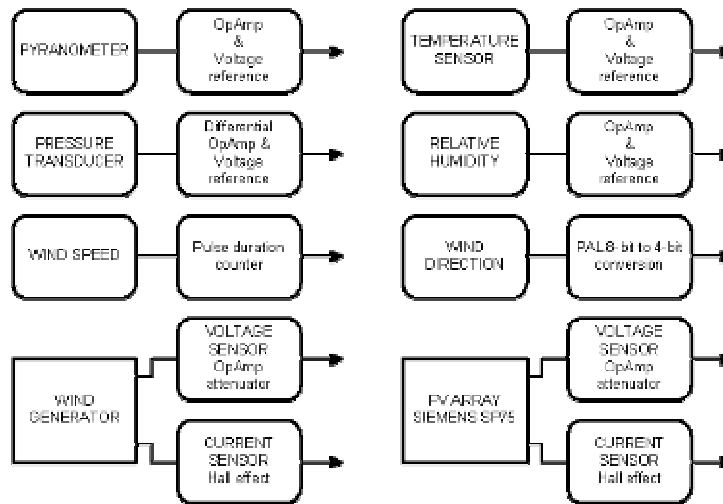
- Ένα πυρανόμετρο τύπου Delta-T GS1 CM3 με ευαισθησία $22.08\mu\text{V}/(\text{W}/\text{m}^2)$ και φασματική περιοχή 305-2800 nm. Είναι ISO δεύτερης κλάσης.
- Ένας NATIONAL SEMICONDUCTORS LM35 αισθητήρα θερμοκρασίας με ευαισθησία $10\text{ mV}/^\circ\text{C}$.
- Ένας SCC 15AN αισθητήρα από την SenSym ο οποίος χρησιμοποιείται για τις μετρήσεις ατμοσφαιρικής πίεσης με μέγιστη πίεση 15 psi (1.03421355 bar) και ευαισθησία $6\text{ mV}/\text{psi}$ ($87.02264634\text{ mV}/\text{bar}$).
- Ένα MERCATOR RHU217-AT σχετικής υγρασίας module με εύρος μέτρησης (span) 0-3.3 V για 0-100% RH.
- Το ανεμόμετρο παρέχει ένα τετραγωνικό παλμό κάθε μια περιστροφή του στροφέα. Είναι τοποθετημένο σε ένα ιστό 12 μέτρων μαζί με το ανιχνευτή κατεύθυνσης ανέμου. Επειδή η ανεμογεννήτρια είναι τοποθετημένη σε ύψος 15 μέτρων, η ταχύτητα ανέμου που διαβάζεται στα 12 μέτρα υπολογίζεται για το ύψος των 15 μέτρων.



Σχήμα 7 Το σύστημα παρακολούθησης

- Ο ανιχνευτής κατεύθυνσης ανέμου κωδικοποιεί τις 16 κύριες κατευθύνσεις σε πακέτα των 8-bits.
- Δύο Hall-effect αισθητήρες χρησιμοποιούνται για τις μετρήσεις του ρεύματος της PV διάταξης και της ανεμογεννήτριας.
- Η τάση εξόδου της PV διάταξης και της ανεμογεννήτριας μετρώνται μέσω τελεστικών ενισχυτών απομόνωσης.

Οι αισθητήρες και τα interface blocks φαίνονται στο Σχήμα 8



Σχήμα 8 Οι αισθητήρες και τα *interface blocks*

Όλα τα interfacing blocks είναι συνδεδεμένα με την κάρτα μετάδοσης δεδομένων μέσω χαμηλοπερατών RC φίλτρων και προστατευτικών διόδων.

Κεφάλαιο 2

Εργαλεία Ανάπτυξης Συστήματος

2.1 Επιλογή γλώσσας προγραμματισμού

Το σύστημα RES-ADMIN αποτελείται από αρκετά υποσυστήματα, που το καθένα έχει ειδικές απαιτήσεις και ανάγκες που πρέπει να καλυφθούν. Εξετάστηκαν δύο δυνατές υλοποιήσεις. Η πρώτη υλοποίηση προέβλεπε όλο το λογισμικό να αναπτυχθεί σε μια γλώσσα προγραμματισμού, η οποία θα καλύπτει όσο το δυνατόν καλύτερα τις ανάγκες του συστήματος. Η δεύτερη υλοποίηση προέβλεπε τη χρησιμοποίηση διαφορετικής γλώσσας

προγραμματισμού για κάθε υποσύστημα, ανάλογα με το ποια γλώσσα είναι πιο κοντά στις ανάγκες του κάθε υποσυστήματος.

Αποφασίσθηκε να κατασκευαστεί το σύστημα σε μια γλώσσα προγραμματισμού, αφού εκτιμήθηκε ότι υπάρχει η δυνατότητα μια γλώσσα να καλύψει πλήρως τις ανάγκες του συστήματος χωρίς να συγκεντρώνει τα μειονεκτήματα που φέρουν συστήματα κατασκευασμένα από διαφορετικές γλώσσες προγραμματισμού. Τέτοιες αδυναμίες είναι:

- Προβλήματα συμβατότητας. Είναι εξαιρετικά δύσκολο να επιτευχθεί απροβλημάτιστη επικοινωνία μεταξύ προγραμμάτων κατασκευασμένων από διαφορετικές γλώσσες προγραμματισμού. Επιπλέον παρατηρείται το φαινόμενο ακόμα και η ίδια γλώσσα να παρουσιάζει διαφορετική συμπεριφορά σε διαφορετικά περιβάλλοντα.
- Είναι ιδιαίτερα δύσκολο για το μηχανικό που αναπτύσσει την εφαρμογή να προγραμματίζει παράλληλα σε διαφορετικά περιβάλλοντα αφού η εναλλαγή των κανόνων ανάλογα με την κάθε γλώσσα προκαλεί μεγάλη μείωση της απόδοσής του.

Σημαντικό ρόλο έπαιξε το γεγονός ότι οι προδιαγραφές του συστήματος ενώ αρχικά προέβλεπαν τη συλλογή και παρουσίαση δεδομένων μέσω του Internet από ένα πειραματικό σταθμό, αργότερα αποφασίστηκε ότι είναι δυνατό να μεγιστοποιηθεί η λειτουργικότητα του συστήματος αν προστεθεί η δυνατότητα υποστήριξης περισσότερων σταθμών. Παράλληλα αποφασίσθηκε ότι τα λαμβανόμενα δεδομένα θα πρέπει να είναι προσπελάσιμα σε μεγάλο αριθμό ατόμων και όχι μόνο στο διαχειριστή του συστήματος. Η παρουσίαση των δεδομένων αποφασίσθηκε να πραγματοποιείται μέσω των προγραμμάτων πλοήγησης του δικτύου (Web browsers).

Το γεγονός ότι το λογισμικό θα έπρεπε να είναι συμβατό με παραπάνω από μια πλατφόρμες, αφού θα υπήρχαν πολλοί πειραματικοί σταθμοί καθώς και η ότι τα δεδομένα θα έπρεπε να παρουσιάζονται μέσω του browser, κατέστησαν την επιλογή της Java ως γλώσσας προγραμματισμού μονόδρομο. Η Java είναι η μοναδική γλώσσα που θα μπορούσε να καλύψει τις ανάγκες του έργου αυτού, αφού συγκεντρώνει πολύ ενδιαφέροντα χαρακτηριστικά

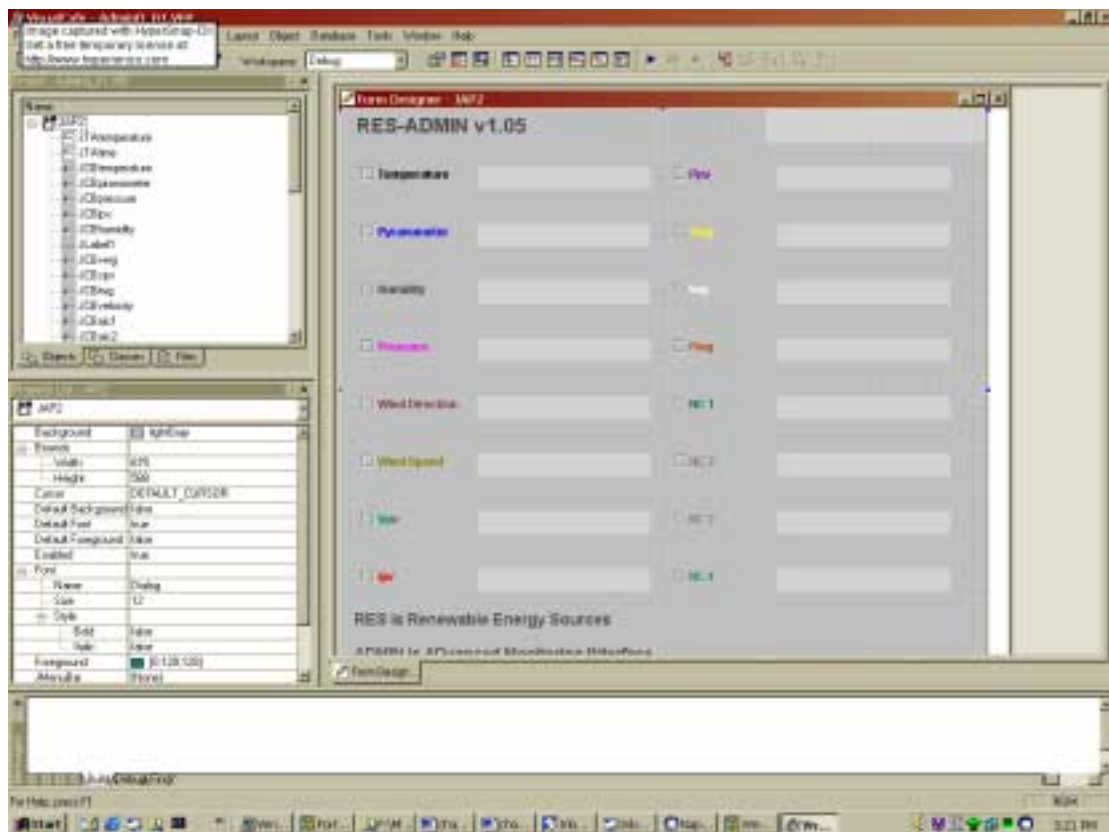
όπως το ότι είναι ανεξάρτητη του λειτουργικού συστήματος και του υλικού που χρησιμοποιείται (Platform Independency), απαραίτητο για την υποστήριξη διαφορετικών σταθμών καθώς και ότι είναι δυνατόν να εκτελείται μέσα από ένα Web browser ως Applet. Τα χαρακτηριστικά αυτά είναι ιδιαίτερα σημαντικά γιατί καλύπτουν σχεδόν στο σύνολο τους τις ανάγκες του συστήματος RES-ADMIN όπως διατυπώθηκαν παραπάνω. Σε κάθε υποσύστημα υπάρχουν βέβαια πολύ περισσότερες παράμετροι που πρέπει να ληφθούν υπόψη και για αυτό θα γίνει ξεχωριστή αναφορά σε καθένα από αυτά, καθώς και το πως ανταποκρίνεται η Java σε αυτές στα αντίστοιχα κεφάλαια.

Όπως θα έχει γίνει είδη αντιληπτό από το πρώτο Κεφάλαιο (Σχήμα 7) το σύστημα RES-ADMIN αποτελεί μία κατ'εξοχήν Client / Server εφαρμογή, η οποία αναλαμβάνει να προωθήσει τα λαμβανόμενα από τους σταθμούς δεδομένα στους Web browsers που χρησιμοποιούν οι χρήστες (push-pull). Η ανάγκη για υποστήριξη μεγάλου αριθμού σταθμών που θα αποστέλλουν δεδομένα αλλά και απεριόριστου αριθμού χρηστών που θα τα παρακολουθούν είναι προφανές ότι οδηγεί στην κατασκευή ενός αρκετά πολύπλοκου συστήματος. Για το λόγο αυτό κρίθηκε απαραίτητο να εξεταστούν όλες οι τεχνικές που χρησιμοποιούνται για την υλοποίηση τέτοιων συστημάτων. Οι τεχνικές αυτές είναι οι εξής

- Χρησιμοποίηση αρχιτεκτονικής COM, DCOM, DNA της Microsoft για την επικοινωνία του προγράμματος συλλογής και μετάδοσης δεδομένων από τους σταθμούς (Collector) στο πρόγραμμα λήψης διαχείρισης και δρομολόγησης δεδομένων (RES-ADMIN Server), καθώς και την τεχνολογία ActiveX για την παρουσίαση των δεδομένων στον browser. Η υλοποίηση αυτή παρουσίαζε πολλά κενά αφού απαιτούσε όλα τα μηχανήματα να χρησιμοποιούν λειτουργικό σύστημα Windows 95/98/NT4.0/2000 κάτι το οποίο είναι μη αποδεκτό, αφού με τρόπο αυτό θα αποκλείονταν όλα τα σύστημα που χρησιμοποιούν λειτουργικό σύστημα διαφορετικό από αυτά της Microsoft. Επιπρόσθετα το μηχάνημα στο οποίο εκτελείται ο RES-ADMIN Server θα πρέπει να είναι απαραίτητα ο Web Server, για λόγους που θα αναλυθούν παρακάτω, και αυτό δημιουργεί μεγάλο πρόβλημα γιατί ένας πολύ μεγάλος αριθμός Web Servers υποστηρίζεται από UNIX \ LINUX συστήματα που

χρησιμοποιούν το λογισμικό Apache. Είναι αυτονόητο ότι τέτοια συστήματα δεν θα μπορούσαν να φιλοξενήσουν τον RES-ADMIN Server. Η παρουσίαση των δεδομένων μέσω ActiveX θα εμφάνιζε ακόμα μεγαλύτερα προβλήματα αφού εκτός του ότι ισχύουν ακριβώς οι ίδιοι περιορισμοί για το λειτουργικό σύστημα αποκλείοντας ένα μεγάλο κομμάτι χρηστών από την εφαρμογή RES-ADMIN δεν υποστηρίζεται παρά μόνο ο browser Internet Explorer της Microsoft. Η μη υποστήριξη του ιδιαίτερα διαδεδομένου Netscape Communicator συρρικνώνει ακόμα περισσότερο το κομμάτι των χρηστών που τελικά είναι σε θέση να παρακολουθήσουν την εφαρμογή RES-ADMIN. Τέλος όσον αφορά την τεχνολογία ActiveX θα πρέπει να αναφερθεί ότι παρουσιάζει σημαντικότερες ελλείψεις στον τομέα της ασφάλειας. Η εκτέλεση ActiveX controls στο σύστημα του χρήστη δεν του παρέχει καμία εγγύηση ότι δεν θα εκτελεστεί επιβλαβές πρόγραμμα στον υπολογιστή του.

- Χρήση της γλώσσας προγραμματισμού Java. Η γλώσσα προγραμματισμού Java προσφέρει εξαιρετικές δυνατότητες για την ανάπτυξη κατακεντρωμένων εφαρμογών. Ωστόσο η κατασκευή και διαχείριση σύνθετων αντικείμενων σε ένα κατακεντρωμένο σύστημα μπορεί να αποβεί ιδιαίτερα πολύπλοκη. Θα γίνει εκτενής αναφορά παρακάτω.
- Τεχνολογία RMI. Η τεχνολογία RMI (Remote Method Invocation), είναι μια ιδιαίτερα αποτελεσματική τεχνική η οποία επιτρέπει σε συναρτήσεις γραμμένες σε κάποιον κεντροποιημένο Server να εκτελούνται σε κάποια απομακρυσμένο Client. Ωστόσο η τεχνική αυτή εμφανίζει κάποιες δυσκολίες στην εγκατάσταση της και θεωρήθηκε ότι θα ήταν κάπως δύσκολο να λειτουργήσει απροβλημάτιστα σε ένα τόσο δυναμικό σύστημα όπως αυτό του RES-ADMIN όπου ελάχιστα στοιχεία είναι εκ των προτέρων γνωστά. Ενώ η εγκατάσταση και ρύθμιση του RMI σε κάθε νέο σταθμό που αναμένεται να συνδεθεί και η πραγματοποίηση των ανάλογων ρυθμίσεων στον Server πιθανόν να προκαλούσαν αρκετές δυσλειτουργίες.



Σχήμα 9 To Visual Cafe

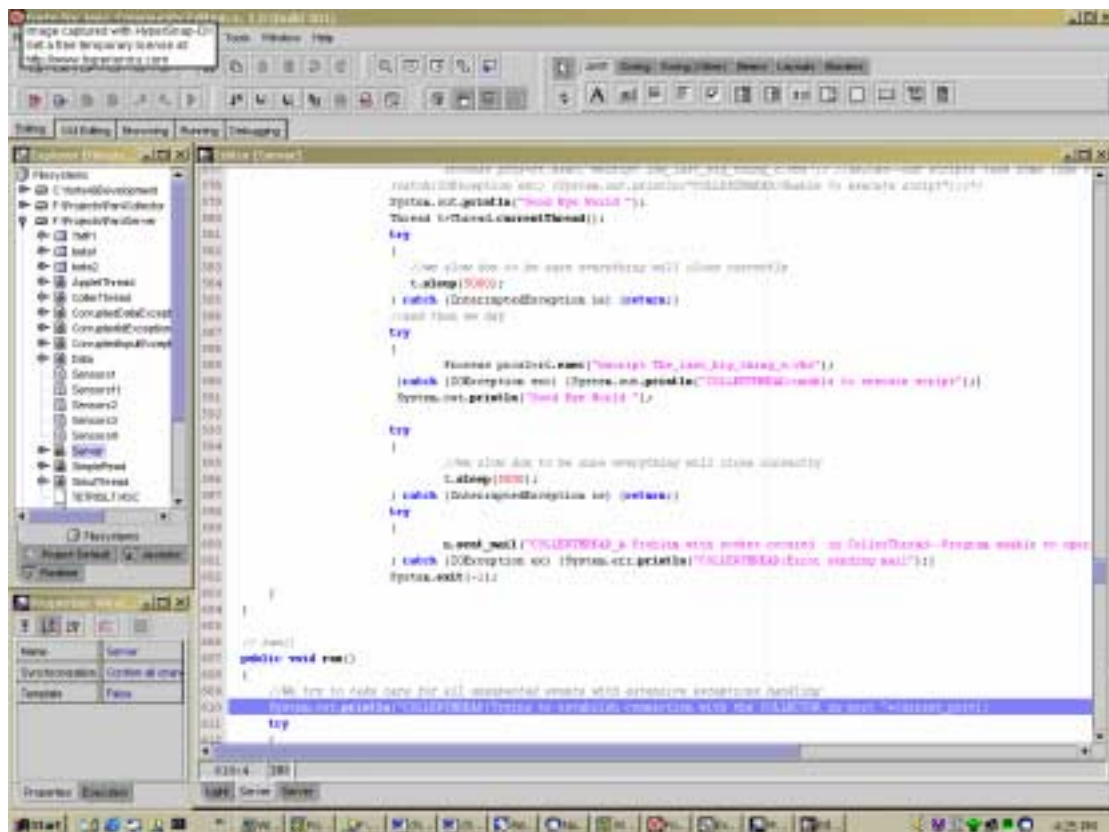
- Τεχνολογία CORBA. Η CORBA (Common Object Request Broker Architecture), αποτελεί μία από τις καλύτερες δυνατές λύσεις για διαχείριση αντικείμενων σε κατανεμημένα συστήματα. Υποστηρίζει αρκετές γλώσσες προγραμματισμού συμπεριλαμβανομένων των πιο γνωστών (C, C++, Java, Lisp, Python, Smalltalk, Ada και IDLscript). Παρέχει την δυνατότητα να χρησιμοποιούνται κοινά αντικείμενα και από εφαρμογές οι οποίες είναι γραμμένες σε διαφορετικές γλώσσες προγραμματισμού. Ωστόσο για να καταστεί αυτό εφικτό απαιτείται αρκετή προεργασία. Συνοπτικά θα μπορούσε να αναφερθεί ότι η CORBA είναι περισσότερο χρήσιμη σε εφαρμογές πολύ μεγαλύτερης κλίμακας από αυτήν του RES-ADMIN.

Από τις παραπάνω τεχνολογίες προτιμήθηκε η χρησιμοποίηση αποκλειστικά της Java, ώστε να εξασφαλιστεί η πλήρης συμβατότητα με όλα τα συστήματα. Επιπλέον καθίσταται δυνατή η προσθήκη νέων σταθμών και

γενικότερα η τροποποίηση της εφαρμογής με πολύ πιο εύκολο τρόπο από ότι με την χρήση των υπολοίπων τεχνικών.

2.2 Επιλογή εργαλείων ανάπτυξης της εφαρμογής

Η εφαρμογή RES-ADMIN αναπτύχθηκε σε δύο κυρίως εργαλεία. Για την κατασκευή του RES-ADMIN Applet, χρησιμοποιήθηκε το Visual Café της Webgain (Σχήμα 9). Το πρόγραμμα αυτό προσφέρει εξαιρετες δυνατότητες δημιουργίας γραφικών και βοήθησε τα μέγιστα στο να κατασκευαστεί το επιθυμητό γραφικό περιβάλλον για το RES-ADMIN Applet. Επιπλέον ελήφθη υπόψη ο ιδιαίτερα καλός debugger που διαθέτει.



Σχήμα 10 Το εργαλείο ανάπτυξης εφαρμογών Forte

Το μεγαλύτερο μέρος του κώδικα αναπτύχθηκε χρησιμοποιώντας το JDK (Java Development Kit) της Sun, το οποίο δεν διαθέτει κάποιο γραφικό περιβάλλον. Ωστόσο στις περιπτώσεις που χρειάστηκε να υπάρχει εποπτική παρακολούθηση του προγράμματος, δηλαδή γραφική απεικόνιση των

κλάσεων και των συναρτήσεων που χρησιμοποιήθηκαν αυτή πραγματοποιήθηκε με το Forte for Java –Community Edition (Σχήμα 10), το οποίο διαθέτει εξαιρετικό περιβάλλον και μπορεί να συνεργαστεί άψογα με όλες τις επίσημες εκδόσεις της Java, δίνοντας την δυνατότητα στον προγραμματιστή να ελέγξει τη συμβατότητα του κώδικά του.

2.3 Πλατφόρμα ανάπτυξης της εφαρμογής

Το σύστημα αναπτύχθηκε σε περιβάλλον Windows 98/ NT 4.0 / 2000. Ο Web Server είναι ο IIS της Microsoft και υποστηρίζεται από τα Windows NT 4.0 Server της Microsoft.

Οι ελάχιστες απαιτήσεις του συστήματος δεν κατέστη δυνατόν να εντοπιστούν, ωστόσο οποιοσδήποτε υπολογιστής τύπου Pentium με 16 MB RAM είναι σε θέση να εκτελέσει την εφαρμογή RES-ADMIN. Θα πρέπει ωστόσο να σημειωθεί ότι η ανάπτυξη του συστήματος RES-ADMIN, λόγω των απαιτητικών εργαλείων, απαιτεί πολύ ισχυρότερους επεξεργαστές (τουλάχιστον Celeron/Pentium II ή AMD Athlon) με 32 MB RAM τουλάχιστον.

Οι δοκιμές που πραγματοποιήθηκαν σε διαφορετικά λειτουργικά συστήματα από αυτά στα οποία αναπτύχθηκε η εφαρμογή, όπως το Linux δεν έδειξαν κάποια δυσλειτουργία.

Κεφάλαιο 3

Το Πρόγραμμα Συλλογής και Μετάδοσης Δεδομένων

3.1 Σύντομη περιγραφή

Το πρόγραμμα συλλογής και μετάδοσης δεδομένων είναι υπεύθυνο για την ορθή λήψη των δεδομένων και για την αξιόπιστη περαιτέρω μετάδοσή τους.

3.2 Γενικά χαρακτηριστικά

Σε κάθε καταναμεμημένο σύστημα επικοινωνίας ιδιαίτερα σημαντικό είναι το τμήμα που αναλαμβάνει την εισαγωγή των δεδομένων στο σύστημα. Οι επιλογές που θα γίνουν στο σχεδιασμό του τμήματος αυτού είναι καθοριστικές για την αξιοπιστία και την ορθή λειτουργία του συστήματος.

Κατά τον σχεδιασμό του προγράμματος συλλογής και μετάδοσης δεδομένων οι επιλογές έγιναν έχοντας ως στόχο την διασφάλιση της αξιοπιστίας του συστήματος. Θεωρήθηκε επίσης ότι είναι ιδιαίτερα σημαντικό να προσδιοριστούν επακριβώς οι συνθήκες κάτω από τις οποίες θα λειτουργήσει η εφαρμογή ούτως ώστε να ανταποκριθεί με τον καλύτερο δυνατό τρόπο σε αυτές.

Προέκυψε λοιπόν ότι το πρόγραμμα συλλογής και μετάδοσης δεδομένων (ή αλλιώς **Collector**) σε πολλές περιπτώσεις είναι δυνατόν να βρίσκεται σε γεωγραφικά απομακρυσμένες περιοχές σε σχέση με το πρόγραμμα λήψης, διαχείρισης και δρομολόγησης δεδομένων (ή αλλιώς **RES-ADMIN Server**). Για το λόγο αυτό είναι ιδιαίτερα σημαντικό να έχει την μέγιστη δυνατή αυτονομία και να μην εξαρτάται απόλυτα από τον RES-ADMIN Server. Ο Collector θα πρέπει επίσης να είναι σε θέση να δέχεται τα δεδομένα από ένα δοκιμασμένο και αξιόπιστο μέσο εισαγωγής δεδομένων, το οποίο να είναι ευρέως διαδεδομένο και προσιτό. Επίσης επειδή οι πληροφορίες που λαμβάνει το σύστημα μπορεί να προέρχονται από θορυβώδεις περιβάλλον είναι απαραίτητο να θωρακισθεί από τον θόρυβο στον μέγιστο δυνατό βαθμό. Σημαντική παράμετρο αποτελεί η δυνατότητα του συστήματος να μπορεί να ανακάμπτει στην περίπτωση που προκύψει κάποια μη αναμενόμενη κατάσταση. Τέλος θεωρήθηκε βασικό το λογισμικό που αναπτύχθηκε να είναι πλήρως ανεξάρτητο τόσο από το λειτουργικό σύστημα όσο και από τον επεξεργαστή που αυτό χρησιμοποιεί. Τα παραπάνω οδήγησαν στις εξής σχεδιαστικές αποφάσεις.

3.3 Τεχνικά χαρακτηριστικά του υποσυστήματος

Όπως αναφέρθηκε νωρίτερα καθοριστικής σημασίας ήταν η επιλογή μιας γλώσσας προγραμματισμού που θα επέτρεπε την χρήση σε πολλά και διαφορετικά περιβάλλοντα χωρίς την ανάγκη για επανεγγραφή του κώδικα ή έστω μερική μετατροπή του, για να υποστηρίζει διαφορετικά συστήματα (porting). Δεδομένης της δυνατότητας του συστήματος να υποστηρίζει την ταυτόχρονη σύνδεση πολλών Collector με ένα Server καθίσταται αυτονόητο ότι το λογισμικό του Collector θα τρέχει σε πολλά διαφορετικά μηχανήματα με διαφορετικό λειτουργικό σύστημα αλλά και υλικό (Hardware). Η ανάγκη μετατροπής του κώδικα ώστε να καταστεί συμβατός με κάθε πιθανή πλατφόρμα θα προκαλούσε ουσιαστικά την αχρήστευση του. Η επιλογή λοιπόν της Java ως γλώσσας προγραμματισμού αποτέλεσε σχεδιαστικό μονόδρομο, δεδομένης της δυνατότητας της να εγγυάται ότι ο κώδικας εκτελείται παντού χωρίς στις περισσότερες περιπτώσεις να απαιτείται ούτε επαναμεταγλώττιση (recompiling). Η Java είναι η μοναδική γλώσσα προγραμματισμού που είναι ανεξάρτητη πλατφόρμας (Platform Independent) και αυτό συντέλεσε αποφασιστικά στην υιοθέτηση της ως εργαλείο ανάπτυξης της εφαρμογής Collector.

Ένα από τα πλέον σημαντικά προβλήματα που έπρεπε να αντιμετωπιστεί ήταν αυτό της απομακρυσμένης γεωγραφικά ή τεχνικά δύσκολα προσβάσιμης περιοχής. Για το λόγο αυτό επιλέχθηκε ο Collector να είναι σε θέση να λειτουργήσει και χωρίς να είναι συνδεδεμένος με τον Server. Συγκεκριμένα ο Collector είναι σε θέση να αναμείνει όσο χρειαστεί προκειμένου να συνδεθεί ο Server και μόλις συμβεί αυτό να ξεκινήσει άμεσα την μετάδοση των δεδομένων. Για την υλοποίηση αυτής της δομής έγινε χρήση των κλάσεων Socket και ServerSocket της Java (θα ακολουθήσει εκτεταμένη αναφορά σε αυτές παρακάτω). Η σύνδεση των Collectors με τον Server για την μεταφορά των δεδομένων πραγματοποιείται με το ευρύτατα διαδεδομένο, σταθερό και πρακτικό πρωτόκολλο TCP/IP. Οι δυνατότητες που προσφέρει αυτή η επιλογή είναι απεριόριστες, αφού επιτρέπει την χρήση της υπάρχουσας υποδομής του δικτύου στο οποίο λειτουργεί η εφαρμογή είτε αυτό είναι κάποιο τοπικό δίκτυο π.χ. το Ethernet είτε είναι το Internet. Έτσι

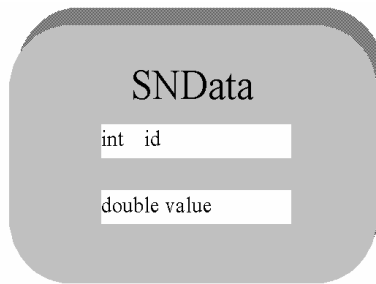
λοιπόν αξιοποιείται ένα ιδιαίτερα αξιόπιστο πρωτόκολλο χωρίς να απαιτείται η περαιτέρω υλοποίηση λογισμικού ή υλικού, ενώ επιπλέον παρέχεται η δυνατότητα μέσω του Internet να εκμηδενισθούν πλήρως οι αποστάσεις καθιστώντας εφικτή την σύνδεση απομακρυσμένων Collectors και Server, από διαφορετικές πόλεις ή ακόμα και από διαφορετικές χώρες. Επιπλέον δεν θα μπορούσε να παραβλεφθεί ότι οι μοντέρνες αρχιτεκτονικές δόμησης συστημάτων λογισμικού έχουν ως βασικό στόχο την ενσωμάτωση και χρήση του Internet. Ακολουθώντας λοιπόν τα σύγχρονα πρότυπα ανάπτυξης εφαρμογών υλοποιήθηκε μια καθαρά δικτυακή εφαρμογή η οποία εξασφαλίζει στον μηχανικό που θα δουλεύει με αυτή ότι δεν θα είναι υποχρεωμένος να γνωρίζει τον τρόπο εκτέλεσης του κάθε υποσυστήματος.

Επόμενο σημείο το οποίο θα έπρεπε να αναφερθεί είναι το μέσο εισόδου. Υπάρχει η ανάγκη για μια αξιόπιστη και δοκιμασμένη συσκευή, η οποία θα είναι και προσιτή οικονομικά αλλά και καθολικά διαθέσιμη. Μια (αν όχι η μοναδική) συσκευή που πληροί αυτές τις προδιαγραφές είναι η σειριακή θύρα RS-232. Τα πλεονέκτημα της είναι ότι υπάρχει σε όλους σχεδόν τους υπολογιστές και συνεπώς δεν τίθεται θέμα διάδοσης, ενώ το κόστος της είναι εξαιρετικά χαμηλό. Επιπλέον είναι από τις λίγες συσκευές που έχει χρησιμοποιηθεί και δοκιμαστεί τόσο εκτεταμένα. Το κυριότερο μειονεκτήματα που αντιμετωπίζει είναι η χαμηλή ταχύτητα μετάδοσης που παρέχει. Στην περίπτωση του προγράμματος συλλογής και μετάδοσης δεδομένων τέτοιο πρόβλημα δεν προέκυψε, αφού χρησιμοποιήθηκαν ταχύτητες μετάδοσης πολύ πιο κάτω από τα όρια της σειριακής θύρας. Βάση αυτών αποφασίστηκε η είσοδος των δεδομένων να πραγματοποιηθεί από την RS-232. Η αδυναμία της Sun να παράσχει κάποια βιβλιοθήκη στην standard έκδοση της Java προκειμένου να καταστεί δυνατή η πρόσβαση στην σειριακή θύρα επιλύθηκε κάνοντας χρήση κάποιων εξωτερικών επεκτατικών (extended) βιβλιοθηκών της Sun για την Java (javax.com extension package). Το μειονέκτημα αυτής της προσέγγισης είναι ότι η βιβλιοθήκη αυτή δεν είναι μέρος του standard JDK (Java Development Kit) και ως εκ τούτου δεν είναι πλήρως τεκμηριωμένη, αλλά ούτε έχει πιστοποιηθεί ότι συνεργάζεται απρόσκοπτα με όλες τις τελευταίες εκδόσεις της Java. Σε κάθε περίπτωση πραγματοποιήθηκαν εκτεταμένες δοκιμές της σειριακής θύρας μέσω ειδικού λογισμικού που

κατασκευάστηκε για τον σκοπό αυτό και δεν διαπιστώθηκε η ύπαρξη κανενός προβλήματος. Επιπλέον θα πρέπει να προστεθεί ότι και η ίδια η Sun σε εκτεταμένα Test που έκανε σε beta εκδόσεις της Java2 1.3 δεν παρατήρησε καμία ασυμβατότητα ή αστάθεια και είναι πολύ πιθανόν να έχει ήδη λάβει το πακέτο αυτό την επίσημη πιστοποίηση της Java και για τις τελευταίες εκδόσεις. Θα περιγράψει αναλυτικότερα η λειτουργία της βιβλιοθήκης αυτής παρακάτω.

Μεγίστης σημασίας είναι η κατασκευή ενός συστήματος ανθεκτικού στο θόρυβο. Οι συνθήκες που αντιμετωπίστηκαν κατά τη λειτουργία του συστήματος στο Πολυτεχνείο ήταν αρκετά δύσκολες. Αν και η συχνότητα της ασύρματου επικοινωνίας είναι στα 433 MHz αντιμετωπίστηκαν πολλές περιπτώσεις ισχυρού ηλεκτρομαγνητικού θορύβου από διαφορετικές πηγές. Για παράδειγμα εντονότατο θόρυβο στο σύστημα εισήγαγε η χρήση ασύρματης ενδοσυνεννόησης, ενώ δεν μπορεί να αποκλεισθεί και το ενδεχόμενο ο θόρυβος να προέρχεται από κινητά τηλέφωνα, από συναγερμούς αυτοκινήτων ή ακόμα και από τηλεοπτικές και ραδιοφωνικές εκπομπές. Τα προβλήματα που προκάλεσε ο θόρυβος ήταν οξύτατα και για να αντιμετωπιστούν αποτελεσματικά χρειάστηκε να ταξινομηθούν σε δυο κατηγορίες.

Στην πρώτη κατηγορία κατατάσσεται ο θόρυβος που προκαλεί αλλοιώσεις στα bits που λαμβάνονται από το υποσύστημα. Σε αυτή την περίπτωση είναι δυνατόν να προκύψουν από λανθασμένες μετρήσεις έως ασυνάρτητοι χαρακτήρες. Το πρόβλημα αυτό αντιμετωπίστηκε λαμβάνοντας τα δεδομένα τρεις φορές από το ασύρματο σύστημα (redundancy), και παράλληλα παρακολουθώντας τη μέθοδο, για την λήψη δεδομένων, συνέχεια μέσω εκτεταμένου Exception Handling προκειμένου να καταστεί δυνατός ο εντοπισμός οποιοδήποτε σφάλματος. Συγκεκριμένα αν κάποιο δεδομένο δεν περιέχει έγκυρες τιμές αλλά ασυνάρτητους χαρακτήρες αμέσως απορρίπτεται και ακολουθεί η λήψη και σύγκριση των άλλων δύο δεδομένων. Σε περίπτωση που και άλλο δεδομένο παρουσιάσει σφάλμα θεωρείται εξ ολοκλήρου χαμένο. Εννοείται ότι αν δεν προκύψει σφάλμα μη έγκυρου χαρακτήρα, τότε συγκρίνεται το δεδομένο που έχει ληφθεί τρεις φορές και αν δυο από τις τιμές του συμφωνούν τότε γίνεται αποδεκτό.



Σχήμα 11 Η κλάση *SNData*

Φυσικά επειδή το δεδομένο μας αποτελείται από δύο πεδία, το ID δηλαδή το αναγνωριστικό του κάθε δεδομένου καθώς και το πεδίο με την τιμή του δεδομένου (Σχήμα 11), σε περίπτωση που προκύψει μη αναστρέψιμο σφάλμα (δηλαδή χαθούν δύο ή παραπάνω τιμές του αναγνωριστικού ID) απορρίπτεται και η αντίστοιχη τιμή του δεδομένου. Γίνεται κατανοητό ότι για να γίνει αποδέκτη λανθασμένη τιμή στο σύστημα θα πρέπει να συμβούν τα ακόλουθα.

- Από τους 256 ASCII χαρακτήρες που υπάρχουν μόνο 14 είναι αποδεκτοί στο υποσύστημα συλλογής και μετάδοσης δεδομένων. Συγκεκριμένα πέραν των 10 ψηφίων (0-9) είναι αποδεκτός ο χαρακτήρας (-) καθώς και ο χαρακτήρας (,) ενώ ακόμα αποδεκτοί είναι οι χαρακτήρες (+) και (.). Θα πρέπει δηλαδή ο θόρυβος να αλλοιώσει έναν έγκυρο χαρακτήρα, να τον μετατρέψει σε κάποιον άλλον από τους 12 αποδεκτούς χαρακτήρες από τους συνολικούς 256 (Ας σημειωθεί ότι από τους 14 χαρακτήρες δεν είναι όλοι πάντα αποδεκτοί αφού κανένας αριθμός δεν μπορεί για παράδειγμα να περιέχει δύο(-)) και
- Να αλλοιώσει ακριβώς τον ίδιο χαρακτήρα και να τον μετατρέψει στον ακριβώς ίδιο έγκυρο χαρακτήρα και στο επόμενο δεδομένο που ακολουθεί καθώς και
- Η τιμή που θα προκύψει να είναι έγκυρη από λογικής πλευράς αφού για παράδειγμα αρνητικό ID είναι αυτονόητο ότι δεν μπορεί να γίνει αποδεκτό.

Ο παραπάνω έλεγχος δεν κατέστη δυνατό να παρακαμφθεί παρ' ότι πραγματοποιήθηκε κάθε δυνατή προσπάθεια υποβάλλοντας το σύστημα σε ισχυρότατο θόρυβο.

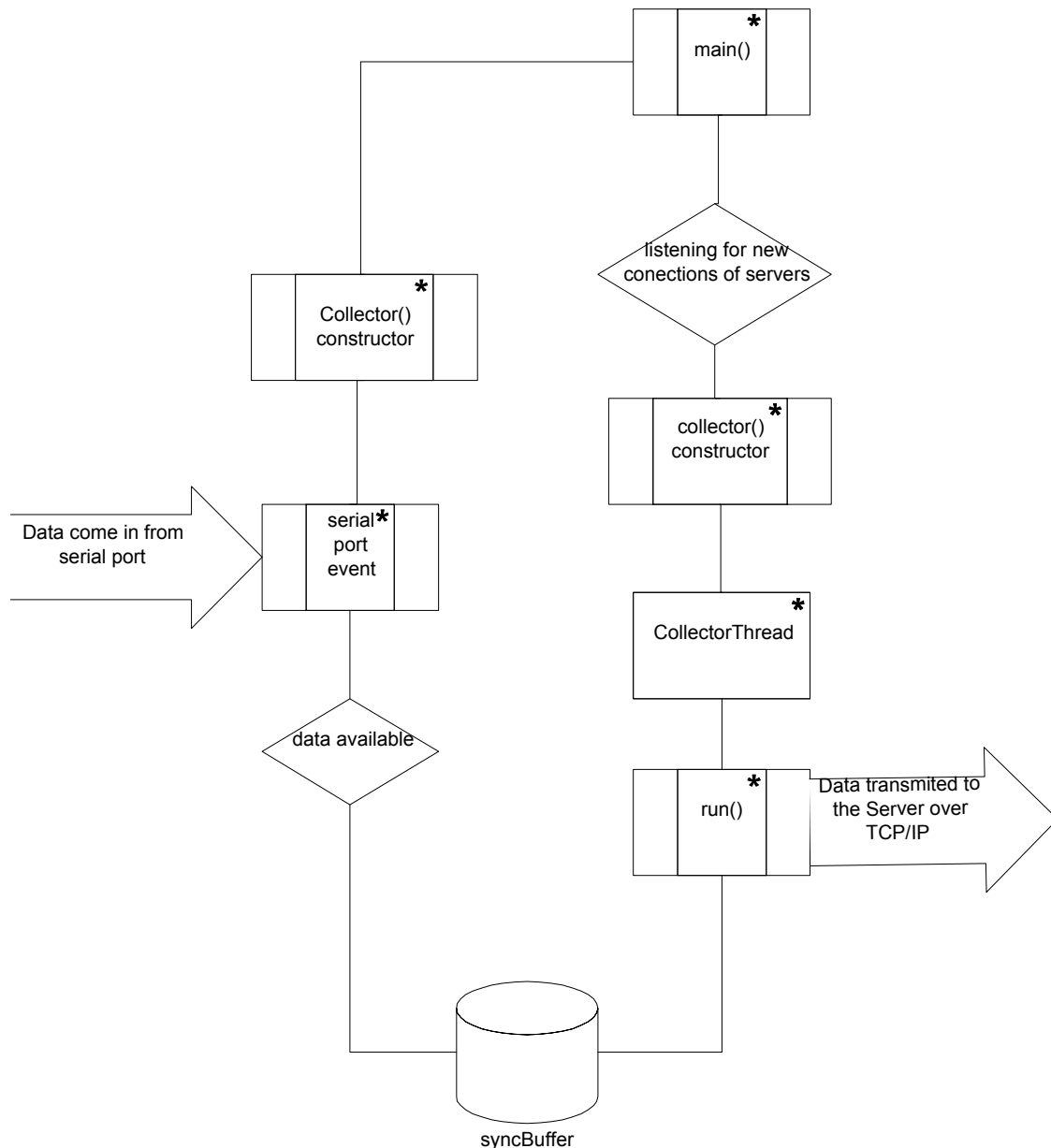
Η δεύτερη και πολύ πιο επικίνδυνη κατηγορία θορύβου έχει να κάνει με τον θόρυβο που προσθέτει bits στο bitstream που μεταδίδεται. Αυτό έχει ως αποτέλεσμα το bytearray το οποίο λειτουργεί σαν buffer για την προσωρινή αποθήκευση των δεδομένων που λαμβάνονται και είναι συγκεκριμένου μεγέθους να γεμίζει με άχρηστα bytes. Αν στην συνέχεια αφιχθούν τα κανονικά bytes των δεδομένων που αναμένονται θα αποθηκευτεί ένα κομμάτι τούς στο σωστό buffer και το υπόλοιπο κομμάτι θα αποθηκευτεί στο buffer που προορίζεται για το επόμενο δεδομένο. Αυτό θα εξακολουθήσει να συμβαίνει για όλα τα δεδομένα που ακολουθούν. Αν συνυπολογιστεί και το γεγονός ότι όπως προαναφέρθηκε το κάθε δεδομένο έχει δύο πεδία, το πεδίο για το αναγνωριστικό ID και το πεδίο για την τιμή του δεδομένου, και συνεπώς χρησιμοποιούνται διαφορετικού μήκους bytearrays για να κρατηθεί το ID από ότι για να κρατηθεί η τιμή, προκύπτει ότι θα αρκούσαν λίγα bits προσθετικού θορύβου για να θέσουν το σύστημα σε μόνιμη αστάθεια. Η κατάσταση είναι ιδιαίτερα δύσκολη αφού πρέπει να γίνει κατανοητό ότι προκειμένου να διαφυλαχθεί η δυναμικότητα του συστήματος, δεν είναι γνωστό εκ των πρότερων τι είδους δεδομένα αναμένονται (το ID τους), αλλά ούτε και η χρονική στιγμή που αυτά θα αφιχθούν.

Η λύση σε αυτή την ομολογουμένως δύσκολη περίπτωση δόθηκε με την χρήση χαρακτήρων έλεγχου (control characters). Οι χαρακτήρες αυτοί μεταδίδονται τόσο στην αρχή όσο και στο τέλος της μετάδοσης του κάθε πακέτου. (Με τον όρο πακέτο εννοείται η τριπλή μετάδοση του ID του δεδομένου καθώς και της τιμής του.) Αξιοποιώντας τους χαρακτήρες αυτούς και με την βοήθεια εκτεταμένου Exception Handling προσδιορίζεται επακριβώς αν συνέβη κάποιο σφάλμα στην μετάδοση του πακέτου, αφού οι χαρακτήρες έλεγχου δεν είναι δυνατόν να βρίσκονται μέσα σε ένα bytearray αλλά πρέπει να έχουν ληφθεί πριν την μετάδοση του πακέτου και αμέσως μετά από αυτή. Οτιδήποτε άλλο θεωρείται απαράδεκτο και οδηγεί στην απόρριψη του πακέτου. Το πρωτόκολλο αυτό θα αναλυθεί με λεπτομέρεια κατά την τεχνική ανάλυση του προγράμματος.

Τέλος ένα ιδιαίτερα σημαντικό χαρακτηριστικό που πρέπει να συγκεντρώνει ένα σύστημα που απαιτείται να λειτουργεί εικοσιτέσσερις ώρες το 24ωρο είναι η αξιοπιστία. Η ανάγκη για 24ώρη διαθεσιμότητα του συστήματος επιβάλλει την ύπαρξη δομών που θα επιτρέπουν την επαναλειτουργία του μετά από κάποιο μη αναστρέψιμο σφάλμα. Ειδικά όταν πρόκειται για μια κατανεμημένη εφαρμογή είναι εξαιρετικά δύσκολος ο ταυτόχρονος έλεγχος όλων το υποσυστημάτων αφού είναι πολύ πιθανόν να βρίσκονται σε απομακρυσμένες ή μη προσιτές περιοχές. Προκείμενου να αντιμετωπισθεί το πρόβλημα αυτό ενσωματώθηκαν τεχνικές επαναλειτουργίας του συστήματος, ούτως ώστε να αποφευχθεί μια γενική και μόνιμη κατάρρευση του, λόγω κάποιου σφάλματος, αστοχίας ή κάποιου αλλού επιμέρους προβλήματος ενός ή και παραπάνω υποσυστημάτων. Συγκεκριμένα, χρησιμοποιώντας εκτεταμένα Exception Handling γίνεται άμεσα γνωστό το σημείο που αντιμετωπίζει πρόβλημα η εφαρμογή. Προσδιορίζοντας λοιπόν το είδος του προβλήματος καθώς και την περιοχή του, πραγματοποιούνται όλες οι απαιτούμενες ενέργειες προκειμένου να επανατεθεί το σύστημα σε λειτουργία με τις ελάχιστες δυνατές απώλειες ακόμα και αν αυτό έχει υποστεί μη αναστρέψιμο σφάλμα. Για να καταστεί κάτι τέτοιο εφικτό γίνεται εκτεταμένη χρήση scripting. Έχουν υλοποιηθεί δηλαδή κατάλληλα scripts που ενεργοποιούνται όταν παρουσιαστεί πρόβλημα. Θα πρέπει να σημειωθεί βέβαια ότι σε αρκετές περιπτώσεις δεν είναι δυνατόν να προσδιοριστεί ένα σφάλμα με απόλυτη βεβαιότητα λόγω του ότι το σύστημα όπως έχει προαναφερθεί είναι κατανεμημένο, αλλά να υπάρχουν σαφείς ενδείξεις ότι κάτι τέτοιο συμβαίνει. Σε αυτήν περίπτωση και για να αποφευχθεί κάποια μόνιμη αναμονή του συστήματος (deadlock) που μπορεί να προκύψει θεωρείται προτιμότερη πρακτική η επανεκκίνηση του προγράμματος προκειμένου να αποφευχθεί ο κίνδυνος του deadlock. Συγκεκριμένα σε περίπτωση που ο Collector αδυνατεί να λάβει δεδομένα για ένα αρκετά μεγάλο χρονικό διάστημα (το οποίο ορίζεται σαν παράμετρος κατά την εκτέλεση του προγράμματος, αφού ο κάθε Collector μπορεί να δέχεται δεδομένα από διαφορετική πηγή και συνεπώς η ταχύτητα μετάδοσης της κάθε πηγής να είναι διαφορετική), διακόπτεται η λειτουργία του προγράμματος στέλνοντας το κατάλληλο σήμα έλεγχου (control signal) στον Server του και

επανεκκινώντας. Τα παραπάνω θέματα θα συζητηθούν εκτεταμένα στην τεχνική ανάλυση της εφαρμογής

3.4 Τεχνική ανάλυση



Σχήμα 12 Flowchart του προγράμματος Collector

Η τεχνική ανάλυση του προγράμματος μας βασίζεται στον αντικειμενοστραφή προγραμματισμό (object oriented programming).

Αρχικά το πρόγραμμα κάνει χρήση της παραμέτρου `implements` προκειμένου να γίνει δυνατή η `threaded` εκτέλεση του αλλά ταυτόχρονα να

αξιοποιηθεί η τεχνολογία Listener. Θα πρέπει να σημειωθεί ότι καλείται η παράμετρος implements γιατί η Java σε αντίθεση με την C++, δεν επιτρέπει την πολλαπλή κληρονομικότητα (multiply inheritance). Έτσι, προκειμένου να αξιοποιηθούν τόσο τα Threads όσο και οι Listeners των οποίων οι χρησιμότητα θα γίνει κατανοητή παρακάτω, επιλέχθηκε το implements που εξασφαλίζει το μεγαλύτερο μέρος της λειτουργικότητας του extends. Πολύ σύντομα θα αναφερθεί ότι η εντολή extends η οποία συντάσσεται ως εξής: “class A extends B”, το οποίο σημαίνει ότι η κλάση A θα κληρονομήσει όλα τα χαρακτηριστικά (μεταβλητές και μεθόδους) της κλάσης B. Αντίθετα δηλώνοντας class A implements B μας παρέχεται η δυνατότητα να χρησιμοποιήσουμε τις μεθόδους της κλάσης B υπό προϋποθέσεις. Η συγκεκριμένη τεχνική συναντάται ιδιαίτερα συχνά στην Java και στην περίπτωση που επιθυμούνται περαιτέρω λεπτομέρειες μπορεί να γίνει αναφορά σε κάποιο εγχειρίδιο της γλώσσας προγραμματισμού Java.

3.4.1 Η μέθοδος main()

Ακολουθώντας το πρόγραμμα καλεί την βασική συνάρτηση (ή μέθοδο όπως συνηθίζεται να αναφέρεται στον αντικειμενοστραφή προγραμματισμό η συνάρτηση) όλων των προγραμμάτων main(). Θα πρέπει να αναφερθεί στο σημείο αυτό ότι η συνάρτηση αυτή πρέπει να δεχτεί 4 ορίσματα που εισάγονται σαν παράμετροι κατά την εκτέλεση του προγράμματος. Τα 4 αυτά ορίσματα είναι τα εξής

- Θύρα (port) που επικοινωνεί ο Collector με τον Server. Η επιλογή της κατάλληλης θύρας είναι αρκετά σημαντική. Η πρόταση του συγγραφέα είναι να επιλέγονται θύρες μεγαλύτερες από το 1000, γιατί μέχρι αυτό τον αριθμό, οι θύρες χρησιμοποιούνται κυρίως για λειτουργίες του συστήματος (FTP, mail, echo). Ακόμα στην περίπτωση που υπάρχουν παραπάνω από ένας Collectors πρέπει οπωσδήποτε να επικοινωνούν με τον Server σε διαφορετικά ports για ευνόητους λόγους. Θα πρέπει επίσης να σημειώσουμε την σπάνια περίπτωση κατά την οποία ο Administrator του συστήματος μπορεί να αντιμετωπίσει κάποιο πρόβλημα με τα διάφορα προγράμματα καταπολέμησης ίων (anti-virus). Συγκεκριμένα η ύπαρξη μόνιμης ανοιχτής επικοινωνίας δύο υπολογιστών μέσω κάποιου

συγκεκριμένου port, μπορεί να προκαλέσει στα anti-virus την εσφαλμένη υπόθεση ότι πρόκειται για κάποιο επιβλαβές πρόγραμμα τύπου Trojan. Το πρόβλημα αυτό επιλύεται συνήθως ορίζοντας κάποιο διαφορετικό port. Αν το πρόβλημα παραμένει τότε θα πρέπει ίσως να απενεργοποιηθεί το anti-virus που χρησιμοποιείται.

- Σειριακή θύρα που χρησιμοποιεί το πρόγραμμα. Επειδή όπως έχει ήδη εκτεταμένα αναφερθεί το λογισμικό αυτό επιβάλλεται να λειτουργεί σε πολλά και ανομοιογενή περιβάλλοντα, είναι προφανές ότι δεν είναι γνωστό εκ των προτέρων σε ποια θύρα θα συνδεθεί το module της ασύρματης επικοινωνίας. Για το λόγο αυτό ο χειρίστης του μηχανήματος είναι σε θέση να επιλέξει την θύρα που επιθυμεί αυτός να συνδέσει το module.
- Ρυθμός λήψης δεδομένων από την σειριακή θύρα (baud rate). Το πρόγραμμα δίνει την δυνατότητα να επιλεγεί από τον μηχανικό που το χρησιμοποιεί το baud rate της σειριακής θύρας. Η δυνατότητα αυτή είναι πραγματικά πολύ χρήσιμη, αν γίνει κατανοητό ότι ανάλογα με το είδος των δεδομένων μπορεί να υπάρχουν εντελώς διαφορετικές ανάγκες στην ταχύτητα μετάδοσης. Δεδομένα τα οποία μεταβάλλονται διαρκώς και είναι ιδιαίτερα σημαντική η άμεση λήψη και επεξεργασία τους μπορούν να μεταδοθούν και να τα ληφθούν με πολύ υψηλότερη ταχύτητα από αλλά δεδομένα που μεταβάλλονται λιγότερο γρήγορα ή δεν είναι τόσο σημαντικά για την συγκεκριμένη μελέτη. Θα πρέπει να τονιστεί ότι σε κάθε περίπτωση πρέπει η ταχύτητα μετάδοσης δεδομένων από το σύστημα μετάδοσης να συμπίπτει με την ταχύτητα λήψης δεδομένων που έχουμε ορίσει. Σε αντίθετη περίπτωση είναι σχεδόν βέβαιο, ότι τα δεδομένα που θα λαμβάνει το σύστημα θα είναι παντελώς άχρηστα. Ακόμα θα πρέπει να τονιστεί ότι το λογισμικό έχει πιστοποιηθεί για να λειτουργεί κάτω από συγκεκριμένες κοινά αποδεκτές ταχύτητες, όπως τα 300, 600, 1200, 2400, 4800bps και σε καμία περίπτωση δεν είναι δυνατόν να εγγυηθεί ο κατασκευαστής ότι το σύστημα θα λειτουργεί με ταχύτητες εκτός των προδιαγραφών.
- Χρόνος ανενεργούς λειτουργίας. Σε περίπτωση που ο Collector δεν καταφέρει για κάποιο χρονικό διάστημα να λάβει δεδομένα τότε αυτόματα

επανεκκινείται. Είναι προφανές ότι επειδή τα συστήματα συλλογής δεδομένων μπορεί να βρίσκονται σε απομακρυσμένες ή δυσπρόσιτες περιοχές είναι πιθανόν να προκύψει κάποια μη αναμενόμενη κατάσταση βάζοντας το πρόγραμμα σε κατάσταση μόνιμης αναμονής. Για το λόγο αυτό μετά την πάροδο του χρονικού διαστήματος το οποίο έχει οριστεί από τον μηχανικό που το χρησιμοποιεί, το πρόγραμμα επαναεκκινείται προσπαθώντας να αποκαταστήσει την επικοινωνία.

Μετά την λήψη και επεξεργασία των ορισμάτων, και αφού γίνουν οι απαραίτητες αρχικοποιήσεις των μεταβλητών δημιουργείται η βασική δομή που θα χρησιμοποιηθεί, για να αποθηκευτούν και να μεταδοθούν τα δεδομένα αργότερα. Η δομή αυτή ονομάζεται `syncBuffer` και είναι ένα αντικείμενο (Object) τύπου `Vector`. Το `Vector` συμπεριφέρεται σαν μια δυναμική λίστα, όπου είναι δυνατόν να προστεθούν και να αφαιρεθούν τα στοιχεία δυναμικά. Δηλαδή το μέγεθος του `Vector` αλλάζει *on the fly*, ενώ η διαχείριση της μνήμης γίνεται αυτόματα από τον `Java Virtual Machine (JVM)` χωρίς να υπάρχει η ανάγκη δέσμευσης και αποδέσμευσης μνήμης ανάλογα με την λειτουργία που εκτελείται. Επιπλέον ιδιαίτερη σημασία είναι το γεγονός ότι το αντικείμενο αυτό έρχεται με πλήθος ετοιμών συναρτήσεων, που επιτρέπουν την αντικατάσταση ενός κόμβου του `Vector` από έναν άλλο, ενώ παρέχονται και οι απαιτούμενες πληροφορίες τόσο για το μέγεθος του όσο και για τα υπόλοιπα χαρακτηριστικά του. Το `syncBuffer` το οποίο χρησιμοποιείται είναι λοιπόν ένα αντικείμενο τύπου `Vector` το οποίο σε κάθε θέση του φιλοξενεί αντικείμενα τύπου `SNData` (Σχήμα 11). Η κλάση αυτή δημιουργήθηκε προκειμένου να αναπαραστήσει τα δεδομένα τα οποία διαχειρίζεται και περιλαμβάνει δύο πεδία. Το πεδίο για το αναγνωριστικό του κάθε δεδομένου (`ID`) καθώς και το πεδίο για την τιμή του δεδομένου (`data`). Η ονομασία της κλάσης αυτής συνιστάται να είναι διαφορετική από την ονομασία που αναπαριστά τα δεδομένα στο πρόγραμμα `Server`. Σε περίπτωση που δύο κλάσεις εμφανίζονται με το ίδιο όνομα είναι δυνατόν να προκύψουν δυσλειτουργίες ακόμα και αν η κάθε κλάση χρησιμοποιείται από ένα μόνο πρόγραμμα και βρίσκεται και σε διαφορετικό `directory` του δίσκου. Το πρόβλημα μπορεί να παρουσιαστεί μόνο στην περίπτωση που τόσο ο `Collector` όσο και ο `Server`

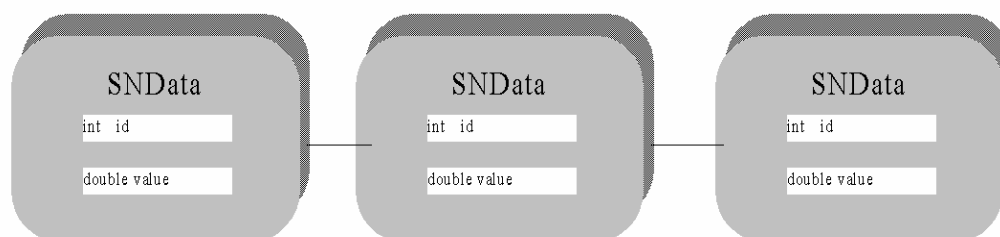
ΕΚΤΕΛΟΥΝΤΑΙ

ΣΤΟΝ

ΙΔΙΟ

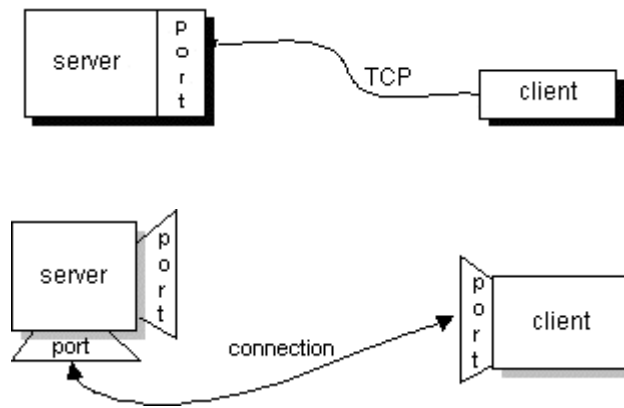
ΥΠΟΛΟΓΙΣΤΗ.

Vector syncBuffer



Σχήμα 13 Το syncBuffer που χρησιμοποιείται στο πρόγραμμα

Στην συνάρτηση αυτή καλείται ο constructor που διαχειρίζεται την σειριακή θύρα καθώς και ο constructor που υλοποιεί την επικοινωνία με τον Server. Οι λειτουργίες των constructors θα περιγραφεί εκτεταμένα παρακάτω, αλλά θα ήταν χρήσιμο να γίνει αναφορά στο σημείο αυτό στα sockets τα οποία χρησιμοποιούνται ευρύτατα σε όλο το σύστημα προκειμένου να καταστεί εφικτή η επικοινωνία των υποπρογραμμάτων. Τα sockets αρχικά ξεκίνησαν ως μια δομή που επέτρεπε την επικοινωνία δύο διεργασιών (processes) σε ένα σύστημα UNIX αργότερα όμως έγινε αντιληπτό ότι και με την βοήθεια των πρωτοκόλλων TCP και IP είναι δυνατόν να επιτευχθούν συνδέσεις μεταξύ διαφορετικών συστημάτων. Η υλοποίηση των sockets στην Java πραγματοποιείται με την χρήση δύο κλάσεων, της κλάσης `ServerSocket` και της κλάσης `ClientSocket` ή απλά `Socket` (Σχήμα 14). Το `Socket` θα μπορούσε να θεωρηθεί συνοπτικά σαν το άκρο της επικοινωνίας δύο μηχανημάτων. Ο τρόπος λειτουργίας είναι αρκετά απλός και προϋποθέτει την ύπαρξη ενός `ServerSocket` στο ένα άκρο της επικοινωνίας, και την ύπαρξη απλών `Sockets` στο άλλο άκρο. Το `ServerSocket` αναμένει να συνδεθεί σε αυτό ένα `client Socket`.



Σχήμα 14 *serverSocket-clientSocket*

Στη μέθοδο `main` ο `Collector` δημιουργεί ένα `ServerSocket` και αναμένει από τον `Server` να συνδεθεί σε αυτόν. Όπως έχει αναφερθεί και παραπάνω ο λόγος που προτιμήθηκε η υλοποίηση αυτή είναι για να εξασφαλιστεί η μέγιστη δυνατή αυτονομία του `Collector` αφού είναι και το δυσκολότερα προσβάσιμο υποσύστημα. Όταν επιτευχθεί η σύνδεση με τον `Server` τότε καλείται ο `constructor` που είναι υπεύθυνος για την μετάδοση δεδομένων στον `Server` και ξεκινάει η `threaded` λειτουργία του.

Στην περίπτωση που προκύψει σφάλμα κατά την σύνδεση του `Server` τότε αυτόματα καλείται η συνάρτηση `IOException` η οποία όπως αναφέρεται αναλυτικά και παρακάτω παύει την λειτουργία του `Collector` και τον επανεκκινεί προκειμένου να πραγματοποιηθεί εκ νέου απόπειρα συνδέσεως των δύο υποσυστημάτων.

Σε κάθε περίπτωση και καθ όλη την διάρκεια εκτελέσεως του προγράμματος εμφανίζονται μηνύματα που ενημερώνουν τον διαχειριστή για την κατάσταση της εφαρμογής.

3.4.2 Ο `Constructor` για τον `Collector` που διαχειρίζεται την σειριακή θύρα

Εφ' όσον δημιουργηθεί το `syncBuffer` καλείται ο `constructor Collector()` ο οποίος είναι υπεύθυνος για την δημιουργία της κλάσης του `Collector` που θα αναλάβει τον χειρισμό της σειριακής θύρας. Στο σημείο αυτό η ροή του προγράμματος θα μεταφερθεί στον `constructor Collector()`, ο οποίος θα αναλάβει άμεσα να ελέγξει αν η σειριακή θύρα που ζητείται είναι ελεύθερη. Σε

περίπτωση που η σειριακή θύρα είναι κατειλημμένη, θα προσπαθήσει να εντοπίσει την εφαρμογή που κάνει χρήση της θύρας εάν αυτό είναι δυνατό, και βέβαια το πρόγραμμα θα τερματιστεί αφού χωρίς την σειριακή θύρα δεν θα υπάρχει τρόπος εισαγωγής δεδομένων στο σύστημα. Αν αντίθετα η θύρα είναι ελεύθερη, τότε αυτόματα ο Collector δεσμεύει την θύρα και ενημερώνει το λειτουργικό για αυτό, ώστε σε περίπτωση που κάποια άλλη εφαρμογή επιχειρήσει να αποκτήσει πρόσβαση στην σειριακή θύρα που χρησιμοποιείται, να ενημερωθεί αμέσως ότι είναι δεσμευμένη από την εφαρμογή Collector. Στην συνέχεια αφού έχουν ολοκληρωθεί οι προετοιμασίες για τον έλεγχο της σειριακής θύρας, αναλαμβάνει την παρακολούθηση της ροής του προγράμματος ο `serialPort.EventListener`.

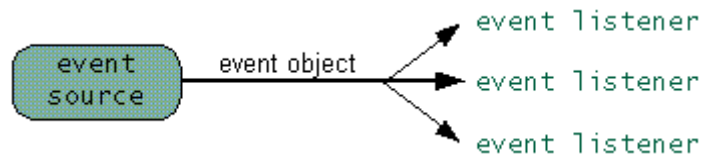
Η τεχνική των Listeners είναι μια αρκετά πολύπλοκη τεχνική και η εκτεταμένη ανάλυση της πιθανότητα ξεφεύγει από τους σκοπούς του συγγράμματος αυτού, για τον λόγο αυτό θα παρατεθούν κάποια γενικά χαρακτηριστικά που θα βοηθήσουν στην σφαιρική κατανόηση του προγράμματος. Θα πρέπει να γίνει άμεσα κατανοητό ότι το πρόγραμμα συλλογής και μετάδοσης δεδομένων είναι ένα μη σειριακό πρόγραμμα. Σε ένα σειριακό πρόγραμμα η ροή του προγράμματος είναι φυσικά σειριακή καθιστώντας απόλυτα προβλέψιμη την εκτέλεση του. Αντίθετα στην εφαρμογή Collector πολλά πράγματα συμβαίνουν ασύγχρονα, δηλαδή σε μη προβλέψιμες και άγνωστες χρονικές στιγμές. Αυτό καθίστα τον έλεγχο τους με τον παραδοσιακό τρόπο προγραμματισμού (τον σειριακό) αδύνατο. Για τον λόγο αυτό εισήχθη η τεχνική των Events και των Listeners (Σχήμα 15). Τα Events είναι τα γεγονότα τα οποία αναμένονται να συμβούν χωρίς να είναι γνωστό πότε, και οι Listeners είναι τα κομμάτια του κώδικα αυτά τα οποία παρακολουθούν την εξέλιξη του προγράμματος περιμένοντας να εντοπίσουν τα Events. Το μοντέλο προγραμματισμού αυτό έχει καθιερωθεί ως `event-driving programming`. Στο προγραμματιστικό μοντέλο αυτό η λογική που ακολουθείται είναι τελείως διαφορετική από αυτή του σειριακού. Συγκεκριμένα στο `event-driving` μοντέλο, το πρόγραμμα αντί να εκτελεί συγκεκριμένη λειτουργία, αναμένει να συμβούν τα διάφορα events. Μόλις λοιπόν ένα event γίνει αντιληπτό, αμέσως η ροή του προγράμματος μεταφέρεται στο σημείο του κώδικα που είναι εξουσιοδοτημένο να διαχειρίζεται το συγκεκριμένο event και,

αφού γίνουν οι απαιτούμενες ενέργειες, η ροή του προγράμματος επιστρέφει στην αρχική της θέση πριν το event.

3.4.3 Η Μέθοδος serialEvent

Στον κώδικα υπάρχει η μέθοδος serialEvent η οποία δέχεται ως όρισμα ένα event και είναι υπεύθυνη για την διαχείριση όλων των events που συμβαίνουν στην σειριακή θύρα που το πρόγραμμα Collector χρησιμοποιεί. Η μέθοδος αυτή παρέχεται από το επεκτατικό πακέτο javax.comm που αναφέρθηκε παραπάνω. Τα events που είναι δυνατόν να διαχειρισθούν είναι τα εξής

- BI: Break interrupt
- OE: Overrun error
- FE: Framing error
- PE: Parity error
- CD: Carrier detect
- CTS: Clear to send
- DSR: Data set ready
- RI: Ring indicator
- OUTPUT_BUFFER_EMPTY: Το output buffer είναι άδειο. Το event αυτό προκαλείται όταν μια έγγραφη τελειώνει και το buffer του συστήματος αδειάζει.
- DATA_AVAILABLE: Υπάρχουν δεδομένα διαθέσιμα στην σειριακή θύρα. Το event αυτό θα δημιουργηθεί όταν νέα δεδομένα αφιχθούν στην σειριακή θύρα. Ακόμα και αν το πρόγραμμα δεν παραλάβει τα δεδομένα αυτά δεν θα ξαναεμφανιστεί το event αυτό μέχρι την στιγμή που θα αφιχθούν νέα δεδομένα στη θύρα.



Σχήμα 15 Event-Handling

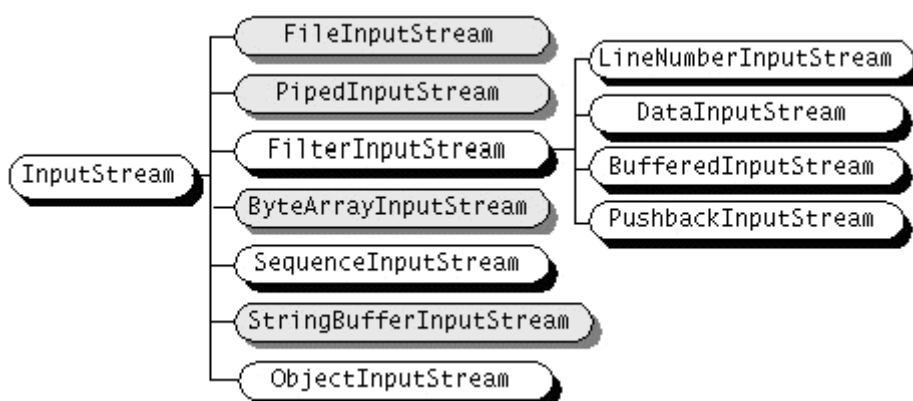
Το τελευταίο event είναι για την εφαρμογή Collector το πιο σημαντικό γιατί είναι αυτό που ενημερώνει για την άφιξη των νέων δεδομένων και αποτελεί το αποτελεσματικότερο εργαλείο για την κατάλληλη διαχείρισή τους. Συγκεκριμένα, κάθε φορά που εμφανίζονται τα δεδομένα στη σειριακή θύρα, ξεκινάει η επεξεργασία τους έχοντας δημιουργήσει ένα InputStream (Σχήμα 16) στην σειριακή θύρα.



Σχήμα 16 Η δομή ενός Stream

Για την ακρίβεια χρησιμοποιώντας την δομή `DataInputStream` (με όρισμα το `InputStream` που δημιουργήθηκε προηγουμένως) και μέσω αυτής διαβάζονται τα δεδομένα. Η διαδικασία αυτή μπορεί να φαίνεται αρκετά πολύπλοκη αλλά είναι απόλυτα συνηθισμένη τακτική στην Java να διαβάζονται και να γράφονται δεδομένα μέσω streams που το ένα επικαλύπτει το άλλο προσφέροντας έτσι πολλαπλή λειτουργικότητα στην δομή αυτή (Σχήμα 17). Εν πάση περιπτώσει αυτό που πρέπει να γίνει κατανοητό είναι ότι τα δεδομένα της σειριακής θύρας διαβάζονται μέσω του `dataIS` που είναι ένα `DataInputStream`. Τα δεδομένα διαβάζονται σαν bytes γιατί παρόλο που η Java υποστηρίζει την δυνατότητα λήψης των δεδομένων (μέσω του `DataInputStream` βέβαια) και σαν ακέραιους και σαν πραγματικούς και σαν String κάτι τέτοιο δεν υποστηρίζεται από το πρόγραμμα που αναλαμβάνει την ασύρματη αποστολή των δεδομένων. Συγκεκριμένα το LabView επιτρέπει την

αποστολή των δεδομένων μόνο με την μορφή String. Δυστυχώς παρ ότι η Java είναι σε θέση να υποστηρίξει δεδομένα τύπου String, στην συγκεκριμένη περίπτωση παρουσιάσθηκε μια ασυμβατότητα που οδήγησε στην επιλογή λήψης των δεδομένων με την μορφή των bytes. Τα bytes αυτά αποθηκεύονται στους κατάλληλους πίνακες που έχουν οριστεί, στην συνέχεια μετατρέπονται σε χαρακτήρες (char) και ακολούθως σε ακραίους ή πραγματικούς ανάλογα αν πρόκειται για το αναγνωριστικό (ID) ή την τιμή του δεδομένου.



Σχήμα 17 Επικάλυψη Streams

Σε αυτό το σημείο θα επιχειρηθεί η ανάλυση του πρωτοκόλλου λήψης δεδομένων ενώ θα περιγράψει και η αποτελεσματικότητα του στην αντιμετώπιση του θορύβου. Το πρωτόκολλο αυτό βασίζεται στην πολλαπλή μετάδοση των δεδομένων καθώς και στην χρησιμοποίηση χαρακτήρων ελέγχου. Προκείμενου να καταστεί δυνατός ο έλεγχος στα δεδομένα τα οποία μεταδίδονται και λαμβάνονται από το σύστημα, επιχειρήθηκε η ομαδοποίηση τους σε πακέτα. Συγκεκριμένα κάθε πακέτο που μεταδίδεται περιέχει τρεις φορές το ID του δεδομένου και τρεις φορές την τιμή του δεδομένου. Το κάθε ID και η κάθε τιμή αποθηκεύονται σε ξεχωριστό buffer υλοποιημένο ως bytearray. Τα bytes μετατρέπονται σε characters όπου αποθηκεύονται σε chararrays, εν συνεχεία το chararray δημιουργεί ένα String το οποίο τελικά θα μετατραπεί σε ακέραιο ή σε πραγματικό ανάλογα με το αν πρόκειται για ID ή για τιμή δεδομένου. Ουσιαστικά στο σημείο αυτό ξεκινάει ο έλεγχος για την εγκυρότητα των δεδομένων. Κρίνεται σκόπιμο να αναφερθεί ότι για τον αποτελεσματικό έλεγχο έχουν υλοποιηθεί επιπλέον τέσσερις Exceptions για

να διαχειρίζονται όλες τις καταστάσεις αναμενόμενες ή μη που μπορεί να προκύψουν. Οι Exceptions που χρησιμοποιούνται για τον σκοπό αυτό είναι η

- **NumberFormatException:** Έτοιμη Exception της Java. Πυροδοτείται όταν γίνει προσπάθεια να μετατραπεί ένα String που περιέχει μη έγκυρους μαθηματικά χαρακτήρες σε πραγματικό ή ακέραιο αριθμό. Η σημασία της είναι πολύ μεγάλη για την εφαρμογή γιατί με αυτόν τον τρόπο υπάρχει η δυνατότητα για άμεση αντιμετώπιση δεδομένων αλλοιωμένων από τον θόρυβο. Όπως έχει αναφερθεί νωρίτερα είναι ιδιαίτερα πιθανό κάποιο δεδομένο το οποίο έχει εκτεθεί στον θόρυβο να περιέχει και μη αποδεκτούς χαρακτήρες. Στην περίπτωση αυτή θα εντοπιστεί και απορριφθεί μέσω αυτής της Exception.
- **UnknownIdException:** Η Exception αυτή πυροδοτείται όταν κατά την σύγκριση αναγνωριστικών τα οποία δεν έχουν υποπέσει σε **NumberFormatException**, προκύψει αδυναμία επιβεβαίωσης ορθής λαμβανόμενης τιμής. Με τον όρο ορθή τιμή εννοείται ότι πρόκειται για ορθή μαθηματικά τιμή και ως τέτοια θεωρείται κάθε τιμή που δεν έχει προκαλέσει **NumberFormatException**. Συγκεκριμένα, αν δεν προκύψει ισότητα σε δύο από τρία αναγνωριστικά του πακέτου, τότε, είναι αδύνατον να γνωρίζουμε αν κάποια από αυτά παραλειφθεί σωστά, αφού δεν μπορεί να προκύψει κανενός είδους επιβεβαίωση. Σε αυτή τη περίπτωση είναι προφανές ότι ούτε η τιμή του δεδομένου μπορεί να γίνει αποδέκτη, αφού δεν είναι γνωστό το αναγνωριστικό του. Επειδή όμως θα ακολουθήσει η τριπλή λήψη της τιμής του δεδομένου του πακέτου, η **UnknownIdException** θα αναλάβει και την συλλογή των άχρηστων bytes που αναπαριστούν την τιμή του δεδομένου που έχουμε ήδη απορρίψει.
- **UnknownDataException:** Η Exception αυτή προκαλείται εάν δεν καταστεί δυνατόν να ληφθούν ορθά τουλάχιστον δύο από τις τρεις τιμές που εμπεριέχονται στο πακέτο των δεδομένων ή από τις τιμές που ελήφθησαν αν δεν προκύψει ισότητα μεταξύ δύο τουλάχιστον εξ αυτών. Είναι δυνατόν να προκύψει **UnknownDataException** και στην περίπτωση που συμβούν αλληπάλληλες **UnknownIdExceptions**. Οι συνεχόμενες **UnknownIdExceptions** ίσως οφείλονται σε κάποιο **Deadlock** ή σε κάποια άλλη μη αναμενόμενη κατάσταση που μπορεί να έχει περιέλθει το

σύστημα και για να αντιμετωπιστεί το ενδεχόμενο αυτό προκαλείται `UnknownDataException` όταν πραγματοποιηθούν τρεις `UnknownIDExceptions`.

- `PanicStartException`: Η `Exception` αυτή προκαλείται από τους χαρακτήρες έλεγχου που χρησιμοποιεί το πρωτόκολλο. Υπάρχουν δύο χαρακτήρες έλεγχου ο χαρακτήρας “s” που πρέπει να λαμβάνεται απαραίτητα πριν από την λήψη του κάθε πακέτου καθώς και ο χαρακτήρας “e” ο οποίος λαμβάνεται πάντα μετά την λήψη του πακέτου. Σε περίπτωση που ο χαρακτήρας “s” ληφθεί σε οποιοδήποτε άλλο σημείο τότε αυτόματα προκαλείται η `PanicStartException`, όπου αναλαμβάνει να συγκεντρώσει όλα τα εισερχόμενα bytes έως ότου να ληφθεί ο χαρακτήρας έλεγχου “e”. Τότε επιβεβαιώνεται η λήξη της μετάδοσης του αλλοιωμένου από το θόρυβο πακέτου και το πρόγραμμα επιστρέφει στην κανονική του ροή αναμένοντας τον χαρακτήρα έλεγχου “s” προκειμένου να λάβει σωστά αυτή τη φορά το νεοεισερχόμενο πακέτο. Σε περίπτωση όμως που αντί για τον αναμενόμενο χαρακτήρα έλεγχου “s” εισαχθεί στο σύστημα οποιοσδήποτε άλλος χαρακτήρας προκαλείται και πάλι η `PanicStartException` με την βεβαιότητα ότι πρόκειται για θόρυβο. Από τον σχεδιασμό του πρωτοκόλλου είναι δεδομένο ότι μεταξύ του “e” του “s” δεν πρέπει να υπεισέρχεται κανένας άλλος χαρακτήρας εκτός του καθορισμένου συνόλου.
- `PanicStopException`: Ο χαρακτήρας έλεγχου “e” πρέπει να λαμβάνεται αποκλειστικά μετά την ολοκλήρωση της μετάδοσης του πακέτου. Σε οποιαδήποτε άλλη περίπτωση είναι σίγουρο ότι έχει υπεισέλθει θόρυβος στο σύστημα και συνεπώς τα δεδομένα του πακέτου που έχουν ληφθεί μέχρι εκείνη την στιγμή είναι εσφαλμένα και άρα μη αποδεκτά. Το πρόγραμμα μετά από την `PanicStopException` αναμένει τον χαρακτήρα “s” προκειμένου να επιχειρήσει να λάβει σωστά το επόμενο πακέτο.

Όπως θα έχει γίνει ήδη κατανοητό στον έμπειρο σε θέματα λογισμικού αναγνώστη, το μεγαλύτερο μέρος του πρωτοκόλλου βασίζεται στην εκτεταμένη και αποτελεσματική χρήση του `Exception Handling`. Οι τρεις πρώτες `Exceptions` αντιμετωπίζουν κυρίως τον θόρυβο που προκαλεί αλλοιώσεις στα εισερχόμενα δεδομένα. Ιδιαίτερα σημαντική είναι η

NumberFormatException η οποία σε πολύ μεγάλο βαθμό προστατεύει το σύστημα από τις αλλοιώσεις που μπορεί να επιφέρει ο θόρυβος αφού λειτουργεί σαν φίλτρο, αποκόπτοντας τους μη αριθμητικούς χαρακτήρες ή σύμβολα. Στην συνέχεια, συγκρίνοντας τα δεδομένα τα οποία λαμβάνονται πολλαπλές φορές σε κάθε πακέτο εκμηδενίζεται ουσιαστικά η περίπτωση εισόδου εσφαλμένων δεδομένων στο σύστημα. Η περίπτωση του προσθετικού θορύβου (στο σύγγραμμα αυτό ο ορισμός προσθετικός θόρυβος αναφέρεται στο φαινόμενο που απαντάται στη διεθνή βιβλιογραφία ως insertion bits), δηλαδή του θορύβου που προσθέτει άχρηστα bits στο σύστημα (έχει γίνει εκτεταμένη αναφορά παραπάνω) που είναι σίγουρα ιδιαίτερα επικίνδυνη για την ευστάθεια τού συστήματος αντιμετωπίζεται ιδιαίτερα αποτελεσματικά με την χρήση των χαρακτήρων ελέγχου και των PanicStartException και PanicStopException. Θεωρητικά είναι, με την υλοποίηση αυτή, αδύνατο να τεθεί το σύστημα σε κατάσταση deadlock, αλλά και η πρακτική εφαρμογή του πρωτοκόλλου αυτού επέδειξε άριστα αποτελέσματα αφού κατά την διάρκεια της λειτουργίας του δεν παρουσιάσθηκε ούτε μια φορά περίπτωση deadlock, παρ' ότι το σύστημα δέχθηκε ισχυρότατο θόρυβο.

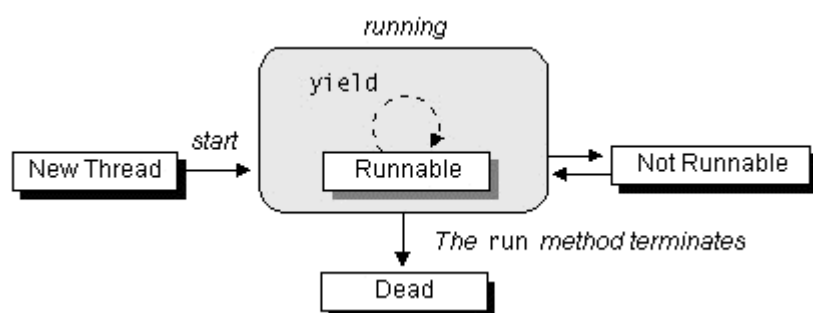
Αξίζει να σημειωθεί ότι γίνεται αναφορά στο είδος των σφαλμάτων που προκύπτουν, έτσι ώστε ο μηχανικός που χειρίζεται το πρόγραμμα να είναι σε θέση να αντιληφθεί γρήγορα το είδος του θορύβου που παρουσιάζεται και να λάβει όλα τα απαιτούμενα μέτρα.

3.4.4 Ο Constructor για τον Collector που διαχειρίζεται την ροή των δεδομένων προς των Server

Η εφαρμογή Collector είναι μια threaded εφαρμογή, επιτρέποντας την παράλληλη εκτέλεση παραπάνω του ενός block κώδικα. Στο σημείο αυτό κρίνεται σκόπιμο να παρατεθούν κάποιες λεπτομέρειες για τον τρόπο που χειρίζεται τα threads η Java. Υπάρχει ένα μνημόνιο που αφορά τις προδιαγραφές POSIX, στο οποίο αναφέρονται οι προδιαγραφές που πρέπει να πληροί ένα σύστημα UNIX. Το μνημόνιο φέρει το χαρακτηριστικό P1003.4a και επικυρώθηκε τον Ιούνιο του 1995. Η υλοποίηση των threads στην Java διαφέρει από τις προδιαγραφές του POSIX μοντέλου. Ο κύριος λόγος που

συνέβη αυτό είναι ότι κατά την διάρκεια του σχεδιασμού και υλοποίησης των threads της Java, δεν είχε ολοκληρωθεί ο σχεδιασμός τους κατά το POSIX μοντέλο, έτσι οι μηχανικοί της Sun αποφάσισαν να ακολουθήσουν την δική τους προσέγγιση στο API (Applications Programming Interface) αυτό. Το αποτέλεσμα που προέκυψε είναι ιδιαίτερα θετικό αφού είναι γενικά αποδεκτό ότι τα threads της Java είναι απλούστερα σε σχέση με αυτά του POSIX προτύπου, αναλαμβάνουν την αυτοδιαχείριση της μνήμης τους, ενώ παράλληλα δεν είναι τόσο δαπανηρά σε πόρους του συστήματος όσο αυτά του POSIX. Τα πλεονεκτήματα του threaded προγραμματισμού είναι τόσα πολλά που τον καθιστούν σχεδόν αναγκαίο για κάθε σοβαρή εφαρμογή. Η σημασία του προγραμματιστικού αυτού τρόπου εγγυάται κυρίως στην καλύτερη αξιοποίηση των πόρων του συστήματος που πραγματοποιείται μέσω ενός threaded προγράμματος. Ένα πρόγραμμα το οποίο περιμένει από το χρήστη να αλληλεπιδράσει με γραφικό περιβάλλον, και τέτοια είναι τα περισσότερα προγράμματα, ή ένα πρόγραμμα με εκτεταμένο I/O είναι σχεδόν απαραίτητο να κάνει χρήση threads. Αυτό θα του προσφέρει την δυνατότητα να μεταφέρει την ροή του προγράμματος από κάποιο κομμάτι που είναι δεν είναι δυνατόν να εκτελεστεί (π.χ το πρόγραμμα περιμένει από τον χρήστη να εισαγάγει δεδομένα) σε κάποιο άλλο που είναι (π.χ. κάποιους μαθηματικούς υπολογισμούς που πρέπει να εκτελεστούν). Ακόμα κάποια μαθηματικά προβλήματα και αλγόριθμοι μοντελοποιούνται πολύ καλύτερα χρησιμοποιώντας threads. Το πλέον σημαντικό πλεονέκτημα που προσφέρουν τα threads είναι η άριστη οργάνωση που παρέχουν σε κάποια είδη εφαρμογών όπως και στην περίπτωση του Collector. Συγκεκριμένα μια Client/Server εφαρμογή είναι πολύ πιο εύκολο να υλοποιηθεί με threads όπου, σε κάθε νέα αίτηση από κάποιο client, θα δημιουργείται και ένα καινούργιο thread από το να υλοποιηθεί κάποιος αλγόριθμος που θα καταγράφει επακριβώς την κατάσταση του κάθε client και τις απαιτούμενες σε σχέση με αυτό ενέργειες. Τέλος για να θεωρηθεί πλήρης η αναφορά στα threads πρέπει να τονιστεί ότι σε σχέση με την χρήση των processes η χρήση των threads είναι πολύ πιο αποτελεσματική. Αυτό συμβαίνει διότι η εναλλαγή από ένα process σε ένα άλλο, το λειτουργικό σύστημα απαιτεί να αποθηκευτούν όλα τα χαρακτηριστικά του προγράμματος όπως ο εικονικός χάρτης μνήμης (virtual memory map), οι περιγραφείς αρχείων (file

descriptors), οι ρυθμίσεις των διακοπών (interrupt settings) και αρκετές άλλες πληροφορίες. Αντίθετα στην περίπτωση των Java threads (Σχήμα 18) η Java Virtual Machine αναλαμβάνει να αποθηκεύσει πληροφορία για κάποιους καταχωρητές, τον stack pointer, τον μετρητή του προγράμματος (program counter) καθώς και κάποια άλλα απαραίτητα στοιχεία. Σε κάθε περίπτωση η διαδικασία εναλλαγής threads είναι σαφώς λιγότερο απαιτητική από αυτή των processes και, σε αρκετές περιπτώσεις αυξάνει την ταχύτητα εκτέλεσης του προγράμματος.



Σχήμα 18 Η λειτουργία ενός Thread

Το πρόγραμμα Collector είναι μια κατ'εξοχήν client/server εφαρμογή και δεδομένης της ανάγκης να δέχεται τα δεδομένα ασύγχρονα από την σειριακή θύρα είναι προφανές ότι συνδυάζει όλα τα χαρακτηριστικά που επιβάλλουν την χρήση threads. Η εφαρμογή λοιπόν παράλληλα με την δημιουργία του constructor Collector που ελέγχει την σειριακή θύρα, εξετάζει εάν υπάρχει κάποιος Server διαθέσιμος για να συνδεθεί με το υποσύστημα Collector. Σε περίπτωση που πράγματι υπάρχει Server διαθέσιμος τότε καλείται ο constructor που είναι υπεύθυνος για την σύνδεση του Collector με τον Server. Ο constructor αυτός δέχεται ως όρισμα το socket που θα συνδεθεί με τον Server καθώς και τη δομή syncBuffer που όπως έχει αναφερθεί παραπάνω είναι η δομή στην οποία αποθηκεύονται δυναμικά τα δεδομένα που λαμβάνονται από την σειριακή θύρα. Θα πρέπει να σημειωθεί ότι ο constructor αυτός καλείται ως thread και η περαιτέρω λειτουργία του προγράμματος είναι threaded.

3.4.5 Η Μέθοδος run

Η μέθοδος αυτή ενδέχεται να μη γίνει άμεσα κατανοητή από τον μη πεπειραμένο στην Java μηχανικό. Το σημείο το οποίο θα μπορούσε να προκαλέσει σύγχυση είναι αυτό στο οποίο καλείται η συνάρτηση run. Η συνάρτηση δεν καλείται σε κάποιο σημείο του προγράμματος όπως θα περίμενε ο αναγνώστης αλλά καλείται αυτόματα από την συνάρτηση start που αναλαμβάνει να εκκινήσει το thread. Θα πρέπει να σημειωθεί ότι αυτός ο μη αναμενόμενος τρόπος κλήσης μια συνάρτησης δεν αποτελεί ιδιαιτερότητα του συγκεκριμένου προγράμματος αλλά είναι ο τυπικός τρόπος εκκίνησης και λειτουργίας των threads στην Java. Πιθανόν θα βοηθούσε τον μηχανικό που χρησιμοποιεί το πρόγραμμα αν προσπαθούσε να θεωρήσει την συνάρτηση run σαν κάτι αντίστοιχο της συναρτήσεως main όπου και σε αυτή την περίπτωση δεν καλείται άμεσα από τον προγραμματιστή αλλά καλείται έμμεσα από την Java κατά την εκκίνηση λειτουργίας του προγράμματος. Όσον αφορά την συνάρτηση run κύριος σκοπός της είναι η μετάδοση των δεδομένων από τον Collector στον Server. Αυτό επιτυγχάνεται με τον εξής τρόπο. Αρχικά ο Collector μέσω της συνάρτησης run δέχεται ένα request από τον Server προκειμένου να αρχίσει η μετάδοση των δεδομένων. Εν συνεχεία στην μέθοδο αυτή γίνεται έλεγχος για την ύπαρξη νέων δεδομένων στο syncBuffer. Ο έλεγχος πραγματοποιείται κάθε 200 milliseconds, αν φυσικά δεν προκύψει άφιξη δεδομένων στην σειριακή θύρα, γιατί τότε η ροή του προγράμματος μεταφέρεται αυτόματα στην μέθοδο SerialEvent που είναι υπεύθυνη για την ορθή λήψη των δεδομένων από την σειριακή θύρα και την αποθήκευσή τους στο syncBuffer. Η χρονική παύση των 200 milliseconds κρίθηκε αναγκαία για να εξασφαλισθεί η οικονομία πόρων του συστήματος. Με την χρονική καθυστέρηση αυτή η απαιτούμενη επεξεργαστική ισχύς κυμάνθηκε γύρω στο 0% με 1%, ενώ στην περίπτωση του συνεχούς ελέγχου χωρίς την χρονική παύση τα αποτελέσματα ήταν καταστροφικά αφού απαιτείτο επεξεργαστική ισχύ κυμαινόμενη στο 100% του επεξεργαστή. Θα πρέπει να τονισθεί ότι τα άριστα αποτελέσματα που παράχθηκαν με το διάλειμμα των 200 milliseconds ελήφθησαν σε μηχανήμα τύπου Pentium II με επεξεργαστή λειτουργούντα στα 400 MHz. Είναι πιθανόν σε περίπτωση που

το πρόγραμμα απαιτηθεί να εκτελεστεί σε κάποιο μηχάνημα παλιότερης τεχνολογίας να χρειαστεί κάποια αύξηση του ορίου αυτού.

Τα δεδομένα μεταδίδονται από τον Collector στον Server χρησιμοποιώντας sockets. Συγκεκριμένα δημιουργώντας τις δομές is τύπου `DataInputStream` και os τύπου `DataOutputStream` οι οποίες κατασκευάστηκαν δεχόμενες ως όρισμα τα `InputStream` και `OutputStream` που προέρχονται από το socket το οποίο έχει συνδεθεί με τον Server παρέχεται η δυνατότητα να μεταδίδονται δεδομένα προς αυτόν και να λαμβάνονται αιτήσεις (request) από αυτόν.

Εφόσον έχει συνδεθεί ο Collector με τον Server και έχει ληφθεί η κατάλληλη αίτηση του Server από τον Collector η συνάρτηση `run` ελέγχει όπως προαναφέρθηκε για την ύπαρξη δεδομένων (όπως έχει τονιστεί επανειλημμένα με τον όρο δεδομένα εννοούμε το αναγνωριστικό του δεδομένου ID και την τιμή του) στο `syncBuffer`, και, αν βρεθούν, τότε αυτά μεταδίδονται εις τριπλούν στον Server. Η τριπλή μετάδοση πραγματοποιείται προκειμένου να ελαχιστοποιηθεί ο κίνδυνος αλλοίωσης των δεδομένων από θόρυβο. Θα πρέπει να σημειωθεί ότι τα δεδομένα μόλις μεταδοθούν διαγράφονται από το `syncBuffer` αφού είναι πλέον ανεπίκαιρα. Με το τρόπο αυτό το μέγεθος του `syncBuffer` παραμένει ιδιαίτερα περιορισμένο και αποφεύγονται προβλήματα υπερχείλισης της μνήμης.

Η συνάρτηση `run` παράλληλα καταγράφει και τον χρόνο της τελευταίας επιτυχημένης χρονικής μετάδοσης, ενώ χρονομετρεί και τον χρονικό διάστημα στο οποίο δεν έχει πραγματοποιηθεί μετάδοση δεδομένων. Αν το χρονικό διάστημα ανενεργούς λειτουργίας, δηλαδή μη μετάδοσης δεδομένων, υπερβεί αυτό που έχει ορίσει ως παράμετρο του προγράμματος ο μηχανικός που χρησιμοποιεί το πρόγραμμα τότε θεωρείται ότι έχει προκύψει μια μη αναμενόμενη κατάσταση για το σύστημα και λαμβάνονται τα απαραίτητα μέτρα. Συγκεκριμένα ο Collector μέσω της συνάρτησης `run` θα μεταδώσει τρεις φορές το κατάλληλο σήμα έλεγχου το οποίο θα προκαλέσει το τερματισμό της λειτουργίας του Server και εν συνεχεία θα διακόψει και την δική του λειτουργία. Βεβαίως λαμβάνοντας το σήμα αυτό ο Server και πριν διακόψει την λειτουργία του αναλαμβάνει να εκτελέσει τα κατάλληλα scripts προκειμένου να επανεκκινήσει το σύστημα. Αξίζει να σημειωθεί ότι εκτός από

την διαδικασία αυτή η οποία έχει ως στόχο να προστατεύει το σύστημα από μη αναμενόμενες καταστάσεις η καταγραφή της τελευταίας επιτυχημένης μετάδοσης καθώς και του χρόνου της ανενεργούς λειτουργίας (μη λήψης δεδομένων από την σειριακή θύρα και μη μετάδοσης τους στον Server) προσφέρει σημαντικότερο επικουρικό έργο στον μηχανικό που χειρίζεται το σύστημα. Είναι ιδιαίτερα σημαντικό για τον διαχειριστή του συστήματος να είναι σε θέση να γνωρίζει τα παραπάνω στοιχεία διότι στις περισσότερες περιπτώσεις η μετάδοση των δεδομένων είναι περιοδική. Έχοντας λοιπόν την δυνατότητα ο διαχειριστής να επιβεβαιώνει την περιοδικότητα αυτή είναι σε πολύ μεγάλο βαθμό σίγουρος για την καλή λειτουργία του συστήματος ενώ αντίθετα στην περίπτωση που διαπιστώσει ότι τα δεδομένα λαμβάνονται σε μη τακτά διαστήματα μπορεί να οδηγηθεί στο συμπέρασμα ότι υφίσταται κάποιο πρόβλημα.

Σημαντικότερη για την εύρυθμη λειτουργία του συστήματος θεωρείται και η χρήση της `IOException`. Κάνοντας χρήση αυτής της `Exception` ο `Collector` είναι σε θέση να γνωρίζει άμεσα αν προέκυψε μη αναμενόμενη διακοπή της σύνδεσης με τον `Server`. Αν παρουσιαστεί το πρόβλημα αυτό τότε ο `Collector` παύει την λειτουργία του και επανεκκινείται προκειμένου να αποκαταστήσει την σύνδεση του με τον `Server`. Θα πρέπει να σημειωθεί ότι και σε αυτή την περίπτωση η επανεκκίνηση του προγράμματος επιτυγχάνεται με την χρήση `scripting`.

Κεφάλαιο 4

Το Πρόγραμμα Λήψης, Διαχείρισης και Δρομολόγησης Δεδομένων

4.1 Σύντομη περιγραφή

Το πρόγραμμα λήψης, διαχείρισης και δρομολόγησης δεδομένων (**RES-ADMIN Server**) έχει ως στόχο την ορθή λήψη των δεδομένων από το πρόγραμμα συλλογής και μετάδοσης δεδομένων (**Collector**), την κατάλληλη

διαχείριση τους και την περαιτέρω δρομολόγηση τους στο πρόγραμμα παρουσίασης δεδομένων (**RES-ADMIN Applet**).

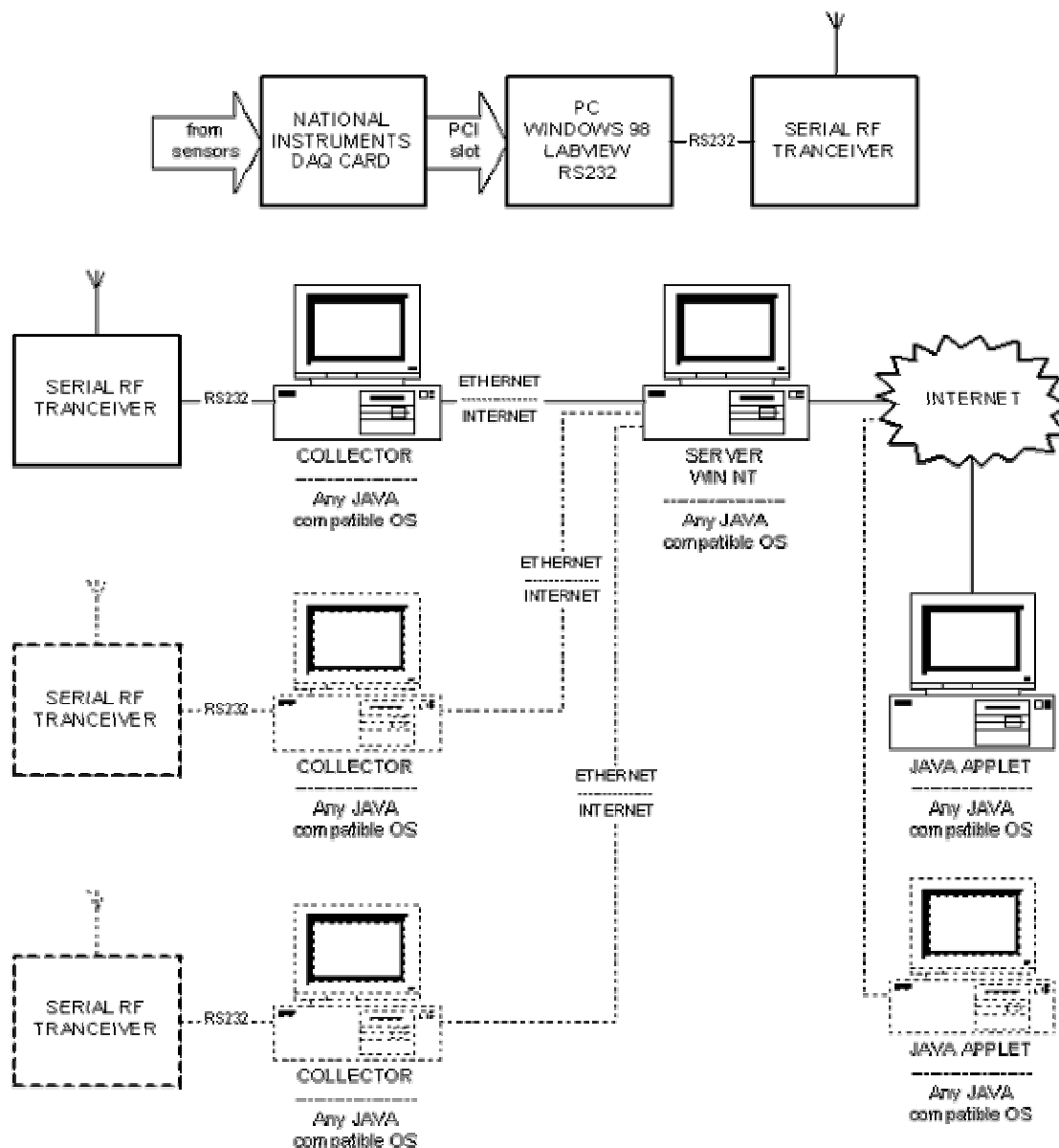
4.2 Γενικά χαρακτηριστικά

Σε κάθε κατανεμημένο σύστημα επικοινωνίας είναι απαραίτητη η ύπαρξη ενδιάμεσων κόμβων που να εξασφαλίζουν τόσο την σύνδεση των υποσυστημάτων που απαρτίζουν το σύστημα όσο και τον κατάλληλο συντονισμό τους. Έχοντας υλοποιήσει το πρόγραμμα συλλογής και μετάδοσης δεδομένων προέκυψε η ανάγκη για τη δημιουργία ενός προγράμματος το οποίο θα αναλάμβανε να συγκεντρώνει τα δεδομένα από τους επιμέρους Collectors και εν συνεχεία να τα μεταδώσει στα Applet των χρηστών που είναι συνδεδεμένοι. Η επιλογή της κατάλληλης γλώσσας προγραμματισμού αποτέλεσε αντικείμενο έντονου προβληματισμού καθώς κύριο σχεδιαστικό μέλημα ήταν η κατασκευή της εφαρμογής ώστε να εγγυάται τον καλύτερο δυνατό συντονισμό όλων των προγραμμάτων που είναι συνδεδεμένα σε αυτή. Ιδιαίτερα σημαντικό είναι να διασφαλιστεί ότι ο Server θα μπορεί να ανακάμπτει σε περίπτωση που προκύψει κάποιο μη αναστρέψιμο σφάλμα. Επιπλέον ελήφθη σοβαρά υπόψη το γεγονός ότι ο Server συγκεντρώνει όλα τα δεδομένα από όλους τους διαφορετικούς Collectors παρέχοντας την δυνατότητα υποβολής τους σε κάποιας μορφής επεξεργασία. Τέλος επειδή είναι υποχρεωτικό εκ των πραγμάτων ο RES-ADMIN Server να εκτελείται σε κάποιο υπολογιστικό σύστημα το οποίο παράλληλα θα λειτουργεί και ως Web Server κρίθηκε αναγκαίο το πρόγραμμα RES-ADMIN Server να είναι ιδιαίτερα αποδοτικό καθώς και να κάνει λελογισμένη χρήση των πόρων του μηχανήματος.

4.3 Τεχνικά χαρακτηριστικά του υποσυστήματος

Το σύστημα αποτελείται από διάφορους Collectors οι οποίοι συνδέονται στον RES-ADMIN Server. Το πρόγραμμα που χρησιμοποιεί ο χρήστης (RES-ADMIN Applet) συνδέεται με την σειρά του στον Server από όπου και λαμβάνει τα δεδομένα. Έχει γίνει λοιπόν κατανοητό ότι πολλοί Collectors συνδέονται στον Server καθώς και ότι πολλά Applets συνδέονται

επίσης στον Server. Ο Server δηλαδή αποτελεί τον συνδετικό κρίκο μεταξύ των Collectors και των Applets. Ο κώδικας του Server είναι γραμμένος σε Java. Επιλέχθηκε και σε αυτήν την περίπτωση η Java, ως εργαλείο προγραμματισμού λόγω του ότι προσφέρει εξαιρετικές δυνατότητες για την δημιουργία networking εφαρμογών. Σε αντίθεση με τις άλλες γλώσσες προγραμματισμού, όπου είναι αρκετά πολύπλοκο και χρονοβόρο να δημιουργηθούν και να διαχειριστούν Sockets καθώς και άλλες δικτυακές δομές των οποίων γίνεται εκτεταμένη χρήση στην εφαρμογή RES-ADMIN, στην Java κάτι τέτοιο είναι απόλυτα εφικτό. Το πλεονέκτημα αυτό και σε συνδυασμό με τα προτερήματα του Platform Independency που αναλύθηκαν λεπτομερώς στα τεχνικά χαρακτηριστικά του Collector οδήγησαν στην παραπάνω επιλογή.



Σχήμα 19 Collectors-Server-Applets

Θα πρέπει ακόμα να σημειωθεί ότι ο Server για λόγους ασφάλειας των δεδομένων τα αποθηκεύει δίνοντας την δυνατότητα να υπάρχει αντίγραφο των μετρήσεων σε περίπτωση που προκύψει κάποιο πρόβλημα με τον μηχανήμα το οποίο έχει αναλάβει να κάνει την δειγματοληψία, καταγραφή και ασύρματη μετάδοση των δεδομένων. Επιπλέον με αυτόν τον τρόπο παρέχεται η δυνατότητα στον τελικό χρήστη να έχει πρόσβαση στις μετρήσεις, επιλέγοντας αν επιθυμεί τις μετρήσεις κάποιας περασμένης ημέρας ή τις σημερινές μετρήσεις μέχρι κάποια συγκεκριμένη ώρα ή τέλος όλες τις μετρήσεις ως την τελευταία που έχει ληφθεί.

Η ανάγκη για δυνατότητα επανάνাকাμψης του Server σε περίπτωση μη αναστρέψιμου σφάλματος αντιμετωπίσθηκε με τη χρήση Exception Handling, η οποία παρέχει την δυνατότητα να εντοπιστεί επακριβώς το σημείο του προγράμματος στο οποίο συνέβη το σφάλμα και στην συνέχεια σε περίπτωση που η σοβαρότητα του είναι τέτοια που δεν επιτρέπει την περαιτέρω ομαλή εκτέλεση, καλώντας τα κατάλληλα scripts να ξεκινήσει η διαδικασία επανεκκίνησης του RES-ADMIN Server. Παράλληλα θα πρέπει να σημειωθεί ότι σαν επιπλέον μέτρο προστασίας απέναντι σε τέτοιου είδους μη αναμενόμενες καταστάσεις το πρόγραμμα είναι σε θέση να ενημερώσει τον διαχειριστή του συστήματος αποστέλλοντας του mail και αναφέροντας την αιτία του προβλήματος.

Τέλος όσον αφορά την λελογισμένη χρήση των πόρων του συστήματος τόσο η επεξεργαστική ισχύ όσο και οι απαιτήσεις σε μνήμη κυμάνθηκαν σε πολύ χαμηλά επίπεδα. Η ανάγκη για μειωμένες απαιτήσεις ήταν δεδομένη αφού θα πρέπει να συνυπολογιστεί το γεγονός ότι το μηχάνημα που θα φιλοξενεί τον RES-ADMIN Server θα εκτελεί ακόμα και χρέη Web Server και FTP Server. Είναι πολύ σημαντικό να γίνει κατανοητό σε αυτό το σημείο ότι ο RES-ADMIN Server που υλοποιήθηκε πρέπει οπωσδήποτε να εκτελείται στον ηλεκτρονικό υπολογιστή που λειτουργεί και ως Web Server διότι κανένας browser δεν θα παρείχε την δυνατότητα στο RES-ADMIN Applet για λόγους ασφαλείας να αποκαταστήσει σύνδεση με κάποιο μηχάνημα πέραν αυτού που φιλοξενεί την σελίδα της; εφαρμογής RES-ADMIN.

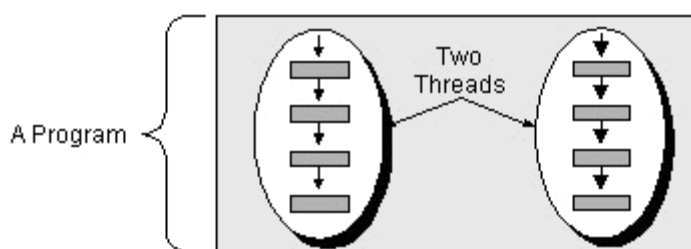
4.4 Τεχνική ανάλυση

Στο πρόγραμμα RES-ADMIN Server όπως και στο πρόγραμμα Collector γίνεται χρήση των threads. Στο πρόγραμμα Server όμως υπάρχει η δυνατότητα τα δύο threads που έχουν υλοποιηθεί προκειμένου να καλύψουν τις ανάγκες των Collector και των Applets αντίστοιχα να κληρονομήσουν πλήρως τα χαρακτηριστικά της κλάσης thread χρησιμοποιώντας την εντολή extends της οποίας η λειτουργία έχει αναλυθεί στο Κεφάλαιο του Collector. Πριν από την εκτεταμένη ανάλυση του προγράμματος θα πρέπει να γίνει κατανοητός ο γενικός τρόπος λειτουργίας του. Το πρόγραμμα Server

χρησιμοποιεί δύο threads το ColletThread και το AppletThread τα οποία διαχειρίζονται μέσω του Server ένα κοινό buffer για την αποθήκευση και μετάδοση των δεδομένων. Το ColletThread αναλαμβάνει να συνδεθεί σε όλους τους Collectors στους οποίους γίνεται αναφορά στο αρχείο των ρυθμίσεων που χρησιμοποιεί, ενώ το AppletThread καλύπτει τις συνδέσεις με τα διάφορα Applets.

4.4.1 Η συνάρτηση main

Το πρόγραμμα RES-ADMIN Server είναι ένα ιδιαίτερα παραμετροποιήσιμο πρόγραμμα προκείμενου να μπορεί να ανταποκριθεί αποτελεσματικά σε πλήθος διαφορετικών συνθηκών. Κατά την εκκίνηση της λειτουργίας του ο Server ανατρέχει στο αρχείο ρυθμίσεων που έχει δημιουργηθεί από τον μηχανικό που χειρίζεται το πρόγραμμα και λαμβάνει τις παρακάτω πληροφορίες για την ορθή εκτέλεση του.



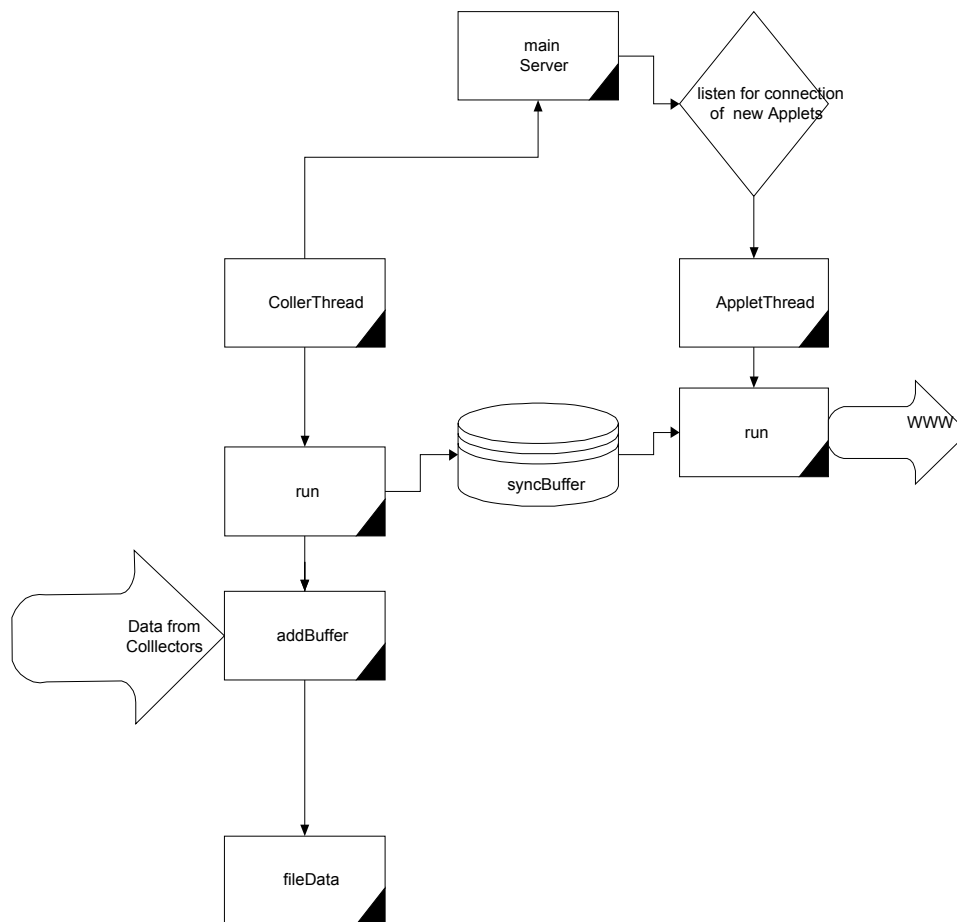
Σχήμα 20 ColletThread και AppletThread. Τα δύο threads του Server

- Αριθμός Collectors που πρέπει να συνδεθεί ο Server. Αυτή η παράμετρος είναι ιδιαίτερα βασική γιατί ενημερώνει το σύστημα για τον αριθμό των Collectors που βρίσκονται ή θα τεθούν σύντομα σε λειτουργία.
- Ταχύτητα μετάδοσης. Η παράμετρος αυτή ενημερώνει το πρόγραμμα για την συχνότητα που πρέπει να μεταδίδει τα δεδομένα στα RES-ADMIN Applets. Η τιμή αυτή είναι σε milliseconds. Ο διαχειριστής του συστήματος είναι σε θέση με αυτόν τον τρόπο να αποφασίσει ανάλογα με το είδος των δεδομένων πόσο συχνά θα ανανεώνονται οι τιμές στο Applet. Αν τα δεδομένα που λαμβάνονται από τον Collector ανανεώνονται πολύ αργά τότε μπορεί η τιμή να είναι αρκετά υψηλή χωρίς να υπάρχει πρόβλημα ενημέρωσης των δεδομένων στο RES-ADMIN Applet. Αντίθετα σε πολύ

σημαντικά δεδομένα για τα οποία απαιτείται συνεχής παρακολούθηση θα πρέπει η τιμή αυτή να είναι ιδιαίτερα χαμηλή προσφέροντας διαρκή ανανέωση. Θα πρέπει να σημειωθεί ωστόσο ότι πολύ χαμηλές τιμές μπορεί να προκαλέσουν σημαντικό φόρτο στο δίκτυο.

- IP του μηχανήματος στο οποίο λειτουργεί ο Collector. Θα πρέπει να είναι γνωστό το IP του μηχανήματος του Collector στο οποίο θα συνδεθεί ο Server.
- Θύρα (port) που χρησιμοποιεί ο Collector. Η θύρα την οποία έχει δεσμεύσει ο Collector για την επικοινωνία του με τον Server.

Μετά την λήψη και την εξακρίβωση της ορθότητας των στοιχείων ο Server ξεκινάει ένα καινούργιο `CollectorThread` για κάθε ένα Collector που είναι σε λειτουργία. Όπως έχει αναφερθεί παραπάνω ο `CollectorThread` είναι ένα thread το οποίο μεταφέρει τη ροή του προγράμματος στο συγκεκριμένο block του κώδικα που είναι το `CollectorThread`. Προτού γίνει αναλυτική αναφορά στο `CollectorThread` θα πρέπει να επισημανθεί ότι το buffer που έχει δημιουργηθεί για την αποθήκευση και μετάδοση των δεδομένων, ονομάζεται `syncBuffer`, είναι ένα αντικείμενο τύπου `Vector` και έχει ακριβώς την ίδια δομή και λειτουργία με αυτό που χρησιμοποιείται στο πρόγραμμα `Collector`. Εκτεταμένη αναφορά καθώς και οι απαραίτητες επεξηγήσεις για τον τρόπο λειτουργίας του έχουν δοθεί στο σχετικό Κεφάλαιο του Collector. Θα πρέπει ακόμα να προστεθεί ότι ο όρος δεδομένα αντιπροσωπεύει και σε αυτήν την περίπτωση την κλάση που έχουμε δημιουργήσει και περιλαμβάνει δύο πεδία, το αναγνωριστικό του δεδομένου (ID) καθώς και την τιμή του. Η κλάση αυτή είναι ακριβώς όμοια με την κλάση `SNData` που χρησιμοποιεί ο Collector. Ο λόγος που αν και οι δύο κλάσεις είναι όμοιες ονομάζονται διαφορετικά είναι ότι παρέχεται αυτονομία στον μηχανικό που επιθυμεί να τροποποιήσει το πρόγραμμα, να μπορεί να μεταβάλει τα δεδομένα της μιας κλάσης χωρίς να προκύπτουν προβλήματα αστάθειας του συστήματος, διότι όπως αναφέρθηκε στο προηγούμενο Κεφάλαιο σε περίπτωση που και τα δύο προγράμματα εκτελούνται στον ίδιο υπολογιστή και περιλαμβάνουν μια κλάση με το ίδιο όνομα αλλά με διαφορετική δομή είναι πολύ πιθανόν να εμφανιστούν ανωμαλίες στο σύστημα.



Σχήμα 21 Flowchart του Server

4.4.2 Ο Constructor του ColletThread

Ο constructor της κλάσης ColletThread δέχεται ως όρισμα το IP του Collector που θα επιχειρήσει να συνδεθεί καθώς και το port που χρησιμοποιεί. Τελευταία παράμετρος είναι το buffer που χρησιμοποιείται για την αποθήκευση και την μετάδοση των δεδομένων και βέβαια είναι το syncBuffer στο οποίο έγινε αναφορά παραπάνω. Είναι σημαντικό να είναι γνωστό στον RES-ADMIN Server το IP του Collector στον οποίο θα συνδεθεί. Αυτό προϋποθέτει την υπάρξει σταθερού IP από την πλευρά του Collector, διότι διαφορετικά θα πρέπει να ενημερωθεί το αρχείο ρυθμίσεων του RES-ADMIN Server για το IP του Collector και ακολούθως να συνδεθεί. Αυτονόητο είναι ότι συνδέσεις που χρησιμοποιούν dynamic IP απαιτούν συνεχείς μεταβολές στο

αρχείο ρυθμίσεων με αποτέλεσμα να απεμπολείται ένα από τα σημαντικότερα πλεονεκτήματα της εφαρμογής, δηλαδή η αυτονομία της.

4.4.3 Η συνάρτηση run

Στην συνάρτηση run δημιουργούνται τα `DataInputStream` και `DataOutputStream` τα οποία επεκτείνουν τα `InputStream` και `OutputStream` και τα οποία έχουν ληφθεί από το `Socket` που έχει δημιουργηθεί για να επιτευχθεί η σύνδεση με τον `Collector`. Ο `Collector` αναμένει νέες αιτήσεις για σύνδεση, χρησιμοποιώντας την κλάση `ServerSocket`. Η τεχνική των streams και των sockets έχει συζητηθεί εκτεταμένα στο Κεφάλαιο της τεχνικής ανάλυσης του `Collector`. Η τεχνική αυτή είναι ιδιαίτερα διαδεδομένη και απαντάται σχεδόν σε κάθε σύγγραμμα που αναφέρεται σε δικτυακές εφαρμογές με Java. Έκτος από την πραγματοποίηση της σύνδεσης με τον ή τους `Collectors`, παράλληλα αρχικοποιείται ένα `buffer` το οποίο είναι σχεδιασμένο να αποθηκεύει χαρακτήρες, και χρησιμοποιείται για την καταγραφή των δεδομένων στο αρχείο. Η διαδικασία αυτή θα συζητηθεί παρακάτω. Όταν πραγματοποιηθεί η σύνδεση του `RES-ADMIN Server` με τον `Collector`, τότε ο `Server` θα επιβεβαιώσει την σύνδεση στέλνοντας ένα `request` στον `Collector`, ο οποίος μόλις το λάβει, θα αρχίσει την μετάδοση των δεδομένων. Εν συνεχεία ο `Server` θα αρχίσει να λαμβάνει δεδομένα, τα οποία θα τα διαχειριστεί χρησιμοποιώντας την μέθοδο `addBuffer`. Θα πρέπει να σημειωθεί ότι στο σημείο αυτό πραγματοποιούνται μικρές διακοπές δίνοντας εντολή στο `thread` να αναστείλει την λειτουργία του για 50 milliseconds. Με τον τρόπο αυτό επιτυγχάνεται η μείωση του φόρτου στον επεξεργαστή. Σε όλη τη διάρκεια της μεθόδου, χρησιμοποιείται εκτεταμένο `Exception Handling` και στην περίπτωση που προκύψει κάποιο πρόβλημα με την λειτουργία των sockets, τότε εκτελούνται τα κατάλληλα `scripts` τα οποία επανεκκινούν τον `Server`.

4.4.4 Η μέθοδος addBuffer

Όπως αναφέρθηκε κατά την ανάλυση της συνάρτησης `run`, μόλις συνδεθεί ο `RES-ADMIN Server` στον `Collector`, τότε αρχίζει η μετάδοση των δεδομένων απ τον `Collector` στον `Server`. Η διαχείριση των δεδομένων

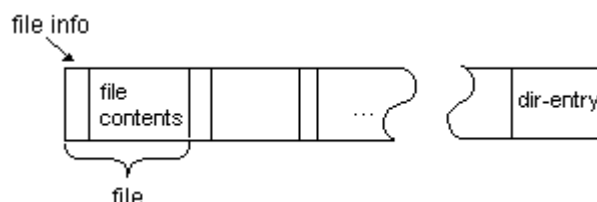
πραγματοποιείται από την μέθοδο `addBuffer` η οποία λαμβάνει σαν όρισμα το `syncBuffer` που έχει δημιουργηθεί για να αποθηκεύει τα δεδομένα. Η διαδικασία που ακολουθείται αρχικά περιλαμβάνει την τριπλή λήψη των δεδομένων καθώς και τον έλεγχο για την εγκυρότητα τους. Η εγκυρότητα πιστοποιείται αν προκύψουν τουλάχιστον δύο ίδιες τιμές από τις τρεις που έχουν ληφθεί, ενώ παράλληλα ελέγχεται και αν πρόκειται για το σήμα ελέγχου που φανερώνει την ύπαρξη προβλήματος στον Collector δίνοντας εντολή στον Server να επανεκκινήσει. Για την αποτελεσματικότερη αντιμετώπιση πιθανόν σφαλμάτων έχουν δημιουργηθεί δύο νέες Exceptions, η `CorruptedIDException` και η `CorruptedDataException`. Η πρώτη πυροδοτείται σε περίπτωση που προκύψει αδυναμία επιβεβαίωσης ορθού ID και αναλαμβάνει να συλλέξει τις άχρηστες τιμές του δεδομένου που θα ακολουθήσει, καθώς και η `CorruptedDataException`. Η διαδικασία αντιμετώπισης των σφαλμάτων αποτελεί υποσύνολο της διαδικασίας ελέγχου που πραγματοποιείται στο μέθοδο λήψης δεδομένων από την σειριακή θύρα, αφού οι συνθήκες μετάδοσης και λήψης των δεδομένων μέσω του δικτύου είναι σαφώς ασφαλέστερες από αυτές της ασύρματης μετάδοσης. Πέραν του ελέγχου που πραγματοποιείται για την εγκυρότητα της μετάδοσης, τα δεδομένα ελέγχονται επιπλέον και για την λογική των ID τους αφού είναι προφανές ότι αρνητικά ID ή υπερβολικά μεγάλα είναι σίγουρα εσφαλμένα.

Μετά την πραγματοποίηση του ελέγχου ακολουθεί η αποθήκευση των δεδομένων. Όπως έχει επανειλημμένα τονιστεί το `syncBuffer` που χρησιμοποιείται για την αποθήκευση των στοιχείων είναι ένα αντικείμενο τύπου `Vector`, το οποίο περιλαμβάνει αρκετές ιδιαίτερα χρήσιμες μεθόδους. Ο τρόπος με τον οποίο αποθηκεύονται τα δεδομένα είναι ο εξής. Αρχικά ελέγχεται αν το `buffer` είναι άδειο και σε αυτήν την περίπτωση εφόσον το δεδομένο έχει ληφθεί ορθά αποθηκεύεται στην πρώτη θέση του `buffer`. Τα υπόλοιπα δεδομένα που θα αφιχθούν υπόκεινται σε διαφορετική διαδικασία. Εξετάζεται το ID το κάθε δεδομένου και συγκρίνεται με το ID όλων των δεδομένων που υπάρχουν στο `buffer`. Στην περίπτωση που δεν βρεθεί το ID αυτό στο `buffer`, τότε προστίθεται το καινούργιο στοιχείο στο `buffer`, αντίθετα αν βρεθεί και άλλο δεδομένο στο `buffer` με το ίδιο ID τότε το νεώτερο δεδομένο αντικαθιστά το παλιότερο. Με αυτόν το τρόπο το μέγεθος του `buffer`

διατηρείται ιδιαίτερα χαμηλό. Το buffer δηλαδή περιέχει τόσους κόμβους όσα και τα είδη των δεδομένων που έχουν ληφθεί. Επιπρόσθετα το buffer είναι πάντα ενημερωμένο με τα πλέον πρόσφατα δεδομένα. Σημαντικό είναι να γίνει αντιληπτό ότι ο αριθμός των θέσεων του buffer είναι ο αριθμός του διαφορετικών ID που έχει λάβει ο Server. Χρησιμοποιώντας την τεχνική αυτή διατηρείται το μέγεθος του buffer σχεδόν σταθερό, αφού ο αριθμός των δεδομένων που δέχεται το σύστημα είναι συνήθως γνωστός και αναμενόμενος. Σε περίπτωση όμως που αφιχθούν νέου είδους δεδομένα στο σύστημα, αυτά θα προστεθούν δυναμικά χωρίς να προκύψει καμία δυσλειτουργία. Τα δεδομένα τα οποία λαμβάνονται εκτός της προσωρινής – δυναμικής αποθήκευσης στο syncBuffer καταγράφονται μόνιμα σε αρχεία στον σκληρό δίσκο που εκτελείται ο RES-ADMIN Server μέσω της συνάρτησης fileData

4.4.5 Η συνάρτηση fileData

Η συνάρτηση fileData καταγράφει τα λαμβανόμενα δεδομένα δίνοντας στον μηχανικό ή στον απλό χρήστη μια πλήρη εικόνα των μετρήσεων που έχουν συγκεντρωθεί. Υπάρχει η δυνατότητα με την χρήση ειδικού script το οποίο έχει υλοποιηθεί σε VbasicScript να διατηρεί το σύστημα αρχεία με τις μετρήσεις κάθε ξεχωριστής ημέρας, καθώς και αρχείο με το σύνολο των σημερινών μετρήσεων ενώ επιπλέον είναι δυνατή η διαχείριση αρχείου της σημερινής ημέρας, το οποίο να περιέχει τις μετρήσεις που έχουν ληφθεί μέχρι κάποια συγκεκριμένη ώρα.



Σχήμα 22 Η δομή ενός αρχείου στην Java

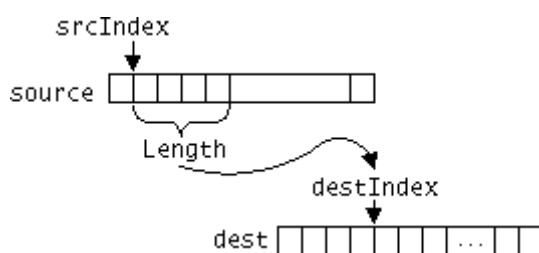
Η διαδικασία της καταγραφής του αρχείου, όπως και ολόκληρη η εφαρμογή RES-ADMIN είναι απόλυτα δυναμική καλύπτοντας και την καταγραφή των δεδομένων στο αρχείο. Για το λόγο αυτό η συνάρτηση

fileData σχεδιάστηκε έτσι ώστε να μπορεί να καταγράφει τα δεδομένα σε μια εύκολα αναγνώσιμη και επεξεργάσιμη μορφή, χωρίς όμως να γνωρίζει εκ των προτέρων ούτε το είδος των δεδομένων, ούτε την σειρά αλλά ούτε και των αριθμό των αφικνούμενων δεδομένων.

Η συνάρτηση fileData δέχεται ως όρισμα την κλάση data, στην οποία έγινε αναλυτική αναφορά στην αρχή του κεφαλαίου. Ο τρόπος λειτουργίας της συνάρτησης βασίζεται στην καταγραφή των μετρήσεων ανά λεπτό το οποίο θεωρείται ικανοποιητικό χρονικό διάστημα για τους σκοπούς της συγκεκριμένης εφαρμογής. Για να επιτευχθεί η δυναμική καταγραφή των μετρήσεων χρησιμοποιείται ένας buffer χαρακτήρων ο οποίος έχει υλοποιηθεί ως πίνακας χαρακτήρων. Ο πίνακας αυτός έχει συγκεκριμένο σταθερό μέγεθος. Όταν κληθεί η συνάρτηση να αποθηκεύσει τα δεδομένα, αρχικά τα επεξεργάζεται κατάλληλα ώστε να είναι σταθερού μήκους πραγματοποιώντας την κατάλληλη στρογγυλοποίηση στα δεκαδικά στοιχεία του αριθμού. Παράλληλα ελέγχοντας το ID του δεδομένου, τοποθετείται το δεδομένο σε συγκεκριμένη θέση μέσα στο char_buffer (ο πίνακας χαρακτήρων στον οποίο έγινε αναφορά παραπάνω). Αυτό επιτυγχάνεται έχοντας δεσμεύσει συγκεκριμένες θέσεις χαρακτήρων στο char_buffer για κάθε ID, έτσι γνωρίζοντας το ID του λαμβανόμενου δεδομένου και πολλαπλασιάζοντας με τον σταθερό αριθμό χαρακτήρων που απαιτεί το κάθε δεδομένο και προσθέτοντας του ένα κατάλληλο offset το οποίο αντιπροσωπεύει τους χαρακτήρες που απαιτούνται για να καταγραφεί η χρονική στιγμή που έχει αφιχθεί το δεδομένο, εγγράφεται στην κατάλληλη θέση του char_buffer. Επιπλέον ελέγχεται η χρονική στιγμή της άφιξης των και αν προκύψει χρονική διάφορα μεγαλύτερη του ενός λεπτού από την τελευταία εγγραφή στο αρχείο τότε το char_buffer μετατρέπεται σε String και καταγράφεται στο αρχείο.

Σε περίπτωση που το αρχείο αποθήκευσης των δεδομένων δεν υπάρχει στο σύστημα τότε δημιουργείται αυτόματα. Εάν προκύψει σφάλμα τόσο κατά την δημιουργία του όσο και κατά την λειτουργία της εφαρμογής τότε εμφανίζεται προειδοποιητικό μήνυμα στην οθόνη, ενώ παράλληλα

αποστέλλεται e-mail στον διαχειριστή του συστήματος ενημερώνοντας τον για την δυσλειτουργία που προέκυψε.



Σχήμα 23 Το *char_buffer* που χρησιμοποιείται από τον Server

4.4.6 Ο Constructor του AppletThread

Στη συνάρτηση main του Server αφού επιτευχθεί η σύνδεση με τους Collectors για την λήψη των δεδομένων, το πρόγραμμα αναμένει κάποιο RES-ADMIN Applet να επιχειρήσει να συνδεθεί σε αυτό. Μόλις αντιληφθεί ότι υπάρχει κάποιο RES-ADMIN Applet το οποίο βρίσκεται σε διαδικασία σύνδεσης δημιουργείται ένα καινούργιο client socket για το Applet και μεταφέρεται άμεσα η ροή του προγράμματος στο thread AppletThread το οποίο αναλαμβάνει να διαχειριστεί τις συνδέσεις με τα Applets. Ο constructor του AppletThread δέχεται σαν ορίσματα το socket στο οποίο έχει συνδεθεί το Applet, τον αριθμό των Collector που υπάρχουν στο σύστημα, το syncBuffer που αποθηκεύονται τα δεδομένα καθώς και την ταχύτητα μετάδοσης των δεδομένων στο Applet. Το syncBuffer χρησιμοποιείται για την αποθήκευση των δεδομένων.

Ο Server παρ' ότι εκτελεί ιδιαίτερα σημαντική αποστολή, την συγκέντρωση των δεδομένων από όλους τους Collectors και την μετάδοση τους σε όλα τα Applets δεν επιβαρύνει ιδιαίτερα το σύστημα. Σύμφωνα με τον Taskmanager (το ειδικό πρόγραμμα του λειτουργικού το οποίο καταγράφει τον φόρτο του επεξεργαστή, την χρησιμοποιούμενη μνήμη καθώς και αναλυτικές πληροφορίες για όλα τα προγράμματα που εκτελούνται) των Windows NT Server η απαιτούμενη επεξεργαστική ισχύ καθ όλη την διάρκεια εκτέλεσης του προγράμματος RES-ADMIN Server καθώς και του προγράμματος Collector δεν ξεπέρασε το 1%. Η επιβάρυνση που προκαλεί η

σύνδεση κάθε νέου RES-ADMIN Applet στον RES-ADMIN Server σύμφωνα πάλι με τον Taskmanager κυμαίνεται γύρω στα 20 Kbytes μνήμης.

4.4.7 Η μέθοδος run του AppletThread

Η μετάδοση των δεδομένων στο RES-ADMIN Applet πραγματοποιείται με την δομή `DataInputStream` η οποία χρησιμοποιεί το `InputStream` το οποίο προέρχεται από το socket το οποίο έχει δημιουργηθεί μεταξύ του RES-ADMIN Server και του RES-ADMIN Applet. Το Applet αμέσως μόλις δημιουργηθεί το socket με τον Server θα μεταδώσει μια αίτηση με την οποία θα ζητάει την εκκίνηση της μετάδοσης των δεδομένων. Ο Server μόλις δεχθεί το request (αίτηση) θα ελέγξει αν υπάρχουν δεδομένα στο `syncBuffer` προς μετάδοση. Εάν δεν υπάρχουν δεδομένα στο `syncBuffer` θα ενημερώσει εμφανίζοντας σχετικό μήνυμα στην οθόνη τον διαχειριστή. Θα πρέπει να σημειωθεί ότι η μοναδική περίπτωση να παρουσιαστεί το φαινόμενο αυτό είναι να ύπαρξη σύνδεση από κάποιο Applet πριν ακόμα αφιχθούν τα πρώτα δεδομένα από τον Collector στον Server. Από την χρονική στιγμή που θα λάβει τα πρώτα δεδομένα ο Server το `syncBuffer` θα περιλαμβάνει πάντα τα πλέον ενημερωμένα δεδομένα. Στην αναμενόμενη περίπτωση που το `syncBuffer` διαθέτει δεδομένα θα αρχίσει η μετάδοση τους. Τα δεδομένα μεταδίδονται συνεχώς προς το Applet ανά τακτά χρονικά διαστήματα ανάλογα με την ταχύτητα μετάδοσης που έχει οριστεί.

Κεφάλαιο 5

Το Πρόγραμμα παρουσίασης των δεδομένων

5.1 Σύντομη περιγραφή

Το πρόγραμμα παρουσίασης δεδομένων (**RES ADMIN-Applet**) είναι υπεύθυνο για την άρτια παρουσίαση των λαμβανομένων μετρήσεων στον τελικό χρήστη.

5.2 Γενικά χαρακτηριστικά

Η χρησιμότητα της συγκέντρωσης των δεδομένων μέσω του συστήματος RES-ADMIN από διαφορετικούς Collectors και η επεξεργασία τους από κεντρικό Server έχει αναλυθεί ήδη σε μεγάλο βαθμό, ωστόσο ιδιαίτερης σημασίας είναι και η παρουσίαση των δεδομένων αυτών στον τελικό χρήστη με άρτιο τρόπο. Ιδιαίτερα σημαντικό είναι τα δεδομένα αυτά να είναι εύκολα και άμεσα προσβάσιμα από τον τελικό χρήστη. Γίνεται αντιληπτό ότι η δημιουργία ενός προγράμματος που θα απαιτούσε από τον χρήστη να το εγκαταστήσει στο σύστημα του και να το ρυθμίσει ανάλογα θα ήταν μη βολικό, αφού ο χρήστης θα μπορούσε να το χρησιμοποιεί μόνο από έναν ηλεκτρονικό υπολογιστή, ενώ σε περίπτωση που θα επιθυμούσε να παρακολουθήσει τα δεδομένα σε διαφορετικό υπολογιστή θα έπρεπε να επαναλάβει την χρονοβόρα διαδικασία της εγκατάστασης και ρύθμισης του συστήματος, εφόσον φυσικά είχε το πρόγραμμα διαθέσιμο. Τέλος είναι σίγουρο ότι η εγκατάσταση ενός προγράμματος το οποίο θα λάμβανε δεδομένα από το διαδίκτυο, θα μπορούσε να προκαλέσει σοβαρότατα προβλήματα, υποβαθμίζοντας σημαντικά την ασφάλεια του συστήματος αλλά και θέτοντας σε κίνδυνο τα προσωπικά δεδομένα του χρήστη.

5.3 Τεχνικά χαρακτηριστικά του υποσυστήματος

Η εγκατάσταση ενός προγράμματος είναι διαδικασία χρονοβόρα και επικίνδυνη, αφού ο χρήστης δεν είναι σε θέση να γνωρίζει τι αλλαγές πραγματοποιούνται στο σύστημα του και κατά πόσο αυτές είναι επιβλαβείς. Επιπλέον η μέθοδος αυτή απαιτεί την συγγραφή ειδικού προγράμματος που να εκτελεί την αυτόματη εγκατάσταση, ενώ θα πρέπει το πρόγραμμα αυτό να τροποποιηθεί κατάλληλα ώστε να υποστηρίζει διαφορετικές πλατφόρμες. Αντίθετα η ύπαρξη του WWW (World Wide Web) προσφέρει σημαντικότερα πλεονεκτήματα σε εφαρμογές που αξιοποιούν την υποδομή του. Η περιήγηση στο διαδίκτυο πραγματοποιείται με την χρήση ενός κατάλληλου προγράμματος πλοήγησης το οποίο ονομάζεται browser. Όλοι σύγχρονοι browsers υποστηρίζουν την γλώσσα προγραμματισμού Java. Αυτό σημαίνει ότι υπάρχει η δυνατότητα ώστε εφαρμογές που είναι γραμμένες σε Java να

εκτελούνται μέσα από κάποιο browser. Τα προγράμματα αυτά απαιτείται να έχουν διαφορετική δομή προκειμένου να πληρούν τις προδιαγραφές που θέτουν τα προγράμματα πλοήγησης. Οι εφαρμογές αυτές ονομάζονται Applets. Οι περιορισμοί που υπόκεινται τα Applets αφορούν κυρίως την ασφάλεια του συστήματος. Η Sun έχει αναπτύξει μια τεχνική την οποία την ονομάζει Sandbox. Η τεχνική αυτή εγγυάται ότι τα Applets είναι σε θέση να πραγματοποιούν μόνο συγκεκριμένες μη επιβλαβείς λειτουργίες, δημιουργώντας ένας ασφαλές περιβάλλον το οποίο περιορίζει την πρόσβαση στους πόρους του χρήστη και εκτελώντας το Applet μέσα σε αυτό.

Η κατακόρυφη αύξηση των υπολογιστών που είναι συνδεδεμένοι στο διαδίκτυο καθώς και των προσφερόμενων υπηρεσιών τα τελευταία χρόνια πέραν των όσων θετικών προσφέρουν έχουν δημιουργήσει και αρκετά κενά ασφάλειας. Η τεχνική Sandbox ουσιαστικά ελαχιστοποίησε τους κινδύνους που ενέχει η χρησιμοποίηση προγραμμάτων προερχόμενων από το διαδίκτυο εάν αυτά είναι Applets. Ο χρήστης ο οποίος χρησιμοποιεί ένα Applet (όπως το RES-ADMIN Applet) είναι απόλυτα βέβαιος ότι το πρόγραμμα αυτό δεν πρόκειται σε καμία περίπτωση να

- Να διαβάσει ή να γράψει κάποιο αρχείο στον υπολογιστή του χρήστη.
- Να φορτώσει κάποια βιβλιοθήκη.
- Να εκτελέσει native κώδικα.
- Να εκτελέσει κάποιο πρόγραμμα στον υπολογιστή του χρήστη.
- Να συνδεθεί με οποιοδήποτε μηχάνημα εκτός από αυτό το οποίο προήλθε, δηλαδή τον Web Server στον οποίο φιλοξενείται η σελίδα που περιέχει το Applet.

Οι περιορισμοί αυτοί θεωρούνται ικανοποιητικοί αφού ουσιαστικά εκμηδενίζουν την περίπτωση εκτέλεσης κακόβουλου κώδικα.

Επιπλέον θα πρέπει να τονισθεί ότι η ασφάλεια των Applets είναι ιδιαίτερα υψηλή, αφού υπάρχουν και άλλες γραμμές άμυνας που εγγυώνται, στο βαθμό που αυτό είναι δυνατό, την ασφαλή λειτουργία του συστήματος. Συγκεκριμένα χρησιμοποιείται η τεχνική bytecode verifier όπου ελέγχεται ο κώδικας του Applet για την ύπαρξη μη έγκυρων εντολών. Γίνεται προσπάθεια

να εντοπιστούν ενδεχόμενες υπερχειλίσσεις ή υποχειλίσσεις της στοίβας (stack overflow, stack underflows), προσβάσεις σε μη έγκυρους καταχωρητές (registers) καθώς και αν η δομή των bytecode εντολών είναι σωστή. Με αυτό τον τρόπο θωρακίζεται το σύστημα από μια ιδιαίτερα συνηθισμένη μέθοδο εισβολής, που περιλαμβάνει την εσκεμμένη εκτέλεση μη έγκυρων εντολών, θέτοντας το σύστημα σε ασταθή λειτουργία και αποκτώντας πρόσβαση στους πόρους με αυτόν τον τρόπο.

Η απαίτηση για την μέγιστη δυνατή ασφάλεια καθώς και η ανάγκη αποφυγής της προβληματικής, όπως επεξηγήθηκε παραπάνω, διαδικασίας εγκατάστασης και χρήσης εκτελέσιμων αρχείων αντιμετωπίστηκαν υλοποιώντας το πρόγραμμα παρουσίασης δεδομένων ως Applet το οποίο παρέχει ικανοποιητική ασφάλεια και δεν απαιτεί διαδικασία εγκατάστασης.

5.4 Τεχνική ανάλυση

Το πρόγραμμα παρουσίασης δεδομένων έχει υλοποιηθεί ως Applet. Θα πρέπει να γίνει κατανοητό ότι τα Applets διαφοροποιούνται αρκετά από τα εκτελέσιμα προγράμματα. Το Applet είναι ένα παραθυροειδές αντικείμενο το οποίο εκτελείται σαν thread. Η παραθυροειδής δομή του RES-ADMIN Applet καθώς και η χρήση διαφόρων αντικειμένων ελέγχου (controls) όπως κουμπιών (buttons) περιοχών κείμενου (text area), ετικετών (labels) και άλλων συνιστούν ένα ασύγχρονο πρόγραμμα το οποίο απαιτεί την χρησιμοποίηση δομών Event-Handling για την ορθή λειτουργία του. Το γεγονός ότι ο χρήστης αλληλεπιδρά με το RES-ADMIN Applet επιβάλλει την ύπαρξη των κατάλληλων Listeners που να είναι σε θέση να παρακολουθούν τις ενέργειες του χρήστη, και να αντιδρούν ανάλογα.

5.4.1 Το Interface του RES-ADMIN Applet



Σχήμα 24 Το RES-ADMIN Applet μόλις φορτωθεί από την σελίδα του εργαστηρίου

Το Interface του προγράμματος παρουσίασης δεδομένων σχεδιάστηκε έχοντας ως πρωταρχικό στόχο το να χρησιμοποιείται ιδιαίτερα εύκολα από τον τελικό χρήστη. Επιπλέον εξίσου σημαντικό θεωρήθηκε να υλοποιηθεί το RES-ADMIN Applet με ένα τρόπο ώστε προβάλλονται οι μετρήσεις που λαμβάνονται σε πραγματικό χρόνο με τον καλύτερο εμποπτικό τρόπο. Η σελίδα του Applet μόλις φορτωθεί από τον Web Server του εργαστηρίου περιέχει 12 μη ενεργοποιημένα τετραγωνίδια επιλογής (check boxes), ενώ επίσης στο κεντρικό πεδίο του Applet υπάρχουν χρήσιμες πληροφορίες που αφορούν τους συντελεστές της εφαρμογής. Τα check boxes είναι διαφορετικού χρώματος το καθένα έτσι ώστε να ξεχωρίζουν εύκολα. Μόλις ο χρήστης μετακινήσει το ποντίκι (mouse) πάνω από κάποιο check box θα του δοθούν διάφορες πληροφορίες για το είδος του αισθητήρα που έχει χρησιμοποιηθεί καθώς και διάφορα άλλα τεχνικά χαρακτηριστικά. Αν ο χρήστης επιλέξει να ενεργοποιήσει κάποιο από τα check boxes τότε αυτόματα θα εξαφανιστούν οι

The screenshot shows a web browser window with the address bar displaying 'http://www.elsa-lab.com/res_admin.html'. The page title is 'RES-ADMIN v1.03'. The main content area is divided into two columns. The left column lists various sensor readings with status indicators (checkmarks or exclamation marks) and units. The right column shows corresponding values and units. A sidebar on the right contains navigation links: 'About', 'Data View', 'Start', and 'Tools'. At the bottom, a note reads: 'RES is Renewable Energy Sources, ADMIN is Advanced Monitoring Interface.'

Parameter	Value	Unit
Temperature	25.43 °C	°C
Humidity	1.833 W/m²	W/m²
Pressure	1.027 mbars	mbars
Power	1.234 W	W

RES is Renewable Energy Sources
ADMIN is Advanced Monitoring Interface

Στην πάνω δεξιά γωνία εμφανίζονται πληροφορίες για την ώρα που λαμβάνονται οι τιμές στην γεωγραφική περιοχή που βρίσκεται ο χρήστης σε σχέση με την ώρα που ώρα ισχύει στην γεωγραφική περιοχή που βρίσκεται ο Server του εργαστηρίου.

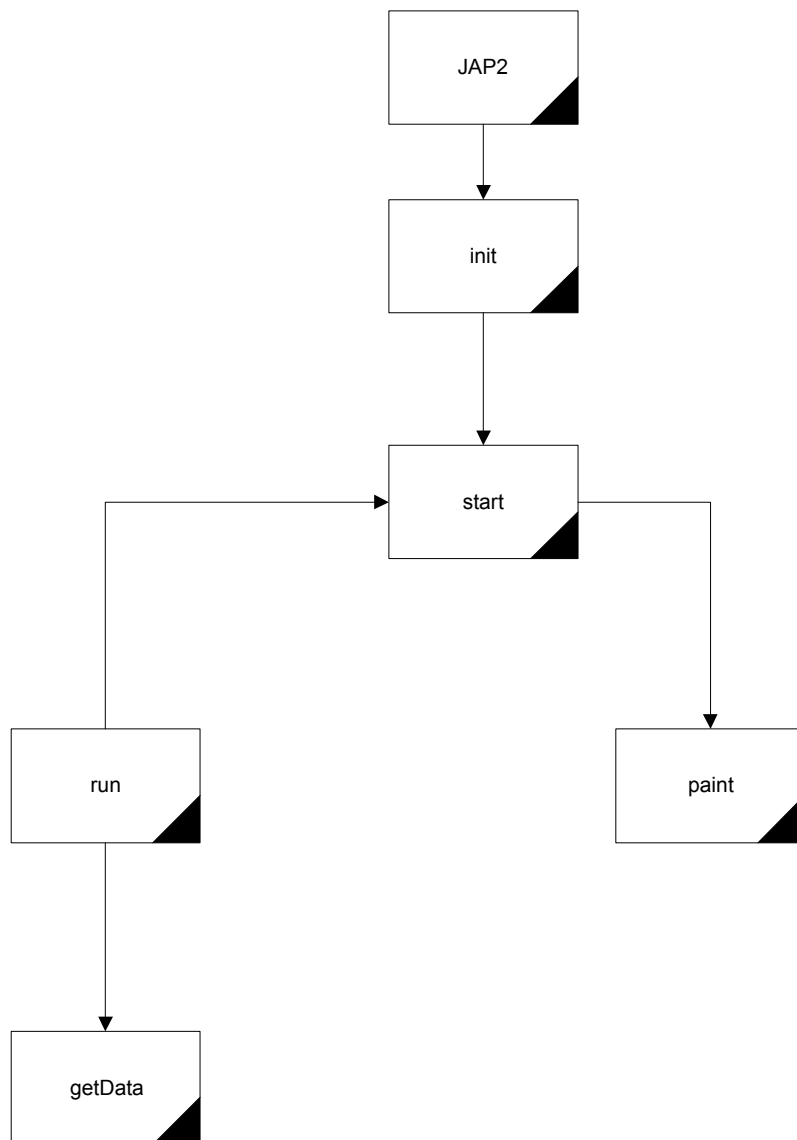
5.4.2 Η συνάρτηση JAP2

Η συνάρτηση JAP2 δεν πραγματοποιεί καμία ουσιαστική λειτουργία. Χρησιμοποιείται για να περιλάβει μια εντολή, ώστε να αποφευχθεί η εμφάνιση του μηνύματος "Swing: checked access to system event queue" σε κάποιους browsers.

5.4.3 Η συνάρτηση init

Η συνάρτηση init είναι η πλέον βασική συνάρτηση, αφού σε αυτή πραγματοποιείται η υλοποίηση των αντικειμένων που αλληλεπιδρούν με τον χρήστη (Graphic User Interface- GUI objects). Η θέση όλων των αντικειμένων, τα επιμέρους χαρακτηριστικά καθώς και η λειτουργικότητα τους ορίζονται στην συνάρτηση αυτή. Τα GUI objects για να χρησιμοποιηθούν πρέπει να ορισθούν επακριβώς στη μέθοδο init, δίνοντας τις συντεταγμένες που αφορούν την σχετική θέση του αντικείμενου στο Applet καθώς και τις ιδιότητες που αυτό πρέπει να έχει.

Κρίνεται σκόπιμο στο σημείο αυτό να γίνει μια συνοπτική αναφορά στο γραφικό περιβάλλον που παρέχει η Java. Στις αρχικές εκδόσεις της Java (JDK 1.0.x) όλες οι γραφικές λειτουργίες που πρέπει να υποστηρίζει μια βιβλιοθήκη (ή πακέτο σύμφωνα με την Sun) καλύπτονταν από το πακέτο AWT (Abstract Window Toolkit). Από την έκδοση JDK 1.1.x εξακολουθούσε να υπάρχει το AWT αλλά υπήρχε η δυνατότητα ενσωμάτωσης σαν εξωτερική βιβλιοθήκη του Swing (το Swing είναι μια γραφική βιβλιοθήκη που περιλαμβάνει αρκετά προηγμένα χαρακτηριστικά και είναι σε θέση να αντικαταστήσει σε μεγάλο βαθμό το AWT, ωστόσο το AWT περιλαμβάνει και ορισμένα επιπλέον αντικείμενα τα οποία δεν υπάρχουν στο Swing). Οι τελευταίες εκδόσεις (Java2) έχουν συμπεριλάβει πλήρως το Swing και θεωρείται πλέον βασικό κομμάτι της Java.



Σχήμα 26 Το flowchart του RES-ADMIN Applet

Το βασικό πλεονέκτημα του AWT είναι ότι επιτρέπει σε εφαρμογές που είναι γραμμένες σε Java να περιλαμβάνουν παραθυρικό περιβάλλον ανεξάρτητα του λειτουργικού στο οποίο εκτελούνται. Είναι αρκετά δύσκολο να γίνει αντιληπτό πως ένα πρόγραμμα σε εντελώς διαφορετικά σύστημα είναι δυνατόν να λειτουργεί και ανάλογα αν εκτελείται σε Windows να εμφανίζει το γραφικό περιβάλλον των Windows, ενώ αν τρέχει σε Solaris να εμφανίζει το γραφικό περιβάλλον του Solaris. Η τεχνική η οποία αναπτύχθηκε για να καταστεί αυτό εφικτό, βασίζεται στο ότι η Java χρησιμοποιεί κοινό Java bytecode για να περιγράψει τις γραφικές λειτουργίες που πρέπει να πραγματοποιηθούν. Ο κώδικας αυτός είναι κοινός για όλες τις πλατφόρμες

που υποστηρίζουν Java. Παράλληλα όμως υπάρχει και η Java runtime library η οποία παρακολουθεί την εκτέλεση του προγράμματος και μόλις αντιληφθεί τις γραφικές λειτουργίες που περιγράφει ο Java bytecode αυτόματα καλεί μια native library η οποία είναι συγκεκριμένη για κάθε λειτουργικό σύστημα και η οποία αναλαμβάνει να εμφανίσει τα γραφικά στην οθόνη. Φυσικά επειδή η native βιβλιοθήκη έχει υλοποιηθεί για το συγκεκριμένο λειτουργικό τα γραφικά αντικείμενα τα οποία εμφανίζει είναι ακριβώς αυτά που χρησιμοποιεί το λειτουργικό. Ο όρος Abstract στα αρχικά AWT αναφέρεται στο ότι η εμφάνιση του γραφικού περιβάλλοντος εξαρτάται αποκλειστικά από το λειτουργικό σύστημα στο οποίο εκτελείται η εφαρμογή.

Η υλοποίηση αυτή παρ' ότι παρείχε υποστήριξη σε πλήθος διαφορετικών πλατφόρμων, επειδή ακριβώς απαιτεί την μετάφραση του Java bytecode σε εντελώς διαφορετικά συστήματα χρησιμοποιώντας εντελώς διαφορετικές native libraries για κάθε σύστημα, εμφάνισε αρκετά προβλήματα ασυμβατότητας και άλλες δυσλειτουργίες οι οποίες μεταφέρθηκαν από το επίπεδο του λειτουργικού στο περιβάλλον εκτέλεσης της Java (Java Runtime Environment-JRE). Για το λόγο αυτό προέκυψε η ανάγκη να δημιουργηθεί ένα προγραμματιστικό περιβάλλον (API) το οποίο να είναι παντελώς ανεξάρτητο από το λειτουργικό σύστημα στο οποίο εκτελείται η εφαρμογή. Η λύση σε αυτό το πρόβλημα δόθηκε με την δημιουργία των Java Foundation Classes (JFC), οι οποίες αποτελούν ένα σύνολο κλάσεων ειδικών για γραφικά. Οι JFC αντιμετώπισαν τις δυσλειτουργίες αυτές με επιτυχία, ενώ παράλληλα εισήγαγαν και κάποια επιπλέον χαρακτηριστικά όπως:

- Κοινής μορφής γραφικά αντικείμενα ανεξάρτητα από το είδος το συστήματος στο οποίο εκτελείται η Java εφαρμογή. Επιπλέον παρέχεται η δυνατότητα στον προγραμματιστή να επιλέξει επακριβώς το γραφικό περιβάλλον με το οποίο θέλει να επενδύσει την εφαρμογή του. Είναι δυνατόν ένα πρόγραμμα το οποίο εκτελείτε σε Windows να έχει το γραφικό περιβάλλον που έχουν οι εφαρμογές του Solaris ή του Macintosh.
- Έχει δημιουργηθεί το Java 2D drawing API, το οποίο παρέχει αρκετά επιπλέον εργαλεία για την δημιουργία απαιτητικών γραφικών. Ακόμα παρέχεται API για χρήστες με ειδικές ανάγκες (π.χ προβλήματα όρασης)

ενώ έχουν προστεθεί βιβλιοθήκες που προσφέρουν “drag ‘n’ drop” δυνατότητες.

- Τέλος είναι πολύ σημαντικό να τονισθεί ότι έχει υλοποιηθεί το Swing σύνολο το οποίο αναφέρθηκε παραπάνω.

Ο τρόπος με τον οποίο λειτουργούν οι JFC είναι εξαιρετικά απλός. Ουσιαστικά η Java δεσμεύει χώρο στην οθόνη αποκλειστικά για δική της χρήση. Όλα τα αντικείμενα που πρέπει να σχεδιαστούν, σχεδιάζονται αποκλειστικά στο κομμάτι εκείνο το οποίο ανήκει στην Java. Με αυτό τον τρόπο δεν απαιτείται καθόλου η χρήση native libraries. Βέβαια έχει υλοποιηθεί μια αρκετά πολύπλοκη δομή κλάσεων που τα διάφορα αντικείμενα κληρονομούν τις ιδιότητες και την λειτουργικότητα των προγόνων τους και επίσης παρέχουν ανάλογα χαρακτηριστικά σε κάποιες υποκλάσεις που προέρχονται από αυτές, έτσι ώστε να καταστεί δυνατή η χρήση γραφικού περιβάλλοντος εντελώς ανεξάρτητου από το λειτουργικό σύστημα στο οποίο υλοποιείται η εφαρμογή. Θα πρέπει επίσης να σημειωθεί ότι μόνο για να υλοποιηθεί το σύνολο Swing χρησιμοποιήθηκαν περισσότερες από 300 κλάσεις. Τα πιο βασικά controls όπως buttons, check boxes, labels και άλλα, έχουν ενσωματωθεί στο Swing και είναι σαφώς καλύτερα, σταθερότερα και αποτελεσματικότερα από τα αντίστοιχα controls του AWT. Ωστόσο το AWT χρησιμοποιείται για να καλύψει άλλες ανάγκες όπως να παρέχει την δυνατότητα εκτύπωσης, κάτι το οποίο το Swing δεν περιλαμβάνει.

Στο σημείο αυτό θα πρέπει να γίνει αναφορά σε μια σημαντική διάκριση που υφίσταται στην Java στα heavyweight components και στα lightweight components. Κάθε αντικείμενο το οποίο εξαρτάται από το υφιστάμενο λειτουργικό σύστημα καθώς και από native βιβλιοθήκες θεωρείται heavyweight component, ενώ αντίθετα τα components τα οποία έχουν υλοποιηθεί με το σύνολο Swing θεωρούνται lightweight components. Έχει παρατηρηθεί ότι η ταυτόχρονη χρήση και ο συνδυασμός heavyweight και lightweight components σε μια εφαρμογή δημιουργεί σημαντικότερα προβλήματα απόδοσης καθώς και διάφορες δυσλειτουργίες. Για το λόγο αυτό συνίσταται να αποφεύγεται ο συνδυασμός τους.

Το RES-ADMIN Applet χρησιμοποιεί αποκλειστικά lightweight components γιατί όπως αναφέρθηκε παραπάνω είναι νεότερης τεχνολογίας, σταθερότερα προσφέρουν καλύτερη απόδοση και είναι παντελώς ανεξάρτητα από το περιβάλλον στο οποίο εκτελείται η εφαρμογή. Το Applet το οποίο έχει υλοποιηθεί έχει οριστεί να κληρονομεί όλα τα χαρακτηριστικά του JApplet. Αυτό είναι ιδιαίτερα σημαντικό διότι JApplet είναι η γενική Swing κλάση που αποτελεί το βασικό component για να κατασκευαστεί ένα Applet με το Swing σύνολο. Επιπλέον το RES-ADMIN Applet περιλαμβάνει και την εντολή implements Runnable προκειμένου να καταστεί δυνατή η threaded εκτέλεση του. Στην συνάρτηση init δίνονται οι συντεταγμένες των διάφορων controls ενώ ορίζεται και η λειτουργικότητά τους. Επίσης δημιουργούνται το client Socket για να επιτευχθεί η επικοινωνία με τον RES-ADMIN Server, ενώ υλοποιούνται και τα κατάλληλα streams για να καταστεί δυνατή η ανάγνωση και η εγγραφή των δεδομένων στο Socket. Τα stream αυτά ονομάζονται is και os είναι τύπου DataInputStream και DataOutputStream και δέχονται ως όρισμα τα InputStream και OutputStream τα οποία λαμβάνουν και στέλνουν δεδομένα στο Socket. Η τεχνική αυτή έχει συζητηθεί αναλυτικά στα προηγούμενα κεφάλαια και ειδικότερα στο Κεφάλαιο του Collector. Αφού πραγματοποιηθεί η σύνδεση με τον RES-ADMIN Server το Applet στέλνει μια αρχική αίτηση για να επιβεβαιώσει την σύνδεση του. Σε περίπτωση που προκύψει αδυναμία σύνδεσης με τον RES-ADMIN Server ή διακοπεί η σύνδεση με τρόπο μη αναμενόμενο, ο χρήστης ενημερώνεται από ειδικό μήνυμα που εμφανίζεται στη status bar του browser όπου περιγράφεται σύντομα το είδος και η αιτία του προβλήματος.

Έκτος από την επίτευξη της σύνδεσης με τον RES-ADMIN Server στην init συνάρτηση υλοποιούνται και οι Listeners οι οποίοι παρακολουθούν την εκτέλεση του προγράμματος και ελέγχουν αν ο χρήστης ενεργοποίησε κάποιο check box. Στην περίπτωση που αντιληφθούν ότι πραγματοποίησε τέτοιου είδους ενέργεια τότε, αναλαμβάνουν να αποκρύψουν το εισαγωγικό κείμενο το οποίο εμφανίζεται στο κεντρικό μέρος του Applet και να εμφανίσουν τα text fields που περιλαμβάνουν τις μετρήσεις που επιθυμεί να παρακολουθήσει. Η λειτουργία των Listeners έχει αναπτυχθεί διεξοδικά στο Κεφάλαιο του Collector.

5.4.4 Η συνάρτηση start

Η συνάρτηση start δεν καλείται πότε άμεσα από τον προγραμματιστή, αλλά καλείται αυτόματα κάθε φορά που ο browser επισκέπτεται την σελίδα που περιέχει το Applet και αναλαμβάνει να εκκινήσει ή να συνεχίσει οποίο thread περιέχει το Applet. Στην περίπτωση της εφαρμογής RES-ADMIN αναλαμβάνει να εκκινήσει ή να συνεχίσει τη λειτουργία του thread το οποίο έχει υλοποιηθεί στο RES-ADMIN Applet.

5.4.5 Η συνάρτηση run

Η συνάρτηση run είναι η συνάρτηση που περιέχει της λειτουργίες που εκτελεί το thread του RES-ADMIN Applet. Στην μέθοδο αυτή ελέγχεται αν χρειάζεται επαναδημιουργηθούν το socket και τα streams για την επικοινωνία με τον RES-ADMIN Server. Μόλις πιστοποιηθεί ότι Server και Applet είναι συνδεδεμένα το RES-ADMIN Applet μεταδίδει μια αίτηση για να αρχίσει η μετάδοση των δεδομένων από τον Server. Τα δεδομένα λαμβάνονται από μια ειδική συνάρτηση η οποία ονομάζεται getData και της οποίας η περιγραφή ακολουθεί παρακάτω. Μετά από την λήψη κάθε δεδομένου προκαλείται παύση της λειτουργίας του thread κατά 200 milliseconds προκειμένου να εξοικονομηθεί επεξεργαστική ισχύς και να μην δαπανηθούν αναίτια πόροι του συστήματος.

5.4.6 Η συνάρτηση stop

Η συνάρτηση stop καλείται είτε όταν ο χρήστης επιλέξει να τερματίσει την χρήση της εφαρμογής, είτε όταν εγκαταλείψει την σελίδα που βρίσκεται το Applet. Ακόμα η μέθοδος stop χρησιμοποιείται και σε οποιαδήποτε περίπτωση είναι επιθυμητή η αναστολή της λειτουργίας του thread του Applet. Στο RES-ADMIN Applet όταν ο χρήστης εγκαταλείψει την σελίδα που περιέχει την εφαρμογή καλείται η συνάρτηση stop προκειμένου να διακόψει την επικοινωνία με τον RES-ADMIN Server κλείνοντας το socket που είχε δημιουργηθεί για την μετάδοση των δεδομένων.

5.4.7 Η συνάρτηση getData

Η συνάρτηση `getData` αναλαμβάνει να λάβει τα δεδομένα από τον RES-ADMIN Server και να τα μετατρέψει σε πραγματικές μετρήσεις με φυσική σημασία. Όπως έχει επανειλημμένα τονιστεί η εφαρμογή RES-ADMIN είναι ένα καθαρά δυναμικό σύστημα το οποίο είναι σε θέση να μεταδώσει και να διαχειριστεί κάθε είδους δεδομένα χωρίς κανένα απολύτως περιορισμό. Τα δεδομένα μεταφράζονται σε μετρήσεις με φυσική σημασία μόνο στο Applet, έτσι με αυτό το τρόπο ελαχιστοποιούνται οι απαιτούμενες αλλαγές σε περίπτωση που χρησιμοποιηθεί το λογισμικό RES-ADMIN για μετάδοση και άλλων τύπων δεδομένων πέραν των μετεωρολογικών που μεταδίδονται τώρα. Στην συνάρτηση `getData` πραγματοποιείται η αντιστοίχιση των ID των δεδομένων με τα φυσικά μεγέθη τα οποία παρακολουθούνται μέσω του RES-ADMIN, ενώ καταγράφεται και η τιμή τους, έτσι ώστε να παρουσιαστεί στο RES-ADMIN Applet. Η μέθοδος φυσικά χρησιμοποιεί τα `streams` που έχουν δημιουργηθεί στη μέθοδο `run` για να διαβάζει από το client Socket τα δεδομένα που μεταδίδει ο RES-ADMIN Server.

5.4.8 Η συνάρτηση paint

Η συνάρτηση `paint` είναι υπεύθυνη για τον σχεδιασμό των γραφικών αντικειμένων στην οθόνη. Επίσης μέσω αυτής της συνάρτησης προστίθεται το κείμενο στα `text fields` και μέσω αυτής πραγματοποιείται και η ανανέωση του. Γενικότερα ο σχεδιασμός όλων των `controls` καθώς και η εμφάνιση σε αυτά των διαφόρων μηνυμάτων πραγματοποιείται από την μέθοδο `paint`.

Κεφάλαιο 6

Βοηθητικά Προγράμματα

6.1 Γενικά

Η εφαρμογή RES-ADMIN είναι ένα κατανεμημένο σύστημα, στο οποίο για να εξασφαλιστεί η μέγιστη δυνατή αυτονομία χρειάστηκε να υλοποιηθούν τεχνικές ανάκαμψης από μη αναμενόμενες καταστάσεις καθώς και συστήματα έγκαιρης ειδοποίησης του διαχειριστή της εφαρμογής για τυχόν δυσλειτουργίες. Επιπλέον λόγω του όγκου των προγραμμάτων, της πολυπλοκότητας τους καθώς και της δυσκολίας δοκιμής τους σε πραγματικές συνθήκες κατασκευάστηκαν κατάλληλα προγράμματα προκειμένου να

προσομοιωθούν οι καταστάσεις που θα αντιμετώπιζε το σύστημα. Τα προγράμματα αυτά χρησιμοποιήθηκαν επίσης για να δοκιμαστεί η εφαρμογή και σε εξαιρετικά ακραίες συνθήκες. Σημαντική ήταν η συνεισφορά των βοηθητικών προγραμμάτων και στον έλεγχο καλής λειτουργίας του υλικού (Hardware) αφού υλοποιήθηκαν ειδικά για να ελέγχουν ορισμένα σημαντικά για την εφαρμογή υλικά του συστήματος όπως η σειριακή θύρα. Συγκεκριμένα χρησιμοποιήθηκαν

- Ένα πρόγραμμα ειδοποίησης του διαχειριστή του RES-ADMIN.
- Ένα πρόγραμμα διαχείρισης των αρχείων του συστήματος RES-ADMIN.
- Τρία scripts για να εκκινούν τα υποπρογράμματα σε περίπτωση που αυτό καταστεί αναγκαίο.
- Τρία προγράμματα για τον έλεγχο της σειριακής θύρας (το ένα ήταν έτοιμο και το παρείχε η Sun)
- Ένα πρόγραμμα έλεγχου καλής λειτουργίας του RES-ADMIN Server σε ακραίες συνθήκες.
- Ένα πρόγραμμα έλεγχου καλής λειτουργίας του RES-ADMIN Applet (το παρείχε η Sun).

Τα παραπάνω προγράμματα βοήθησαν τα μέγιστα στο να εντοπιστούν τα οποία σφάλματα των προγραμμάτων του RES-ADMIN καθώς θα ήταν εξαιρετικά δύσκολο να γίνουν αλλαγές του συστήματος όταν αυτό βρίσκεται σε πλήρη λειτουργία, αφού σημαντικό μέρος του εξοπλισμού της εφαρμογής βρίσκεται στον πειραματικό σταθμό του τμήματος, ο οποίος είναι σε απόσταση τουλάχιστον 100 μέτρων από το εργαστήριο που βρίσκονται τα υπόλοιπα μέρη του υποσυστήματος. Όπως γίνεται εύκολα αντιληπτό η μεταφορά μέρος του εξοπλισμού από και προς το εργαστήριο προκειμένου να εντοπιστούν και να διορθωθούν τα διάφορα σφάλματα της εφαρμογής είναι μη εφικτή. Για το λόγο αυτό πριν το σύστημα τεθεί σε λειτουργία πραγματοποιήθηκαν εκτεταμένες δοκιμές καθώς και όλες οι απαραίτητες διαδικασίες ώστε να πιστοποιηθεί η καλή λειτουργία του.

6.2 Πρόγραμμα ειδοποίησης

Το πρόγραμμα ειδοποίησης mail χρησιμοποιείται τόσο από το υποσύστημα RES-ADMIN Server όσο και από το υποσύστημα Collector προκειμένου να ενημερώσει τον διαχειριστή για κάποιο σημαντικό, μη αναστρέψιμο πρόβλημα που προέκυψε. Τέτοιο πρόβλημα μπορεί να είναι:

- Η μη αναμενόμενη διακοπή της σύνδεσης του Collector με τον RES-ADMIN Server.
- Η αδυναμία του RES-ADMIN Server να καταγράψει τα λαμβανόμενα δεδομένα σε αρχείο
- Η αδυναμία του Collector να λάβει δεδομένα από την σειριακή θύρα για χρονικό διάστημα μεγαλύτερου αυτού που έχει ορίσει ο διαχειριστής του συστήματος.
- Η δέσμευση του port που έχει οριστεί να επικοινωνούν Collector RES-ADMIN Server από κάποιο άλλο πρόγραμμα,
- Η δέσμευση της σειριακής θύρα που έχει οριστεί να λαμβάνει δεδομένα ο Collector από κάποιο άλλο πρόγραμμα.
- Η αδυναμία του RES-ADMIN Server να εντοπίσει το αρχείο των ρυθμίσεων.
- Η δέσμευση του port επικοινωνίας του RES-ADMIN Server με το RES-ADMIN Applet.

Σε περίπτωση που συμβεί ένα από τα παραπάνω ενδεχόμενα το σύστημα χρησιμοποιώντας το υποπρόγραμμα mail θα αποστέλλει e-mail στον διαχειριστή περιγράφοντας το πρόβλημα το οποίο έχει δημιουργηθεί. Θα πρέπει ωστόσο να σημειωθεί ότι για να λειτουργήσει ικανοποιητικά αυτή η υπηρεσία θα πρέπει ο mail server στον οποίο θα αποσταλεί το e-mail να χρησιμοποιεί κάποιο πρόγραμμα mail βασισμένο στο SMTP πρωτόκολλο. Σε περίπτωση που ο mail server πληροί αυτές τις προϋποθέσεις ο διαχειριστής του συστήματος θα είναι σε θέση να ειδοποιείται για τις όποιες δυσλειτουργίες των προγραμμάτων μέσω e-mail.

6.3 Προγράμματα επανεκκίνησης του συστήματος

Το σύστημα RES-ADMIN κατά την δίμηνη λειτουργία του έχει επιδείξει άριστη συμπεριφορά, αφού σε καμία περίπτωση δεν παρουσιάστηκε μη αναστρέψιμο σφάλμα οφειλόμενο είτε σε θόρυβο είτε σε κάποιο πρόβλημα του λογισμικού. Θεωρήθηκε όμως απαραίτητο να υλοποιηθούν και τα κατάλληλα βοηθητικά προγράμματα, ώστε να είναι δυνατή η επαναφορά του συστήματος εάν χρειαστεί. Τα προγράμματα αυτά έχουν κατασκευαστεί σε VBScript. Η χρήση της VBScript επιτρέπει να αυτοματοποιηθεί η απαιτούμενη διαδικασία επανεκκίνησης του προγράμματος. Εκτός από την VBScript χρησιμοποιούνται επίσης αρχεία δέσμης (batch files).

Εάν η αιτία του προβλήματος δεν είναι τόσο δραστική (τερματισμός του προγράμματος με την εντολή kill) ώστε να μην επιτρέπει καμία ενέργεια, υπάρχει η δυνατότητα να εκτελεστεί κάποιο script το οποίο καλώντας ένα αρχείο δέσμης επανεκκινεί το πρόγραμμα.

6.3.1 Επανεκκίνηση του προγράμματος Collector

Σε περίπτωση που παρουσιαστεί σφάλμα το οποίο δεν είναι σε θέση να αντιμετωπιστεί από τον Collector, τότε ακριβώς πριν ο Collector τερματίσει την λειτουργία του, καλεί το script "The_last_big_thing_c.vbs". Το script αυτό θα εκτελέσει ένα αρχείο δέσμης μέσα από το dos-prompt το οποίο ονομάζεται "Gods_hand_collector.bat". Το αρχείο αυτό θα αναλάβει να επαναθέσει σε λειτουργία τον Collector μετά από κάποιο χρονικό διάστημα, ώστε να αποφευχθεί το ενδεχόμενο να μην έχουν απελευθερωθεί πλήρως οι πόροι του συστήματος. Οι προκαθορισμένες παράμετροι που οριστεί στο αρχείο Gods_hand_collector.bat είναι οι εξής

- Θύρα επικοινωνίας με τον Server 3333
- Σειριακή θύρα 2 (δηλαδή COM 2)
- Ταχύτητα μετάδοσης 1200bps
- Χρόνος ανενεργού επικοινωνίας 20 λεπτά

Οι ρυθμίσεις αυτές βέβαια είναι δυνατόν να τροποποιηθούν ώστε να καλύψουν απόλυτα τις ανάγκες του μηχανικού που χρησιμοποιεί το πρόγραμμα.

6.3.2 Επανεκκίνηση του προγράμματος Server

Η διαδικασία επανεκκίνησης του Server είναι ακριβώς η ίδια με αυτή του Collector. Εάν προκύψει σφάλμα το οποίο ο Server δεν είναι σε θέση να αντιμετωπίσει τότε τερματίζει την λειτουργία του αφού πρώτα καλέσει, μέσα από τον Java κώδικα, το script “The_last_big_thing_s.vbs”. Το script αυτό με την σειρά του θα εκτελέσει το αρχείο δέσμης “Gods_hand_server.bat”. Ακολουθώντας την διαδικασία αυτή μετά από κάποιο χρονικό διάστημα, προκειμένου να καταστεί απολύτως βέβαιο ότι θα έχουν απελευθερωθεί όλοι οι πόροι του συστήματος, ο Server θα τεθεί σε λειτουργία.

6.4 Το πρόγραμμα διαχείρισης αρχείων

Η εφαρμογή RES-ADMIN προσφέρει στον τελικό χρήστη την δυνατότητα να έχει πρόσβαση σε όλη τα δεδομένα που έχει συγκεντρώσει το σύστημα. Ο χρήστης είναι σε θέση να επιλέξει μεταξύ των μετρήσεων που έχουν ληφθεί κάποια προηγούμενη ημερομηνία. Το αρχείο αυτό φέρει ως όνομα την ημερομηνία την οποία αντιπροσωπεύει, και σε αυτό περιέχονται όλες οι μετρήσεις που έχουν ληφθεί την συγκεκριμένη ημέρα. Παρέχεται επίσης η δυνατότητα στον χρήστη να λάβει τα δεδομένα της παρούσης ημέρας, τα οποία βρίσκονται σε ένα αρχείο το οποίο περιλαμβάνει όλες τις τιμές, μέχρι τις πλέον πρόσφατες που έχουν ληφθεί από το σύστημα, ενώ επίσης κρατούνται αρχεία με τις μετρήσεις που έχουν ληφθεί ανά ώρα.

Γίνεται εύκολα αντιληπτό ότι το σύστημα συγκεντρώνει αρκετά μεγάλη ποσότητα πληροφορίας και η απλή αποθήκευση της σε καμία περίπτωση δεν θα επαρκούσε για να καλύψει τις ανάγκες της εφαρμογής RES-ADMIN. Για το λόγο αυτό κρίθηκε αναγκαία η δημιουργία κατάλληλου προγράμματος, το οποίο θα είχε ως αποκλειστικό σκοπό να αποθηκεύει και να διαχειρίζεται την πληροφορία με τέτοιο τρόπο ώστε να είναι άμεσα προσιτή στον τελικό χρήστη. Το πρόγραμμα που υλοποιήθηκε επειδή οφείλει να εκτελεί εργασίες,

όπως αντιγραφή, μετονομασία και διαγραφή αρχείων, οι οποίες σχετίζονται άμεσα με το λειτουργικό κατασκευαστικό με script. Η χρησιμοποίηση scripting γλώσσας προγραμματισμού θεωρήθηκε προτιμότερη, από μια καθαρά «προγραμματιστική» γλώσσα, αφού θα απαιτούσε περισσότερη προσπάθεια για να υλοποιηθεί με σαφώς υποδεέστερα αποτελέσματα, αφού η επικοινωνία των παραδοσιακών γλωσσών προγραμματισμού με το λειτουργικό σύστημα είναι συνήθως προβληματική. Αντιθέτως οι scripting γλώσσες επειδή είναι σχεδιασμένες για να εκτελούν εργασίες που άπτονται του λειτουργικού συστήματος είναι αποτελεσματικότερες από τις παραδοσιακές γλώσσες προγραμματισμού.

Η γλώσσα προγραμματισμού που επιλέχθηκε για την κατασκευή του προγράμματος διαχείρισης των αρχείων είναι η VBScript. Θα πρέπει στο σημείο αυτό να αναφερθεί ότι ο Web Server του εργαστηρίου είναι ο IIS ο οποίος υποστηρίζεται από τα Windows NT 4.0 Server Edition, οπότε θα πρέπει το script να λειτουργεί κάτω από αυτό το περιβάλλον. Είναι γεγονός ότι μέχρι πρότινος ήταν εξαιρετικά δύσκολη η δημιουργία scripts σε πλατφόρμα Windows, αφού δεν υπήρχε κατάλληλη γλώσσα για το λειτουργικό αυτό, σε αντίθεση με τα UNIX συστήματα που παρείχαν πολλές και εξαιρετικές δυνατότητες scripting. Η συγγραφή scripts για τα Windows περιοριζόταν στην χρήση των ξεπερασμένων αρχείων δέσμης (batch files), τα οποία όμως έχουν πλήθος περιορισμών. Η δημιουργία της VBScript από την Microsoft, συνέβαλε ουσιαστικά στο να ξεπεραστεί αυτό το εμπόδιο παρέχοντας ένα αρκετά καλό εργαλείο στους προγραμματιστές. Ωστόσο θα πρέπει να σημειωθεί ότι πρόκειται περί μίας πολύ νέας γλώσσας η οποία δεν είναι επαρκώς δοκιμασμένη, άλλα ούτε παρέχει και ικανοποιητική τεκμηρίωση και βιβλιογραφία.

Το πρόγραμμα διαχείρισης αρχείων εκτελείται ως δαίμονας “daemon”. Η χαρακτηριστική λειτουργία ενός δαίμονα είναι ότι παραμένει ανενεργός για μεγάλα χρονικά διαστήματα, ενεργοποιείται, εκτελεί τις εργασίες τις οποίες οφείλει να εκτελέσει και στην συνέχεια αυτοαπενεργοποιείται μέχρι να παρέλθει ένα προκαθορισμένο χρονικό διάστημα και να επανενεργοποιηθεί και πάλι. Το πρόγραμμα διαχείρισης, το οποίο ονομάζεται “Theater.vbs”, λειτουργεί ως εξής. Αμέσως μόλις ενεργοποιηθεί ελέγχει την ημερομηνία του

συστήματος και δημιουργεί ένα αρχείο, του οποίου το όνομα αποτελείται από την τρέχουσα ημερομηνία, την ώρα και το χαρακτηριστικό γράμμα ελέγχου “H”, το οποίο υποδηλώνει ότι πρόκειται για αρχείο το οποίο κρατάει τα δεδομένα τα οποία έχουν ληφθεί μέχρι την συγκεκριμένη ώρα. Αν δηλαδή η ημερομηνία είναι Δευτέρα, 1 Ιουλίου 2000 και ώρα είναι 22:00 το όνομα του αρχείου θα γίνει “22Monday, July 01,2000H.txt”. Ακολούθως θα αντιγράψει το περιεχόμενο του αρχείου “update.txt”, το οποίο είναι το αρχείο στο οποίο καταγράφει τα δεδομένα που λαμβάνει ο RES-ADMIN Server, στο νεοδημιουργηθέν αρχείο και θα απενεργοποιηθεί για μια ώρα περίπου. Εάν όταν επανενεργοποιηθεί το πρόγραμμα είναι νωρίτερα από τις 23:50, τότε θα επαναλάβει την ίδια διαδικασία δημιουργώντας πάλι ένα αρχείο με την συγκεκριμένη ώρα μπροστά από την ημερομηνία, βάσει της διαδικασίας που περιγράφηκε παραπάνω. Σε περίπτωση που είναι αργότερα από 23:50 τότε το πρόγραμμα ενεργοποιείται και απενεργοποιείται κάθε λεπτό, έως ότου φτάσει στις 00:00. Εκείνη την χρονική στιγμή θα δημιουργήσει ένα αρχείο που θα έχει ως όνομα την ημερομηνία δηλαδή “Monday, July 01,2000.txt” θα αντιγράψει σε αυτό όλες τις μετρήσεις που έχουν καταγραφεί από το αρχείο “update.txt”. Στην συνέχεια θα διαγράψει όλα τα αρχεία τα οποία έχουν χρησιμοποιηθεί για να καταγράφονται οι μετρήσεις ανά ώρα και τέλος θα διαγραφεί και το αρχείο “update.txt”. Όταν ο RES-ADMIN Server επιχειρήσει να καταγράψει μια καινούργια μέτρηση και διαπιστώσει ότι το αρχείο update.txt δεν υφίσταται θα το δημιουργήσει εκ νέου.

Με αυτόν τον τρόπο το πρόγραμμα διαχείρισης αρχείων παρέχει στο σύστημα την δυνατότητα να αποθηκεύει όλα τα δεδομένα που λαμβάνει σε αρχεία, τα οποία φέρουν το όνομα της αντίστοιχης ημερομηνίας στην οποία λήφθηκαν οι μετρήσεις, καθιστώντας ιδιαίτερα εύκολο των εντοπισμό και την επεξεργασία των στοιχείων που ενδιαφέρουν τον τελικό χρήστη.

6.5 Το πρόγραμμα εκκίνησης εφαρμογής RES-ADMIN

Το σύστημα RES-ADMIN όπως επανειλημμένα αναφέρθηκε παρουσίασε αξιοσημείωτη σταθερότητα αφού κατά την διάρκεια της δίμηνης λειτουργίας του, δεν εμφάνισε ούτε μια περίπτωση αστοχίας του λογισμικού,

ωστόσο παρουσιάστηκαν αρκετές γενικευμένες διακοπές ρεύματος στο Πολυτεχνείο με αποτέλεσμα να διακόπτεται η λειτουργία του. Λόγω του προβλήματος αυτού προέκυψε η ανάγκη δημιουργίας κάποιου προγράμματος το οποίο να αυτοματοποιεί την διαδικασία εκκίνησης των εφαρμογών Collector, RES-ADMIN Server καθώς και των scripts που απαιτούνται για να λειτουργήσει ομαλά το σύστημα RES-ADMIN.

Το script το οποίο αναπτύχθηκε για να αυτοματοποιήσει αυτή την διαδικασία, σε καμία περίπτωση δεν αποτελεί ξεχωριστό πρόγραμμα, αντίθετα πρόκειται για την ενσωμάτωση όλων των υποπρογραμμάτων και scripts που χρησιμοποιούνται για την διαχείριση του συστήματος, σε μια εφαρμογή η οποία αναλαμβάνει να εκκινήσει όλα τα υποσυστήματα παρέχοντας τις κατάλληλες ρυθμίσεις. Η λειτουργικότητα του script είναι προφανής αφού με τον τρόπο αυτό υπεραπλουστεύεται η διαδικασία εκκίνησης του συστήματος, ενώ ελαχιστοποιείται και η πιθανότητα από παραλήψεις ή σφάλματα άπειρου διαχειριστή να τεθεί το πρόγραμμα σε λειτουργία με λανθασμένες ρυθμίσεις ή παραλείποντας κάποιο σημαντικό υποπρόγραμμα.

Οι διαδικασίες που εκτελούνται από το script “Start_me_up.vbs” είναι οι εξής. Αρχικά εκκινείται ο Collector χρησιμοποιώντας των κώδικα που υλοποιήθηκε το “The_last_big_think_c.vbs” script. Μετά την πάροδο κάποιου ασφαλούς χρονικού διαστήματος, προκειμένου να καταστεί βέβαιο ότι ο Collector έχει τεθεί σε λειτουργία, εκκινείται ο Server, χρησιμοποιώντας αντίστοιχα των κώδικα που κατασκευάστηκε το “The_last_big_think_s.vbs” script. Το επόμενο αναγκαίο υποσύστημα για την ομαλή λειτουργία του RES-ADMIN είναι ο δαίμονας ο οποίος αναλαμβάνει να διαχειριστεί τα αρχεία της εφαρμογής RES-ADMIN. Σε αυτή την περίπτωση χρησιμοποιείται ο κώδικας του προγράμματος διαχείρισης αρχείων “Theater.vbs”.

Η κατασκευή του script “Start_me_up.vbs” είναι ιδιαίτερα σημαντική διότι επιτρέπει την αυτόματη εκκίνηση, χωρίς να είναι απαραίτητη η παρουσία φυσικού προσώπου, του συστήματος RES-ADMIN σε περίπτωση που λόγω κάποιας απρόβλεπτης κατάστασης, όπως η πτώση του ηλεκτρικού ρεύματος τερματιστεί η λειτουργία του. Η λειτουργία του συστήματος χωρίς την ανθρώπινη παρουσία αποτέλεσε υψηλή προτεραιότητα. Στόχος του

κατασκευαστή είναι ο διαχειριστής απλά να εφαρμόζει μία φορά τις αρχικές ρυθμίσεις στο σύστημα και να μην απαιτείται η περαιτέρω συμμετοχή του, εκτός αν προκύψει κάποιο πρόβλημα για το οποίο θα ενημερώνεται μέσω e-mail οπότε και θα πρέπει να προβεί στις απαραίτητες ενέργειες προκειμένου να αποκαταστήσει την ομαλή λειτουργία του προγράμματος.

Είναι σημαντικό να γίνει κατανοητό ότι η διαχείριση ενός κατανεμημένου συστήματος είναι μια εξαιρετικά πολύπλοκη διαδικασία. Η υλοποίηση ενός απόλυτα ανεξάρτητου από τον ανθρώπινο παράγοντα κατανεμημένου συστήματος θεωρείται εξαιρετικά δύσκολη έως ανέφικτη. Στην εφαρμογή RES-ADMIN δόθηκε ιδιαίτερη σημασία στην κατασκευή ενός πραγματικά σταθερού συστήματος καθώς και στην ανάπτυξη δομών οι οποίες να επιτρέπουν τόσο την άμεση ενημέρωση του διαχειριστή για την ύπαρξη πιθανών προβλημάτων όσο και αυτόματη αποκατάσταση των δυσλειτουργιών όπου αυτό είναι εφικτό.

Τέλος θα πρέπει να σημειωθεί γενικά για την χρήση των scripts ότι αποτελούν το μοναδικό κομμάτι της εφαρμογής RES-ADMIN το οποίο δεν είναι Platform Independent, ενώ εξαρτάται άμεσα από το την τοπολογία του συστήματος RES-ADMIN, δηλαδή πόσοι Collectors βρίσκονται σε λειτουργία, αν εκτελούνται στον υπολογιστή που εκτελείτε ο RES-ADMIN Server ή σε κάποιο άλλο απομακρυσμένο σύστημα. Σε αυτήν την περίπτωση θα πρέπει τα scripts να τροποποιηθούν κατάλληλα. Θα πρέπει ωστόσο να γίνει κατανοητό ότι σε περίπτωση που αποφασισθεί να χρησιμοποιηθεί η εφαρμογή RES-ADMIN σε πλατφόρμα διαφορετική από την παρούσα θα πρέπει είτε με ελάχιστο κόπο να αφαιρεθούν όλα τα scripts και η λειτουργικότητα που αυτά προσφέρουν και να επιφορτιστεί περισσότερο ο διαχειριστής με την επίβλεψη του συστήματος (αν και θα μπορεί να ενημερώνεται μέσω e-mail για τις οποίες δυσλειτουργίες), είτε να μετατραπούν τα scripts ώστε να καταστούν απολύτως συμβατά με την αντίστοιχη πλατφόρμα. Θα πρέπει ακόμα να αναφερθεί ότι τα UNIX, Linux, Solaris συστήματα διαθέτουν εξαιρετικές γλώσσες scripting πολύ ανώτερες από την VBScript των Windows.

6.6 Προγράμματα ελέγχου της σειριακής θύρας

Η σειριακή θύρα είναι το μέσο εισαγωγής δεδομένων στο σύστημα. Η καλή λειτουργία της θεωρείται βασική προϋπόθεση για την ομαλή λειτουργία του συστήματος. Είναι προφανές ότι πριν τεθεί σε λειτουργία ολόκληρη η εφαρμογή, και πριν αναπτυχθούν όλα τα υποπρογράμματα θα πρέπει να υπάρχει η βεβαιότητα ότι η σειριακή θύρα λειτουργεί απροβλημάτιστα. Για το λόγο αυτό χρησιμοποιήθηκαν και αναπτύχθηκαν τα κατάλληλα προγράμματα για τον έλεγχο της.

6.6.1 Το πρόγραμμα BlackBox

Το πρόγραμμα BlackBox παρέχεται από την Sun μαζί με το πακέτο `javax.commapi` το οποίο χρησιμοποιείται για την πρόσβαση στην σειριακή θύρα. Το πρόγραμμα αυτό είναι ιδιαίτερα χρήσιμο, αφού επιτρέπει την δέσμευση όλων των ελεύθερων θυρών του υπολογιστή και την παρακολούθηση όλων το εισερχόμενων και εξερχόμενων δεδομένων σε αυτή σε οποιαδήποτε δυνατή ταχύτητα. Το BlackBox αποτελεί το βασικό τρόπο ελέγχου της λειτουργίας της σειριακής θύρας.

6.6.2 Το πρόγραμμα SimpleRead

Το πρόγραμμα BlackBox αν και καλύπτει σε μεγάλο βαθμό την ανάγκη για παρακολούθηση των δραστηριοτήτων της σειριακής θύρας, δεν καλύπτει απόλυτα τις απαιτήσεις της εφαρμογής RES-ADMIN. Ο σχεδιασμός του πρωτοκόλλου επικοινωνίας με το LabView απαίτησε πλήθος δοκιμών και τροποποιήσεων ώστε να καταστεί δυνατή η ορθή λήψη των δεδομένων από τον Collector. Συγκεκριμένα το LabView είναι σε θέση να αποστέλλει δεδομένα μόνο με την μορφή `string`. Το πρόγραμμα Collector εκμεταλλευόμενο τις δυνατότητες που του παρείχε η Java είναι σε θέση να διαβάζει `string`. Δυστυχώς κατά τις δοκιμές που πραγματοποιήθηκαν αποδείχτηκε ότι κάτι τέτοιο δεν είναι δυνατόν, αφού εμφανίστηκαν ασυμβατότητες μεταξύ των δύο προγραμμάτων. Έτσι χρειάστηκε να πραγματοποιηθούν πιο πολύπλοκες υλοποιήσεις προκειμένου να καταστεί

εφικτή η λήψη των δεδομένων από τον Collector (λεπτομέρειες παραθέτονται στο Κεφάλαιο του Collector).

Το πρόγραμμα SimpleRead χρησιμοποιήθηκε σαν πλατφόρμα δοκιμών προκειμένου να εντοπιστούν οι κατάλληλες μετατροπές ώστε να επιτευχθεί η επιθυμητή συμβατότητα. Το SimpleRead, το οποίο αποτελεί ένα μικρό μέρος του προγράμματος Collector, εμφανίζει τα λαμβανόμενα δεδομένα στην οθόνη. Η διαδικασία αυτή αποδείχθηκε ιδιαίτερα χρήσιμη αφού παρέχει την δυνατότητα της άμεσης παρακολούθησης των μετατροπών που πραγματοποιούνται, καθώς και των αποτελεσμάτων που επιφέρουν.

6.6.3 Το πρόγραμμα SimpleWrite

Η εφαρμογή RES-ADMIN δεν προβλέπεται να στέλνει δεδομένα μέσω της σειριακής θύρας παρά μόνο να λαμβάνει. Ωστόσο κρίθηκε αναγκαία η κατασκευή ενός προγράμματος το οποίο θα εγγράφει στην σειριακή θύρα τόσο για την πραγματοποίηση των απαιτούμενων ελέγχων καλής λειτουργίας της όσο και για την υλοποίηση των αναγκαίων προσομοιώσεων.

Το SimpleWrite παρέχει την δυνατότητα στον μηχανικό που αναπτύσσει την εφαρμογή να μεταδίδει μέσω της σειριακής θύρας στον Collector κάθε πιθανή μορφή δεδομένων. Είναι ιδιαίτερα σημαντικό να υποβληθεί το σύστημα σε εκτεταμένους ελέγχους έτσι ώστε να πιστοποιηθεί ότι δεν πρόκειται να προκύψει αστοχία του σε οποιαδήποτε περίπτωση εσφαλμένων ή αλλοιωμένων δεδομένων. Η εφαρμογή RES-ADMIN λειτουργεί σε ιδιαίτερα εχθρικό περιβάλλον και η αλλοίωση των δεδομένων από τον θόρυβο θεωρείται αναμενόμενη, είναι λοιπόν αναγκαίο να διασφαλιστεί ότι ακόμα και τα πλέον κατεστραμμένα από τον θόρυβο δεδομένα δεν θα προκαλέσουν πρόβλημα στον Collector, δεδομένου ότι πολλά συστήματα είναι δυνατόν να καταρρεύσουν σε περίπτωση που δεχθούν δεδομένα τα οποία δεν είναι έγκυρα.

Το πρόγραμμα SimpleWrite παρέχει επίσης την δυνατότητα να πραγματοποιηθούν οι κατάλληλες προσομοιώσεις. Συγκεκριμένα είναι δυνατόν να εκτελείται κανονικά το πρόγραμμα Collector και να λαμβάνει τα δεδομένα από την σειριακή θύρα, τα οποία θα προέρχονται από το

SimpleWrite. Με αυτόν τον τρόπο η προσομοίωση στην οποία μπορεί να υποβληθεί το σύστημα είναι πλήρης αφού περιλαμβάνει όλα τα στάδια της εφαρμογής RES-ADMIN, από την λήψη των δεδομένων από την σειριακή θύρα ως την παρουσίαση τους στο RES-ADMIN Applet.

6.7 Το Δοκιμαστικό πρόγραμμα Light

Ο έλεγχος και η πιστοποίηση ενός κατακευκμένου συστήματος όπως το RES-ADMIN είναι ιδιαίτερα δύσκολη και επίπονη διαδικασία, αφού πρέπει να δοκιμαστούν εκτεταμένα όλα τα υποσυστήματα. Στην περίπτωση του Collector απαιτήθηκε να ελεγχθεί εκτεταμένα η καλή λειτουργία της σειριακής θύρας και για αυτό χρησιμοποιήθηκαν τα κατάλληλα προγράμματα BlackBox, SimpleRead και SimpleWrite. Στην περίπτωση του RES-ADMIN Server υπάρχει η απαίτηση για επιβεβαίωση της καλής συνεργασίας του τόσο με τους Collectors όσο και με τα RES-ADMIN Applets.

Η ανάγκη αυτή οδήγησε στην κατασκευή του προγράμματος Light το οποίο αποτελεί ένα απλοποιημένο υποσύνολο του προγράμματος Collector. Συγκεκριμένα το Light δεν ασχολείται καθόλου με την σειριακή θύρα. Αντίθετα η χρησιμότητα του εστιάζεται στην παραγωγή όσων τυχαίων αποτελεσμάτων και με οποίο ρυθμό επιθυμεί ο μηχανικός που υλοποιεί την εφαρμογή. Αξιοποιώντας το πρόγραμμα Light είναι εφικτό να πραγματοποιηθεί πλήθος απαραίτητων δοκιμών, όπως την σύνδεση παραπάνω του ενός Collectors στον RES-ADMIN Server (στην ουσία δεν συνδέονται Collectors άλλα προγράμματα τύπου Light, κάτι το οποίο όμως δεν αλλοιώνει την αξιοπιστία της δοκιμής αφού πέραν της πρόσβασης στην σειριακή θύρα η λειτουργικότητα τους είναι ακριβώς η ίδια με αυτήν του Collector). Σε δοκιμές που πραγματοποιήθηκαν συνδέθηκαν μέχρι και τρεις διαφορετικοί Collectors-Light χωρίς να παρουσιαστεί κανένα απολύτως πρόβλημα.

Ενδιαφέρον παρουσίασε η μελέτη της συμπεριφοράς του RES-ADMIN Server κάνοντας εντελώς διαφορετικές υποθέσεις. Εξετάσθηκε τόσο η συνηθισμένη περίπτωση κατά την οποία ο κάθε Collector μεταδίδει εντελώς διαφορετικά δεδομένα και τα οποία βέβαια λαμβάνονται και υφίστανται επεξεργασία ως συνηθισμένες μετρήσεις, όσο και η περίπτωση κατά την

οποία όλοι οι Collectors μεταδίδουν ιδίου τύπου δεδομένα. Η παράδοση αυτή φαινομενικά προσπάθεια αποδείχτηκε ιδιαίτερα χρήσιμη, αφού αυτό που τελικά κρατούσε και αξιοποιούσε ο Server ήταν η τελευταία λαμβανόμενη μέτρηση. Κάτι τέτοιο σίγουρα μπορεί να μην έχει ιδιαίτερη σημασία σε μια μετεωρολογική εφαρμογή, θα μπορούσε όμως να έχει τεράστια χρησιμότητα για εντελώς διαφορετικά είδη εφαρμογών, όπως η παρακολούθηση ενός αντικείμενου από έναν ικανοποιητικό αριθμό αισθητήρων όπου ο Server θα μπορεί να ανιχνεύσει την τροχιά του αντικείμενου. Η υλοποίηση ενός τέτοιου συστήματος που θα αποτελούσε μια εναλλακτική λύση του ραντάρ θα απαιτούσε μικρές ή ίσως και καθόλου αλλαγές στον υπάρχον λογισμικό. Θα πρέπει να τονιστεί ότι και στις δύο περιπτώσεις που προαναφέρθηκαν, η απόδοση τόσο του Server όσο και του Collector ήταν ικανοποιητικότερη και δεν παρουσιάστηκε κανένα πρόβλημα στην λειτουργία τους.

Το δεύτερο απαραίτητο πεδίο των δοκιμών πέρα από την ανταπόκριση του Server σε πλήθος συνδεδεμένων σε αυτόν Collectors είχε να κάνει με την απόκριση του Server κάτω από δυσμενείς συνθήκες και οι πιθανές δυσλειτουργίες που θα προέκυπταν από τέτοιου είδους καταστάσεις. Το πρόγραμμα Light επιτρέπει στον μηχανικό που αναπτύσσει την εφαρμογή να αυξήσει υπερβολικά την ταχύτητα αποστολής των δεδομένων από τον Collector στον Server και να παρατηρήσει τις αντιδράσεις του. Θα πρέπει να αναφερθεί ότι πολλά χρήσιμα συμπεράσματα και αρκετές μετατροπές πραγματοποιήθηκαν στο πρόγραμμα RES-ADMIN κατά το στάδιο των δοκιμών αυτών.

6.7 Το Πρόγραμμα ελέγχου Applets

Η χρησιμοποίηση του Applet στην εφαρμογή RES-ADMIN προσέφερε πολλά στην λειτουργικότητά της. Ωστόσο η κατασκευή των Applets εμπεριέχει και αρκετούς περιορισμούς, ενώ προϋποθέτει και την σωστή συνεργασία τους με τους Web browsers. Σε περίπτωση που κατά την διαδικασία δοκιμής ενός Applet εμφανιστούν κάποια προβλήματα είναι εξαιρετικά δύσκολο να προσδιοριστεί με βεβαιότητα εάν τα προβλήματα αυτά προέρχονται από σφάλματα στην κατασκευή του Applet ή αντίθετα οφείλονται σε κάποια

ασυμβατότητα του browser που μπορούσαν εύκολα να επιλυθούν απλά αναβαθμίζοντας σε κάποια καινούργια έκδοση του συγκεκριμένου browser.

Το πρόγραμμα Appletviewer είναι κατασκευασμένο από την Sun και περιέχεται σε όλες τις εκδόσεις του JDK. Το Appletviewer αναλαμβάνει να εξετάσει μια σελίδα HTML και αν εντοπίσει σε αυτή κάποιο Applet το εκτελεί. Η σημασία του Appletviewer έγκειται στο ότι προέρχεται από την Sun και η μηχανή Java (JVM) που χρησιμοποιεί είναι κατασκευασμένη από την Sun, ώστε να γίνεται άμεσα αντιληπτό αν ο κώδικας είναι προβληματικός ή όχι. Σε περίπτωση που κώδικας λειτουργεί στο Appletviewer είναι σίγουρο ότι τα οποία προβλήματα παρουσιαστούν οφείλονται στον Web browser. Είναι δυνατόν διάφοροι browser να χρησιμοποιούν μηχανές Java οι οποίες να μην απόλυτα συμβατές με αυτή της Sun ή να χρησιμοποιούν πεπαλαιωμένες εκδόσεις της. Για το λόγο αυτό είναι απαραίτητο να υπάρχει ένα πρότυπο το οποίο να χρησιμοποιείται σαν αναφορά. Το Appletviewer αποτελεί αυτό ακριβώς το πρότυπο και συνίσταται από τον κατασκευαστή κάθε τροποποίηση στον κώδικα του RES-ADMIN Applet να ελέγχεται πρώτα σε αυτό.

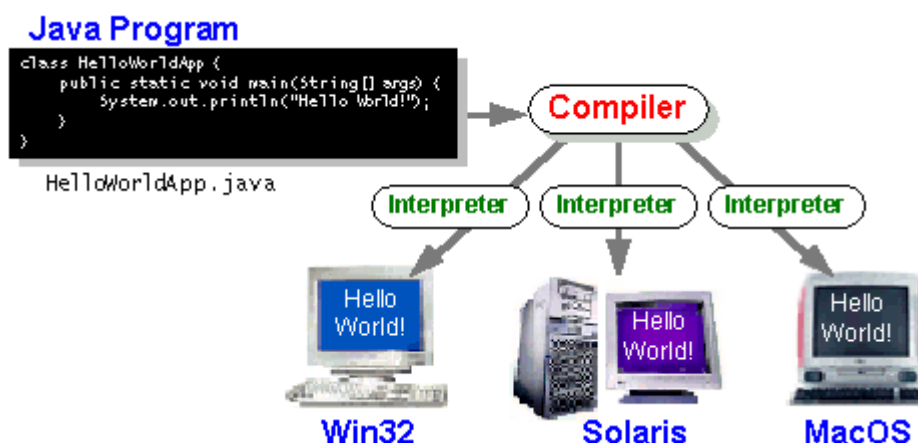
Κεφάλαιο 7

Εγχειρίδιο Χρήσης

7.1 Γενικά

Η εγκατάσταση των προγραμμάτων Collector, Server και RES-ADMIN Applet δεν παρουσιάζει σημαντικές δυσκολίες. Στο παρόν κείμενο θεωρείται σαν δεδομένη πλατφόρμα το πρότυπο Wintel (Windows/Intel). Θα πρέπει να σημειωθεί ότι ο όρος Windows αναφέρεται στις εκδόσεις Windows 95, Windows 98, Windows NT 4.0 της Microsoft, ενώ ο όρος Intel δεν αφορά αποκλειστικά τους επεξεργαστές της εταιρίας Intel αλλά καλύπτει γενικά την αρχιτεκτονική i386 περιλαμβάνοντας όλους τους συγχρόνους επεξεργαστές

που την ακολουθούν. Σε κάθε περίπτωση οι επεξεργαστές των εταιριών Intel, Cyrix, AMD από την αρχιτεκτονική 486 και μεταγενέστερα μπορούν να θεωρηθούν απόλυτα συμβατές με τις προδιαγραφές του προγράμματος. Η εφαρμογή δημιουργήθηκε και πιστοποιήθηκε κάνοντας χρήση των παραπάνω λειτουργικών και υλικών ωστόσο δοκιμές που πραγματοποιήθηκαν και σε άλλες πλατφόρμες όπως το Linux παρουσίασαν πολύ καλή συμπεριφορά της εφαρμογής, ενώ αναμένεται και σε πλατφόρμες όπως το Solaris ή σε άλλα τύπου UNIX συστήματα να μην εμφανιστούν δυσλειτουργίες. Για το λόγο αυτό θα δοθούν όπου χρειαστεί και ειδικές πληροφορίες και για τις υπόλοιπες πλατφόρμες.

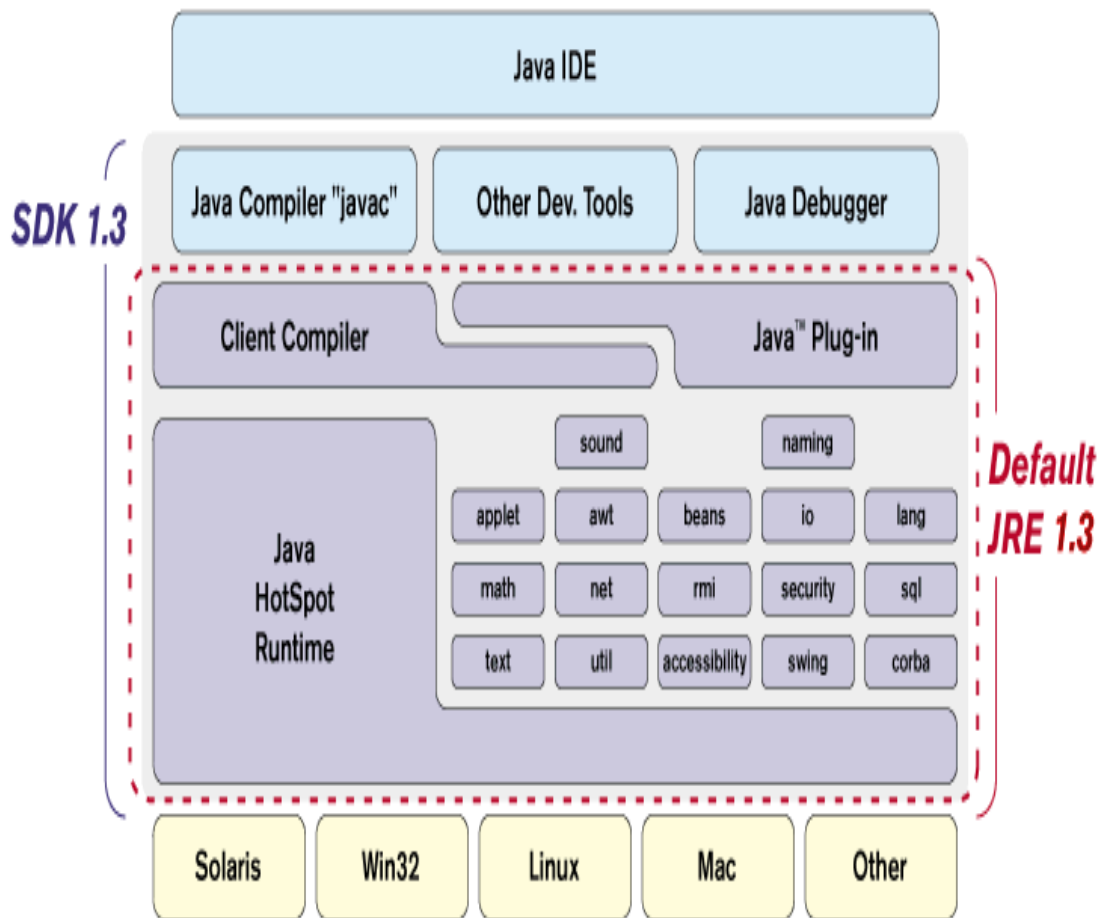


Σχήμα 27 Πλατφόρμες στις οποίες έχει λειτουργήσει ή αναμένεται να λειτουργήσει χωρίς προβλήματα η εφαρμογή RES-ADMIN

Το προγράμματα Collector, RES-ADMIN Server και RES-ADMIN Applet μπορούν να εγκατασταθούν σε οποιονδήποτε υπολογιστή ο οποίος πληροί τις προϋποθέσεις που αναφέρθηκαν παραπάνω. Θα πρέπει επίσης να έχει εγκατασταθεί το Java Development Kit ή αλλιώς JDK το οποίο μπορεί να βρεθεί στον δικτυακό τόπο της Sun. (<http://java.sun.com>) ή το Java Runtime Environment ή αλλιώς JRE, διαθέσιμο και αυτό από τον ίδιο δικτυακό τόπο. Συνιστάται από τον κατασκευαστή να προτιμηθεί η έκδοση Java2 version1.3. Τα προγράμματα αναμένεται να εκτελεστούν απροβλημάτιστα και με τις

προηγούμενες εκδόσεις. Η έκδοση 1.2.2 έχει χρησιμοποιηθεί και δοκιμαστεί εκτεταμένα και δεν παρουσίασε καμία απολύτως δυσλειτουργία. Οι εκδόσεις Java version 1.1.x έχουν δοκιμασθεί μερικώς κατά το αρχικό στάδιο κατασκευής της εφαρμογής και δεν παρατηρήθηκαν προβλήματα, ωστόσο επειδή δεν έχουν ενσωματώσει όλες τις τεχνολογίες που χρησιμοποιήθηκαν στην εφαρμογή όπως για παράδειγμα την τεχνολογία SWING απαιτείται από τον μηχανικό να ακολουθήσει αρκετά πολύπλοκες διαδικασίες προκειμένου να ενσωματώσει τις τεχνολογίες αυτές. Δεν υφίσταται κανένας απολύτως λόγος για να επιλεγεί η έκδοση αυτή. Τέλος όσον αφορά τις εκδόσεις 1.0.x θα πρέπει να τονιστεί ότι είναι ιδιαίτερα απαρχαιωμένες, υπολείπονται ιδιαίτερα των μεταγενέστερων και σε καμία περίπτωση δεν μπορούν να δοθούν εγγυήσεις για την ομαλή λειτουργία των προγραμμάτων της εφαρμογής RES-ADMIN σε αυτές.

Θα πρέπει να σημειωθεί ότι προκειμένου να μεγιστοποιηθεί η απόδοση του συστήματος RES-ADMIN συνιστάται ιδιαίτερα η επιλογή της εκδόσεως Java2 version1.3 η οποία προσφέρει σημαντικότερα πλεονεκτήματα. Σημαντικότερο όλων είναι αναμφίβολα είναι η ενσωμάτωση της τεχνολογίας Java HotSpot Client VM. Η τεχνολογία αυτή αυξάνει δραστικά την απόδοση του συστήματος. Συγκεκριμένα μετρήσεις που έγιναν από την Sun έδειξαν αύξηση της τάξεως του 40% στον χρόνο εκκινήσεων των εφαρμογών καθώς και μείωση κατά 25% της δεσμευμένης από το σύστημα μνήμης.



Σχήμα 28. Αναπαράσταση της Java 2 Platform, Standard Edition v 1.3

Η αύξηση αυτή κάθε άλλο παρά αμελητέα είναι αφού επιτρέπει την καλύτερη αξιοποίηση του υλικού. Ειδικότερα στην περίπτωση του προγράμματος RES-ADMIN Server είναι σημαντικότερη η βελτίωση αυτή. Ο RES-ADMIN Server όπως έχει γίνει είδη κατανοητό εκτελείται σε ένα μηχάνημα το οποίο λειτουργεί παράλληλα και ως Web Server τουλάχιστον. Γίνεται εύκολα αντιληπτό ότι πρόκειται για ένα ιδιαίτερα επιβαρημένο μηχάνημα και συνεπώς θα πρέπει ο επιπλέον φόρτος που θα δεχτεί από το πρόγραμμα RES-ADMIN Server να μην είναι ιδιαίτερα μεγάλος. Σύμφωνα με τις μετρήσεις που πραγματοποιήθηκαν από τον γράφοντα η επεξεργαστική ισχύς που απαιτήθηκε από το πρόγραμμα RES-ADMIN Server είναι ικανοποιητικά χαμηλή και κυμαίνεται στο 0% με 1%.

Εκτός από το Server σημαντικοί περιορισμοί στην κατανάλωση πόρων του συστήματος υπάρχουν και από το πρόγραμμα RES-ADMIN Applet. Το

RES-ADMIN Applet επειδή εκτελείται μέσω ενός προγράμματος πλοήγησης του διαδικτύου (browser) υπάρχει η απαίτηση να μην είναι μεγάλο σε μέγεθος προκειμένου να μεταφέρεται από τον Web Server στον υπολογιστή του χρήστη σύντομα αλλά και να μην απαιτείται υπερβολική επεξεργαστική ισχύ. Είναι γεγονός ότι στους τομείς αυτούς παρατηρούνται ακόμα αρκετές αδυναμίες όχι μόνο στο Applet της εφαρμογής ADMIN αλλά γενικότερα για όλα τα Applets για αυτό συνιστάται η χρήση της τελευταίας έκδοσης της Java που εμφανίζεται βελτιωμένη σε σχέση με τις προηγούμενες.

Δευτερευόντως η έκδοση 1.3 προσφέρει και κάποια ακόμη πλεονεκτήματα που αφορούν βελτιώσεις στον τομέα της ασφάλειας καθώς και την ενσωμάτωση κάποιων τεχνολογιών για την αποτελεσματικότερη δημιουργία λογισμικού για κατανεμημένα συστήματα. Για τους ανωτέρω λόγους συνίσταται η χρήση της έκδοσης 1.3.

Η εγκατάσταση της Java 2 platform, Standard Edition (J2SE) πραγματοποιείται πολύ γρήγορα και εύκολα, με ειδική αυτοματοποιημένη διαδικασία. Μετά την ολοκλήρωση της εγκατάστασης το σύστημα είναι σε θέση να εκτελεί και να μεταγλωττίζει (compile) Java προγράμματα.

7.2 Εκτέλεση του προγράμματος Collector

Ειδικά για το πρόγραμμα Collector απαιτείται και η εγκατάσταση της εξωτερικής βιβλιοθήκης Java Communications API το οποίο είναι διαθέσιμο από τον δικτυακό τόπο της Sun. Το πακέτο αυτό περιλαμβάνει λεπτομερείς οδηγίες για την εγκατάσταση του. Το Java Communications API υποστηρίζει τις πλατφόρμες Windows και Solaris σύμφωνα με την Sun, ενώ παράλληλα με κάποιες τροποποιήσεις υποστηρίζει και το Linux. Θα πρέπει ωστόσο να σημειωθεί ότι στην πλατφόρμα του Linux δεν υπάρχει επίσημη πιστοποίηση από την Sun. Όσον αφορά την υποστήριξη της πλατφόρμας των Windows θα πρέπει να τονισθεί ότι λειτουργήσε απρόσκοπτα στα Windows 98 καθώς και στα Windows NT 4.0, αντίθετα δεν κατέστη δυνατό να λειτουργήσει στα Windows 2000 καθώς δεν μπορούσε να εντοπίσει τις σειριακές θύρες του συστήματος.

Ο μηχανικός που επιθυμεί να εγκαταστήσει τον Collector σε Linux ή άλλο UNIX σύστημα συνίσταται να επισκεφτεί τον δικτυακό τόπο <http://www.frii.com/~jarvi/rxtx/> όπου περιλαμβάνονται πληροφορίες για την διαδικασία μετατροπής της βιβλιοθήκης ώστε να καταστεί συμβατή με την UNIX/Linux πλατφόρμα. Η πλατφόρμα Solaris υποστηρίζεται πλήρως από την Sun απαιτείται όμως να γίνει μια μικρή μετατροπή στον κώδικα προκειμένου να καταστεί το πρόγραμμα συμβατό με τον εν λόγω λειτουργικό σύστημα. Συγκεκριμένα στην συνάρτηση main θα πρέπει η εντολή «if portID.getName().equals("COM"+sport))» να αντικατασταθεί από την εντολή «if (portID.getName().equals("/dev/term/a"))» η οποία έχει σημειωθεί ως σχόλιο προκειμένου να μην προκαλεί πρόβλημα στην εκτέλεση του προγράμματος. Ο λόγος αυτής της παρέμβασης είναι ο διαφορετικός τρόπος με τον οποίο ονομάζονται η θύρες στις πλατφόρμες win/Solaris-Linux-UNIX καθιστώντας ανέφικτη την δημιουργία κώδικα απόλυτα συμβατού με όλες τις πλατφόρμες.

Η διαδικασία εκτέλεσης (αφού πραγματοποιηθούν οι παραπάνω ενέργειες για να επιτευχθεί συμβατότητα με τα υπόλοιπα λειτουργικά συστήματα) είναι κοινή για όλες τις πλατφόρμες. Ανοίγοντας ένα παράθυρο εντολών (MS-DOS prompt για Windows ή XTerm για UNIX/LINUX) ο μηχανικός δίνει την εντολή «java Collector 3333 2 1200 10». Τα ορίσματα τα οποία ακολουθούν το όνομα του προγράμματος Collector σαφώς μπορούν να προσαρμοσθούν κατάλληλα στις ανάγκες του διαχειριστή του συστήματος και αντιπροσωπεύουν τα εξής

- Θύρα στην οποία πραγματοποιείται η σύνδεση του συστήματος. Η θύρα αυτή καλό είναι να λαμβάνει τιμές μεγαλύτερες του 1000, αφού χαμηλότερες τιμές συνήθως αντιπροσωπεύουν λειτουργίες του συστήματος.
- Σειριακή θύρα στην οποία είναι συνδεδεμένο το module το οποίο εισάγει τα δεδομένα στο σύστημα. Συνήθως λαμβάνει τιμές από 1 έως 4.
- Ταχύτητα λήψης δεδομένων σε bps (bits per seconds). Η τιμή αυτή πρέπει να συμφωνεί απόλυτα με την ταχύτητα αποστολής των δεδομένων από το σύστημα μετάδοσης. Προτεινόμενες ταχύτητες

300,600,1200,2400 bps. Καμία εγγύηση δεν μπορεί να δοθεί ότι διαφορετικές ταχύτητες θα λειτουργήσουν.

- Μέγιστος χρόνος ανενεργούς λειτουργίας. Σε περίπτωση που για κάποιο μη αναμενόμενο λόγο το σύστημα αδυνατεί να λάβει δεδομένα διατάσσεται να διακόψει την λειτουργία του και παράλληλα να επανεκκινήσει, ενώ ενημερώνει και τον διαχειριστή του συστήματος.

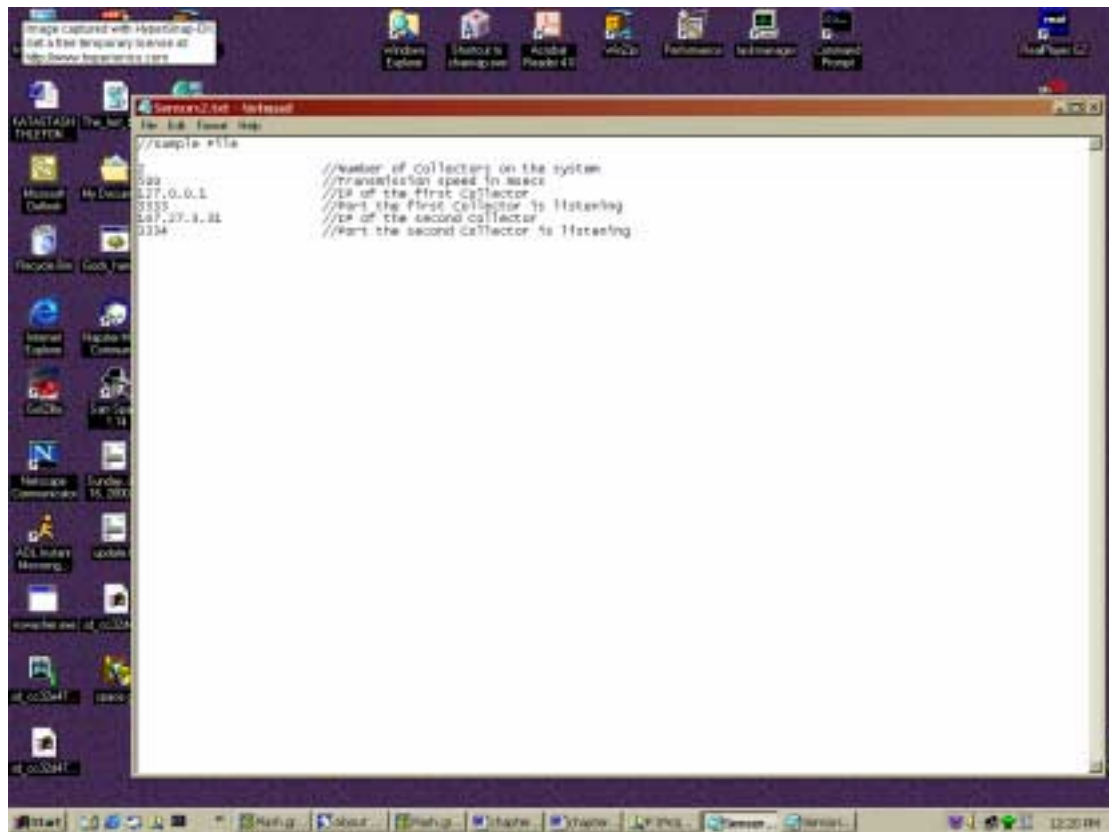
Εκτεταμένη συζήτηση για τον τρόπο λειτουργίας των παραπάνω παραμέτρων έχει πραγματοποιηθεί στο Κεφάλαιο 3 στην τεχνική ανάλυση του Collector.

7.3 Εκτέλεση του προγράμματος RES-ADMIN Server

Το πρόγραμμα RES-ADMIN Server δεν απαιτεί καμία ιδιαίτερη διαδικασία για να εκτελεστεί σε όλες τις Java συμβατές πλατφόρμες. Η εκτέλεση του πραγματοποιείται μέσω κάποιου παραθύρου DOS, όπως και ο Collector και καλείται ως εξής «java Server Sensors1.txt» Το μοναδικό όρισμα που δέχεται είναι ένα αρχείο τύπου text το οποίο περιλαμβάνει όλες τις ρυθμίσεις που χρειάζεται ο Server. Το αρχείο περιλαμβάνει τουλάχιστον 4 παραμέτρους.

- Η πρώτη παράμετρος δηλώνει τον αριθμό των Collectors που είναι συνδεδεμένοι στο σύστημα.
- Η δεύτερη παράμετρος ορίζει την ταχύτητα μετάδοσης των δεδομένων στο δίκτυο.
- Η τρίτη παράμετρος το IP του Collector στο δίκτυο
- Η τέταρτη παράμετρος την θύρα (port) στην οποία αναμένει σύνδεση ο Collector

Τα δύο τελευταία ορίσματα θα πρέπει να επαναληφθούν ανάλογα με τον αριθμό των Collectors στο σύστημα αφού είναι αυτονόητο ότι πρέπει να είναι γνωστό το IP και η θύρα που αναμένει σύνδεση ο κάθε Collector.



Σχήμα 29. Δείγμα αρχείου ρυθμίσεων με τις κατάλληλες επεξηγήσεις.

7.4 Εκτέλεση του RES-ADMIN Applet

Το RES-ADMIN Applet όπως είναι αναμενόμενο εκτελείται κατευθείαν μέσα από το browser του χρήστη χωρίς αυτός να χρειαστεί να κάνει απολύτως τίποτα. Μόλις επιλέξει να φορτώσει την σελίδα της εφαρμογής RES-ADMIN τότε το Applet θα ξεκινήσει να λειτουργεί αυτόματα. Η μοναδική απαίτηση που υπάρχει από μέρος του χρήστη είναι να υπάρχει εγκατεστημένο στο σύστημα του το Java Plug In. Το Java Plug In είναι ένα εργαλείο το οποίο επιτρέπει στους browser να εκτελούν προγράμματα Java τα οποία περιλαμβάνουν αρκετά προηγμένα χαρακτηριστικά, όπως η δυνατότητα παρακολούθησης Applets τα οποία έχουν υλοποιηθεί χρησιμοποιώντας την τεχνολογία Swing. Εάν ο χρήστης δεν διαθέτει αυτό το Plug In του παρέχεται η δυνατότητα είτε να κατεβάσει από τον Server του εργαστηρίου Ηλεκτρικών Κυκλωμάτων και Ανανεώσιμων Πηγών Ενέργειας <http://www.elci.tuc.gr/> είτε να μεταφερθεί αυτόματα στον δικτυακό τόπο της

Sun απ όπου θα υπάρχει η τελευταία έκδοση του Java Plug In διαθέσιμη. Θα πρέπει να σημειωθεί ότι σύμφωνα με τις ανακοινώσεις των κύριων κατασκευαστών browser (Netscape, Microsoft) αλλά και σύμφωνα με τις τελευταίες beta εκδόσεις των εν λόγω προγραμμάτων (Communicator 6.0) υπάρχει πλήρης υποστήριξη στις τελευταίες εκδόσεις της γλώσσας Java χωρίς να απαιτείται η ύπαρξη Plug In, καθιστώντας ακόμα πιο εύκολη την χρήση του RES-ADMIN Applet.

Κεφάλαιο 8

Συμπεράσματα

Θα πρέπει να γίνει κατανοητό ότι η εφαρμογή RES-ADMIN είναι γραμμένη σε Java προκειμένου να είναι διαθέσιμη με όλα τα λειτουργικά συστήματα και αρχιτεκτονικές. Σημαντικές δυσλειτουργίες ή ασυμβατότητες δεν αναμένονται να υπάρξουν, ωστόσο λόγω του ότι πρόκειται για ένα ιδιαίτερα πολύπλοκο έργο το οποίο περιλαμβάνει τόσο λειτουργίες πολύ χαμηλού επιπέδου (low level) οι οποίες μέχρι στιγμής δεν έχουν χρησιμοποιηθεί ευρύτατα (πρόσβαση στην σειριακή θύρα, χρησιμοποιώντας Java) όσο και τα πλέον προηγμένα χαρακτηριστικά της γλώσσας Java (χρησιμοποίηση JApplet Swing) είναι δυνατόν σε περίπτωση που επιχειρηθεί να εκτελεστεί σε τελείως διαφορετική πλατφόρμα να απαιτηθεί να γίνουν

κάποιες μικρές τροποποιήσεις όχι τόσο στον κώδικα που έχει υλοποιηθεί όσο σε επιμέρους ρυθμίσεις του λειτουργικού συστήματος καθώς και να χρειαστεί να γίνει σωστή και πλήρης εγκατάσταση του JDK και των απαιτούμενων επεκτατικών πακέτων.

Το σύστημα RES-ADMIN επέδειξε αξιόλογη σταθερότητα, παραμένοντας σε λειτουργία για δύο ολόκληρους μήνες επί 24ώρου βάσεως. Δεν παρουσιάσθηκε στις τελικές εκδόσεις καμία απολύτως αστοχία οφειλόμενη σε πρόβλημα του λογισμικού ή του υλικού. Η κεντρική σελίδα του εργαστηρίου στην οποία φιλοξενείται η εφαρμογή RES-ADMIN δέχτηκε παραπάνω από διακόσιες επισκέψεις από χρήστες εκτός του Πολυτεχνείου Κρήτης. Επιπλέον πραγματοποιήθηκε μεγάλος αριθμός δοκιμών σε όλα τα υποσυστήματα του RES-ADMIN και δεν προκλήθηκε σε καμία περίπτωση κατάρρευση του συστήματος ή μη αναμενόμενη λειτουργία του. Για τις δοκιμές αυτές χρησιμοποιήθηκαν τόσο λογισμικό κατασκευασμένο στο εργαστήριο ειδικά για αυτό τον σκοπό, όσο και υλικό που προκαλεί αποδεδειγμένα θόρυβο όπως ισχυρά συστήματα ενδοσυνεννόησης. Η εφαρμογή θα παραμείνει μόνιμα σε λειτουργία, ενώ μελλοντικός στόχος είναι και η προσθήκη νέων υποσταθμών στο σύστημα. Το σύστημα RES-ADMIN χρησιμοποιείται πλέον επαγγελματικά σε καθημερινή βάση από ραδιοφωνικό σταθμό παρέχοντας με αυτό ακριβή μετεωρολογική ενημέρωση στους ακροατές του.

Σε περίπτωση εμφάνισης κάποιας αστοχίας οφειλόμενης σε κάποια άγνωστη αιτία το σύστημα RES-ADMIN πραγματοποιεί όλες τις απαιτούμενες ενέργειες προκειμένου να επανέλθει σε ομαλή λειτουργία. Η διαδικασία που ακολουθείται είναι η επανεκκίνηση των υποσυστημάτων καθώς και η ενημέρωση του διαχειριστή μέσω e-mail. Τέλος θα πρέπει ακόμα να σημειωθεί ότι έχει αυτοματοποιηθεί πλήρως η διαδικασία εκκίνησης του συστήματος καθώς και η διαχείριση των παραγόμενων αρχείων, χωρίς να απαιτείται η παρακολούθησή του από κάποιον άνθρωπο.

Κεφάλαιο 9

Μελλοντικές Επεκτάσεις

Το σύστημα RES-ADMIN μπορεί να θεωρηθεί πλήρες, αφού καλύπτει σχεδόν συνολικά τις ανάγκες συλλογής και μετάδοσης μετεωρολογικών δεδομένων μέσω του Internet σε πραγματικό χρόνο. Οι υπηρεσίες αυτόματης ενημέρωσης του διαχειριστή, αυτόματης επανεκκίνησης του συστήματος σε περίπτωση αστοχίας υλικού ή λογισμικού αλλά κυρίως η δυναμικότητα που δίδει την εφαρμογή RES-ADMIN παρέχοντας την δυνατότητα για σύνδεση πολλαπλών Collectors, καθώς και η δυναμική υποστήριξη δεδομένων καθιστούν τις οποίες μελλοντικές επεκτάσεις χρήσιμες άλλα όχι και καθοριστικές.

Αντίθετα ουσιαστικότερη θεωρείται η προσπάθεια αξιοποίησης του λογισμικού RES-ADMIN σε διαφορετικές εφαρμογές. Η δομή του λογισμικού είναι τέτοια που μπορεί βάσιμα να υποστηριχτεί ότι ακόμη και σε εντελώς διαφορετικές εφαρμογές οι ελάχιστες αλλαγές που θα απαιτηθούν περιορίζονται μόνο στο πρόγραμμα παρουσίασης δεδομένων για λόγους οι οποίοι είναι προφανείς.

Το λογισμικό RES-ADMIN θα μπορούσε άμεσα να προσφέρει λύσεις σε τηλεϊατρικές εφαρμογές, όπου όλα τα απαιτούμενα ιατρικά όργανα (καρδιογράφοι, πιεσόμετρα) συνδέονται σε κάποιο Η/Υ και οι λαμβανόμενες μετρήσεις παρέχονται σε πραγματικό χρόνο στους γιατρούς που παρακολουθούν την εξέταση ή την εγχείριση. Ακόμα τεραστία ευκολία θα προσέφερε η εφαρμογή RES-ADMIN και σε διάφορους οργανισμούς και ιδρύματα τα οποία ασχολούνται με φυσικά φαινόμενα. Θα μπορούσε να αναφερθεί σαν παράδειγμα ή μελέτη της σεισμικής δραστηριότητας μίας χώρας στην οποία όπως είναι φυσικό οι διάφοροι σταθμοί είναι διεσπαρμένοι γεωγραφικά. Η χρησιμοποίηση του λογισμικού RES-ADMIN είναι προφανές ότι θα καθιστούσε εφικτή την συγκέντρωση όλων των μετρήσεων στον κεντρικό RES-ADMIN Server και από εκεί την παρακολούθηση τους σε πραγματικό χρόνο, μέσω του RES-ADMIN Applet, από όλους τους ενδιαφερόμενους ερευνητές.

Ιδιαίτερα χρήσιμη θα μπορούσε να χαρακτηριστεί η μελέτη της λειτουργίας του RES-ADMIN σε εναλλακτικές συνθήκες. Όπως έδειξαν περιορισμένες δοκιμές που πραγματοποιήθηκαν, η αποστολή δεδομένων ίδιου τύπου από διαφορετικούς Collectors στον ίδιο Server δημιουργεί ένα καινούργιο πλέγμα εφαρμογών. Σε αυτή την περίπτωση θα μπορούσε πιθανόν να χρησιμοποιηθεί το σύστημα RES-ADMIN για να ανιχνευθεί η θέση κάποιου κινούμενου αντικείμενου. Ο κάθε Collector θα λαμβάνει τα δεδομένα όχι περιοδικά αλλά μόλις ο αισθητήρας καταγράψει μια αλλαγή και θα τα αποστέλλει άμεσα στον RES-ADMIN Server. Με αυτόν τον τρόπο ο RES-ADMIN Server όντας ενημερωμένος με τις τελευταίες μετρήσεις θα είναι σε θέση να εξάγει τα απαιτούμενα συμπεράσματα για την θέση του κινητού στο συγκεκριμένο παράδειγμα ή ότι άλλο χρειαστεί.

Τέλος για να διατηρηθεί το σύστημα στην αιχμή της τεχνολογίας θα πρέπει να ενσωματωθούν τα νέα πρότυπα που καθιερώνονται στο χώρο της ηλεκτρονικής, της πληροφορικής αλλά και των τηλεπικοινωνιών. Σε συνεργασία με τους μηχανικούς υλικού (Hardware Engineers) είναι απαραίτητη η υποστήριξη της GSM τεχνολογίας, ενώ θα πρέπει να θεωρηθεί αναμενόμενη στην επόμενη έκδοση του RES-ADMIN η υποστήριξη της Java Micro Edition η οποία θα επιτρέπει την εκτέλεση μιας τροποποιημένης έκδοσης του RES-ADMIN Applet σε κινητά τηλέφωνα και PDAs. Με αυτόν τον τρόπο το σύστημα RES-ADMIN είναι βέβαιο ότι θα εκπληρώσει τους στόχους που αρχικά τέθηκαν, δηλαδή την υλοποίηση μιας υπηρεσίας δυναμικής παροχής δεδομένων σε πραγματικό χρόνο, προσιτής σε όλους τους χρήστες που διαθέτουν οποιαδήποτε συσκευή συνδεδεμένη στο δίκτυο.

Βιβλιογραφία

1. Just Java-Peter van der Linden
2. Java sourcebook, A complete gui-Anuff Ed
3. Επικοινωνίες δεδομένων-Αλεξόπουλος Α
4. Internet Security professional, Atkins Derek
5. The visual J++ handbook, Maso Brian
6. Java 2 networking, Couch Justin
7. The mythical man month essays-Brooks Frederick
8. Java for the World Wide Web-Dori Smith