

*Πανεπιστήμιο Κρήτης
Σχολή Θετικών Επιστημών
Τμήμα Επιστήμης Υπολογιστών*

*Χρήση αυτόνομων οντοτήτων λογισμικού στη διαχείριση πόρων ενός συστήματος
επείγουσας προνοσοκομειακής ιατρικής*

Ζαμπούλης Ξενοφών

Μεταπτυχιακή Εργασία

Ηράκλειο Ιούνιος 1998

Χρήση αυτόνομων οντοτήτων λογισμικού στη διαχείριση πόρων ενός συστήματος επείγουσας προνοσοκομειακής ιατρικής

Ζαμπούλης Ξενοφών

Μεταπτυχιακή Εργασία
Τμήμα Επιστήμης Υπολογιστών
Πανεπιστήμιο Κρήτης

Περίληψη

Επίκεντρο της εργασίας είναι ο σχεδιασμός και η ανάπτυξη ενός πληροφοριακού συστήματος το οποίο θα δρα επικουρικά στη διαχείριση πόρων και διεργασία αντιμετώπισης περιστατικών ενός συστήματος επείγουσας προνοσοκομειακής ιατρικής. Επιπλέον στόχος της εργασίας είναι η ανάπτυξη υπηρεσιών προστιθέμενης αξίας πάνω στην πληροφορία που διαθέτει το σύστημα, έτσι ώστε να γίνει δυνατός ο υπολογισμός προβλέψεων οι οποίες αφορούν τη μελλοντική συμπεριφορά του συστήματος και στη συνέχεια την παροχή συμβουλευτικής βοήθειας, προς το χρήστη, με στόχο τη βελτίωση της απόδοσης της διεργασίας αντιμετώπισης περιστατικών. Για την επίτευξη των παραπάνω στόχου χρησιμοποιούνται μέθοδοι μηχανικής μάθησης για τον προσδιορισμό περιοχών υψηλής συχνότητας εμφάνισης επειγόντων περιστατικών με βάση το ιστορικό του συστήματος, ενώ παρουσιάζεται και αλγόριθμος εύρεσης συντομότερης διαδρομής σε γράφο, ο οποίος λαμβάνει υπ' όψιν ιδιαίτερα χαρακτηριστικά του συγκεκριμένου γράφου, όπως υπογράφοι οι οποίοι παρουσιάζουν δενδρική μορφή. Συμπληρωματικό εργαλείο του συστήματος είναι ο προσομοιωτής του συστήματος επείγουσας προνοσοκομειακής ιατρικής περίθαλψης, ο οποίος αναπτύχθηκε για τους σκοπούς της εργασίας. Κύριος ερευνητικός στόχος της εργασίας είναι η μελέτη αρχιτεκτονικών αυτόνομων πρακτόρων και τρόπων απόδοσης αυτόνομης και ευφυούς συμπεριφοράς σε οντότητες λογισμικού, πάνω στην οποία βασίζεται ο σχεδιασμός του συστήματος το οποίο αναπτύχθηκε. Ακόμη η εργασία αυτή στοχεύει στον καθορισμό ενός γενικότερου πλαισίου ορισμού μιας αυτόνομης οντότητας και στη δημιουργία βιβλιοθήκης που να το υποστηρίζει, δίνοντας τη δυνατότητα δημιουργίας οντοτήτων, μέσω της εξειδίκευσης του πλαισίου.

Επόπτης Καθηγητής:

Στέλιος Ορφανουδάκης
Καθηγητής, Τμήμα Επιστήμης Υπολογιστών, Πανεπιστήμιο Κρήτης

Use of autonomous software entities in the resource management of an emergency prehospital medical system

Zamboulis Xenophon

Master Thesis
Computer Science Department
University of Crete

Abstract

The project to be presented deals with the design and implementation of an information system that acts in an auxiliary way in the resource management and emergency episode handling of an emergency pre hospital medical care system. Additional goal of the project is the development of added value services on the information that the system handles, so that it would calculate estimations on the future behavior of the system and assist the user, in order to improve the performance of emergency episode handling. For the achievement of the previous goal machine learning methods are being used in order to locate regions that have a history of presenting emergency episodes, as well as a shortest path algorithm that takes into consideration certain characteristics of the graph, such as tree structured subgraphs. The system uses a simulator of the emergency pre hospital medical care system which has been developed for the goals of the project. Main research goal of the project is the study of autonomous agent architectures and ways of developing software entities that present autonomous and intelligent behavior. The system that has been developed is based on this study. The project is oriented in defining a framework that defines an autonomous software entity and in creating a software library that supports it, giving the ability of creating such entities by specializing it.

Supervisor:

Stelios Orphanoudakis

Professor, Computer Science Department, University of Crete

Περιεχόμενα

Περίληψη

Abstract

1 Εισαγωγή.....	10
Ορισμός προβλήματος	10
2 Αυτόνομοι πράκτορες (Autonomous agents).....	12
2.1 Ορισμοί - Συζήτηση.....	12
2.2 Χαρακτηριστικά αυτόνομων όντων	13
2.3 Αρχιτεκτονική συστημάτων αυτόνομων όντων και απαιτήσεις	14
2.3.1 Αυτονομία	14
2.3.2 Προσαρμοσιμότητα	15
2.3.3 Ανεκτικότητα σε βλάβες – Καλή λειτουργία (Robustness).....	15
2.3.4 Αντίληψη – Γνώση – Δράση (Perception – Cognition – Action)	15
2.4 Σύγκριση με την τεχνητή νοημοσύνη.....	17
2.5 Σχεδιασμός και υλοποίηση.....	18
2.5.1 Βασικές αρχές σχεδιασμού.....	18
2.5.2 Χαρακτηριστικά σχεδιασμού.....	20
2.5.3 Υλοποίηση.....	21
3 Η τεχνολογία αυτόνομων οντοτήτων ως τεχνική σχεδίασης λογισμικού.....	24
4 Μοντέλο μιας τυπικής οντότητας	28
4.1 Λομή οντότητας.....	28
4.1.1 Αισθητήρια	28
4.1.2 Γνωσιακό τμήμα.....	29
4.1.3 Αντικείμενα δράσης.....	31
4.2 Προγραμματιστική διεπιφάνεια χρήσης.....	31
5 Ανάθεση στόχου σε πράκτορα.....	32
5.1 Σχέδια δράσης.....	32
5.2 Εφαρμογή σχεδίων δράσης.....	32
5.3 Πλεονεκτήματα και λόγοι χρήσης των σχεδίων δράσης	33
5.4 Συμπεράσματα	34
6 Σχεδίαση ενός πράκτορα με παραπάνω του ενός στόχου.....	35
6.1 Πλεονεκτήματα κατανεμημένης σχεδίασης.....	35
6.2 Ζητήματα σχεδίασης	36
6.3 Αντικρουόμενοι στόχοι και προτεραιότητες.....	38
6.4 Υλοποίηση	39
7 Εφαρμογή στο προνοσοκομειακό σύστημα υγείας.....	41
7.1 Χαρακτηριστικά συστήματος.....	41
7.2 Ορισμός στόχων του συστήματος.....	43

7.3 Κατασκευή μοντέλου περιβάλλοντος	43
8 Παρουσίαση οντοτήτων	45
8.1 Αισθητήρες	45
8.1.1 Γενική περιγραφή	45
8.1.2 Σχεδιασμός	46
8.1.3 Υλοποίηση	46
8.1.4 Αισθητήριο αντίληψης και παρακολούθησης εξέλιξης περιστατικών	47
8.1.5 Αισθητήριο παρακολούθησης κινητών μονάδων	48
8.1.6 Αισθητήριο παρακολούθησης βάσης δεδομένων	51
8.2 Γνωσιακό επίπεδο	51
8.2.1 Στόχοι	51
8.2.2 Σχεδίαση	51
8.2.3 Δύο όψεις του γνωσιακού επιπέδου	52
8.2.4 Παρουσίαση μηχανισμού διαχείρισης σχεδίων δράσης των οντοτήτων	56
8.2.5 Οντότητες εύρεσης καταλληλότερου πόρου	57
8.2.6 Οντότητες υλοποίησης στόχων	60
8.3 Αντικείμενα δράσης	66
9 Μεθοδολογία απλοποίησης γράφου και εύρεσης συντομότερων διαδρομών σε αυτόν	68
9.1 Εισαγωγή	68
9.2 Πρώτη μέθοδος	69
9.2.1 Απλοποίηση γράφου	69
9.2.2 Υπολογισμός συντομότερης διαδρομής	70
9.3 Δεύτερη μέθοδος	72
9.3.1 Απλοποίηση γράφου	72
9.3.2 Υπολογισμός συντομότερης διαδρομής	74
9.3.3 Βελτιστοποίηση μεθόδου	75
9.4 Σύνδεση με το υπόλοιπο σύστημα	76
10 Αλγόριθμος εντοπισμού περιοχών υψηλής συχνότητας εμφάνισης περιστατικών	77
10.1 Αλγόριθμος	77
10.2 Γενική τοποθέτηση προβλήματος	77
10.3 Χαρακτηρισμός παραδειγμάτων	77
10.4 Υπολογισμός ελάχιστου συνδετικού δέντρου	79
10.5 Επαναληπτικός διαχωρισμός του ελάχιστου συνδετικού δέντρου	80
10.6 Βελτιστοποίηση (κλάδεμα) του δέντρου ταξινόμησης	81
11 Περιβάλλον Εφαρμογής	82
11.1 Εισαγωγή	82
11.2 Εργαλείο κατασκευής αρχείων σεναρίων	83
11.2.1 Σκοπός	83
11.2.2 Δομή αρχείου σεναρίου	83
11.2.3 Λειτουργικότητα εργαλείου κατασκευής αρχείων σεναρίου	83
11.2.4 Υλοποίηση	85
11.3 Εξομοιωτής	86
11.3.1 Σκοπός	86
11.3.2 Λειτουργικότητα	87
11.3.3 Αρχιτεκτονική	87

12 Μελλοντική εργασία.....	90
12.1 Χρήση γλώσσας επικοινωνίας πρακτόρων μεταξύ των οντοτήτων	90
12.2 Παραγωγή σχεδίων δράσης	90
12.3 Σχεδίαση οντότητας ελέγχου σημάτων κυκλοφορίας.....	90
12.4 Βελτίωση αλγορίθμου ευρετηριασμού του χάρτη του οδικού δικτύου.....	91
13 Συμπεράσματα.....	92
Βιβλιογραφία	94
Παράρτημα Α Πειραματικά αποτελέσματα	97
Απλοποίηση γράφου	97
Παράδειγμα 1	97
Παράδειγμα 2	98
Εύρεση συντομότερης διαδρομής.....	100
Εύρεση περιοχών υψηλής συχνότητας εμφάνισης περιστατικών.....	104
Παράδειγμα 1	105
Παράδειγμα 2	106
Υπολογισμός περιπολιών.....	106

1 Εισαγωγή

Ορισμός προβλήματος

Η εργασία αυτή ασχολείται με το σχεδιασμό και την ανάπτυξη ενός πληροφοριακού συστήματος το οποίο θα δρα επικουρικά στη διεργασία αντιμετώπισης περιστατικών ενός συστήματος επείγουσας προνοσοκομειακής ιατρικής περίθαλψης καθώς και στη διαχείριση των πόρων του συστήματος αυτού. Επιπλέον στόχος της εργασίας είναι η ανάπτυξη υπηρεσιών προστιθέμενης αξίας πάνω στην πληροφορία που διαθέτει το σύστημα, έτσι ώστε να γίνει δυνατή η κατασκευή προβλέψεων που αφορούν τη μελλοντική συμπεριφορά του συστήματος και στη συνέχεια η παροχή συμβουλευτικής βοήθειας, προς το χρήστη, με στόχο τη βελτίωση της απόδοσης της διεργασίας αντιμετώπισης περιστατικών. Τέλος μέρος της εργασίας αποτελεί η απεικόνιση της διαχειριζόμενης από το σύστημα πληροφορίας, όσο και των εξαγομένων συμπερασμάτων. Ως μέτρο απόδοσης του συστήματος θεωρούμε τόσο το χρόνο απόκρισης του όσο και την αντιστοίχιση που κάνει μεταξύ αιτήσεων και πόρων οι οποίοι τελικά διατίθενται.

Επιπρόσθετος στόχος της εργασίας είναι η μελέτη αρχιτεκτονικών αυτόνομων πρακτόρων και οι τρόποι απόδοσης αυτόνομης και ευφυούς συμπεριφοράς σε τέτοια συστήματα. Ακόμη η εργασία αυτή στοχεύει στον καθορισμό ενός γενικότερου πλαισίου ορισμού μιας αυτόνομης οντότητας και στη δημιουργία βιβλιοθήκης που να το υποστηρίζει, δίνοντας τη δυνατότητα δημιουργίας οντοτήτων, μέσω της εξειδίκευσης του πλαισίου.

Περιγραφή προσέγγισης

Η προσέγγιση η οποία ακολουθήθηκε στο σχεδιασμό του συστήματος, είναι αυτή των αυτόνομων πρακτόρων. Η λόγοι οι οποίοι μας οδήγησαν σε μια τέτοια επιλογή ήταν:

- Το ενδιαφέρον για μελέτη της τεχνικής
- Η ανάγκη ευφυούς διαχείρισης μεγάλου όγκου δεδομένων
- Η ύπαρξη πολλαπλών και, ορισμένες φορές, αντικρουόμενων στόχων
- Η ανάγκη εξυπηρέτησης πολλών αιτήσεων παράλληλων περιστατικών καθώς και ο συνδυασμός τους
- Η πολυπλοκότητα της εργασίας
- Η προσδοκία λήψης πρωτοβουλιών από το σύστημα

Το σύστημα το οποίο αναπτύχθηκε αποσκοπεί στη βελτίωση της ποιότητας των υπηρεσιών ενός συστήματος επείγουσας προνοσοκομειακής ιατρικής περίθαλψης καθώς και στην καλύτερη διαχείριση των πόρων του. Ως πόρος ορίζεται κάθε επαναχρησιμοποιήσιμο, σχετικά σταθερό κομμάτι υλικού ή λογισμικού ενός συστήματος το οποίο είναι χρήσιμο στους χρήστες ή στις διεργασίες τις οποίες αυτοί εκτελούν [1]. Οι πόροι είναι δυνατόν να ταξινομηθούν σε φυσικούς και λογικούς. Οι φυσικοί πόροι αποτελούνται από τα υλικά τμήματα του συστήματος ενώ οι λογικοί σε συλλογές δεδομένων, υπηρεσίες κ.α.

Δεδομένης της ιατρικής φύσης της εφαρμογής, προέχει η ποιότητα των υπηρεσιών προς τους αιτούντες και κατά δεύτερο λόγο μας ενδιαφέρει η οικονομικότερη διαχείριση των πόρων. Επιπλέον πρέπει να έχουμε υπ' όψιν πως το προνοσοκομειακό σύστημα επείγουσας ιατρικής περίθαλψης, αποτελεί ένα καταμεμημένο σύστημα, του οποίου οι πόροι είναι δυνατόν να μετακινούνται γεωγραφικά (πχ. ασθενοφόρα). Ωστόσο για να είμαστε σε θέση να βελτιώσουμε το επίπεδο των προσφερομένων υπηρεσιών, θα πρέπει να έχουμε στη διάθεση μας όσο το δυνατόν περισσότερους πόρους του συστήματος. Έτσι το σύστημα θα πρέπει να προγραμματίζει τόσο την οργάνωση όσο και τη διάθεση των πόρων, αλλά και τη δυνατότητα και καταλληλότητα χρήσης τους. Επίσης το σύστημα θα πρέπει να ελέγχει τη χρήση του κάθε πόρου, τόσο λόγω κανόνων πρόσβασης αλλά και λόγω πολιτικής χρήσης τους, όπως και επίσης να είναι βέβαιο για τη λειτουργία και διαθεσιμότητα τους και όταν κάτι τέτοιο δεν συμβαίνει να υποδεικνύει το γεγονός αυτό. Η προσέγγιση μας στη διαχείριση πόρων μπορεί να χαρακτηριστεί από τα εξής σημεία:

- Τμηματοποίηση των προβλημάτων σε μικρότερα και απλούστερα και ανάθεση τους σε εξειδικευμένες για αυτό το σκοπό οντότητες
- Απουσία κεντρικού σημείου ελέγχου του συστήματος και άρα κεντρικό σημείο αποτυχίας του (στη συνέχεια της εργασίας θα δούμε και άλλους λόγους για την επιλογή μας αυτή, οι οποίοι αποσκοπούν στην ανάπτυξη ευφυούς ή τουλάχιστο αυτόνομης συμπεριφοράς από το σύστημα)
- Πρόβλεψη για αίτηση πόρων στο εγγύτερο μέλλον, έτσι ώστε να είναι δυνατή η διάθεση τους στο ελάχιστο δυνατό χρονικό διάστημα
- Αποφυγή σφαλμάτων, συμπεριφορά η οποία είναι απαραίτητη για σύστημα ζωτικής σημασίας όπως τα πληροφοριακά συστήματα υγείας

2 Αυτόνομοι πράκτορες λογισμικού (Autonomous software agents)

Στην ενότητα αυτή θα παρουσιάσουμε την βιβλιογραφική έρευνα η οποία έγινε προκειμένου να καταλάβουμε τις βασικές έννοιες της τεχνολογίας των αυτόνομων πρακτόρων λογισμικού (η πρακτόρων για συντομία). Θα ορίσουμε την έννοια του αυτόνομου πράκτορα όπως μπορεί να βρεθεί αυτή στη βιβλιογραφία και θα παρουσιάσουμε τα βασικά χαρακτηριστικά του, ενώ για πιο απλές οντότητες οι οποίες παρουσιάζουν αυτονομία στη συμπεριφορά θα υιοθετήσουμε τον όρο οντότητα λογισμικού. Στη συνέχεια θα παρουσιάσουμε ζητήματα και μεθοδολογίες σχεδιασμού αρχιτεκτονικών αυτόνομων όντων. Η όλη αναφορά αυτή, χρησιμεύει έτσι, ώστε να καταλήξουμε στα στοιχεία σχεδίασης τα οποία θα επιλέξουμε να χρησιμοποιήσουμε στην αρχιτεκτονική μας και τις μεθοδολογίες υλοποίησης που θα διαλέξουμε να χρησιμοποιήσουμε.

2.1 Ορισμοί - Συζήτηση

Η έννοια του αυτόνομου όντος είναι αρκετά ασαφής. Θα προσπαθήσουμε να την προσδιορίσουμε παρουσιάζοντας ορισμούς από τη βιβλιογραφία και στη συνέχεια ορίζοντας το πλαίσιο στο οποίο θα αναφερόμαστε εμείς σε ένα ον ως αυτόνομο, σε αυτή την εργασία. Μιας και η έννοια του αυτόνομου όντος δεν είναι πλήρως καθορισμένη στη βιβλιογραφία, είναι δυνατόν κανείς να βρει πληθώρα ορισμών:

“Ένα αυτόνομο ον είναι ένα σύστημα το οποίο προσπαθεί να ικανοποιήσει ένα σύνολο στόχων σε ένα πολύπλοκο, δυναμικό περιβάλλον. Μπορεί να αντιληφθεί το περιβάλλον μέσω των αισθητήρων (sensors) του και να δράσει σε αυτό χρησιμοποιώντας δρώντα αντικείμενα (actuators)”

“Ένας αυτόνομο ον είναι μια οντότητα λογισμικού η οποία μπορεί να λειτουργήσει αυτόνομα, εκτελώντας ενέργειες και αντιπροσωπεύοντας το χρήστη”

“Ένας αυτόνομο ον είναι ένα κομμάτι λογισμικού που υλοποιεί την αντιπροσώπευση του χρήστη. Η τεχνολογία των αυτόνομων όντων είναι περισσότερο προγραμματιστική τεχνική παρά μια νέα τεχνολογία η ένα προϊόν”

Οι τρεις παραπάνω ορισμοί περιγράφουν τόσο τη συμπεριφορά ενός αυτόνομου όντος όσο και στοιχεία της αρχιτεκτονικής του (πχ. ύπαρξη αισθητήρων). Γενικά καταλαβαίνουμε πως ένα αυτόνομο ον είναι μια οντότητα λογισμικού στην οποία ο χρήστης αναθέτει μια η παραπάνω εργασίες και περιμένει από αυτή να τις επιτελέσει με τον τρόπο που θα επέλεγε και αυτός (ή τουλάχιστο με τρόπο με τον οποίο έχει ορίσει αυτός στην οντότητα ως σωστό). Υπό άλλη οπτική γωνία μπορεί να πει κανείς πως ο χρήστης αναθέτει στο αυτόνομο ον να επιτελέσει τον έλεγχο ή την εργασία, αντ’ αυτού και για αυτό το λόγο οι οντότητες αυτές χαρακτηρίζονται ως *αυτόνομοι πράκτορες (autonomous agents)*.

Η ανάπτυξη των αυτόνομων όντων, έχει άμεση σχέση με την ανάγκη χειρισμού σύνθετων εφαρμογών από τον υπολογιστή με όσο το δυνατόν λιγότερη αλληλεπίδραση με το χρήστη. Δηλαδή θα θέλαμε εφαρμογές οι οποίες να μπορούν να χειριστούν προβλήματα τα οποία δεν έχουμε ορίσει πλήρως, ωστόσο έχουμε επιλέξει τις κατευθυντήριες γραμμές (στόχους) επίλυσης τους. Η πλήρης ανάθεση της εργασίας στην οντότητα λογισμικού καθώς και η φαινομενικά, από τον εξωτερικό παρατηρητή, ευφυής συμπεριφορά του πράκτορα είναι δυνατόν να της αποδώσει τον όρο *ευφυής πράκτορας (intelligent agent)*. Παρ’ όλα αυτά προσπαθώντας να σχεδιάσει κανείς οντότητες που απλά εμφανίζουν ευφυή χαρακτηριστικά, είναι δυνατόν να δει πως αυτά είναι δυνατόν να προέρχονται από τη σύνθεση πολλών απλών και στοιχειώδους ευφυΐας. Υποστηρίζουμε σε αυτή την εργασία τη φιλοσοφία που υπαγορεύει πως η ευφυΐα είναι ένα *ολιστικό χαρακτηριστικό* με την έννοια του ότι προέρχεται από τη σύνθεση μικρότερων και απλών συνιστωσών, όπου ή κάθε μια έχει ένα στοιχειώδες βαθμό ευφυΐας [2].

Στη φύση συναντά κανείς ευφυή συμπεριφορά η οποία προέρχεται από τη σύνθεση πολλών απλών συστατικών στοιχειώδους ευφυΐας. Για παράδειγμα το νευρικό δίκτυο ενός έμβριου όντος το οποίο συντίθεται από νευρικά κύτταρα στοιχειώδους ευφυΐας ή μια κοινωνία μυρμηγκιών [3][4], στην οποία κάθε μέλος της δεν μπορεί να βρει το δρόμο της η τροφή από μόνο του, αλλά μόνο όλα μαζί το καταφέρνουν. Αναφέρουμε τα παραδείγματα αυτά για υποστηρίζοντας την άποψη του ότι η ευφυΐα αποτελεί χαρακτηριστικό κατανεμημένης και παράλληλης επεξεργασίας της πληροφορίας, από εξειδικευμένες οντότητες για την κάθε εργασία. Σε επόμενη ενότητα θα μελετήσουμε διαφορές του πεδίου έρευνας της τεχνητής νοημοσύνης και αυτής των αυτόνομων όντων.

Από την άλλη πλευρά είναι δυνατόν να αντιμετωπίσουμε την τεχνολογία των αυτόνομων όντων ως τεχνική σχεδίασης λογισμικού, όπου αποσυνθέτουμε μια σύνθετη και πολύπλοκη εργασία σε απλούστερες τις οποίες γνωρίζουμε με ποιο τρόπο να χειριστούμε, παρόμοια με την τεχνική αυτή του δομημένου ή του οντοκεντρικού προγραμματισμού. Θα αναφερθούμε εκτεταμένα στη χρήση της τεχνολογίας αυτόνομων οντοτήτων ως τεχνική σχεδίασης λογισμικού σε επόμενη ενότητα.

Στη συνέχεια της εργασίας αυτής, αρκετές φορές είναι δυνατόν κανείς να συναντήσει το ερέθισμα για συζήτηση πάνω σε τρόπους υλοποίησης μιας «ευφυούς» οντότητας ή για το εάν η ευφυΐα είναι ένα επιπρόσθετο χαρακτηριστικό το οποίο πηγάζει από τον τρόπο σχεδίασης μιας οντότητας ή δικτύου. Θα αναφερθούμε ξανά σε αυτό το θέμα στην ενότητα των συμπερασμάτων, ωστόσο μια τέτοια συζήτηση είναι πέρα από τους στόχους αυτής της εργασίας.

2.2 Χαρακτηριστικά αυτόνομων όντων

Παρόλη την πληθώρα ορισμών που μπορεί να συναντήσει κανείς στη βιβλιογραφία καθώς και είδη οντοτήτων, είναι χρήσιμο να μελετήσουμε ορισμένα κοινά χαρακτηριστικά των αυτόνομων όντων [5]:

Αυτονομία: Το σύστημα θα πρέπει να μπορεί να λειτουργήσει επιτυχάνοντας στόχους χωρίς καθοδήγηση από το χρήστη. Η έννοια περιλαμβάνει και την ανάληψη πρωτοβουλιών, την περιοδική και αυθόρμητη εκτέλεση ενεργειών.

Προσαρμοσιμότητα: Το σύστημα θα πρέπει να μπορεί να προσαρμόζεται στις αλλαγές του περιβάλλοντος και να διατηρεί σταθερή την απόδοση του.

Δυνατότητα για μάθηση: Το σύστημα θα πρέπει να μαθαίνει από τις εμπειρίες που έχει, έτσι ώστε να είναι σε θέση να επιλέξει την καλύτερη ενέργεια για κάθε περίπτωση, ή να μην επαναλάβει παλιότερα σφάλματα

Επικοινωνία με το χρήστη (discourse): Το σύστημα θα πρέπει να είναι σε θέση να επικοινωνήσει με το χρήστη για να ζητήσει οδηγίες για τις επιθυμίες ή ανάγκες του.

Καλή λειτουργία – Διαχείριση Πόρων: Το σύστημα θα πρέπει να λειτουργεί ακόμα και όταν συμβαίνουν σφάλματα ή όταν είναι υπερφορτωμένο. Η απόκριση του θα πρέπει να είναι σε πραγματικό χρόνο ακόμα και αν δεν είναι η βέλτιστη, ενώ θα πρέπει να είναι σε θέση να διαχειρίζεται τους υπολογιστικούς του πόρους.

Συνεργασία με άλλες οντότητες: Πολλές φορές το σύστημα στην προσπάθεια να βελτιώσει τη απόδοση του είναι δυνατόν να επικοινωνήσει με άλλα αυτόνομα όντα έτσι ώστε να πάρει έτοιμη πληροφορία ή γνώση εξοικονομώντας έτσι υπολογιστικούς πόρους και χρόνο. Είναι επιθυμητή ιδιότητα του συστήματος του να μπορεί να συνεργάζεται με άλλα όντα τα οποία μπορεί να υπάρχουν στο περιβάλλον του. Για την επίτευξη αυτής της ιδιότητας είναι δυνατόν να χρησιμοποιηθούν ειδικές γλώσσες επικοινωνίας των αυτόνομων όντων (όπως π.χ. η KQML, ATP κ.α.) [6]. Ωστόσο παρόλη την επάρκεια προτάσεων και για πρωτόκολλα και γλώσσες επικοινωνίας μεταξύ αυτόνομων όντων, δεν υπάρχει σήμερα κάποιο αποδεκτό από όλους πρότυπο (standard).

2.3 Αρχιτεκτονική συστημάτων αυτόνομων όντων και απαιτήσεις

Η ανάγκη για συστήματα τα οποία να ικανοποιούν ταυτόχρονα πολλούς στόχους σε ένα δυναμικό και μη προβλέψιμο περιβάλλον οδήγησε στον ορισμό κάποιων χαρακτηριστικών τα οποία είναι επιθυμητά (αν όχι αναγκαία) να πληρούνται από αυτά. Επίσης, η τεχνολογία λογισμικού των αυτόνομων όντων έχει συνδεθεί με μια γενικότερη αρχιτεκτονική λογισμικού ή αλλιώς μοντελοποίηση της συμπεριφοράς της οντότητας και η οποία ταξινομεί τις λειτουργίες της σε αυτές που σχετίζονται με την αντίληψη, νοητική διεργασία και πράξη, οι οποίες διεκπαιρώνονται και από τα αντίστοιχα τμήματα λογισμικού της οντότητας (Perception – Cognition – Action) [7]. Στην παράγραφο αυτή θα δούμε γιατί είναι απαραίτητο να πληρούνται αυτά τα χαρακτηριστικά καθώς και την αρχιτεκτονική σχεδιασμού των συστημάτων αυτών.

Ένα σύστημα το οποίο έχει σχεδιαστεί με χρήση αυτόνομων όντων, συνήθως πρέπει να ανταποκριθεί σε διάφορες απαιτήσεις οι οποίες μπορεί να είναι αντικρουόμενες μεταξύ τους. Το πως μπορεί ένα αυτόνομο ον να συνδυάσει στόχους έτσι ώστε να κάνει ενέργειες οι οποίες ωφελούν την εκπλήρωση όσο το δυνατόν παραπάνω στόχων, καθώς και το να ικανοποιήσει αντικρουόμενους στόχους (ή το να βρει τον καλύτερο συμβιβασμό μεταξύ τους), αποτελούν τρέχοντα ερευνητικά θέματα.

Για να μπορεί το σύστημα να ανταποκριθεί στις απαιτήσεις για τις οποίες κατασκευάστηκε θα πρέπει να γνωρίζει τις επιθυμίες του χρήστη. Οι επιθυμίες αυτές καθώς και άλλες υπολογιστικές ανάγκες του συστήματος μαζί με τις εντολές προς εκτέλεση του χρήστη, μπορούν να αποτελέσουν ένα σύνολο στόχων. Για παράδειγμα ένα αυτόνομο ον το οποίο χρησιμοποιείται για ανάκτηση πληροφορίας από το Internet (Διαδύκτιο) θα μπορούσε να συμπεριλαμβάνει στο σύνολο στόχων του, την επιθυμία του χρήστη για περιοδική ειδοποίηση του για όσον αφορά νέες δημοσιεύσεις για κάποιο επιστημονικό θέμα, την υπολογιστική ανάγκη για την ταχύτερη δυνατή δρομολόγηση των πακέτων δικτύου και την ομαλή συνεργασία του με άλλες εφαρμογές στο ίδιο υπολογιστικό περιβάλλον καθώς βέβαια και τις εντολές προς ικανοποίηση του χρήστη π.χ. “βρες όλα τα έγγραφα τα οποία περιέχουν την φωτογραφία του προσώπου Α και αναφέρονται στο ζήτημα Δ”. Το ζητούμενο είναι ένας εξωτερικός παρατηρητής του συστήματος, να αντιλαμβάνεται πως αυτό ενεργεί βάση ενός συνόλου στόχων προς εκπλήρωση.

Ωστόσο, δεν είναι απαραίτητος ο ορισμός των στόχων αυτών σε επίπεδο λογισμικού. Το σύστημα θα μπορούσε να προσπαθεί να ικανοποιήσει του στόχους αυτούς προσπαθώντας να επιτύχει καλύτερες καταστάσεις, (όπου η κάθε κατάσταση βαθμολογείται ανάλογα με το ποσοστό το κατά πόσο έχουν επιτευχθεί κάποιοι από του στόχους του χρήστη). Ακόμα το σύστημα θα μπορούσε να λειτουργεί με τη χρήση ανακλαστικών όπου οι γνωσιακές δομές του αυτόνομου όντος του υπαγορεύουν πως πρέπει να αντιδράσει σε κάθε κατάσταση [8]. Η αντιδράσεις του αυτόνομου όντος είναι ορισμένες έτσι ώστε να ικανοποιούν τους ζητούμενους στόχους. Μια επέκταση αυτής της τεχνικής είναι το να δημιουργούνται “σχέδια δράσης” (action plans)¹ τα οποία υπαγορεύουν μελλοντικές ενέργειες σε συνάρτηση με τα δυνατά συμβάντα τα οποία είναι δυνατόν να αντιληφθεί το ον [9]. Πάντως είναι δυνατόν το σύστημα να έχει ορισμένους σε ένα σύνολο (το οποίο μάλιστα θα μπορεί να είναι δυναμικά μεταβαλλόμενο) τους στόχους προς εκπλήρωση. Σε αυτή την περίπτωση θα πρέπει ο κάθε στόχος να είναι ορισμένος σε κάποια γλώσσα την οποία να μπορεί να καταλάβει το σύστημα.

2.3.1 Αυτονομία

Έχουμε δει πως η ανάγκη για έξυπνη διαχείριση και επιλογή της πληροφορίας καθώς και η δράση σε δυναμικά μεταβαλλόμενα περιβάλλοντα, ήταν δύο από τους λόγους που οδήγησαν στο σχεδιασμό και ανάπτυξη συστημάτων βασισμένων σε αυτόνομα όντα. Στην ουσία δηλαδή ο χρήστης χρησιμοποιεί ένα έξυπνο σύστημα έτσι ώστε να χάνει όσο το δυνατόν λιγότερο χρόνο για αναζήτηση και διαχείριση της πληροφορίας. Ένα σύστημα το οποίο ξέρει, ή μπορεί να μάθει πως να κάνει κάτι τέτοιο χωρίς να καταφεύγει σε διάλογο, μέσω της διεπιφάνειας χρήσης, με το χρήστη, σίγουρα κάνει πιο εύκολη και γρήγορη τη δουλεία του χρήστη. Είναι επιθυμητό μάλιστα, το σύστημα να παίρνει πρωτοβουλίες απαλλάσσοντας το χρήστη από επιπλέον επικοινωνία μαζί του. Ο χρήστης δεν θα ήθελε να βρίσκεται όλη αυτή την ώρα πάνω

¹ Ο όρος action plan θα μπορούσε να μεταφραστεί και ως πλάνο δράσης. Προτιμάμε την ελληνικότερη και πιο ελεύθερη μετάφραση της λέξης σχέδιο αντί πλάνο.

από το υπολογιστή περιμένοντας μήπως το σύστημα αντιμετωπίσει κάποιο δίλημμα έτσι ώστε να του το επιλύσει.

Όπως είδαμε για να είναι δυνατή η αυτονομία ενός αυτόνομου όντος αυτός θα πρέπει να παίρνει κάποιες αποφάσεις. Η επιλογή της απόφασης αυτής θα πρέπει να γίνεται βάση των επιθυμιών του χρήστη οι οποίες είτε έχουν οριστεί από την αρχή, είτε έχουν εκμαθηθεί από το σύστημα παρακολουθώντας τις ενέργειες του χρήστη καθώς και τα αποτελέσματα των ίδιων του των ενεργειών καθώς και την πιθανή τους βαθμολογία από το χρήστη. Για παράδειγμα, ένα αυτόνομο ον του οποίου χρήστης είναι ένας βιολόγος θα περιμένει να δει έγγραφα τα οποία σχετίζονται με τα φυσικά δέντρα όταν δώσει μια επερώτηση με περιεχόμενο “δέντρα”, σε αντίθεση με έναν επιστήμονα υπολογιστών ο οποίος περιμένει, μάλλον, να δει έγγραφα τα οποία σχετίζονται με δομές δεδομένων (το συγκεκριμένο παράδειγμα έχει συσχετίζεται βέβαια και με την εκμάθηση του πλαισίου αναφοράς ή context από την οντότητα). Το σύστημα μπορεί να μάθει μια τέτοια προτίμηση του χρήστη είτε παρακολουθώντας τις ενέργειες του χρήστη πάνω στα αποτελέσματα της επερώτησης (π.χ. να παρακολουθεί ποια από αυτά διάβασε και σε ποιο βαθμό), είτε ζητώντας από το χρήστη να βαθμολογήσει ή σχολιάσει το κάθε ένα αποτέλεσμα της επερώτησης. Η αυτονομία καθώς και η τάση για μάθηση των προτιμήσεων του χρήστη είναι δυνατόν να συμπεριληφθούν στους στόχους του συστήματος.

2.3.2 Προσαρμοσιμότητα

Είναι εύκολο να διαπιστώσει κανείς, πως οι κυριότεροι τομείς εφαρμογής έξυπνων συστημάτων σχετίζονται με δυναμικά περιβάλλοντα. Θα θέλαμε τα συστήματα που λειτουργούν σε αυτό το πλαίσιο να προσαρμόζονται στις αλλαγές του περιβάλλοντος τους ώστε να μην μειώνεται η απόδοση τους λόγω των αλλαγών αυτών.

Από την άλλη πλευρά με την πάροδο του χρόνου εκτός από το περιβάλλον μπορεί να μεταβάλλονται και οι επιθυμίες του χρήστη (στην πραγματικότητα και ο ίδιος ο χρήστης είναι κομμάτι του περιβάλλοντος του αυτόνομου όντος) με αποτέλεσμα το σύστημα να πρέπει να προσαρμόζει όχι μόνο εσωτερικές γνωσιακές δομές, αλλά και τρόπο αντίληψης, μέσω των αισθητήρων.

Τέλος απαίτηση των χρηστών είναι ότι, το σύστημα θα πρέπει να μπορεί να προσαρμόζεται και στις καταστάσεις στις οποίες το ίδιο βρίσκεται. Έτσι θα πρέπει να μπορεί να διαχειριστεί τους υπολογιστικούς του πόρους ανάλογα με την τρέχουσα κατάσταση. Έτσι όταν ο χρήστης κάνει μια επείγουσα επερώτηση, το σύστημα θα πρέπει να ασχολείται όσο το δυνατόν μόνο με αυτό αφήνοντας για μετά άλλες εργασίες. Η τεχνική μπορεί να βρεθεί στη βιβλιογραφία ως “εστίαση της προσοχής” (focus of attention) [56].

2.3.3 Ανεκτικότητα σε βλάβες – Καλή λειτουργία (Robustness)

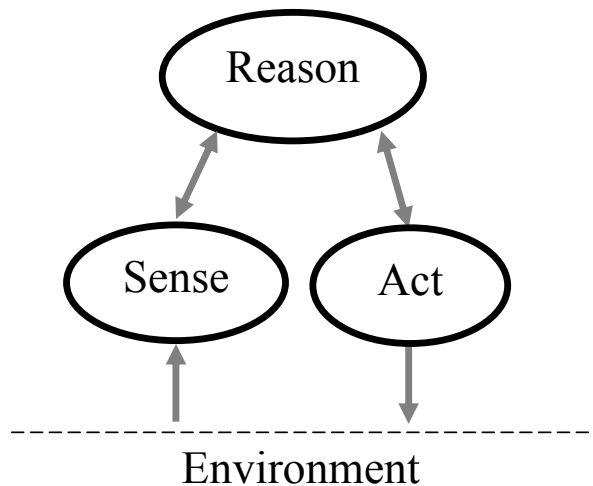
Απαίτηση των χρηστών είναι και η καλή λειτουργία του συστήματος ακόμα και σε στιγμές μεγάλου φόρτου εργασίας ή σε βλάβες. Το σύστημα θα πρέπει να συνεχίσει να λειτουργεί ικανοποιητικά και να αποκρίνεται σε πραγματικό χρόνο, ενώ να μπορεί να εφαρμόσει εναλλακτικές λύσης σε περίπτωση βλάβης ή αποτυχίας μιας ενέργειας. Η τεχνική που συνήθως συναντά κανείς είναι το σύστημα να υπολογίζει κάποια ικανοποιητική λύση, αλλά όχι ίσως τη βέλτιστη. Σε αυτή την περίπτωση, το αποτέλεσμα που θα του επιστραφεί ίσως να μην είναι το βέλτιστο, αλλά το πιο ικανοποιητικό δεδομένων των συνθηκών υπολογισμού.

2.3.4 Αντίληψη – Νοητική Διεργασία – Δράση (Sense – Reasoning – Action)

Η υιοθέτηση του παραδείγματος μοντελοποίησης της συμπεριφοράς μιας οντότητας ως μια συνεχή ακολουθία λειτουργιών αντίληψης, νοητικής διεργασίας και δράσης, αποτελεί βασικό χαρακτηριστικό σχεδιασμού όσον αφορά την αρχιτεκτονική του συστήματος. Μάλιστα το παράδειγμα αυτό υπάγορεύει και αντίστοιχη αρχιτεκτονική, αφού οι λειτουργίες αυτές πραγματοποιούνται από αντίστοιχα τμήματα λογισμικού (software modules) και τα οποία μπορούν να χαρακτηριστούν ως αισθητήρες, νοητικό ή γνωσιακό τμήμα (perception – cognition – action) και αντικείμενα δράσης. Θα θέλαμε να σταθούμε για λίγο στο νοητικό ή γνωσιακό τμήμα. Ο χαρακτηρισμός του παίρνει το όνομα του τόσο από το ότι είναι το τμήμα στο οποίο κρατούνται γνωσιακές δομές οι οποίες χρησιμοποιούνται για την αποθήκευση γνώσης, εμπειρίας ή άλλης πληροφορίας, όσο και από το ότι είναι αυτό το οποίο υλοποιεί τις

συμπερασματικές διαδικασίες, δηλαδή αυτό που ονομάζουμε «νόηση» της οντότητας. Στη συνέχεια της εργασίας τόσο λόγω συντομίας όσο και λόγω του ότι στη βιβλιογραφία ο όρος συναντάται με χρήση της λέξης cognitive (της οποίας η μετάφραση είναι γνωσιακός), θα αναφερόμαστε ορισμένες φορές στο τμήμα λογισμικού αυτό και ως απλά γνωσιακό. Μάλιστα ενδιαφέρον είναι ο προβληματισμός πάνω στην άποψη του ότι οι γνωσιακές δομές μιας οντότητας καθορίζουν πλήρως τον τρόπο με τον οποίο γίνεται η συμπερασματική διαδικασία, με την έννοια του ότι η μέθοδος η οποία ακολουθείται είναι αποθηκευμένη σε αυτές, ή αλλιώς οι γνωσιακές δομές είναι κατασκευασμένες με τέτοιο τρόπο ώστε η διαδικασία νόησης να υπαγορεύεται από τη δομή τους (αυτό συμβαίνει κυρίως στα έμβια όντα, αν δεχθεί κανείς πως οι αποφάσεις αποτελούν το αποτέλεσμα της λειτουργίας ενός δικτύου νευρώνων) Παραθέτουμε στη συνέχεια την περιγραφή των τμημάτων τα οποία προαναφέραμε.

Παρατηρώντας ένα σύστημα, όπως και αυτό το οποίο περιγράφουμε, μπορεί να διακρίνει κανείς τρία επίπεδα: (1) αυτό στο οποίο ανήκουν όλα τα τμήματα του συστήματος από τα οποία παίρνει πληροφορίες από το περιβάλλον (αισθητήρες, καταχωρητές γεγονότων), (2) αυτό με στο οποίο παίρνονται αποφάσεις για τη λειτουργία και τις ενέργειες του συστήματος και (3) αυτό στο οποίο ανήκουν τα τμήματα με τα οποία το σύστημα επικοινωνεί και δρα πάνω στο περιβάλλον του. Τα τμήματα αυτά υλοποιούν τις λειτουργίες αντίληψης, γνώσης και δράσης του αυτόνομου όντος (βλ σχήμα 2.1). Το περιβάλλον γίνεται αντιληπτό από το αισθητήριο τμήμα της οντότητας, ενώ το γνωσιακό ή νοητικό επίπεδο, δρα έχοντας στη διάθεση του τα γεγονότα του περιβάλλοντος και πληροφορία αποθηκευμένη στις γνωσιακές δομές. Η συμπερασματική διαδικασία και η εξαγωγή αποφάσεων γίνεται στο γνωσιακό επίπεδο, το οποίο μεταβιβάζει τις αποφάσεις για δράση πάνω στο περιβάλλον, στα δρώντα αντικείμενα τα οποία και εκτελούν τις δοθείσες εντολές.



Σχήμα 2.1 Η τυπική λειτουργία του παραδείγματος της αίσθησης, συμπερασματικής διαδικασίας και πράξης.

Στην σχεδιαστική αυτή τεχνική βασίζεται η πλειοψηφία των συστημάτων τα οποία μελετάμε. Ωστόσο πρέπει να τονιστεί πως αυτή αποτελεί τεχνική σχεδιασμού και μοντελοποίησης των τμημάτων τα οποία αποτελούν ένα αυτόνομο ον και όχι λύση πάνω στο πρόβλημα της τεχνητής ευφυΐας. Μάλιστα όπως είναι δυνατόν να διαπιστώσει κανείς, σχεδόν κάθε πρόγραμμα σε έναν υπολογιστή μπορεί να μοντελοποιηθεί κατά αυτό τον τρόπο.

Το τμήμα του αυτόνομου όντος το οποίο είναι υπεύθυνο για την αντίληψη και περιλαμβάνει τους αισθητήρες (sensors) του συστήματος, δημιουργεί τα αντικείμενα τα οποία περιγράφουν τα αντιλαμβανόμενα γεγονότα και τα καταχωρεί στις δομές γεγονότων που αντιλαμβάνεται το σύστημα, ενώ ταυτόχρονα τα χαρακτηρίζει ανάλογα με τη σημασία τους και τα φιλτράρει

(όσον αφορά το ποια θα προωθηθούν στο γνωσιακό επίπεδο), ανάλογα με το βαθμό ενδιαφέροντος τους. Οι αισθητήρες είναι δυνατόν να προσαρμόσουν τη λειτουργία τους επιλέγοντας ανάλογα με εντολές που δέχονται από το νοητικό επίπεδο.

Το γνωσιακό επίπεδο επηρεάζει τις επιλεγόμενες προς εκτέλεση ενέργειες και πιθανά και τη διαδικασία αντίληψης, ανάλογα με τη σχεδίαση. Σε αυτό το τμήμα το σύστημα αντιλαμβάνεται τη σημασιολογική σημασία των γεγονότων, λύνει προβλήματα και δημιουργεί σχέδια δράσης. Συνήθως μέλη του τμήματος αυτού αποτελούν ένα "blackboard system" [10] του οποίου οι δομές είναι προσβάσιμες από όλες τις οντότητες του γνωσιακού επιπέδου. Επίσης το τμήμα αυτό είναι υπεύθυνο για την επιλογή στρατηγικών μάθησης καθώς και για την διαδικασία προσαρμογής στο περιβάλλον, όλων των επιπέδων (ακόμα και του ίδιου του εαυτού του). Το τμήμα αυτό είναι δυνατόν να εμπεριέχει τις οντότητες / αντικείμενα τα οποία αποφασίζουν για το που θα ψάξει ή το πως θα φιλτράρει ή το τι συμπέρασμα θα ενάγει για κάποια ιδιότητα του χρήστη κ.τ.λ.

Το επίπεδο δράσης το οποίο είναι υπεύθυνο για την εκτέλεση εξωτερικών για το αυτόνομο ον ενεργειών περιλαμβάνει τα τμήματα τα οποία ενεργούν στο περιβάλλον του. Τα αντικείμενα του επιπέδου αυτού, δέχονται εντολές από οντότητες του γνωσιακού επιπέδου, τις οποίες προσπαθούν να εφαρμόσουν.

Συνήθως τα τρία παραπάνω τμήματα λειτουργούν παράλληλα επικοινωνώντας με μηνύματα, ενώ οι διαδικασίες αντίληψης και δράσης είναι κατά πολύ ταχύτερες από αυτές του γνωσιακού επιπέδου.

2.4 Σύγκριση με την τεχνητή νοημοσύνη

Στην παράγραφο αυτή θα δούμε τις βασικές διαφορές μεταξύ της τεχνολογίας λογισμικού των αυτόνομων όντων και της τεχνητής νοημοσύνης.

1. Η παραδοσιακή τεχνητή νοημοσύνη στην πλειονότητα των εφαρμογών της έχει μέχρι στιγμής δώσει έμφαση σε προβλήματα τα οποία απαιτούν προχωρημένη γνώση κάποιου θέματος (π.χ. ιατρική διάγνωση, σκάκι, κτλ.). Τα συστήματα αυτά παρέχουν μεγαλύτερη ανάλυση και γνώση σε βάθος ενός προβλήματος. Σε αντίθεση τα συστήματα αυτόνομων όντων, διαμοιράζουν τη γνώση και ικανότητες, έτσι ώστε κάθε αυτόνομο τμήμα να είναι εξειδικευμένο σε μια μικρή και απλή εργασία και οι γνωσιακές του δομές περιέχουν πληροφορία μόνο για την εργασία αυτή. Η υψηλότερου βαθμού νόηση και γνώση είναι αποτέλεσμα της συνεργασίας πολλών τέτοιων αυτόνομων οντοτήτων.
2. Στον τομέα της τεχνητής νοημοσύνης συναντά κανείς "κλειστές" εφαρμογές οι οποίες δεν έχουν άμεση αλληλεπίδραση με το περιβάλλον. Συνήθως σε τέτοια συστήματα την επικοινωνία μεταξύ περιβάλλοντος και συστήματος την κάνουν οι χρήστες της εφαρμογής οι οποίοι περιγράφουν το πρόβλημα σε κάποια συμβολική γλώσσα που αντιλαμβάνεται το σύστημα. Το σύστημα τότε αποκρίνεται πάλι σε αυτή τη συμβολική γλώσσα υπαγορεύοντας τις ενέργειες οι οποίες πρέπει να γίνουν. Στην τεχνολογία των αυτόνομων όντων το ίδιο το ον είναι κομμάτι του περιβάλλοντος και είναι απευθείας συνδεδεμένος με αυτό, μέσω των αισθητήρων και των τμημάτων του υπεύθυνων για την εκτέλεση ενεργειών. Πολλές φορές το σύστημα αλληλεπιδρά και με άλλα παρόμοια συστήματα τα οποία βρίσκονται στο περιβάλλον.
3. Στις περισσότερες εφαρμογές της τεχνητής νοημοσύνης, το σύστημα ασχολείται με ένα πρόβλημα κάθε φορά το οποίο δίδεται από το χρήστη. Πολλές φορές μάλιστα δεν υπάρχουν χρονικοί περιορισμοί όσον αφορά την επίλυση του προβλήματος, ενώ το σύστημα δεν αντιμετωπίζει διακοπές για επίλυση άλλων προβλημάτων. Επιπλέον από την οπτική πλευρά του συστήματος το περιβάλλον παραμένει σταθερό κατά την επίλυση ενός προβλήματος. Σε αντίθεση στις εφαρμογές βασισμένες σε αυτόνομα όντα το σύστημα πρέπει να αποκρίνεται σε πραγματικό χρόνο στα προβλήματα τα οποία του δίδονται ενώ είναι δυνατόν να αντιμετωπίζει παραπάνω από ένα προβλήματα. Οι δύο απαιτήσεις αυτές μπορεί να το οδηγήσει το σύστημα στην λήψη αντικρουόμενων αποφάσεων ή στόχων τους οποίους πρέπει να συμβιβάσει.

4. Στα συστήματα τεχνητής νοημοσύνης δίδεται έμφαση στο τι γνώση έχει ήδη το σύστημα από την αρχή, ενώ οι γνωσιακές του δομές είναι συνήθως στατικές. Το σύστημα είναι ενεργό μόνο κατά την επίλυση κάποιου προβλήματος. Αντίθετα στον ερευνητικό τομέα των αυτόνομων όντων, έμφαση δίνεται στο τι συμπεριφορά θα παρουσιάσει το σύστημα όταν τεθεί στο περιβάλλον. Οι γνωσιακές δομές όχι μόνο είναι μεταβλητές και προσαρμόσιμες, αλλά είναι και αυτές οι οποίες καθορίζουν τη συμπεριφορά του συστήματος μιας και οι αποφάσεις που λαμβάνει το σύστημα εξαρτώνται άμεσα από τις γνώσεις και εμπειρία που έχει το σύστημα.
5. Τέλος, η τεχνητή νοημοσύνη δεν ασχολείται με το πως το σύστημα αποκτά τη γνώση και με το πως μεταβάλλεται αυτή με την πάροδο του χρόνου. Στις εφαρμογές οι οποίες χρησιμοποιούν αυτόνομα όντα μεγάλη έμφαση δίνεται στον τρόπο με τον οποίο το σύστημα μαθαίνει, ενώ το ίδιο το σύστημα συνήθως ξεκινά με μικρή η καθόλου αρχική γνώση.

Τα κυρίαρχα προβλήματα με τα οποία ασχολείται η έρευνα των αυτόνομων όντων είναι ο σχεδιασμός μιας γενικότερης αρχιτεκτονικής η οποία να εγγυάται συστήματα τα οποία είναι προσαρμόσιμα, παρουσιάζουν ανεκτικότητα στις βλάβες και είναι σε θέση να επιτύχουν ένα σύνολο στόχων. Για να γίνει αυτό θα πρέπει να δοθεί ικανοποιητική λύση σε δύο υπό - προβλήματα:

1. Το πρόβλημα της επιλογής ενέργειας.
2. Το πρόβλημα της μάθησης από εμπειρία

Κάτι άλλο που πρέπει να τονιστεί είναι ότι μέχρι στιγμής δεν έχει εμφανιστεί κάποια γενικότερη αρχιτεκτονική να δίνει λύση στα παραπάνω προβλήματα ενώ ταυτόχρονα να είναι εφαρμόσιμη σε ένα ευρύ πεδίο εφαρμογών. Έτσι, οι περισσότερες αρχιτεκτονικές οι οποίες αναπτύσσονται, παρουσιάζουν εξειδίκευση όσον αφορά το πεδίο εφαρμογής τους. Θα αναφερθούμε αναλυτικότερα στα δύο παραπάνω προβλήματα σε επόμενες παραγράφους.

2.5 Σχεδιασμός και υλοποίηση

2.5.1 Βασικές αρχές σχεδιασμού

Η μελέτη των συστημάτων αυτόνομων όντων δίνει έμφαση σε δύο βασικές παρατηρήσεις οι οποίες αποτελούν και τη βάση για αρχές σχεδιασμού και περαιτέρω έρευνα στον τομέα. Η πρώτη είναι πως είναι ευκολότερο να σχεδιάσει κανείς συστήματα όταν γνωρίζει το περιβάλλον χρήσης τους, ενώ η δεύτερη ότι η επικοινωνία μεταξύ αυτόνομων τμημάτων του συστήματος μπορεί να αποδώσει στο σύστημα επιπλέον χαρακτηριστικά για τα οποία δεν έχει προγραμματιστεί το σύστημα.

Όσον αφορά τη γνώση του περιβάλλοντος χρήσης είναι δυνατό να επαληθεύσει κανείς την ορθότητα της αρχής σχεδιασμού σε πολλαπλά επίπεδα.

1. Οι μέθοδοι οι οποίοι μοντελοποιούνται όπως αντίληψη, προγραμματισμός ενεργειών, μάθηση, κτλ. είναι μέρος ενός πλήρους συστήματος το οποίο ονομάζεται αυτόνομο ον. Σχεδιάζοντας συστήματα με έναν ολοκληρωμένο τρόπο, είναι δυνατόν να κάνει απλούστερο το σχεδιασμό. Για παράδειγμα, ένα σύστημα μπορεί να εξαρτάται λιγότερο από τον προγραμματισμό ενεργειών, διότι μπορεί να αποθηκεύει παλιά σχέδια δράσης για επαναχρησιμοποίηση. Ένα σύστημα το οποίο έχει αισθητήρες, έχει λιγότερη ανάγκη μοντελοποίησης του περιβάλλοντος, μιας και μπορεί να το παρατηρήσει όποτε είναι αναγκαίο κτλ.
2. Το αυτόνομο ον, αποτελεί τμήμα του περιβάλλοντος στο οποίο βρίσκεται. Γενικά στα συστήματα αυτόνομων όντων, δεν υπάρχει ανάγκη μοντελοποίησης του περιβάλλοντος, μιας και το ον το παρατηρεί μέσω των αισθητήρων του [11]. Το εξωτερικό περιβάλλον μπορεί επίσης να χρησιμοποιηθεί και ως εξωτερική μνήμη. Για παράδειγμα το σύστημα μπορεί να χρησιμοποιήσει το περιβάλλον για υπενθυμίσει του ποιες από τις προγραμματισμένες εργασίες έχουν γίνει και ποιες όχι [12]. Πολλές φορές το περιβάλλον

έχει κάποια ορισμένα χαρακτηριστικά τα οποία είναι εκμεταλλεύσιμα από το σύστημα, εάν βέβαια είναι γνωστά κατά το σχεδιασμό του [13].

3. Το σύστημα είναι δυνατόν να έχει αντίληψη του χρόνου. Υπονοείται εδώ ότι το σύστημα είναι δυνατόν να βελτιώσει την επίδοση του με την πάροδο του χρόνου. Επίσης κατά την πάροδο του χρόνου το σύστημα σε διάλογο με το χρήστη μπορεί να πάρει πληροφορίες για τη διαλεύκανση ασαφειών. Για παράδειγμα, ένα σύστημα το οποίο αναγνωρίζει φυσική γλώσσα δεν είναι ανάγκη να μπορεί να καταλήγει σε συμπέρασμα όταν έχει κάποια ασάφεια στη γραμματική ανάλυση. Είναι δυνατόν να ρωτά το χρήστη, για τη σωστή σημασιολογική σημασία της πρότασης, βελτιώνοντας παράλληλα τις γνωσιακές του δομές.
4. Τέλος, κάθε τέτοιο σύστημα είναι δυνατόν να αποτελεί μέλος μιας “κοινωνίας”. Άλλα αυτόνομα όντα στο ίδιο περιβάλλον είναι δυνατόν να ασχολούνται με τα ίδια η παρόμοια προβλήματα. Έτσι το σύστημα δεν είναι ανάγκη να μπορεί να κάνει τα πάντα από μόνο του, μιας και μπορεί να καταφεύγει σε άλλα για την απόκτηση γνώσης ή πληροφορίας. Στη βιβλιογραφία μπορεί κανείς να μελετήσει παραδείγματα συστημάτων τα οποία κατάφεραν να μάθουν να κάνουν ορισμένες εργασίες (κυρίως όσον αφορά tasks σε επίπεδο διεπιφάνειας χρήσης), παρακολουθώντας και μιμούμενα άλλους χρήστες [14].

Η συνέπεια των παραπάνω ήταν ότι η έρευνα πάνω στα συστήματα αυτόνομων όντων να επικεντρωθεί στη μοντελοποίηση συστημάτων ειδικά στο περιεχόμενο της εφαρμογής για την οποία προορίζονται.

Η ιδέα του ότι η επικοινωνία μεταξύ αυτόνομων τμημάτων του συστήματος μπορεί να αποδώσει στο σύστημα επιπλέον χαρακτηριστικά για τα οποία δεν έχει προγραμματιστεί το σύστημα, μπορεί να βρει εφαρμογή σε διάφορα επίπεδα σχεδιασμού [15][16].

1. Η δυναμική της αλληλεπίδρασης μεταξύ ενός αυτόνομου όντος και του περιβάλλοντος στο οποίο βρίσκεται, μπορεί να αναδείξει νέες δομές και συμπεριφορές του συστήματος επιπρόσθετες από αυτές για τις οποίες είχε προγραμματιστεί. Στη βιβλιογραφία μπορεί να δει κανείς πως πολύπλοκη συμπεριφορά, όπως αυτή της επίτευξης στόχων, μπορεί να μοντελοποιηθεί ως αναδεικνυόμενη, από την αλληλεπίδραση, μεταξύ ενός πολύπλοκου περιβάλλοντος και ενός όντος προγραμματισμένου μόνο για ανακλαστικές ενέργειες. Το νόημα των παραπάνω είναι ότι οι γνωσιακές δομές οι οποίες καθορίζουν τη συμπεριφορά ενός αυτόνομου όντος, δεν είναι ανάγκη να είναι πολύπλοκες για να προκαλέσουν σύνθετη συμπεριφορά [17].
2. Η αλληλεπίδραση μεταξύ ενός αυτόνομου όντος και του περιβάλλοντος είναι επίσης δυνατό να αναδείξει νέες δυνατότητες, όσον αφορά τις δυνατές ενέργειες του συστήματος. Για παράδειγμα ένα αυτόνομο όν το οποίο μπορεί να βρίσκει έγγραφα τα οποία έχουν μοιάζουν με ένα πρότυπο έγγραφο δεν είναι να έχει κάποια μέθοδο εξειδικευμένη για αυτή την εργασία. Είναι δυνατόν να έχει μικρότερες οντότητες, μια η οποία να βρίσκει λέξεις κλειδιά από ένα έγγραφο, μια που αναγνωρίζει τη δομή του εγγράφου, μια η οποία βάση των λέξεων κλειδιών, αναγνωρίζει το γνωσιακό πεδίο πάνω στο οποίο αναφέρεται το έγγραφο και μια η οποία γνωρίζει που να αναζητήσει έγγραφα ανάλογα με το γνωσιακό πεδίο στο οποίο αναφέρονται. Πολλές φορές μάλιστα είναι δυνατόν να μην υπάρχει απολύτως κανένα τμήμα του αυτόνομου όντος υπεύθυνο για την επιλογή ενέργειας. Αντίθετα υπάρχουν πολλά μικρότερα, αυτόνομα το καθένα, τα οποία ζητούν από το σύστημα να πραγματοποιήσει κάποια συγκεκριμένη ενέργεια [18].
3. Η ίδια ιδέα βρίσκει εφαρμογή και σε μεγαλύτερη κλίμακα, όπου η αλληλεπίδραση μεταξύ πολλών αυτόνομων όντων, κάτω από ένα σύνολο “κοινωνικών” κανόνων συμπεριφοράς, μπορεί να αναδείξει δυνατότητα για πολύπλοκη συμπεριφορά και ενέργειες. Η ιδέα αυτή, την οποία μπορεί να συναντήσει στην ίδια τη φύση [3][4] μπορεί να βρει εφαρμογή σε πολύπλοκα υπολογιστικά συστήματα διαχειριζόμενα από αυτόνομα όντα[19].

Το συμπέρασμα το οποίο μπορεί να εξαγάγει κανείς, είναι ότι η αναδεικνυόμενη πολυπλοκότητα του συστήματος, είναι πολλές φορές πιο ανεκτική σε λάθη και ευέλικτη από τον top – down σχεδιασμό. Αυτό συμβαίνει διότι κανένα από τα τμήματα του συστήματος δεν είναι υπεύθυνο για την παραγωγή αυτής της πολύπλοκης συμπεριφοράς, ενώ κανένα δεν θεωρείται ως πιο κρίσιμο από κάποιο άλλο. Εάν κάποιο από αυτά δεν είναι σε θέση να λειτουργήσει, απλά η απόδοση του συστήματος σημειώνει απλά κάποια επιδείνωση. Δεδομένου του ότι όλα τα

τμήματα του συστήματος, λειτουργούν εν παραλλήλω, το σύστημα είναι πιο πιθανό να μπορεί να προσαρμοστεί σε αλλαγές του περιβάλλοντος. Πολλές φορές το σύστημα υπολογίζει πολλές λύσεις για κάποιο πρόβλημα, παράλληλα, έτσι ώστε με την αλλαγή κάποιων μεταβλητών του περιβάλλοντος, το σύστημα να έχει ήδη υπολογισμένη κάποια εναλλακτική λύση [20].

2.5.2 Χαρακτηριστικά σχεδιασμού

Πολλές από τις αρχιτεκτονικές αυτόνομων όντων, οι οποίες έχουν προταθεί έχουν κάποια κοινά χαρακτηριστικά μεταξύ τους. Στην παράγραφο αυτή θα δούμε τις βασικότερες τέτοιες ομοιότητες και θα τις αντιπαραθέσουμε με χαρακτηριστικά της παραδοσιακής τεχνητής νοημοσύνης.

Task – Oriented Modules

Στην τεχνητή νοημοσύνη, το σύστημα συνήθως διαχωρίζεται σε λειτουργικές οντότητες (functional modules) οι οποίες αναπτύσσονται ανεξάρτητα και η κάθε μια προορίζεται για κάποια συγκεκριμένη εργασία, ενώ χρησιμοποιούν μια κεντρική οντότητα (αυτή που περιέχει τη γνώση του συστήματος), στην οποία έχουν πρόσβαση, τόσο για να πάρουν δεδομένα, όσο και για να επικοινωνήσουν με τις υπόλοιπες οντότητες του συστήματος.

Σε αντίθεση, ένα αυτόνομο ον μπορεί να χαρακτηριστεί ως ένα σύνολο από οντότητες, από τις οποίες η κάθε μια είναι υπεύθυνη για μια μικρή αλλά συγκεκριμένη εργασία, οι οποίες ονομάζονται συμπεριφορές [20]. Οι οντότητες αυτές επικοινωνούν ξεχωριστά η κάθε μια μεταξύ τους, με μικρά απλά μηνύματα, ενώ η επικοινωνία είναι δυνατόν να γίνει και έμμεσα, μέσω του περιβάλλοντος (μια οντότητα μπορεί να αλλάξει κάτι στο περιβάλλον, γεγονός το οποίο μπορεί να ενεργοποιήσει μια άλλη οντότητα).

Task – Specific Solutions

Στην παραδοσιακή τεχνητή νοημοσύνη, τα διάφορα τμήματα ενός συστήματος, είναι όσο το δυνατόν γίνεται πιο γενικά και ανεξάρτητα από το πεδίο εφαρμογής τους, με απώτερο σκοπό να είναι δυνατόν να επαναχρησιμοποιηθούν σε άλλα συστήματα με διαφορετικό πεδίο εφαρμογής.

Από την άλλη πλευρά, ένα αυτόνομο ον δεν έχει γενικά τμήματα, ή άλλα τμήματα τα οποία να μη γνωρίζουν το πεδίο εφαρμογής για το οποίο προορίζονται. Δεν υπάρχει κάποιο κεντρικό κομμάτι το οποίο να ελέγχει όλες τις ενέργειες οι οποίες γίνονται από το αυτόνομο ον.

Η αναπαράσταση του περιβάλλοντος αποκτά μικρότερη σημασία

Όπως έχουμε δει και προηγουμένως, στην τεχνολογία αυτόνομων όντων, δεν έχουμε κάποια κεντρική αναπαράσταση του περιβάλλοντος την οποία να μοιράζονται όλα τα τμήματα του συστήματος. Το σύστημα δεν προσπαθεί να ενώσει όλα τα σήματα που λαμβάνει από τους αισθητήρες για να δημιουργήσει κάποιο μοντέλο του περιβάλλοντος. Κάθε οντότητα χρησιμοποιεί γνώση και πληροφορία την οποία χρειάζεται αυτό και μόνο, ενώ κάθε ένα από αυτά μπορεί να έχει διαφορετικό τρόπο αναπαράστασης της γνώσης.

Απουσία κεντρικής μονάδας ελέγχου

Στον τομέα της τεχνητής νοημοσύνης, οι οντότητες οι οποίες αποτελούν το σύστημα, είναι ιεραρχημένες, έτσι ώστε να υπάρχει κάποια η οποία τελικά να διαχειρίζεται τις ενέργειες του συστήματος. Οι οντότητες περιμένουν να έρθει η σειρά τους, ώστε να μπορούν να εκτελέσουν κάποια ενέργεια και η κεντρική μονάδα ελέγχου, αποφασίζει ποια σειρά είναι.

Η αρχιτεκτονική ενός αυτόνομου όντος, είναι όσο το δυνατόν κατανεμημένη. Όλες οι οντότητες του συστήματος, λειτουργούν παράλληλα, ενώ δεν υπάρχει κάποια οντότητα η οποία να ελέγχει κάποια άλλη. Ωστόσο, πρέπει να υπάρχει κάποιος τρόπος για την επιλογή ή την ένωση πολλών ενεργειών σε μια. Συνήθως αυτό γίνεται οργανώνοντας τις οντότητες αυτές σε κάποιο δίκτυο είτε τύπου “winner take all” [18], είτε κάποιο προκαθορισμένο με προτεραιότητες [20]. Θα επανέρθουμε αναλυτικότερα στο θέμα αυτό σε επόμενη παράγραφο. Τέλος λόγω της κατανεμημένης φύσης του συστήματος, αυτό

είναι δυνατόν να αντιδρά πιο γρήγορα σε αλλαγές του περιβάλλοντος, ή των ίδιων του των στόχων.

Αποτελέσματα του σχεδιασμού

Τα συστήματα τα οποία σχεδιάζονται με τις παραπάνω αρχές, τείνουν να παρουσιάσουν μεγαλύτερη προσαρμοσιμότητα και καλύτερη λειτουργία. Λειτουργούν γρηγορότερα διότι (1) έχουν λιγότερα επίπεδα επεξεργασίας της πληροφορίας, (2) είναι περισσότερο κατανοητά και δεν υπάρχει ανάγκη συγχρονισμού (3) απαιτούν λιγότερο υπολογιστικό φόρτο μιας και υπάρχουν εξειδικευμένες όσον αφορά το πεδίο εφαρμογής οντότητες. Είναι ικανά να προσαρμοστούν ευκολότερα σε αλλαγές του περιβάλλοντος, αφού χρησιμοποιούν αυτό ως πηγή πληροφορίας και παρουσιάζουν καλύτερη λειτουργία διότι (1) κανένα από τα τμήματα του συστήματος δεν είναι πιο κρίσιμο από κάποιο άλλο (δεν υπάρχει σημείο ολικής αποτυχίας), (2) δεν ασχολούνται με την πλήρη κατανόηση της κατάστασης του συστήματος (το οποίο συνήθως απαιτεί μεγάλο υπολογιστικό φόρτο), (3) προσαρμόζονται κατά την πάροδο του χρόνου.

2.5.3 Υλοποίηση

2.5.3.1 Το πρόβλημα της επιλογής ενέργειας

Ορισμός προβλήματος

Το πρόβλημα της επιλογής ενέργειας είναι δυνατόν να διατυπωθεί ως εξής: “Δεδομένου, ενός αυτόνομου όντος ο οποίος έχει χρονικά μεταβαλλόμενους στόχους, ένα σύνολο από πιθανές ενέργειες (από τις οποίες μερικές είναι δυνατό να εκτελεστούν) και πληροφορία από τους αισθητήρες, ποιες ενέργειες πρέπει να εκτελέσει το σύστημα έτσι ώστε να βελτιστοποιήσει την επίτευξη των στόχων ;”. Μάλιστα, πρέπει να τονιστεί εδώ πως στόχος του συστήματος μπορεί να είναι και το να προσπαθεί, να μάθει πως να επιτύχει καλύτερα τους στόχους του.

Είναι δυνατόν να υπολογίσουμε την βέλτιστη πολιτική επιλογής ενέργειας, για ένα αυτόνομο ον ο οποίος βρίσκεται σε ένα ντετερμινιστικό και πιθανοκρατικό περιβάλλον [21]. Οι λόγοι που καθιστούν αδύνατη την εφαρμογή μιας τέτοιας λύσης είναι το ότι το σύστημα πρέπει να λειτουργήσει (1) έχοντας περιορισμένους υπολογιστικούς πόρους (απόκριση σε πραγματικό χρόνο, ταχύτητα υπολογισμού, μνήμη), (2) έχοντας πιθανώς ατελής η λανθασμένες πληροφορίες (πληροφορία που προέρχεται από τους αισθητήρες), (3) ένα δυναμικό, μη ντετερμινιστικό, μη πιθανοκρατικό περιβάλλον, (4) στόχους οι οποίοι αλλάζουν με την πάροδο του χρόνου.

Λύσεις που έχουν δοθεί μέχρι τώρα

Hand – Built, Flat Networks [17,22,23,24,20]

Σε αυτή την αρχιτεκτονική, οι στόχοι δεν υπάρχουν στο σύστημα, αλλά μόνο στη σκέψη του σχεδιαστή. Οι στόχοι, είναι δυνατόν να είναι πολλαπλοί και διαφορετικής φύσης και οι πληροφορίες από τους αισθητήρες απεριόριστη (σε σχέση με τις υπολογιστικές δυνατότητες του συστήματος). Ο μηχανισμός του δικτύου (ο οποίος έχει προγραμματιστεί “με το χέρι” από το σχεδιαστή), καθορίζει το ποια οντότητα θα καταφέρει να οδηγήσει το σύστημα σε κάποια συμπεριφορά, χρησιμοποιώντας τα τμήματα του συστήματος τα οποία εκτελούν κάποια ενέργεια. Η αρχιτεκτονική αυτή, δεν επιτρέπει, συνένωση ενεργειών, γεγονός που καθιστά αδύνατη την ταυτόχρονη εκτέλεση ενεργειών από δύο διαφορετικές οντότητες. Ακόμη, το κύριο μειονέκτημα της μεθόδου είναι, ότι στην ουσία ο ίδιος ο προγραμματιστής επιλύει το πρόβλημα της επιλογής ενέργειας κατά τον σχεδιασμό.

Compiled – Built, Flat Networks [25,18,26,27]

Η σχεδίαση αυτή είναι επέκταση της προηγούμενης, και βασίζεται στη δυναμική δημιουργία του δικτύου από το ίδιο το σύστημα. Δηλαδή το ίδιο το σύστημα αναλαμβάνει να δημιουργήσει το δίκτυο το οποίο συνενώνει τις οντότητες του συστήματος. Έτσι κάθε φορά που μεταβάλλονται οι στόχοι του συστήματος, το δίκτυο επαναδημιουργείται μεταβάλλοντας κατά αυτό τον τρόπο τη συμπεριφορά του συστήματος. Ωστόσο σε αυτή την αρχιτεκτονική ο σχεδιαστής πρέπει να ορίσει από πριν τους στόχους του συστήματος, το πως μπορούν να

συνενωθούν στόχοι σε άλλους στόχους ή σε ενέργειες και το ποιες είναι οι οντότητες οι οποίες δρουν στο περιβάλλον και ποια είναι τα αναμενόμενα αποτελέσματα τους.

Hand – Built, Hierarchical Networks [28,29,30,21]

Η αρχιτεκτονική αυτή προτείνει την ιεραρχική κατάταξη των ενεργειών, η οποία κυμαίνεται από ενέργειες υψηλού επιπέδου σε άλλες χαμηλότερου, Μόνο οι βασικές ενέργειες του πιο χαμηλού επιπέδου είναι εκτελέσιμες. Οι υψηλότερου επιπέδου ενέργειες αποτελούνται από άλλες χαμηλότερου. Στην ουσία τα συστήματα αυτά χρησιμοποιούν κάποια αφηρημένη τεχνική επιλογής ενέργειας, η οποία μεταφράζεται σε ένα σύνολο ενεργειών χαμηλότερου επιπέδου. Η κατηγορία αρχιτεκτονικών αυτή, υποστηρίζει πιο πολύπλοκα και μεγάλα σύνολα στόχων, αλλά συχνά απαιτείται από το σχεδιαστή να σχεδιάσει ο ίδιος το δίκτυο της ιεραρχίας.

2.5.3.2 Το πρόβλημα της μάθησης από την εμπειρία

Ορισμός προβλήματος

Η μάθηση η οποία προέρχεται από την εμπειρία είναι απαραίτητη για ένα σύστημα το οποίο πρέπει να βελτιώνεται με την πάροδο του χρόνου, ενώ πρέπει να παρουσιάζει καλή λειτουργία και αυτονομία για μεγάλα χρονικά διαστήματα. Το πρόβλημα είναι δυνατόν να διατυπωθεί ως εξής: “Δεδομένου ενός αυτόνομου όντος με (1) ένα σύνολο ενεργειών, (2) πληροφορία από τους αισθητήρες, (3) πολλαπλούς και χρονικά μεταβαλλόμενους στόχους, πώς μπορεί αυτό να βελτιώσει τη συμπεριφορά επιλογής ενέργειας βασιζόμενο στην εμπειρία;”.

Οι μέθοδοι της μάθησης θα πρέπει να δίνουν απάντηση στο πώς μαθαίνει το σύστημα, ποια είναι η στρατηγική εκμάθησης και ποιες είναι οι μηχανισμοί επιλογής ενέργειας που χρησιμοποιεί το σύστημα.

Λύσεις που έχουν δοθεί μέχρι τώρα

Reinforcement Learning [31,32,33,34]

Η γενική ιδέα της τεχνικής αυτής είναι η εξής: Δεδομένου ενός αυτόνομου όντος με (1) ένα σύνολο στόχων, (2) ένα σύνολο καταστάσεων στις οποίες μπορεί να βρεθεί, (3) μια μεταβλητή η οποία βαθμολογεί κάθε εκτελούμενη ενέργεια σε σχέση με το αποτέλεσμα της, το σύστημα μαθαίνει μια πολιτική επιλογής ενέργειας η οποία αντιστοιχεί μεταξύ καταστάσεων και ενεργειών, τέτοια ώστε να μεγιστοποιεί το άθροισμα των βαθμολογιών από τις εκτελούμενες ενέργειες.

Αναλυτικότερα, το σύστημα κρατά ένα διδιάστατο πίνακα του οποίου η μια διάσταση περιέχει τις πιθανές καταστάσεις στις οποίες μπορεί να βρεθεί το σύστημα, ενώ η άλλη τις πιθανές ενέργειες που μπορεί να εκτελέσει. Η τιμή του στοιχείου A_{ij} του πίνακα περιέχει το μέσο όρο βαθμολογίας του συγκεκριμένου ζευγαριού κατάστασης i – ενέργειας j . Το σύστημα διαλέγει την ενέργεια με την καλύτερη βαθμολογία, μαθαίνει ανανεώνοντας τον πίνακα με νέες βαθμολογίες, ενώ ως στρατηγική εκμάθησης χρησιμοποιεί την τυχαία επιλογή ενός στοιχείου του πίνακα από τη σειρά που αντιστοιχεί στην κατάσταση που βρίσκεται (το οποίο αντιστοιχεί σε ένα ζεύγος κατάστασης – ενέργειας), και εκτελεί την αντίστοιχη ενέργεια.

Classifier Systems [35,36,37]

Μια άλλη τεχνική είναι αυτή της χρήσης κανόνων ταξινόμησης (classifiers) και είναι δυνατόν να χαρακτηριστούν ως επέκταση της προηγούμενης. Σε αυτή την περίπτωση το σύστημα πάλι προσπαθεί να μεγιστοποιήσει τη βαθμολογία που παίρνει για ενέργειες σε συγκεκριμένες καταστάσεις. Το σύστημα εδώ περιέχει ένα σύνολο από κανόνες για να ταξινομήσει τα εισερχόμενα γεγονότα, οι οποίοι ονομάζονται classifiers και στοιχεία για την απόδοση κάθε κανόνα, καθώς και μια ένδειξη της “δύναμης” του κανόνα και η οποία σχετίζεται με τη βαθμολογία που παίρνει ο κανόνας μετά τη χρήση του. Το σύστημα διαλέγει για κάθε κατάσταση, ένα σύνολο από κανόνες τους οποίους χρησιμοποιεί για να επιλέξει τις προς εκτέλεση ενέργειες. Όταν το σύστημα εκτελεί μια ενέργεια, μοιράζει τη βαθμολογία που πήρε στους κανόνες τους οποίους χρησιμοποίησε, ενώ χρησιμοποιεί τους κανόνες με τη μεγαλύτερη “δύναμη”. Η στρατηγική εκμάθησης είναι η εξής: το σύνολο κανόνων, έχει σταθερό πληθάρημο. Κάθε ένα συγκεκριμένο χρονικό διάστημα, οι κανόνες με μικρή δύναμη αντικαθιστούνται από καινούριους οι οποίοι δοκιμάζονται. Οι νέοι αυτοί κανόνες μπορεί να

είναι συνδυασμοί οι μεταλλάξεις (ο αλγόριθμος προέρχεται από την κλάση των γενετικών αλγορίθμων).

Model Builders [38,39,40]

Σε αυτή τη μέθοδο, το σύστημα κατασκευάζει ένα πιθανοκρατικό μοντέλο του ποιες μπορεί, και με ποια πιθανότητα, να είναι οι συνέπειες που θα ακολουθήσουν την εκτέλεση μιας ενέργειας. Το μοντέλο αυτό χρησιμοποιείται για να αποφασιστεί ποια ενέργεια είναι σχετικότερη με την κατάσταση και με τους στόχους του συστήματος. Το σύστημα μαθαίνει παρακολουθώντας τις αλλαγές που συμβαίνουν στο περιβάλλον μετά από κάθε του ενέργεια ενώ δοκιμάζει νέες ή τυχαίες ενέργειες προσπαθώντας να επεκτείνει το ήδη υπάρχον μοντέλο.

3 Η τεχνολογία αυτόνομων οντοτήτων ως τεχνική σχεδίασης λογισμικού

Εξετάζοντας την αρχιτεκτονική των αυτόνομων οντοτήτων από άλλη οπτική γωνία, είναι δυνατόν να βρούμε κοινά στοιχεία με τον οντοκεντρικό προγραμματισμό. Αυτό συμβαίνει διότι πράγματι ο σχεδιασμός έχει γίνει με τέτοιο προσανατολισμό. Στην ενότητα αυτή θα παρουσιάσουμε τα στοιχεία οντοκεντρικού σχεδιασμού συστημάτων τα οποία χρησιμοποιούμε καθώς και νέες τεχνικές τις οποίες εισάγουμε στο σχεδιασμό προκειμένου να υλοποιήσουμε συνεργαζόμενες οντότητες λογισμικού με αυτόνομη συμπεριφορά. Παράλληλα θα παρουσιάσουμε τη σχεδίαση συστημάτων με αυτόνομες οντότητες ως τεχνική σχεδίασης λογισμικού, ενώ στο τέλος της ενότητας θα δούμε πως αλληλεπιδρούν οι οντότητες αυτές στο σύστημα μας.

Η κυριότερη κατευθυντήριος γραμμή του οντοκεντρικού σχεδιασμού συστημάτων η οποία χρησιμοποιήθηκε, ήταν αυτή του διαχωρισμού των εργασιών του συστήματος και η αντιστοίχησή τους σε ανάλογες οντότητες. Οι οντότητες αυτές μπορεί να είναι σύνθετες με την έννοια του ότι αποτελούνται και από άλλα αντικείμενα ή οντότητες. Έτσι για κάθε εργασία υπάρχει η αντίστοιχη οντότητα ενώ κάθε δομή δεδομένων η οποία χρησιμοποιείται είναι και αυτή ένα αντικείμενο. Η αντιστοιχία δηλαδή βρίσκεται στον κατακερματισμό της σύνθετης εργασίας σε απλές και την ανάθεση της κάθε εργασίας σε ένα αντικείμενο (φυσικά κάθε εργασία είναι δυνατόν να αποτελείται από υπό εργασίες κ.τ.λ.) το οποίο περιέχει τόσο την πληροφορία όσο και τις μεθόδους για την επιτέλεση της. Κάθε τέτοιο αντικείμενο είναι δυνατόν να αντιμετωπιστεί ως ένα κλειστό σύστημα (ή αλλιώς “μαύρο κουτί”) με την έννοια του ότι αναλαμβάνει εξ' ολοκλήρου την εργασία η οποία του ανατέθηκε και του ότι επιτελεί την εργασία ή επιστρέφει το αποτέλεσμα του υπολογισμού το οποίο του ανατέθηκε, χωρίς περαιτέρω αλληλεπίδραση με την αιτούσα οντότητα. Κατά αυτόν τον τρόπο μπορεί να πει κανείς πως διαχωρίζουμε την εφαρμογή σε τμήματα το κάθε ένα εξειδικευμένο σε μια εργασία και αποκλειστικά υπεύθυνο για την επιτέλεση της. Ταυτόχρονα διασπάμε και τα δεδομένα της εφαρμογής μας διαμοιράζοντας τα στις οντότητες οι οποίες τα χρησιμοποιούν. Στη βιβλιογραφία μπορεί να συναντήσει κανείς τον όρο *Agent Oriented Programming* [41], ο οποίος αντιμετωπίζεται ως εξειδίκευση του οντοκεντρικού προγραμματισμού.

Σχεδιάζοντας το σύστημα κατά αυτόν τον τρόπο μπορούμε να αντιμετωπίσουμε την κάθε εργασία του συστήματος μας μεμονωμένα και κατά το σχεδιασμό να επικεντρώσουμε τη προσοχή μας στο σχεδιασμό μιας οντότητας η οποία επιτελεί μια μεμονωμένη εργασία. Η τεχνική αυτή είναι εντελώς αντίστοιχη με το δομημένο προγραμματισμό κατά τον οποίο διασπάμε μια σύνθετη εργασία σε απλούστερες προκειμένου έχουμε να υλοποιήσουμε ένα σύνολο από απλές εργασίες παρά μια σύνθετη ή οποία θα ήταν πολύπλοκη στον προγραμματισμό. Στην περίπτωση των αντικειμένων ωστόσο, κάθε αντικείμενο εκτός από τις μεθόδους και πληροφορία για την εκτέλεση της εργασίας περιέχει και μεταβλητές κατάστασης, οι οποίες του επιτρέπουν να μεταβάλει τη συμπεριφορά του με την πάροδο του χρόνου.

Θα σταθούμε για λίγο στην τελευταία πρόταση, επιχειρηματολογώντας υπέρ του ότι ο μόνος τρόπος για μια οντότητα να μπορέσει να βελτιώσει την απόδοσή της και να προσαρμοστεί στο περιβάλλον, είναι να τροποποιεί κάποια εσωτερική δομή του. Εάν ένα αντικείμενο δεν περιείχε μεταβλητές κατάστασης κάθε φορά που θα του ανατίθονταν μια εργασία, υπό τις ίδιες συνθήκες, θα την επιτελούσε με τον ίδιο τρόπο. Προκειμένου το αντικείμενο να μεταβάλει τη συμπεριφορά του ανάλογα με τις εμπειρίες του, θα πρέπει οι τελευταίες να το τροποποιούν κατά τρόπο τέτοιο, ώστε να αποτυπώνουν τουλάχιστον την αποκτηθείσα από την εμπειρία γνώση.

Από τα πλεονεκτήματα του οντοκεντρικού σχεδιασμού εφαρμογών αυτό το οποίο μας ενδιαφέρει περισσότερο, είναι η κατασκευή μιας μεγάλης και σύνθετης οντότητας από απλές. Παράλληλα κερδίζουμε ευκολία στο σχεδιασμό και απλούστερη υλοποίηση καθώς και ευκολότερο εντοπισμό λαθών στην υλοποίηση. Άλλα επιπρόσθετα πλεονεκτήματα, είναι η δυνατότητα επαναχρησιμοποίησης τμημάτων λογισμικού, χρήση οντοτήτων λογισμικού από τρίτους κατασκευαστές κ.τ.λ. Κατά αυτόν τον τρόπο θα προσπαθήσουμε να σχεδιάσουμε μια

σύνθετη οντότητα από αριθμό απλούστερων μικρότερων. Ο λόγος για αυτό είναι η προσπάθεια ανάπτυξης αυτόνομης και προσαρμοστικής συμπεριφοράς η οποία, σύμφωνα με τη φιλοσοφία σχεδιασμού την οποία ακολουθούμε, οφείλεται στην αλληλεπίδραση απλών και αυτόνομων οντοτήτων, χωρίς κάποια από αυτές να έχει τον κεντρικό έλεγχο. Οι οντότητες αυτές μπορούν φυσικά να επικοινωνούν μεταξύ τους, έτσι ώστε να είναι δυνατή η συνεργασία τους σε εργασίες όπου απαιτείται να εμπλακεί παραπάνω από μια οντότητα.

Δεδομένης της έλλειψης κεντρικού σημείου ελέγχου από την οντότητα, και της πολιτικής επιλογής ενέργειας της κάθε οντότητας, οι ενέργειες τις οποίες επιτελεί το σύστημα καθορίζονται από τις ενέργειες τις οποίες αποφασίζουν προς εκτέλεση οι οντότητες του γνωσιακού επιπέδου, και διατάζουν τα δρώντα αντικείμενα. Η ολική συμπεριφορά του συστήματος είναι η συνισταμένη των συμπεριφορών όλων των οντοτήτων που το αποτελούν.

Προκειμένου όμως να μιλήσουμε για ολική συμπεριφορά του συστήματος θα πρέπει να ορίσουμε τον τρόπο με τον οποίο αποδίδουμε συμπεριφορά σε έστω μια οντότητα. Στα περισσότερα συστήματα λογισμικού η ροή ελέγχου καθορίζεται από ένα και μόνο νήμα ελέγχου το οποίο και αποτελεί τη διεργασία του συστήματος. Προκειμένου να αποδώσουμε ξεχωριστή συμπεριφορά σε κάθε οντότητα του συστήματος μας έχουμε δύο επιλογές:

- Να δημιουργήσουμε μια κεντρική οντότητα η οποία να μοιράζει τον έλεγχο (δηλαδή ποσοστό υπολογιστικού χρόνου του επεξεργαστή) περιοδικά στις οντότητες και με ποσοστό ανάλογο με την προτεραιότητα της εργασίας που πρόκειται να επιτελέσει η οντότητα
- Να δώσουμε σε κάθε οντότητα τουλάχιστον ένα νήμα ελέγχου, κάνοντας την εφαρμογή μας πολυπρογραμματιστική (multithreaded) και αναθέτοντας προτεραιότητες σε κάθε νήμα.

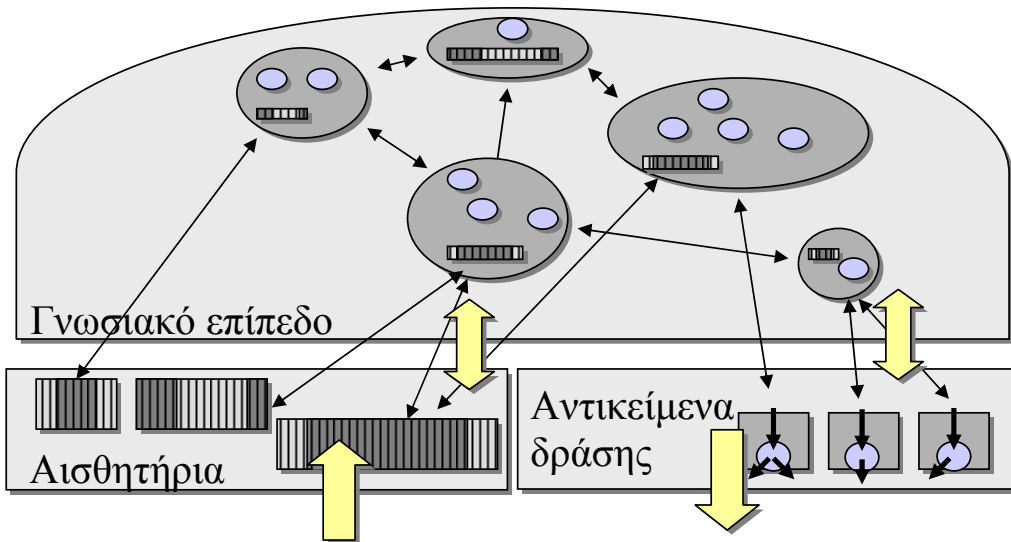
Οι κυριότεροι λόγοι οι οποίοι μας προτρέπουν να χρησιμοποιήσουμε τη δεύτερη λύση είναι η αποφυγή κεντρικού σημείου ελέγχου, η αποφυγή της πλεονάζουσας επικοινωνίας μεταξύ των οντοτήτων, η επεκτασιμότητα (στην πρώτη περίπτωση για να προσθέσουμε μια οντότητα θα πρέπει να επαναπρογραμματίσουμε την οντότητα που διαμοιράζει τον έλεγχο, ενώ στη δεύτερη απλά προσθέτουμε μια οντότητα η οποία δρα παράλληλα με τις άλλες) καθώς και η ευκολία προγραμματισμού. Κατά αυτό τον τρόπο επεκτείνουμε την έννοια του αντικειμένου που υπάρχει στον κλασικό οντοκεντρικό προγραμματισμό και του δίνουμε την δυνατότητα αυτόνομης συμπεριφοράς, με την έννοια ότι το ίδιο το αντικείμενο φροντίζει για το πότε θα εκτελέσει κάποια εργασία η όχι, ανάλογα με τα μηνύματα τα οποία δέχεται η τις εσωτερικές νοητικές αποφάσεις τις οποίες λαμβάνει. Θα αναφερόμαστε σε αντικείμενα αυτού του είδους ως *ενεργά αντικείμενα*, τα οποία μάλιστα είναι και η υποδομή πάνω στην οποία θα στηριχτεί η υλοποίηση μιας τυπικής αυτόνομης οντότητας. Κάτι που θα πρέπει να σημειωθεί είναι ότι έχοντας πολλά νήματα ελέγχου να δρουν πάνω σε διαμοιραζόμενη μνήμη, είναι δυνατόν να αντιμετωπίσουμε προβλήματα συντονισμού και ταυτόχρονης πρόσβασης στο ίδιο τμήμα μνήμης από παραπάνω από μια οντότητες. Οι κυριότεροι τρόποι αντιμετώπισης των προβλημάτων αυτών βασίζονται σε τεχνικές των λειτουργικών και καταναμημένων συστημάτων [42].

Ο τρόπος με τον οποίο αλληλεπιδρούν οι οντότητες καθώς και οι εντολές οι οποίες δίνουν στα αντικείμενα δράσης, καθορίζουν τη συμπεριφορά του συστήματος. Στην εργασία αυτή η αλληλεπίδραση μεταξύ των οντοτήτων γίνεται σε συνεργατικό επίπεδο, δηλαδή οι οντότητες μοιράζονται τις γνώσεις και τα πιστεύω τους, ενώ μια είναι δυνατόν να ζητήσει από κάποια άλλη την εκτέλεση μιας εργασίας. Η τελευταία ιδιότητα ειδικά αποτελεί αρκετά σημαντικό στοιχείο για το χαρακτηρισμό του τρόπου αλληλεπίδρασης των οντοτήτων στην εφαρμογή μας, όπου κάθε οντότητα είναι εξειδικευμένη σε μια εργασία. Οι οντότητες οι οποίες επιτελούν σύνθετες εργασίες βασίζονται στις υπόλοιπες οι οποίες θα εκτελέσουν τμήματα των εργασιών αυτών όποτε τους το ζητήσουν οι πρώτες. Άλλο μοντέλο αλληλεπίδρασης το οποίο μπορεί να βρεθεί στη βιβλιογραφία [43], είναι αυτό των ανταγωνιστικών πρακτόρων όπου η απόδοση του πράκτορα αντιστοιχίζεται με κάποιας μορφής οικονομικό κέρδος όπου κάθε ένας προσπαθεί να το μεγιστοποιήσει προς όφελος του ίδιου και όχι υπέρ του συνόλου των πρακτόρων.

Μιλώντας για αλληλεπίδραση μεταξύ των οντοτήτων θέλουμε να αναφερθούμε στον τρόπο με τον οποίο επικοινωνούν οι οντότητες αυτές. Γενικά δεδομένου ότι οι οντότητες εκτελούνται σε διαμοιραζόμενη μνήμη, είναι δυνατόν να επικοινωνήσουν καλώντας η μια μεθόδους της άλλης. Ωστόσο αυτό μειώνει την επεκτασιμότητα του συστήματος, ενώ κάνει ουσιαστικά αδύνατη τη χρήση οντοτήτων από τρίτους κατασκευαστές και ακόμα δυσκολότερη την επιλογή χρήσης μιας οντότητας μεταξύ πολλών. Επιπλέον θα δούμε πως δεν είναι απαραίτητο όλες οι οντότητες να εκτελούνται στο ίδιο μηχάνημα. Οι προηγούμενοι λόγοι μας ωθούν στην επικοινωνία των οντοτήτων με μηνύματα και κατά συνέπεια στον ορισμό κάποιας γλώσσας επικοινωνίας μεταξύ των οντοτήτων.

Τόσο στη βιβλιογραφία όσο και σε άλλα υπάρχοντα συστήματα μπορεί να βρει κανείς αρκετές γλώσσες επικοινωνίας πρακτόρων (agent communication languages), με επικρατέστερες την KQML [6] και TCL [44]. Καμία από αυτές ωστόσο δεν αποτελεί πρότυπο έτσι ώστε να το υιοθετήσουμε. Επιπλέον η επικοινωνία η οποία λαμβάνει χώρα μεταξύ των οντοτήτων του συστήματος μας, έχει σχεδιαστεί έτσι ώστε, ειδικά για αυτό το λόγο, να είναι ιδιαίτερα απλή. Κάθε μήνυμα περιέχει είτε μια αυστηρά ορισμένη δομή δεδομένων, είτε την αίτηση για κλήση μιας μεθόδου κάποιου αντικειμένου (αυτό στο οποίο απευθύνεται το μήνυμα). Τα δύο είδη μηνυμάτων αυτά είναι δυνατόν να κωδικοποιηθούν σε πολλές από τις υπάρχοντες γλώσσες επικοινωνίας πρακτόρων έτσι ώστε να μπορούμε να χρησιμοποιήσουμε κάποια για αυτό το σκοπό.

Το παρακάτω σχήμα (σχήμα 3.1) απεικονίζει την αρχιτεκτονική μιας σύνθετης οντότητας, σχεδιασμένη με τις κατευθυντήριες γραμμές τις οποίες προαναφέραμε. Παρατηρούμε τη γενική μορφή μια αυτόνομης οντότητας, της οποίας η λειτουργία ακολουθεί τη διαδικασία της αίσθησης, συμπερασματικής διαδικασίας και πράξης. Κάθε μια από αυτές τις λειτουργίες αναλαμβάνεται από ένα τμήμα της οντότητας, εξειδικευμένο για αυτή τη διαδικασία, σχεδιασμένο όμως με τις ίδιες αρχές. Τα αισθητήρια κάνουν αντιληπτά τα γεγονότα του εξωτερικού προς την οντότητα κόσμου και μετά από επεξεργασία (ταξινόμηση, επιλογή) είναι υπεύθυνα για τη διάθεση τους προς τις οντότητες του γνωσιακού επιπέδου. Το ίδιο το γνωσιακό επίπεδο αποτελείται από οντότητες εξειδικευμένες σε ένα σύνολο εργασιών η κάθε μια. Οι οντότητες αυτές λειτουργούν αυτόνομα κάνοντας χρήση πολλών νημάτων ελέγχου η κάθε μια και δημιουργώντας δυναμικά νέες οντότητες στις οποίες αναθέτουν εργασίες. Το περιβάλλον μιας οντότητας του γνωσιακού επιπέδου, είναι οι άλλες οντότητες αυτού με τις οποίες συνεργάζεται ανταλλάσσοντας δεδομένα και εργασίες, καθώς και τα αισθητήρια και αντικείμενα δράσης του συστήματος. Η δομή μιας οντότητας του γνωσιακού επιπέδου, είναι αντίστοιχη με αυτή της ίδιας της οντότητας, αν και υπάρχει εξειδίκευση ανάλογα με την εργασία στην οποία απασκοπεί. Τα αισθητήρια της οντότητας λαμβάνουν εντολές από τις οντότητες του γνωσιακού επιπέδου και εκτελώντας αυτές καθιστούν εφικτή τη δράση της οντότητας στο περιβάλλον. Στο σχήμα με έντονα και ανοιχτόχρωμα βέλη απεικονίζεται η ροή πληροφορίας και μηνυμάτων σε γενικό επίπεδο, ενώ με μικρότερα βέλη απεικονίζονται οι ανταλλαγές μηνυμάτων μεταξύ των αντικειμένων ειδικότερα.



Σχήμα 3.1 Γενική αρχιτεκτονική μιας αυτόνομης οντότητας.

4 Μοντέλο μιας τυπικής οντότητας

Στο τμήμα αυτό του κειμένου θα περιγράψουμε τη δομή και λειτουργία μιας τυπικής οντότητας του συστήματος πάνω στο σχεδιασμό της οποίας στηρίζεται και αυτός των οντοτήτων όλων των επιπέδων. Θα δούμε τα στοιχεία και τρόπο υλοποίησης των επί μέρους τμημάτων του, καθώς και τον τρόπο με τον οποίο του αποδίδουν αυτά την επιθυμητή συμπεριφορά. Η προσπάθεια περιγραφής μας εδώ, επικεντρώνεται στην ανάδειξη των στοιχείων αυτών τα οποία είναι κοινά μεταξύ των οντοτήτων και τα οποία κληρονομεί κάθε οντότητα του συστήματος. Το αποτέλεσμα της εύρεσης των κοινών στοιχείων αυτών χρησιμοποιήθηκε στον ορισμό μιας βασικής κλάσης οντοτήτων η οποία περιγράφει μια αυτόνομη οντότητα και από την οποία κληρονομούν οι υπόλοιπες τα χαρακτηριστικά αυτά.

Σε επόμενο τμήμα της εργασίας, το οποίο περιγράφει αναλυτικά τις οντότητες του συστήματος, θα δούμε τα ιδιαίτερα χαρακτηριστικά της κάθε μιας, τα οποία και την εξειδικεύουν στην ανάλογη εργασία για την οποία έχει σχεδιαστεί. Στο τέλος της ενότητας αναφερόμαστε στην προγραμματιστική διεπιφάνεια χρήσης των οντοτήτων αυτών.

4.1 Δομή οντότητας

Η οντότητα αποτελείται από τρία αντικείμενα τα οποία υλοποιούν το γνωσιακό τμήμα, καθώς και τα τμήματα αίσθησης και δράσης της οντότητας. Τα αντικείμενα δρουν αυτόνομα επικοινωνώντας μεταξύ τους, μέσω κλήσης μεθόδων, έτσι ώστε τα αντιλαμβανόμενα γεγονότα να μεταφέρονται από τους αισθητήρες στο γνωσιακό κομμάτι και οι εντολές λειτουργίας από το γνωσιακό στα αντικείμενα δράσης. Η δομή των αντικειμένων αυτών προσεγγίζει αρκετά τη δομή της ίδιας της οντότητας, με την έννοια του ότι ακολουθείται μερικώς η αρχιτεκτονική διαχωρισμού των μελών της οντότητας λογισμικού, σε αυτά της αίσθησης της δράσης και του γνωσιακού τμήματος. Θα δούμε στη συνέχεια για ποιους λόγους είναι επιθυμητή η δομή αυτή.

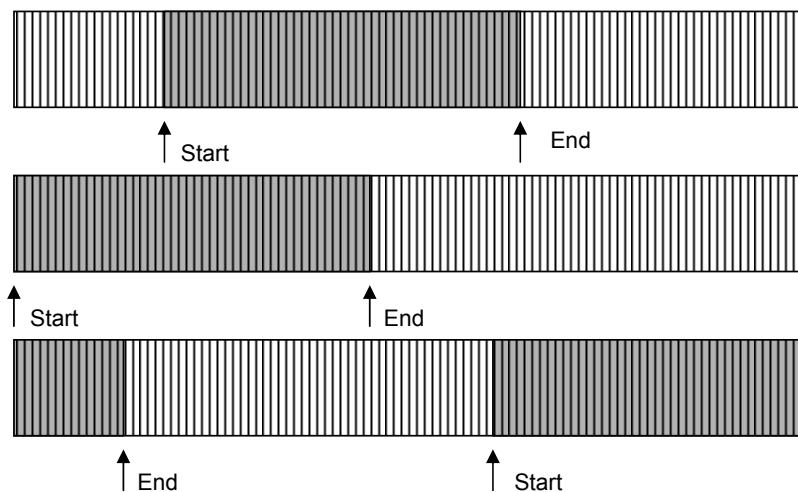
Η συμπεριφορά καθενός από τα προηγούμενα τρία τμήματα, καθορίζεται από τουλάχιστο ένα νήμα ελέγχου το οποίο είτε εξυπηρετεί αιτήσεις οι οποίες λαμβάνονται μέσω των αισθητήρων, τα οποία είναι δυνατόν να είναι παραπάνω από ένα για κάθε οντότητα, είτε φροντίζει για την εφαρμογή κάποιου σχεδίου δράσης το οποίο έχει ανατεθεί στην οντότητα. Το νήμα ελέγχου, ειδικά στην περίπτωση του γνωσιακού αντικειμένου, είναι δυνατόν να δημιουργήσει άλλα νήματα ή οντότητες στα οποία θα αναθέσει κάποια εργασία, και τα οποία θα εκτελούνται παράλληλα, οπότε όμως θα αναλάβει και τη διαχείριση αυτών, όπως και των υπολογιστικών αποτελεσμάτων που πιθανώς να παράγουν αυτά.

4.1.1 Αισθητήρια

Η βασική λειτουργία του αισθητήριου αντικειμένου είναι η συλλογή μηνυμάτων (ή γεγονότων) από το περιβάλλον της οντότητας και η εναπόθεση τους με ταξινομημένο τρόπο, σε δομή στην οποία έχουν πρόσβαση οι υπόλοιπες οντότητες του συστήματος. Ταυτόχρονα το αντικείμενο φροντίζει για τη διαχείριση του περιεχομένου της δομής αυτής, δηλαδή για την απομάκρυνση (δίνοντας πιθανά εντολή για την αρχειοθέτηση τους σε άλλη μνήμη εάν αυτό είναι απαραίτητο) από την δομή των παλαιότερων γεγονότων και ελευθέρωση της μνήμης τα οποία αυτά είχαν δεσμεύσει. Η δομή φύλαξης των γεγονότων αποτελεί δομή με πεπερασμένου μεγέθους και αυτό διότι η συνεχής συνάθροιση δεδομένων σε αυτή χωρίς την περιοδική απομάκρυνση μερικών, θα εξαντλούσε τελικά τη μνήμη του υπολογιστή. Θα αναφερόμαστε στη δομή αυτή στη συνέχεια ως *καταχωρητή γεγονότων (event buffer)*. Η πρόσβαση του γνωσιακού τμήματος της οντότητας γίνεται απευθείας από την γνωσιακή οντότητα, εκμεταλλευόμενη το γεγονός της διαμοιραζόμενης μνήμης μεταξύ των οντοτήτων, και διότι θέλουμε να αποφύγουμε τον επιπλέον υπολογιστικό φόρτο τον οποίον θα προκαλούσε η υλοποίηση της επικοινωνίας αυτής με χρήση μηνυμάτων. Στη συνέχεια θα δικαιολογήσουμε την επιλογή μας αυτή και θα δούμε τον τρόπο με τον οποίο αποφεύγουμε τα συνηθισμένους κινδύνους χρήσης διαμοιραζόμενης μνήμης από πολλά νήματα ελέγχου [42].

Το νήμα ελέγχου του αντικείμενου αίσθησης μιας οντότητας τίθεται σε λειτουργία με την άφιξη ενός νέου μηνύματος ή αίτησης. Στη συνέχεια και εάν υπάρχει ανεπάρκεια χώρου στη δομή, φροντίζει για την απομάκρυνση παλαιότερων δεδομένων από αυτή, ενώ τέλος ταξινομεί το νέο μήνυμα στη δομή. Εάν η διεργασία συλλογής και εναπόθεσης μηνύματος στη δομή, αποτελεί χρονοβόρο διαδικασία, τότε για κάθε νέο μήνυμα, δημιουργείται ένα νέο νήμα ελέγχου το οποίο και αναλαμβάνει τη διαδικασία. Η ταξινόμηση στη δομή γίνεται με βάση τη χρονολογική σειρά λήψης του μηνύματος εκτός και εάν υπάρχει ανάγκη για άλλο τρόπο ταξινόμησης, οπότε και αυτός ορίζεται ξεχωριστά για κάθε τέτοια περίπτωση.

Ο καταχωρήτης γεγονότων αποτελεί αντικείμενο το οποίο διαχειρίζεται εσωτερική δομή και έχει μεθόδους για εκχώρηση και διαγραφή, καθώς και για πρόσβαση στην πληροφορία της δομής. Αποτελείται από έναν κυκλικό πίνακα σταθερού μεγέθους με δύο δείκτες οι οποίοι υποδεικνύουν την αρχή και τέλος του ο οποίος υλοποιεί μια ουρά προτεραιότητας, και ο οποίος επιτρέπει εισαγωγή γεγονότος στο συμβολικό τέλος του και διαγραφή από την αρχή του. Το παρακάτω σχήμα (σχήμα 4.1) περιγράφει τρεις τέτοιες καταστάσεις. Όπως έχει αναφερθεί χρησιμοποιούμε δομή σταθερού μεγέθους για να αποφύγουμε εξάντληση της μνήμης του υπολογιστή, ενώ για να αποφύγουμε το πρόβλημα της ταυτόχρονης εγγραφών ή το πρόβλημα της ταυτόχρονης εγγραφής και ανάγνωσης από νήματα ελέγχου χρησιμοποιούμε ατομική εκτέλεση της κάθε συναλλαγής (transaction) με τη δομή.



Σχήμα 4.1 Στο σχήμα βλέπουμε τρεις δυνατές καταστάσεις του κυκλικού πίνακα ο οποίος χρησιμοποιείται από τα αισθητήρια, προκειμένου να αποθηκευτούν τα αντιλαμβανόμενα γεγονότα.

4.1.2 Γνωσιακό τμήμα

Το γνωσιακό τμήμα αποτελείται από μια ή παραπάνω οντότητες οι οποίες αντιλαμβάνονται το περιβάλλον της οντότητας και παράγουν εντολές τις οποίες διαβιβάζουν στα αντικείμενα δράσης της οντότητας, προς εκτέλεση. Το γνωσιακό τμήμα εμπεριέχει είτε μια οντότητα η οποία εξυπηρετεί αιτήσεις προερχόμενες από τα αισθητήρια, είτε κάποια η οποία εφαρμόζει ένα σχέδιο δράσης είτε και τα δύο. Παράλληλα στο γνωσιακό κομμάτι της οντότητας είναι δυνατόν να περιέχονται και άλλα αντικείμενα ή οντότητες οι οποίες συντελούν στη λειτουργία της οντότητας.

Οντότητα εξυπηρέτησης αιτήσεων: μια τέτοια οντότητα, η οποία ταξινομείται σύμφωνα με τα προηγούμενα κάτω από την κατηγορία των οντοτήτων του γνωσιακού επιπέδου, παρακολουθεί ένα από τα αισθητήρια της οντότητας (αυτό στο οποίο αποστέλλονται οι αιτήσεις προς εξυπηρέτηση) και για κάθε νέα αίτηση αναλαμβάνει την εξυπηρέτηση της. Η εξυπηρέτηση της αίτησης είναι δυνατόν να γίνει είτε από το νήμα ελέγχου το οποίο παρακολουθεί το αισθητήριο, σε περίπτωση όπου η εξυπηρέτηση της αίτησης έχει μικρό υπολογιστικό κόστος, είτε από ένα νέο το οποίο θα δημιουργηθεί για αυτό το σκοπό και θα

την αναλάβει. Στην τελευταία περίπτωση το νέο νήμα ελέγχου το οποίο δημιουργείται εκτελεί την εξυπηρέτηση της αίτησης παράλληλα με το πρώτο νήμα, το οποίο συνεχίζει την παρακολούθηση του αισθητηρίου και δημιουργία νέων νημάτων ελέγχου. Κατά τη χρήση πολλών νημάτων ελέγχου από το γνωσιακό τμήμα της οντότητας, αυτό αναλαμβάνει τη διαχείριση των νημάτων καθώς και τη μεταβίβαση του αποτελέσματος της αίτησης στα δρώντα αντικείμενα της οντότητας. Ανάλογα με τον τρόπο λειτουργίας τον οποίο ακολουθεί η οντότητα οι αιτήσεις είναι δυνατόν να εκτελούνται είτε σειριακά (αποτιθέμενες σε ουρά προτεραιότητας), είτε παράλληλα χρησιμοποιώντας προτεραιότητες οι οποίες καθορίζουν το ποσοστό υπολογιστικής ισχύος το οποίο θα αποδίδεται σε κάθε οντότητα. Οι τιμές των προτεραιοτήτων αυτών κυμαίνονται συνήθως μεταξύ δύο ή τριών τιμών (πχ. αργή, ουδέτερη, επείγουσα).

Για την εξυπηρέτηση μιας αίτησης από την οντότητα απαιτείται να εκτελεστεί κάποιος υπολογισμός ή αναζήτηση σε δομή δεδομένων (ή κάποιο άλλο σύνολο υποδιεργασιών). Οι οδηγίες για την υλοποίηση της εξυπηρέτησης της αίτησης βρίσκονται στο γνωσιακό επίπεδο με τη μορφή αντικειμένων καθώς και τη μεταβίβαση κάποιον "αλγόριθμο" και επιστρέφουν κάποιο αποτέλεσμα για κάθε εισερχόμενη αίτηση. Αναλυτικότερα το γνωσιακό επίπεδο περιέχει αντικείμενα μη προσβάσιμα από άλλες οντότητες τα οποία περιέχουν τις κατάλληλες μεθόδους για την εξυπηρέτηση της αίτησης. Το κάθε νήμα ελέγχου το οποίο εξυπηρετεί μια αίτηση χρησιμοποιεί ένα στιγμιότυπο (instance) κάποιου τέτοιου αντικειμένου ώστε να εξυπηρετήσει την αίτηση. Τα αντικείμενα στα οποία αναφερόμαστε, υλοποιούν γνώση η οποία έχει δοθεί στην οντότητα από την κατασκευή της [11][45], και συνήθως αυτή είναι στατική, δηλαδή για την ίδια αίτηση και υπό τις ίδιες συνθήκες θα έχουμε το ίδιο αποτέλεσμα, όσον αφορά τον τρόπο εξυπηρέτησης. Ωστόσο είναι δυνατόν να εφαρμόζεται κάποιος αλγόριθμος μηχανικής μάθησης, ο οποίος να επιτρέπει στο αντικείμενο να βελτιώνει την απόδοσή του, καθώς αυξάνεται η εμπειρία του στο συγκεκριμένο περιβάλλον.

Οντότητα εφαρμογής σχεδίου δράσης: σε κάθε τέτοια οντότητα, έχει ανατεθεί ένα σχέδιο δράσης το οποίο το νήμα ελέγχου της οντότητας, αναλαμβάνει να εφαρμόσει. Κατά την εφαρμογή του σχεδίου δράσης της οντότητας είναι δυνατόν να δημιουργηθούν νέες οντότητες, να επιτελείται επικοινωνία με άλλες ή να παρακολουθούνται αισθητήρια της οντότητας. Οι υποδιεργασίες αυτές καθώς και ο τρόπος με τον οποίο επιτελούνται καθορίζονται από το σχέδιο δράσης το οποίο ανατίθεται δυναμικά στην οντότητα. Ξανά στη περίπτωση δημιουργίας νέων οντοτήτων η νημάτων ελέγχου, η οντότητα η οποία τα δημιούργησε είναι υπεύθυνη για τη διαχείρισή τους, χειρισμό σφαλμάτων και τερματισμό λειτουργίας τους.

Ενδιαφέρον είναι ο συνδυασμός της λειτουργίας των δύο παραπάνω οντοτήτων μαζί. Δηλαδή, είναι δυνατόν να έχουμε μια οντότητα η οποία να εξυπηρετεί αιτήσεις και για κάθε μια την οποία εξυπηρετεί να δημιουργεί μια οντότητα της τελευταίας κατηγορίας, στην οποία ανάλογα με τον τύπο της αίτησης να αναθέτει το κατάλληλο σχέδιο δράσης (σε κάποιο πιο πολύπλοκο σύστημα ίσως να ήθελε κανείς να υπολογίζει το σχέδιο δράσης εκείνη τη στιγμή). Η οντότητα δηλαδή η οποία ανέλαβε την εξυπηρέτηση της αίτησης είναι μεγαλύτερου βαθμού πολυπλοκότητας από μια της πρώτης κατηγορίας και, ανάλογα βέβαια και με το ανατιθέμενο σχέδιο δράσης, πιο αυτόνομη, με την έννοια του ότι ανέλαβε την εξυπηρέτηση μιας μη τετριμμένης διαδικασίας της οποίας η εξυπηρέτηση δεν καλύπτεται με έναν απλό υπολογισμό ενώ η ίδια η αίτηση είναι δυνατόν να απαιτεί μια οντότητα να ασχολείται με ένα γεγονός ή περιστατικό καθ' όλη τη διάρκεια εξέλιξής του.

Χαρακτηριστικό παραδείγματα τέτοιας οντότητας, είναι το γνωσιακό επίπεδο της οντότητας εξυπηρέτησης περιστατικών η οποία για κάθε εισερχόμενη αίτηση δημιουργεί μια νέα οντότητα η οποία θα την εξυπηρετήσει. Ανάλογα με τον τύπο του περιστατικού, σε κάθε νέα οντότητα η οποία δημιουργείται, ανατίθεται ένα σχέδιο δράσης το οποίο είναι εξαρτάται από τον τύπο του περιστατικού. Η οντότητα η οποία δημιουργείται αναλαμβάνει την εξυπηρέτηση του περιστατικού, διεργασία η οποία εμπεριέχει επικοινωνία με άλλες οντότητες και δημιουργία νέων οντοτήτων, ενώ διαχειρίζεται το περιστατικό από τη στιγμή εκδήλωσής του έως τη λήξη του.

4.1.3 Αντικείμενα δράσης

Τα αντικείμενα δράσης της οντότητας προσπαθούν να εφαρμόσουν εντολές οι οποίες υπαγορεύονται από το γνωσιακό επίπεδο. Συνήθως η λειτουργία του περιορίζεται είτε στην αποστολή αποτελέσματος κάποιου υπολογισμού προς την αντίστοιχη αιτούσα οντότητα.

4.2 Προγραμματιστική διεπιφάνεια χρήσης

Οι βασικές λειτουργίες μιας απλής οντότητας υλοποιούνται σε αφηρημένες κλάσεις (abstract classes) [53] τις οποίες μπορεί να χρησιμοποιήσει ένας προγραμματιστής για να υλοποιήσει κάποια εξειδικευμένη οντότητα. Αναλυτικότερα κατά την υλοποίηση μιας νέας οντότητας, ο προγραμματιστής μπορεί να δημιουργήσει μια υποκλάση της δοθείσας κλάσης, κληρονομώντας όλη τη βασική λειτουργικότητα της οντότητας και έχοντας να συμπληρώσει μόνο το τμήμα της οντότητας το οποίο την εξειδικεύει στη συγκεκριμένη εργασία για την οποία σχεδιάζεται. Επιπλέον ο προγραμματιστής είναι δυνατόν να επιλέγει μεταξύ εναλλακτικών τρόπων υλοποίησης της κάθε οντότητας ή να ορίσει παραμέτρους λειτουργίας της.

Έτσι για τα αισθητήρια αντικείμενα μια οντότητας ο προγραμματιστής είναι δυνατόν να επιλέξει μεταξύ τρόπου απομάκρυνσης δεδομένων από τον καταχωρητή γεγονότων, περίοδο ελέγχου του αισθητηρίου, τρόπου ταξινόμησης των δεδομένων στο καταχωρητή γεγονότων και μέγεθος του καταχωρητή αυτού.

Για τις οντότητες του γνωσιακού επιπέδου, είναι δυνατόν να επιλέξει κανείς μεταξύ οντοτήτων οι οποίες εξυπηρετούν αιτήσεις ή άλλες οι οποίες υλοποιούν σχέδια δράσης ή ακόμη και να κάνει συνδυασμούς μεταξύ των δύο. Άλλες επιλογές είναι αυτή της παράλληλης με προτεραιότητες ή σειριακή εκτέλεση αιτήσεων κ.τ.λ.

Τέλος τα αντικείμενα δράσης εμπεριέχουν το μηχανισμό μετάδοσης ενός μηνύματος καθώς και αυτόν της κωδικοποίησης του σε ακολουθία bytes για την περίπτωση αποστολής τους πάνω από δίκτυο.

5 Ανάθεση στόχου σε πράκτορα

Έχοντας δει τις γενικές αρχές σχεδιασμού και λειτουργίας μιας οντότητας του συστήματος, θα μελετήσουμε τους τρόπους με τους οποίους είναι δυνατόν να προσαρμόσει τη συμπεριφορά της ανάλογα με τις ανάγκες του συστήματος. Κάτι τέτοιο είναι επιθυμητό σε μεταβαλλόμενα περιβάλλοντα, εφόσον θα ήθελε κανείς να αυξήσει την απόδοση ενός συστήματος εκμεταλλευόμενος τη γνώση αυτών των αλλαγών. Επιπλέον η μεταβολή της συμπεριφοράς είναι δυνατόν να είναι επιθυμητή για το χειρισμό απρόβλεπτων καταστάσεων, τη δυνατότητα εστίασης της προσοχής του συστήματος σε συγκεκριμένες εργασίες, καθώς και σφαλμάτων συστήματος.

Σε ένα σύστημα, η παραγόμενη από την οντότητα συμπεριφορά, εξαρτάται από τις οδηγίες οι οποίες της έχουν ανατεθεί από τον προγραμματιστή. Συνήθως οι οδηγίες αυτές είναι ήδη υπάρχουσες (και μεταφρασμένες σε γλώσσα μηχανής) στο γνωσιακό επίπεδο της κάθε οντότητας. Με αυτό τον τρόπο η οντότητα συμπεριφέρεται ακολουθώντας την ίδια πολιτική καθόλη τη διάρκεια ύπαρξής της. Ωστόσο, στην προσπάθεια βελτιστοποίησης της απόδοσης του συστήματος είναι δυνατόν να θέλει κανείς να μεταβάλλει την τρέχουσα πολιτική, προς όφελος της απόδοσης του συστήματος (στο σύστημα μας ένα τέτοιο παράδειγμα, είναι η αλλαγή του αλγόριθμου προσδιορισμού διαδρομής περιπολίας, μετά από στατιστικές μετρήσεις). Η νέα πολιτική η οποία θα ακολουθηθεί, είναι δυνατόν να επιλέγεται από ένα σύνολο προμετεφρασμένων, είτε να δημιουργείται δυναμικά με υπολογιστικό τρόπο (πχ. από κάποιον αλγόριθμο).

Στη συνέχεια θα περιγράψουμε τον τεχνικό τρόπο τον οποίο χρησιμοποιούμε έτσι ώστε να μπορούμε να αναθέτουμε οδηγίες σε μια οντότητα, οι οποίες περιγράφουν την επιθυμητή από αυτή συμπεριφορά. Στη συνέχεια θα αναφερόμαστε στην περιγραφή αυτή πολιτικής και ενεργειών ως *σχέδιο δράσης*.

5.1 Σχέδια δράσης

Το σχέδιο δράσης είναι δυνατόν να αναπαρασταθεί με ένα διάγραμμα ροής το οποίο περιγράφει τις προς εκτέλεση ενέργειες καθώς και την επιλογή μεταξύ ενεργειών βάση κάποιας πληροφορίας, όπως η τιμή κάποιας μεταβλητής, ο τερματισμός η όχι κάποιας διεργασίας κ.τ.λ. Στην περίπτωση των οντοτήτων, οι ενέργειες αυτές υλοποιούνται από μεθόδους αντικειμένων της οντότητας (βλ. Μοντέλο μιας τυπικής οντότητας).

Για να υλοποιήσει κανείς ένα τέτοιο διάγραμμα θα πρέπει να δημιουργήσει τις δομές των κόμβων καθώς και τις συνδέσεις μεταξύ αυτών. Κάθε κόμβος είναι δυνατόν να αναφέρεται σε μια η παραπάνω προγραμματιστικές εντολές, φυλλάσσοντας μια αναφορά προς τη μέθοδο ενός αντικειμένου καθώς και τις παραμέτρους με τις οποίες θα κληθεί αυτή. Επιπλέον, επειδή μια εντολή είναι δυνατόν να εκτελεστεί παράλληλα (εάν ανατεθεί σε ένα ξεχωριστό νήμα ελέγχου) φυλάσσεται και ο επιθυμητός τρόπος εκτέλεσης της. Η επιλογή μεταξύ ενεργειών (δηλαδή τα σημεία απόφασης του διαγράμματος ροής), υλοποιούνται με `if - then - else` δηλώσεις. Επίσης λόγω του ότι η επιλογή μιας απόφασης είναι δυνατόν να εξαρτάται από το αποτέλεσμα ενός υπολογισμού, το σχέδιο δράσης (ως προγραμματιστικό αντικείμενο), περιέχει μια δομή ευρετηρίου, η οποία αντιστοιχίζει ονόματα μεταβλητών με τιμές και την οποία χρησιμοποιεί για την αποθήκευση και ανάκληση υπολογιστικών αποτελεσμάτων.

5.2 Εφαρμογή σχεδίων δράσης

Η οντότητα η οποία θα ακολουθήσει ένα σχέδιο δράσης χρησιμοποιεί ένα νήμα ελέγχου για αυτό το σκοπό. Το νήμα ελέγχου αρχίζει προσπελώνοντας τον αρχικό κόμβο και εκτελώντας τις μεθόδους των αντικειμένων στα οποία αναφέρεται. Ανάλογα με την εξέλιξη των εκτελούμενων εντολών ή άλλων γεγονότων, το νήμα ελέγχου βρίσκει τον επόμενο κόμβο προς εκτέλεση και συνεχίζει την ίδια διαδικασία.

Οι μέθοδοι οι οποίοι αναφέρονται στο σχέδιο δράσης, αποτελούν μεθόδους αντικειμένων του γνωσσιακού επιπέδου της οντότητας η οποία το εκτελεί. Στις μεθόδους αυτές είναι δυνατόν να συγκαταλέγονται μέθοδοι, οι οποίες επιτελούν επικοινωνία με άλλες οντότητες προκειμένου να λάβουν αποτελέσματα υπολογισμών, πληροφορία από τα αισθητήρια κ.α. Ανάλογα με τον τρόπο εκτέλεσης ο οποίος έχει οριστεί, μια μέθοδος είναι δυνατόν να εκτελεστεί παράλληλα με την εκτέλεση του σχεδίου δράσης. Ωστόσο σε περίπτωση όπου το σχέδιο δράσης χρησιμοποιεί στη συνέχεια το αποτέλεσμα του υπολογισμού θα πρέπει να υπάρχει συγχρονισμός, έτσι ώστε είτε αυτό να είναι διαθέσιμο τη χρονική στιγμή χρήσης του, είτε το νήμα ελέγχου το οποίο θα το χρησιμοποιήσει να περιμένει για την περάτωση του υπολογισμού του.

5.3 Πλεονεκτήματα και λόγοι χρήσης των σχεδίων δράσης

Ο κυριότερος λόγος για την επιλογή της χρήσης σχεδίων δράσης, ήταν η διάθεση για την ανάπτυξη μεταβλητής συμπεριφοράς από μια προγραμματιστική οντότητα. Στη συνέχεια θα θέλαμε βέβαια η μεταβολή αυτή να γίνεται με την πρόθεση της προσαρμογής της οντότητας στο περιβάλλον και τη βελτιστοποίηση της απόδοσης της. Η αλλαγή του σχεδίου δράσης μιας οντότητας είναι δυνατόν να αποτελεί την αντίδραση του συστήματος απέναντι στην παρατήρηση μιας μεταβολής στο περιβάλλον, την αλλαγή πολιτικής με σκοπό τη βελτιστοποίηση της απόδοσης, την εστίαση της προσοχής σε κάποια διεργασία, ή ακόμα και τις οδηγίες για την ανάρρωση από κάποιο σφάλμα συστήματος.

Στη γενική περίπτωση με τη χρήση σχεδίων δράσης είναι δυνατόν να έχουμε μια οντότητα η οποία να είναι ειδικευμένη σε κάποιο σύνολο εργασιών (αυτές τις οποίες υλοποιούν οι οντότητες του γνωσσιακού της επιπέδου). Το σχέδιο δράσης υπαγορεύει τον τρόπο με τον οποίο θα γίνουν, την πολιτική δηλαδή την οποία θα ακολουθεί η οντότητα στη επιτέλεση των εργασιών οι οποίες τις ανατίθενται.

Το σχέδιο δράσης το οποίο ακολουθεί η κάθε οντότητα είναι δυνατόν να επιλέγεται από σύνολο προκατασκευασμένων ή να δημιουργείται δυναμικά από αλγόριθμο τον οποίο εκτελεί η ίδια ή άλλη οντότητα.

Ωστόσο εκτός από τους λόγους δυνατότητας αλλαγής της συμπεριφοράς μιας οντότητας, είναι δυνατόν να δούμε και κάποια πλεονεκτήματα στη χρήση τους σε προγραμματιστικό επίπεδο. Αυτά είναι:

- *Η δυνατότητα χρήσης μιας οντότητας με διαφορετικούς τρόπους.* Είναι δυνατόν να έχουμε μια οντότητα γενικής χρήσης της οποία να χρησιμοποιούμε κάθε φορά ανάλογα με την εφαρμογή μας. Η χρήση τέτοιων οντοτήτων είναι παρόμοια με αυτή μιας βιβλιοθήκης μεθόδων. Για παράδειγμα στο σύστημα μας η οντότητα εύρεσης συντομότερων διαδρομών, διαθέτει μεθόδους τόσο για εύρεσης διαδρομών με ποικίλους τρόπους, αλλά και άλλες όπως η εύρεση της μεγαλύτερης απόστασης στο γράφο ή η εκτίμηση του χρόνου που θα κάνει μια κινητή μονάδα να φτάσει στον προορισμό της. Έτσι είναι δυνατόν να έχουμε βιβλιοθήκες από οντότητες γενικής χρήσης τις οποίες να εξειδικεύουμε σε πράκτορες, εκφράζοντας μόνο την επιθυμητή από την οντότητα συμπεριφορά και χωρίς να χρειάζεται η γνώση του τρόπου υλοποίησης της οντότητας. Στην εφαρμογή μας, κάτι τέτοιο είναι επιπλέον χρήσιμο, διότι τα ιατρικά πρωτόκολλα αντιμετώπισης επειγόντων περιστατικών, διαφέρουν μεταξύ των υπεύθυνων οργανισμών σε διάφορες χώρες. Κατά αυτό τον τρόπο η ίδια εφαρμογή με κατάλληλη ρύθμιση του σχεδίου αντιμετώπισης περιστατικών, μπορεί να χρησιμοποιηθεί σε διαφορετικά περιβάλλοντα.
- *Η δυνατότητα επέκτασης της εφαρμογής στο μέλλον.* Εάν αλλάξει μια παράμετρος της εφαρμογής που έχουμε και θέλουμε να κάνουμε αλλαγές στην εφαρμογή μας δεν θα είναι απαραίτητη η συγγραφή εκ νέου του πηγαίου κώδικα της εφαρμογής. Είναι δυνατόν να αλλάξουμε μόνο το σχέδιο δράσης, ενώ μάλιστα κάτι τέτοιο είναι δυνατόν να συμβεί καθώς η εφαρμογή βρίσκεται σε λειτουργία.
- *Ο εύκολος σχεδιασμός και υλοποίηση μιας οντότητας.* Έχοντας στη διάθεση κανείς τις μεθόδους οι οποίες υλοποιούν τις επιθυμητές από την οντότητα λειτουργίες και χρησιμοποιώντας το προγραμματιστικό πρότυπο μιας αυτόνομης οντότητας, είναι δυνατόν να την υλοποιήσει με μικρό χρόνο ανάπτυξης. Εάν μάλιστα γίνει εργασία πάνω στην

περιγραφή των λειτουργιών των αντικειμένων (κάτι το οποίο είναι εφικτό χρησιμοποιώντας μια γλώσσα περιγραφής της προγραμματιστικής διεπιφάνειας χρήσης ενός αντικειμένου - Interface Definition Languages [45]), είναι δυνατόν η ο σχεδιασμός και υλοποίηση μιας οντότητας να γίνεται μέσω κάποιου αυτόματου περιβάλλοντος.

5.4 Συμπεράσματα

Η χρήση των σχεδίων δράσης δεν εγγυάται από μόνη της την αυτόνομη συμπεριφορά των οντοτήτων. Ωστόσο δημιουργεί την υποδομή τόσο για προσαρμογή της συμπεριφοράς των οντοτήτων στο περιβάλλον και την εύκολη διαβίβαση οδηγιών προς αυτές. Εξάλλου, εάν καταφέρουμε να περιγράψουμε την επιθυμητή από την οντότητα συμπεριφορά, τότε ουσιαστικά θα έχουμε μια οντότητα η οποία θα συμπεριφέρεται και με αυτό τον τρόπο.

Προγραμματιστικά, ο τρόπος υλοποίησης των οντοτήτων, ακολουθεί το μοντέλο των αυτόνομων πρακτόρων με την έννοια του ότι είναι δυνατόν να έχουμε ένα αντικείμενο γενικής φύσης, από το οποίο ζητάμε την εκτέλεση διεργασιών. Μάλιστα με κατάλληλη διεπιφάνεια χρήσης (η οποία να βοηθά το χρήστη να δημιουργεί σχέδια δράσης με αυτοματοποιημένο τρόπο), ο χρήστης είναι δυνατόν να εκφράσει πιο αφηρημένες αιτήσεις προς την οντότητα, χρησιμοποιώντας ως τμήματα του υπό κατασκευή σχεδίου έτοιμες προτάσεις η δηλώσεις.

6 Σχεδίαση ενός πράκτορα με παραπάνω του ενός στόχου

Στην ενότητα αυτή θα περιγράψουμε την αρχιτεκτονική ενός πράκτορα στον οποίο έχουν ανατεθεί παραπάνω του ενός στόχοι. Ως τέτοιου είδους οντότητα μπορεί να περιγραφεί το ίδιο το σύστημα διαχείρισης πόρων το οποίο αποτελεί πράκτορα με πολλαπλούς και αλληλεξαρτώμενους στόχους. Η σχεδίαση μιας τέτοιας οντότητας δεν διαφέρει από την τυπική οντότητα του συστήματος όπως την έχουμε περιγράψει προηγούμενα. Ωστόσο, εκτός από τα θέματα υλοποίησης τα οποία προκύπτουν, το ενδιαφέρον μας τώρα εστιάζεται στον τρόπο συνεργασίας μεταξύ των στόχων αυτών, καθώς και στον καθορισμό προτεραιοτήτων, όσον αφορά την επιλογή ενεργειών του πράκτορα.

Στόχος μας είναι η δημιουργία μιας οντότητας της οποίας το γνωσιακό επίπεδο δεν έχει κάποιο κεντρικό σημείο ελέγχου ή παραγωγής συμπεριφοράς. Θα θέλαμε η, από τον εξωτερικό παρατηρητή, παρατηρούμενη αυτονομία και προσαρμοστικότητα, να είναι αποτέλεσμα της δράσης πολλών εξειδικευμένων σε κάποια εργασία πρακτόρων. Όπως θα δούμε στην ενότητα των συμπερασμάτων, ο τρόπος δημιουργίας σύνθετων οντοτήτων βασίζεται στην κατασκευή τους από απλούστερες. Όσο πιο απλά είναι τα δομικά στοιχεία της οντότητας, τόσο μεγαλώνει το εύρος των δυνατών επιλογών στη συμπεριφορά της οντότητας.

Όπως έχουμε αναφέρει, ένας στόχος στο γνωσιακό επίπεδο, υλοποιείται από μία οντότητα στην οποία έχει ανατεθεί κάποιο καθήκον. Η οντότητα αυτή, λειτουργώντας σύμφωνα με το σχέδιο δράσης της, αναλαμβάνει την προσπάθεια επίτευξης του στόχου, ο οποίος μπορεί να είναι η εξυπηρέτηση κάποιου τύπου αιτήσεων ή άλλου είδους εργασία, όπως η με πρωτοβουλία της οντότητας διαχείριση πόρων του συστήματος (πχ. περιπολία μη απασχολημένων κινητών μονάδων). Η οντότητα επίτευξης στόχων μπορεί να χαρακτηριστεί ως υψηλού επιπέδου οντότητα μιας και δημιουργεί άλλες στις οποίες αναθέτει εργασίες ενώ πολλές φορές επικοινωνεί και με άλλες οντότητες του γνωσιακού επιπέδου.

Η ύπαρξη ενός κεντρικού σημείου ελέγχου της ροής εργασιών τείνει να δεσμεύσει την λειτουργία της οντότητας σε ισχυρά καθορισμένη συμπεριφορά. Επιπλέον έχοντας κανείς να σχεδιάσει ένα σύστημα το οποίο πρέπει να επιτελεί ποικίλες και πολύπλοκες εργασίες, θα ήθελε να διασπάσει αυτές σε απλούστερες. Έτσι επιλέγουμε το να σχεδιάζουμε μια οντότητα για κάθε εργασία και στη συνέχεια να προσπαθήσουμε να βάλουμε τις οντότητες αυτές να συνεργαστούν. Μέχρις εδώ η διαφορά μεταξύ του να σχεδιάσουμε μια οντότητα η οποία μόνη της αναλαμβάνει όλες τις εργασίες του συστήματος με το να δημιουργήσουμε πολλές οι οποίες αναλαμβάνουν μια εργασία η κάθε μια, είναι μόνο προγραμματιστική και μπορεί κάποιος να αποδείξει πως με κατάλληλο σχεδιασμό είναι δυνατόν να δημιουργήσει οντότητα η οποία να έχει την ίδια συμπεριφορά ανεξάρτητα με τον τρόπο σχεδιασμού. Στη συνέχεια θα δούμε ωστόσο τα πλεονεκτήματα της αποσύνθεσης πολύπλοκων εργασιών σε απλούστερες και την ανάθεση στις αντίστοιχες οντότητες καθώς όμως και ζητήματα τα οποία πρέπει να λάβουμε υπ' όψιν προκειμένου να λειτουργεί σωστά.

6.1 Πλεονεκτήματα κατανεμημένης σχεδίασης

- *Ευκολότερος σχεδιασμός.* Ο διαχωρισμός πολύπλοκων εργασιών σε απλούστερες μας επιτρέπει τον ευκολότερο σχεδιασμό του συστήματος. Είναι προφανές ότι σχεδιάζοντας μια οντότητα για κάθε στόχο ξεχωριστά, η εργασία γίνεται ευκολότερη από το να προσπαθούσε κανείς να κατασκευάσει μια η οποία να έκανε όλες τις εργασίες. Ωστόσο το πλεονέκτημα αυτό δεν αποτελεί από μόνο του αίτιο για την συγκεκριμένη επιλογή σχεδίασης.
- *Επεκτασιμότητα.* Εάν θα θέλαμε να επεκτείνουμε το σχεδιαζόμενο σύστημα, έτσι ώστε να κάνει περισσότερες εργασίες, στην περίπτωση του κεντρικοποιημένου σχεδιασμού, θα έπρεπε να επανασχεδιάσουμε όλη την οντότητα εξ' αρχής, αφού τις περισσότερες φορές οι ανατιθέμενες εργασίες θα παρουσιάζουν κάποια εξάρτηση μεταξύ τους (κάτι τέτοιο συμβαίνει όχι μόνο λόγω της συνεργασίας των οντοτήτων μεταξύ τους, αλλά και λόγω της διαχείρισης κοινών πόρων από τις οντότητες). Από την άλλη μεριά, έχοντας μια οντότητα για κάθε εργασία, μπορούμε να δημιουργούμε νέες οντότητες τις οποίες να προσθέτουμε στο σύστημα μας.

- *Καλύτερη απόδοση λόγω εξειδίκευσης.* Έχοντας μια οντότητα για κάθε εργασία είναι ευκολότερο κανείς να εξειδικεύσει το σχεδιασμό της οντότητας προς την εργασία αυτή. Επιπρόσθετα η ίδια η οντότητα είναι δυνατόν με χρήση κάποιου αλγόριθμου μηχανικής μάθησης να βελτιώνει την απόδοσή της.
- *Δυνατότητα κατανομής της εργασίας σε πολλούς υπολογιστές.* Δεδομένης της αυτονομίας της κάθε οντότητας είναι δυνατόν να μεταφέρουμε κάποιες από αυτές σε άλλους υπολογιστές, έτσι ώστε να αποφύγουμε τη δυσλειτουργία του συστήματος λόγω ανεπάρκειας υπολογιστικών πόρων (κυρίως αυτού της υπολογιστικής ισχύος του επεξεργαστή). Βέβαια σε αυτή την περίπτωση προκύπτουν ζητήματα τόσο επικοινωνίας, όσο και εύρεσης τρόπου διαμοιρασμού κοινής μνήμης.
- *Δυνατότητα επιλογής οντότητας για εργασία.* Είναι δυνατόν να έχουμε στη διάθεση μας παραπάνω από μια οντότητα για κάθε εργασία. Σε αυτή την περίπτωση το σύστημα αποκτά επιλογή, όσον αφορά την ανάθεση κάποιας εργασίας σε οντότητα. Κάτι τέτοιο είναι δυνατόν να βρει εφαρμογή, σε περιπτώσεις όπου τμήματα του συστήματος, είναι σχεδιασμένα από τρίτους κατασκευαστές ή σε περίπτωση όπου έχουμε στη διάθεση μας παραπάνω από μια οντότητα η οποία να μπορεί να επιτελέσει μια συγκεκριμένη εργασία, οπότε και επιλέγουμε την πιο κατάλληλη υπό τις τρέχουσες συνθήκες.
- *Δυνατότητα καθορισμού προτεραιοτήτων.* Αντιμετωπίζοντας κάθε στόχο ως ξεχωριστή διεργασία, μπορούμε να ορίσουμε το ποσοστό υπολογιστικής ισχύος το οποίο θέλουμε να αντιστοιχίσουμε σε κάθε μια, δίνοντας έτσι προτεραιότητα στους κρίσιμους στόχους και αφήνοντας να εκτελεστούν οι δευτερεύοντες μόνο σε περίπτωση επάρκειας υπολογιστικών πόρων.

Γενικά, εκτός από τα προαναφερθέντα πλεονεκτήματα, στόχος της εργασίας αποτελεί και η δοκιμή της χρήσης αρχιτεκτονικής αποτελούμενης από αυτόνομους πράκτορες για τη λύση κάποιου προβλήματος. Εξάλλου, όπως έχουμε δει προηγούμενα και θα συζητήσουμε στη συνέχεια (στην ενότητα των συμπερασμάτων), πιστεύουμε πως η συνεργασία μεταξύ πολλών και απλών αυτόνομων οντοτήτων, μπορεί ευκολότερα να δώσει αυτονομία και προσαρμοστικότητα στη συμπεριφορά του συστήματος.

6.2 Ζητήματα σχεδίασης

Λόγω της κατανεμημένης σχεδίασης του συστήματος προκύπτουν ζητήματα συγχρονισμού μεταξύ των εργασιών καθώς και ελέγχου ροής τους. Επίσης, με το να αποφεύγουμε την ύπαρξη κεντρικής μονάδας ελέγχου, προκύπτει το ζήτημα της εύρεσης του τρόπου με τον οποίο τελικά θα γίνεται ο έλεγχος. Η λύση η οποία ακολουθούμε και στα δύο ζητήματα είναι η κατάλληλη σχεδίαση του δικτύου επικοινωνίας μεταξύ των οντοτήτων. Με άλλα λόγια ο έλεγχος ροής των διεργασιών βρίσκεται αποτυπωμένος στον τρόπο με τον οποίο επικοινωνούν οι οντότητες μεταξύ τους. Όπως έχουμε δει σε προηγούμενη ενότητα (βλ Ενότητα 2) το δίκτυο αυτό μπορεί να είναι σταθερό και σχεδιασμένο από τον προγραμματιστή του συστήματος ή δυναμικά μεταβαλλόμενο από το ίδιο το σύστημα. Το σύστημα μας ανήκει στην πρώτη κατηγορία κυρίως λόγω της απαίτησης αντιμετώπισης των ιατρικών περιστατικών με συγκεκριμένο τρόπο ο οποίος ορίζεται από το εκάστοτε ιατρικό πρωτόκολλο.

Όσον αφορά τη χρήση κοινών πόρων από τις οντότητες χρησιμοποιούμε μεταβλητές κλειδώματος [42], για να αποφύγουμε αδιέξοδα και συνθήκες ανταγωνισμού (race conditions). Οι πόροι των οποίων η χρήση είναι δυνατόν να διαμοιραστεί μεταξύ πολλών οντοτήτων (πχ. κινητές μονάδες), φέρουν μια λογική μεταβλητή δύο καταστάσεων (boolean) η οποία δείχνει το εάν ο πόρος είναι διαθέσιμος ή όχι.

Ο συγχρονισμός καθώς και η υλοποίηση της απαιτούμενης ροής ελέγχου υλοποιείται μέσω του δικτύου επικοινωνίας των οντοτήτων και πιο συγκεκριμένα μέσω του σχεδίου δράσης της κάθε οντότητας το οποίο καθορίζει το χρόνο και το σκοπό για τον οποίο, ή οντότητα που το εκτελεί θα επικοινωνήσει με κάποια άλλη. Μπορεί να παρατηρήσει κανείς πως μέχρις εδώ δεν έχουμε λύσει το πρόβλημα του συγχρονισμού και ελέγχου ροής, απλά το μεταθέσαμε στην ορθότητα των σχεδίων δράσης τα οποία ακολουθούν οι οντότητες. Ουσιαστικά είναι εύθνη του προγραμματιστή της εφαρμογής η ορθή σχεδίαση του και στην ουσία του συγγραφέα του ιατρικού πρωτοκόλλου.

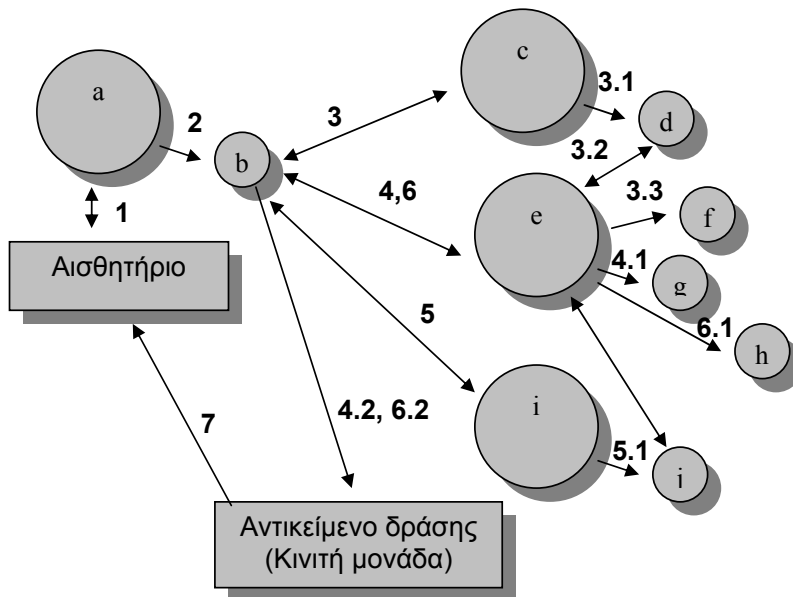
Στην περίπτωση του συστήματος μας, τα σχέδια δράσης των οντοτήτων δημιουργούνται από τον προγραμματιστή και για αυτό το δίκτυο επικοινωνίας και ελέγχου ροής των οντοτήτων ταξινομείται κάτω από την κατηγορία των μη δυναμικά δημιουργούμενων δικτύων (hand built networks). Ωστόσο η ύπαρξη σχεδίων δράσης, σε αντιπαράθεση με τον καθορισμό του τρόπου και χρόνου επικοινωνίας στον πηγαίο κώδικα των οντοτήτων, μας δίνει τη δυνατότητα οργάνωσης του δικτύου και με δυναμικό τρόπο. Κάτι τέτοιο θα ήταν εφικτό, εάν τα σχέδια δράσης παράγονταν δυναμικά από τις οντότητες ανάλογα με τις τρέχουσες συνθήκες του συστήματος.

Ως σχόλιο θα θέλαμε να αναφέρουμε, πως όσο μεγαλώνει η πολυπλοκότητα του δικτύου και των σχεδίων δράσης των οντοτήτων, τόσο πιο «ευφυής» φαίνεται η συμπεριφορά του συστήματος προς τον εξωτερικό παρατηρητή, ειδικά μάλιστα όταν το σχέδιο δράσης υπαγορεύει την επιλογή της καταλληλότερης ενέργειας η σχεδίου δράσης για την κάθε κατάσταση του συστήματος. Παρόλα αυτά, όμως ακόμα και η δυναμική σχεδίαση σχεδίων δράσης στον υπολογιστή θα γινόταν μέσω κάποιου είδους αλγορίθμου, τουλάχιστον με τις συμβατικές μεθόδους προγραμματισμού και όχι ίσως με τη χρήση νευρωνικού δικτύου η αφηρημένης λογικής (fuzzy logic) [58]. Σε αυτή την περίπτωση μπορεί κανείς να επιχειρηματολογήσει, πως δεν έχουμε να κάνουμε με κάποιο σύστημα συνθετικής ευφυΐας, αλλά με ένα λεπτομερώς προγραμματισμένο σύστημα, αφού ακόμα και η μάθηση του συστήματος από εμπειρία βασίζεται σε αλγόριθμους. Η συζήτηση για το εάν η ευφυΐα είναι δυνατόν να υπάρξει σε κάποιο συγκεκριμένο προγραμματιστικό τμήμα ενός προγράμματος ή εάν είναι ένα επιπρόσθετο χαρακτηριστικό το οποίο έχουν τα πολύπλοκα δίκτυα ροής της πληροφορίας (εξάλλου τα δίκτυα τα οποία συναντά κανείς στη φύση είναι πολλές τάξεις μεγέθους πολυπλοκότερα από το δικό μας και φυσικά πολύ πιο εξειδικευμένα) καθώς και για το εάν θεωρούμε ευφυή μια οντότητα η οποία απλώς φέρει ευφυή χαρακτηριστικά παρατηρούμενα από κάποιον εξωτερικό παρατηρητή, είναι προφανώς έξω από τους στόχους και το πεδίο μελέτης αυτής της εργασίας. Αυτό που μας ενδιαφέρει είναι η δημιουργία κάποιας οντότητας η οποία να επιτελεί τις ανατιθέσες εργασίες με βέλτιστο τρόπο σε σχέση με τις συνθήκες του περιβάλλοντος της, ενώ να παρουσιάζει κάποια τουλάχιστο ευφυή χαρακτηριστικά κυρίως όσον αφορά τη μάθηση, τη λήψη πρωτοβουλιών και τη περιορισμένη ανάγκη σε οδηγίες λειτουργίας από το χρήστη.

Θα παραθέσουμε στη συνέχεια, ένα απλό παράδειγμα για την καλύτερη κατανόηση του τρόπου υλοποίησης του δικτύου επικοινωνίας το οποίο χρησιμοποιούμε. Το παράδειγμα προέρχεται από την οντότητα εξυπηρέτησης επειγόντων περιστατικών, ενώ το σχέδιο δράσης περιγράφεται αναλυτικότερα στην συνέχεια όπου παρουσιάζεται και με μεγαλύτερη λεπτομέρεια η οντότητα αυτή. Εδώ θα επικεντρώσουμε τη προσοχή στον τρόπο με τον οποίο επιτυγχάνεται ο χρονισμός στην εκτέλεση των ενεργειών.

Για την εξυπηρέτηση ενός τυπικού περιστατικού, απαιτείται η επιλογή της κινητής μονάδας η οποία θα εξυπηρετήσει το περιστατικό. Αφού επιλεγεί αυτή (και μόνο τότε), είναι δυνατόν να υπολογιστεί η διαδρομή της κινητής μονάδας από την τρέχουσα θέση της σε αυτή του περιστατικού. Η διεργασία της εύρεσης της διαδρομής αυτής θα πρέπει αναγκαστικά να ανασταλεί έως ότου επιλεγεί η κατάλληλη κινητή μονάδα. Παρόμοια στη συνέχεια και αφού έχουμε δώσει εντολή στην κινητή μονάδα να ακολουθήσει την υπολογισθείσα διαδρομή, επιλέγουμε, μέσω της αντίστοιχης οντότητας το ιατρικό κέντρο στο οποίο θα μεταβιβαστεί ο ασθενής και τότε μπορούμε να υπολογίσουμε και τη διαδρομή από τη θέση του περιστατικού προς το ιατρικό κέντρο το οποίο επιλέχθηκε. Κατά την περιγραφή του δικτύου επικοινωνίας που περιγράψαμε θέλουμε να επισημάνουμε την επιπλέον επικοινωνία που συμβαίνει μεταξύ των οντοτήτων και η οποία δεν φαίνεται άμεσα από τα προηγούμενα. Πιο συγκεκριμένα, η επιλογή, για παράδειγμα, καταλληλότερης κινητής μονάδας, γίνεται λαμβάνοντας υπ' όψιν δύο παραμέτρους: αυτή της καταλληλότητας του εξοπλισμού της κινητής μονάδας καθώς και της εγγύτητας της ως προς την τοποθεσία του περιστατικού. Για να αποφανθεί όμως η οντότητα επιλογής κατάλληλης κινητής μονάδας για το ποια είναι η κοντινότερη κινητή μονάδα, θα πρέπει να επικοινωνήσει με την οντότητα εύρεσης συντομότερων διαδρομών, αφού αναφερόμαστε σε χρονικές αποστάσεις μέσω του οδικού δικτύου και όχι σε Ευκλείδεια απόσταση μεταξύ δύο σημείων. Το παρακάτω σχήμα (σχήμα 6.1) αναπαριστά το παράδειγμα που περιγράψαμε προηγούμενα. Τα βήματα τα οποία ακολουθούνται είναι τα εξής: 1. Το νέο γεγονός γίνεται αντιληπτό από το αισθητήριο και στη συνέχεια από την οντότητα (στόχο) εξυπηρέτησης περιστατικών. 2. Η τελευταία δημιουργεί μια νέα οντότητα στην οποία αναθέτει

το περιστατικό. 3. Επιλογή κινητής μονάδας. Η οντότητα που δημιουργήθηκε στο 2 επικοινωνεί με την οντότητα επιλογής κινητής μονάδας, η οποία βάση της πληροφορίας που έχει για τις κινητές μονάδες και της επικοινωνίας που κάνει με την οντότητα εύρεσης συντομότερης διαδρομής (η οποία δημιουργεί μια οντότητα για να εξυπηρετήσει την αίτηση 3.2, 3.3) επιλέγει την κινητή μονάδα και επιστρέφει το αποτέλεσμα. 4. Επιλογή διαδρομής από τη θέση της κινητής μονάδας, στη θέση του περιστατικού (δημιουργία οντότητας εξυπηρέτησης της αίτησης 4.1) και αποστολή εντολής σε αυτή 4.2. 5. Επιλογή ιατρικού κέντρου προορισμού του ασθενή (5.1 οντότητα εξυπηρέτησης της αίτησης). 6. Επιλογή διαδρομής από τη θέση της κινητής μονάδας, από θέση του περιστατικού σε αυτή του προορισμού της και αποστολή της εντολής σε αυτή 6.2. 7. Ανακοίνωση από την κινητή μονάδα προς το σύστημα για εκπλήρωση της αποστολής



a) οντότητα εξυπηρέτησης περιστατικών b) διεργασία εξυπηρέτησης περιστατικού
c) οντότητα επιλογής κινητής μονάδας d) διεργασία επιλογής κινητής μονάδας
e) οντότητα εύρεσης διαδρομών f, g, h) διεργασίες εύρεσης διαδρομών
i) οντότητα επιλογής ιατρικού κέντρου j) διεργασία επιλογής ιατρικού κέντρου

Σχήμα 6.1 Λειτουργία της οντότητας εξυπηρέτησης επειγόντων περιστατικών.

6.3 Αντικρουόμενοι στόχοι και προτεραιότητες

Ένα ζήτημα στο οποίο δεν έχουμε αναφερθεί, είναι η περίπτωση κατά την οποία δύο στόχοι είναι δυνατόν να είναι αντικρουόμενοι όπως επίσης και αυτό του ποιοι στόχοι ενός συστήματος έχουν μεγαλύτερη προτεραιότητα από άλλους. Στη συνέχεια θα περιγράψουμε τον τρόπο με τον οποίο αντιμετωπίζουμε τα δύο αυτά προβλήματα, όπως επίσης και θα διαχωρίσουμε τις περιπτώσεις συστημάτων όπου οι επιλογές του σημαντικότερου στόχου μεταξύ δύο ή παραπάνω καθώς και η προτεραιότητα του κάθε στόχου, είναι γνωστή από την κατασκευή του συστήματος ή όχι.

Όσον αφορά την αντιπαλότητα δύο ή παραπάνω στόχων, οι οντότητες ακολουθούν το σχέδιο δράσης το οποίο τους έχει ανατεθεί και το οποίο υπαγορεύει την εκτέλεση κάποιας ενέργειας

(η οποία συμβάλει στην εκπλήρωση του στόχου), ή όχι. Για παράδειγμα στο σύστημα μας δύο ανταγωνιζόμενοι στόχοι, είναι η οντότητα εξυπηρέτησης περιστατικών ή οποία προσπαθεί να δεσμεύσει κινητές μονάδες για χρήση τους στη μετακομιδή των ασθενών στα ιατρικά κέντρα και η οντότητα καθορισμού περιπολιών η οποία υπαγορεύει τις κινητές μονάδες του συστήματος το να κάνουν περιπολίες. Η λύση του ανταγωνισμού γίνεται δίνοντας προτεραιότητα στην οντότητα εξυπηρέτησης περιστατικών μέσω της μεταβλητής κατάστασης του αντικειμένου που περιγράφει την κινητή μονάδα και δηλώνει το εάν η μονάδα είναι δεσμευμένη σε κάποιο περιστατικό ή όχι. Δηλαδή η αντιμετώπιση μας στο πρόβλημα των αντικρουόμενων στόχων είναι αυτή του ορισμού της επιθυμητής από το χρήστη ενέργειας. Άλλωστε στη συγκεκριμένη περίπτωση και λόγω της ζωτικής κρίσιμότητας (life critical) της εφαρμογής δεν θα μπορούσε να γίνει και αλλιώς τόσο για νομικούς λόγους όσο και κυρίως διότι δεν επιτρέπεται στο σύστημα μας να κάνει λανθασμένες επιλογές (φυσικά βέβαια θα μπορούσε να ισχυριστεί κάποιος πως η εύρεση μιας καλύτερης επιλογής, αυτόματα καθιστά την ήδη υπάρχουσα επιλογή λανθασμένη σε μια τέτοιου είδους εφαρμογή).

Όσον αφορά την προτεραιότητα του ρυθμού εκτέλεσης έναντι κάποιου άλλου το ζήτημα επιλύεται με την αντιστοίχιση ποσοστών υπολογιστικής ισχύος στα νήματα ελέγχου τα οποία υλοποιούν τους στόχους του συστήματος. Έτσι για παράδειγμα μεγάλη προτεραιότητα έχει η οντότητα εξυπηρέτησης επειγόντων περιστατικών έναντι της οντότητας εύρεσης περιοχών υψηλής συχνότητας εμφάνισης περιστατικών. Προκειμένου η πρώτη να αντιληφθεί γρήγορα το περιστατικό δίνει μεγάλη προτεραιότητα στην παρακολούθηση των αντίστοιχων αισθητηρίων, καθώς και στην εύρεση κινητής μονάδας, ιατρικού κέντρου, υπολογισμό διαδρομών κ.τ.λ. ακριβώς λόγω της επείγουσας φύσης της εργασίας. Από την άλλη πλευρά, η οντότητα η οποία παρατηρεί το ιστορικό των περιστατικών και βρίσκει της ζητούμενες περιοχές, δρα στο παρασκήνιο επιτρέποντας στον εαυτό της να εκτελεστεί μόνο σε στιγμές μικρού υπολογιστικού φόρτου.

Από τα προηγούμενα συμπεραίνει κανείς πως η εφαρμογή μας ταξινομείται κάτω από την κατηγορία των εφαρμογών στις οποίες η προτεραιότητα των στόχων μεταξύ τους, είναι προκαθορισμένη και η γνώση αυτή υπάρχει στο σύστημα από την εκκίνηση του, εμπεριεχόμενη στα σχέδια δράσης των οντοτήτων. Σε αυτή την εργασία δεν έχει γίνει συγκεκριμένη μελέτη για εύρεση τρόπου καθορισμού του σημαντικότερου στόχου και ο κύριος λόγος για αυτό είναι, όπως έχουμε αναφέρει ο χαρακτηρισμός των λαθών ως απαράδεκτη από το σύστημα συμπεριφορά. Ωστόσο, για να γίνει κάτι τέτοιο απαιτείται οπωσδήποτε είτε ανάδραση από το χρήστη, είτε ο ορισμός μέτρου απόδοσης τόσο του κάθε στόχου όσο και του συστήματος γενικότερα, σε συνδυασμό με τη χρήση κατάλληλου αλγόριθμου μηχανικής μάθησης.

6.4 Υλοποίηση

Κάθε στόχος του συστήματος υλοποιείται από μια οντότητα η οποία, μέσω του σχεδίου δράσης το οποίο περιγράφει τις επιθυμητές ενέργειες για την επίτευξη του. Κάθε μια από αυτές τις οντότητες εμπεριέχει τουλάχιστο ένα νήμα ελέγχου, ενώ είναι δυνατόν να δημιουργήσει και άλλες οντότητες. Τεχνικά η ύπαρξη του νήματος αυτού είναι που κάνει τις οντότητες αυτόνομες, αφού δεν υπάρχει κάποιο κεντρικό νήμα το οποίο να καθορίζει το ποιο τμήμα λογισμικού θα εκτελεστεί κάθε φορά. Επιπλέον με αυτό τον τρόπο γίνεται δυνατή η εύκολη προσθήκη νέων στόχων στο σύστημα μας. Απλά προσθέτουμε τη νέα οντότητα στόχων στο σύνολο των ήδη υπαρχόντων προσαρμόζοντας ίσως τα ήδη υπάρχοντα σχέδια δράσης.

Η υλοποίηση των προτεραιοτήτων των στόχων γίνεται αναθέτοντας ποσοστά υπολογιστικής ισχύος στα νήματα ελέγχου τα οποία υλοποιούν τον κάθε στόχο. Επιπλέον εάν κάποια από τις οντότητες αυτές δημιουργεί νέες, μεταβιβάζει προς αυτές την αντίστοιχη προτεραιότητα εκτέλεσης. Εάν θέλουμε να αναστείλουμε, λόγω μεγάλου υπολογιστικού φόρτου, την εκτέλεση κάποιου νήματος (το οποίο υλοποιεί κάποιον δευτερεύων στόχο), μειώνουμε την προτεραιότητα του σε μηδέν. Οι τιμές των προτεραιοτήτων δεν κυμαίνονται, όπως ίσως να περίμενε κανείς, σε αριθμητικές τιμές μεγάλης ακρίβειας. Αντίθετα, έχουν συμβολικές τιμές και συγκεκριμένα τις «παρασκηνακή», «κανονική» και «επείγουσα». Δύο είναι οι λόγοι για τους οποίους συμβαίνει αυτό. Ο πρώτος είναι η άγνοια μας για τις ακριβείς τιμές καθώς και ο ότι

τελικά δεν έχει και τόσο μεγάλη σημασία μιας και λόγω των μεγεθών της εφαρμογής η διαφορά είναι δύσκολα αντιλήψιμη. Ο κυριότερος λόγος όμως, είναι το ότι κατά αυτόν τον τρόπο είναι δυνατόν το σύστημα μας να κάνει εύκολα αντιστοιχίσεις ανάμεσα σε καταστάσεις και επιλογές (πχ. όταν η μεταβλητή που υποδεικνύει τον τρέχοντα φόρτο του συστήματος έχει την τιμή «υψηλός», τότε ανάθεσε την τιμή «παρασκευαστική» στην οντότητα εύρεσης περιοχών υψηλής συχνότητας ατυχημάτων). Καταλήγοντας, αναφέρουμε πως αυτός είναι και ο τρόπος ο οποίος υποδεικνύεται από τη θεωρία της μηχανικής μάθησης για την ποιοτική εκτίμηση αριθμητικών μεγεθών, ενώ μάλιστα υπάρχουν και σχετικές μέθοδοι για την αντιστοίχιση εύρους συνεχών αριθμητικών τιμών σε διακριτές λειτουργία η οποία ονομάζεται *διακριτοποίηση* [46].

7 Εφαρμογή στο προνοσοκομειακό σύστημα υγείας

Σε αυτή την ενότητα θα εστιάσουμε την προσοχή μας στις ιδιαίτερες ανάγκες ενός συστήματος επείγουσας προνοσοκομειακής ιατρικής περίθαλψης και θα δούμε την επιρροή που έχουν αυτές στο σχεδιασμό και υλοποίηση του συγκεκριμένου συστήματος. Θα παρουσιάσουμε επίσης τον τρόπο με τον οποίο θα οργανωθεί η εφαρμογή τόσο σε επίπεδο υλοποίησης όσο και στον τρόπο με τον οποίο θα έρχεται αυτή σε επικοινωνία με το χρήστη.

Η αρχιτεκτονική η οποία έχουμε παρουσιάσει μέχρι στιγμής, είναι αρκετά γενική και αφηρημένη, ενώ μπορεί να πει κανείς πως μπορεί να εφαρμοστεί σε μεγάλο αριθμό συστημάτων. Αυτό βέβαια δεν απέχει και σε μεγάλο βαθμό από την πραγματικότητα αφού όπως έχουμε δει η σχεδίαση συστήματος βασισμένου σε αρχιτεκτονική αυτόνομων πρακτόρων, μπορεί να χαρακτηριστεί και ως μεθοδολογία σχεδίασης λογισμικού. Θα αναφέρουμε τις απαιτήσεις μας από το σύστημα οι οποίες εξειδικεύουν την προαναφερθείσα αφηρημένη αρχιτεκτονική καθώς και άλλα ζητήματα τα οποία συμβάλουν στη σχεδίαση της αρχιτεκτονικής του εφαρμοσμένου συστήματος.

7.1 Χαρακτηριστικά συστήματος

- **Σύστημα πραγματικού χρόνου (Real time system)**

Όπως μπορεί κανείς να συμπεράνει και από το όνομα, ένα σύστημα επείγουσας προνοσοκομειακής ιατρικής περίθαλψης, είναι ένα *σύστημα πραγματικού χρόνου* [47]. Με τον όρο αυτό υπονοούμε όλα τα υπολογιστικά συστήματα τα οποία αλληλεπιδρούν με το εξωτερικό τους περιβάλλον με ρυθμούς απόκρισης οριζόμενους από αυτό. Έτσι, πρέπει να λάβουμε υπ' όψιν πως όποιες υπηρεσίες παρέχει το σύστημα προς τους χρήστες του, δεν πρέπει να έχουν τέτοιο χρόνο απόκρισης ώστε να καθυστερούν την ήδη υπάρχουσα διαδικασία αντιμετώπισης περιστατικών, ή οποία υπήρχε πριν την εφαρμογή του συστήματος, και αυτό λόγω της επείγουσας φύσης της εφαρμογής.

Αναλυτικότερα, όσον αφορά το χρονισμό του συστήματος, πρέπει να είναι τέτοιος ώστε να επιταχύνει τις επείγουσες διαδικασίες βελτιώνοντας έτσι την απόδοση του συστήματος. Μάλιστα ο τρόπος αντιμετώπισης πολλών ζητημάτων σχεδιασμού του συστήματος καθορίζεται σε μεγάλο βαθμό από το γεγονός του ότι η βελτιστοποίηση του χρόνου απόκρισης του συστήματος, πάντα βέβαια δίνοντας έξοδο καλύτερη ή έστω το ίδιο καλή, αποτελεί βασικό στόχο του σχεδιασμού του συστήματος μας.

Χαρακτηριστικό παράδειγμα της ανάγκης απόκρισης του συστήματος σε χρόνο τέτοιο ώστε πραγματικά να επιταχύνει τις διαδικασίες του συστήματος είναι η εύρεση συντομότερης διαδρομής από το σύστημα. Όπως θα δούμε και στην αντίστοιχη ενότητα λόγω του μεγάλου υπολογιστικού κόστους της εύρεσης της διαδρομής και προκειμένου το σύστημα να τηρεί τις προϋποθέσεις που αναφέραμε, η οντότητα αυτή προσπαθεί να ευρετηριάσει συγκεκριμένες γεωγραφικές περιοχές του χάρτη, ενώ ταυτόχρονα υπολογίζει τη διαδρομή σε τμήματα, από το πιο επείγον τμήμα, την αρχή της διαδρομής, προς το τελευταίο. Κάθε τμήμα της διαδρομής με το που υπολογίζεται αποστέλλεται στην κινητή μονάδα, ενώ συνεχίζεται ο υπολογισμός της υπόλοιπης διαδρομής καθώς η μονάδα βρίσκεται ήδη σε κίνηση.

- **Μάθηση στο παρασκήνιο**

Από την άλλη πλευρά υπάρχουν διαδικασίες του συστήματος οι οποίες θα θέλαμε να λειτουργούν στο *παρασκήνιο*. Με τον όρο αυτό θα υπονοούμε διεργασίες οι οποίες εκτελούνται χωρίς συνεχή αλληλεπίδραση με το χρήστη (απλά ο χρήστης λαμβάνει το αποτέλεσμα του υπολογισμού ή της υπηρεσίας), σε στιγμές επάρκειας υπολογιστικής ισχύος, ενώ συνήθως κρατάν μεγάλο χρονικό διάστημα σε σχέση με τις υπόλοιπες διεργασίες του συστήματος. Συγκεκριμένα ορισμένες διαδικασίες μάθησης παρατηρούν γεγονότα σε διάστημα μεταξύ πολλών ημερών, η ακόμα και καθ' όλη τη διάρκεια ζωής του συστήματος. Σε αυτή την περίπτωση δεν θα θέλαμε η εκτέλεση των αλγορίθμων μάθησης να γίνεται σε

στιγμές όπου υπάρχει μεγάλος υπολογιστικός φόρτος, λόγω επεξεργασίας επειγόντων αιτήσεων. Αντίθετα το σύστημα θα πρέπει να επιλέγει στιγμές όπου υπάρχει επάρκεια υπολογιστικής ισχύος για τέτοιες εργασίες.

Παραδείγματα τέτοιων διαδικασιών μάθησης, αποτελούν οι διεργασίες εύρεσης στατιστικών στοιχείων για το φόρτο των τμημάτων του οδικού δικτύου (πχ. μέσος όρος φόρτου για συγκεκριμένες χρονικές περιόδους) και ο εντοπισμός περιοχών υψηλής συχνότητας εμφάνισης περιστατικών.

- **Εστίαση της προσοχής**

Τα παραπάνω δύο γνωρίσματα του συστήματος μας οδηγούν στη μελέτη ενός από τα σημαντικότερα στοιχεία ενός ευφυούς πράκτορα, σε αυτό της δυνατότητας *εστίασης της προσοχής*. Συγκεκριμένα θα θέλαμε το σύστημα σε στιγμές όπου υπάρχει ανεπάρκεια υπολογιστικής ισχύος, προκειμένου να αποκριθεί σε πραγματικό χρόνο, να επιλέγει τις εργασίες οι οποίες επείγουν να τελειώσουν. Ταυτόχρονα, ορισμένες φορές θα θέλαμε το σύστημα να αντιλαμβάνεται την κρισιμότητα ορισμένων περιστατικών και όχι μόνο να δίνει περισσότερους υπολογιστικούς πόρους στη διεργασία εξυπηρέτησης του, αλλά και υλικούς πόρους του συστήματος που κανονικά δεν θα δίδονταν σε ένα τυπικό περιστατικό (πχ. η διάθεση μιας απασχολημένης από μη επείγον περιστατικό κινητής μονάδας), όποτε είναι απαραίτητο ή στην περίπτωση του συστήματος μας ορισμένο από το χρήστη.

Ο τρόπος με τον οποίο το σύστημα μας εστιάζει την προσοχή, τουλάχιστο από άποψη υπολογιστικού φόρτου, είναι μάλλον απλός με την έννοια του ότι γνωρίζει από την σχεδίαση του τις επείγουσες και κρίσιμες διεργασίες καθώς αυτές με μικρή προτεραιότητα εκτέλεσης. Απλά σε χρονικές στιγμές μεγάλου υπολογιστικού φόρτου αναστέλλεται η λειτουργία των με χαμηλή προτεραιότητα διεργασιών, ενώ αντίθετα σε στιγμές υπερεπάρκειας του, είναι δυνατόν να εκτελούνται με κανονική προτεραιότητα (θυμίζουμε πως τρεις είναι οι δυνατές προτεραιότητες στο σύστημα μας, ενώ υπάρχει και η μηδενική προτεραιότητα που ουσιαστικά σημαίνει αναστολή της διεργασίας). Η εστίαση της προσοχής σε ποιοτικό επίπεδο όσον αφορά την ποιότητα της προσφερόμενης υπηρεσίας από το σύστημα, όσον αφορά την εξυπηρέτηση ενός περιστατικού καθορίζεται ουσιαστικά από τα σχέδια δράσης της κάθε ομάδας και έχει σχέση με την πολιτική την οποία ακολουθεί ο συγκεκριμένος οργανισμός χρήσης του συστήματος.

Θα θέλαμε να σχολιάσουμε εδώ τη δυσκολία εστίασης της προσοχής του συστήματος σε περίπτωση μη ορισμού των επειγόντων διεργασιών από τον προγραμματιστή. Σε αυτή την περίπτωση το σύστημα θα έπρεπε να δέχεται οδηγίες ή ανάδραση από το χρήστη έως ότου μάθει τις επείγουσες διαδικασίες. Δυσκολότερη θα ήταν ακόμη η ποιοτική εύρεση των απαραίτητων ενεργειών για βελτίωση της ποιότητας της παρεχόμενης υπηρεσίας όπου θα έπρεπε να γίνουν τα κατάλληλα πειράματα τουλάχιστο στον προσομοιωτή.

- **Αντικρουόμενοι στόχοι**

Σε συστήματα αυτόνομων οντοτήτων τα οποία υλοποιούν πολλαπλούς στόχους, είναι δυνατόν να υπάρξουν αντικρουόμενοι στόχοι, πράγμα το οποίο συνήθως μεταφράζεται στο ότι το σύστημα έχει παραπάνω από μια δυνατές ενέργειες προς εκτέλεση, η εκτέλεση της μιας όμως απαγορεύει αυτή των υπολοίπων. Το πρόβλημα μεταφράζεται σε αυτό της επιλογής κατάλληλης ενέργειας το οποίο είχαμε δει στο πρώτο μέρος της εργασίας αυτής. Επιπλέον στο πεδίο της διαχείρισης πόρων, είναι δυνατόν δύο στόχοι να αιτούν τον ίδιο πόρο, ενώ αυτός να είναι δυνατόν να διατεθεί μόνο σε ένα από αυτούς.

Προφανώς για τη λύση του προβλήματος, είτε πρέπει να υπάρξει κάποια συμβιβαστική λύση μεταξύ των δύο, είτε να δώσουμε προτεραιότητα σε κάποιον από τους στόχους. Στο σύστημα μας συμβαίνει το τελευταίο, διότι οι πιο σημαντικές συνθήκες ανταγωνισμού μεταξύ στόχων, προκύπτουν όταν δύο από αυτούς ζητάνε τον ίδιο πόρο. Η λύση στο πρόβλημα αυτό δίνεται μέσω των προτεραιοτήτων του τρόπου επικοινωνίας των οντοτήτων του συστήματος. Αναλυτικότερα οι οντότητες οι οποίες υλοποιούν τους στόχους του συστήματος, έχουν από τη σχεδίαση τους, τη γνώση για το εάν κάποιος στόχος έχει μεγαλύτερη προτεραιότητα και υποχωρούν στην αίτηση του πόρου ή την εκτέλεση ενέργειας. Για παράδειγμα, το σύστημα

δεν θα στείλει κάποια κινητή μονάδα σε περιπολία, προκειμένου να ικανοποιήσει το στόχο της για περιπολίες μονάδων σε ορισμένες περιοχές, ενώ εκκρεμούν περιστατικά τα οποία δεν εξυπηρετούνται.

Ουσιαστικά η λύση και σε αυτή την περίπτωση δίνεται από το χρήστη ή σχεδιαστή του συστήματος.

- **Έλεγχος δράσης και χειρισμός δυσλειτουργιών**

Επειδή το σύστημα μας δεν έχει τα αντικείμενα δράσης του υπό πλήρη έλεγχο, αλλά ουσιαστικά δίνει εντολές προς αυτά προκειμένου να τις πραγματοποιήσουν, τα αντικείμενα αυτά ελέγχονται, μέσω των αισθητηρίων, για το εάν κατάφεραν να εκτελέσουν την εντολή η οποία τους ανατέθηκε ή όχι. Ανάλογα με το εάν το αντικείμενο πέτυχε ή όχι την εκτέλεση της εντολής η οποία του ανατέθηκε, η οντότητα ή οποία έχει το αντικείμενο υπό τον έλεγχο της, συνεχίζει να του δίνει της κατάλληλες εντολές. Γενικά σε περίπτωση σφαλμάτων τα σχέδια δράσης των οντοτήτων εμπεριέχουν εντολές για αντιμετώπιση της κατάστασης.

Αυτό που είναι ενδιαφέρον να παρατηρήσει κανείς, είναι η αντιμετώπιση και των αντικειμένων δράσης ως αυτόνομες οντότητες, όπως και των υπόλοιπων οντοτήτων του συστήματος. Δεδομένου ότι ο χειρισμός των λαθών και της δυσλειτουργίας και των υπόλοιπων τμημάτων του συστήματος, γίνεται με τον ίδιο τρόπο (δηλαδή ακολουθώντας εντολές ειδικά για κάθε είδους λάθος οι οποίες βρίσκονται στα σχέδια δράσης), έχουμε τον ίδιο τρόπο αντιμετώπισης λαθών για όλες τις οντότητες, πράγμα το οποίο συμβάλει στον ομοιόμορφο σχεδιασμό μεταξύ των οντοτήτων. Το τελευταίο είναι αρκετά σημαντικό, διότι συμβάλει στην ομοιόμορφη σχεδίαση ενός πλαισίου (template) για πρότυπο πράκτορα τον οποίο θα μπορούμε να εξειδικεύουμε κάθε φορά ανάλογα με την εργασία την οποία πρόκειται να του αναθέσουμε.

7.2 Ορισμός στόχων του συστήματος

Κυριότερος στόχος του συστήματος είναι η καλύτερη εξυπηρέτηση των επειγόντων περιστατικών και η βέλτιστη διαχείριση των πόρων του, είτε απευθείας στο περιβάλλον είτε δρώντας συμβουλευτικά ή επικουρικά προς το προσωπικό του προνοσοκομειακού συστήματος επείγουσας ιατρικής περίθαλψης. Οι δύο αυτοί γενικοί στόχοι είναι δυνατόν να αποσυντεθούν σε άλλους ειδικότερους. Μάλιστα, μπορεί να διαφωνήσει κανείς για το ποιοι είναι οι ακριβώς οι επί μέρους στόχοι οι οποίοι συνδέουν τους γενικότερους. Δευτερεύοντες στόχους αποτελούν η εκπαίδευση του ιατρικού προσωπικού καθώς και η δοκιμή νέων πολιτικών ή ιατρικών πρωτοκόλλων.

Όσον αφορά την βελτίωση ποιότητας υπηρεσίας των εισερχομένων προς το σύστημα περιστατικών, το σύστημα προσπαθεί να μειώσει το χρόνο απόκρισης του, ενώ παράλληλα προσπαθεί να αντιστοιχίσει τους καταλληλότερους πόρους σε κάθε περιστατικό, κάνοντας ταυτόχρονα οικονομία πόρων, έτσι ώστε να αυξήσει τη διαθεσιμότητα πόρων για νέα περιστατικά.

Η μείωση του χρόνου απόκρισης του συστήματος εξαρτάται από τον υπολογισμό συντομότερης διαδρομής για τις κινητές μονάδες καθώς και από την εύρεση της πλησιέστερης κινητής μονάδας και ιατρικού κέντρου τα οποία θα εξυπηρετήσουν το περιστατικό. Επίσης το σύστημα φροντίζει για την ανάθεση περιπολιών στις περιοχές του περιβάλλοντος οι οποίες έχουν μεγαλύτερη πιθανότητα εμφάνισης περιστατικού στο εγγύτερο μέλλον. Έμμεσα η λειτουργία αυτή ανήκει σε αυτές οι οποίες βελτιστοποιούν το χρόνο απόκρισης του συστήματος.

Για την εκπαίδευση του ιατρικού προσωπικού ή δοκιμή νέων πολιτικών από το σύστημα το σύστημα είναι δυνατόν να λειτουργήσει με εναλλακτικά σχέδια δράσης. Για την εκπλήρωση αυτού του στόχου γίνεται χρήση ενός προσομοιωτή του περιβάλλοντος, ο οποίος προσπαθεί να προσομοιώσει τη συμπεριφορά των πόρων του συστήματος.

7.3 Κατασκευή μοντέλου περιβάλλοντος

Εκτός από τα αισθητήρια τα οποία παρατηρούν το περιβάλλον κατά την εκτέλεση της εφαρμογής, θα θέλαμε το σύστημα να μπορεί αυτόματα να προσαρμόζει τις δομές και εσωτερική κατάσταση του, σε κάθε νέο περιβάλλον στο οποίο προσαρμόζεται. Έτσι έχει αυτοματοποιηθεί η διαδικασία ανάγνωσης των πόρων του συστήματος από τη βάση δεδομένων καθώς και η διαδικασία κατασκευής του γράφου οδικού δικτύου από το αρχείο τεχνικού σχεδίου το οποίο περιγράφει το οδικό δίκτυο. Στην πρώτη περίπτωση το σύστημα προσπελαίνει τη βάση και βρίσκει τις οντότητες οι οποίες αποτελούν τους διαθέσιμους πόρους του συστήματος (κινητές μονάδες, νοσοκομεία, κέντρα υγείας κ.α.). Σε συνδυασμό με την κατασκευή του γράφου του οδικού δικτύου από αρχείο τεχνικού σχεδίου τύπου DXF [55], το σύστημα κατά την εγκατάσταση του είναι δυνατόν να κατασκευάζει αυτόματα το μοντέλο το οποίο χρειάζεται για τη διαχείριση των πόρων του συστήματος. Από την άλλη πλευρά σε περίπτωση εξομοίωσης ο προσομοιωτής του συστήματος κατά τη στιγμή αρχικοποίησης προσπελαίνει αρχείο το οποίο περιγράφει το περιβάλλον προσομοίωσης (αναλυτικότερη αναφορά σε αυτό γίνεται σε επόμενη ενότητα).

Στην πραγματικότητα, το σύστημα δεν κατασκευάζει ένα μοντέλο το οποίο προσομοιώνει το σύστημα. Όπως αναφέρεται και στη βιβλιογραφία, το ίδιο το περιβάλλον είναι το καλύτερο μοντέλο του εαυτού του [11]. Έτσι ναί μεν χρησιμοποιούμε δομές δεδομένων οι οποίες περιγράφουν το περιβάλλον, όμως αφενός οι δομές αυτές δεν είναι ολοκληρωμένες σε ένα ενιαίο μοντέλο, αφετέρου αυτές δημιουργούνται κυρίως από τα αισθητήρια του συστήματος. Αυτό που συμβαίνει είναι πως κάθε οντότητα έχει την πληροφορία για το περιβάλλον η οποία απαιτείται για να λειτουργήσει και η οποία προέρχεται από τα αισθητήρια όργανα του συστήματος. Σε περιπτώσεις κατά τις οποίες παραπάνω από μια οντότητα χρησιμοποιούν την ίδια πληροφορία (όπως για παράδειγμα η βάση δεδομένων των πόρων του συστήματος), τότε υπάρχει ένα αισθητήριο από το οποίο αντλούν οι οντότητες την πληροφορία αυτή.

8 Παρουσίαση οντοτήτων

8.1 Αισθητήρες

Η ενότητα αυτή περιγράφει το σχεδιασμό και λειτουργία των διάφορων αισθητήρων του συστήματος. Οι αισθητήρες του τμήματος διαχείρισης πόρων είναι το μέσο με το οποίο το τμήμα αυτό αντιλαμβάνεται τα γεγονότα που λαμβάνουν χώρα στο υπό διαχείριση σύστημα, ενώ είναι και αυτοί σχεδιασμένοι ακολουθώντας τη μεθοδολογία των αυτόνομων και συνεργαζόμενων οντοτήτων. Η βασική λειτουργία του αισθητήριου αντικειμένου είναι η συλλογή μηνυμάτων (ή γεγονότων) από το περιβάλλον της οντότητας και η εναπόθεση με ταξινομημένο τρόπο, σε δομή στην οποία έχουν πρόσβαση οι υπόλοιπες οντότητες του συστήματος. Η δομή αυτή αποτελεί μέλος της οντότητας προσβάσιμο από οποιαδήποτε άλλη οντότητα του συστήματος. Έτσι ως προς της άλλες οντότητες η οντότητα, η οποία διαχειρίζεται τη δομή αυτή, αποτελεί στην ουσία μια ενεργή δομή δεδομένων, με την έννοια του ότι φροντίζει ή ίδια για την ανανέωση, διάταξη και διαχείριση του περιεχομένου της. Μια τέτοια αντιμετώπιση συμβάλλει στον τρόπο σχεδίασης ο οποίος αποσκοπεί στην κατανομή του ελέγχου σε πολλές οντότητες καθώς και στον αφαιρετικό τρόπο σχεδίασης (programming abstraction), κατά τον οποίο αναθέτουμε μια εργασία εξολοκλήρου σε μια εξειδικευμένη προς αυτή οντότητα. Στη συνέχεια θα περιγράψουμε τις γενικές αρχές λειτουργίας των αισθητήρων, το σχεδιασμό τους, ενώ τέλος θα αναφερθούμε σε συγκεκριμένα παραδείγματα αισθητήρων τους οποίους χρησιμοποιεί το σύστημα μας.

8.1.1 Γενική περιγραφή

Οι αισθητήρες μπορούν να περιγραφούν ως στοιχειώδεις οντότητες, οι οποίες συλλέγουν γεγονότα τα οποία αντιλαμβάνονται και καθιστούν την πληροφορία τους διαθέσιμη προς τις υπόλοιπες οντότητες του γνωσιακού επιπέδου. Τα γεγονότα εναποτίθενται σε κατάλληλες δομές δεδομένων, έτσι ώστε να είναι στη διάθεση των οντοτήτων του γνωσιακού επιπέδου, οι οποίες ελέγχουν τις δομές αυτές περιοδικά και αντιλαμβάνονται την εκδήλωση ενός νέου γεγονότος, από τις αλλαγές σε αυτές. Η αντίληψη ενός νέου γεγονότος γίνεται σε χρόνο μικρότερο ή το πολύ ίσο με αυτό της περιόδου ελέγχου του καταχωρητή γεγονότων ο οποίος προσπελαύνεται. Επιπρόσθετα οι δομές αυτές πρέπει να έχουν πεπερασμένο μέγεθος και να καταστρέφουν (ή τουλάχιστο να κρατάνε ιστορικό) των παλιότερων γεγονότων, διότι στην αντίθετη περίπτωση μετά από αρκετή ώρα λειτουργίας του συστήματος και μόνο η φύλαξη παλαιότερων γεγονότων στους καταχωρητές θα εξαντλούσε τη μνήμη του συστήματος.

Με κατάλληλο προγραμματισμό οι αισθητήρες είναι δυνατόν να αντιδρούν σε συγκεκριμένα γεγονότα (ή συγκεκριμένους τύπους γεγονότων) παρακάμπτοντας τη διαδικασία συλλογής γεγονότων από άλλες οντότητες και ειδοποιώντας αυτές κατευθείαν. Η προαιρετική λειτουργία αυτή για το σύστημα, είναι δυνατόν να παρομοιαστεί με τη λειτουργία των ανακλαστικών αντιδράσεων ενός συστήματος, δηλαδή ενεργειών οι οποίες έχουμε ορίσει να γίνονται από το σύστημα χωρίς ενασχόληση του γνωσιακού επιπέδου. Οι ενέργειες αυτές είναι ορισμένες από το χρήστη και συγκεκριμένα στο περιβάλλον επείγουσας προνοσοκομειακής ιατρικής περίθαλψης, αντιστοιχούν σε ενέργειες ορισμένες από το ιατρικό πρωτόκολλο ή πολιτική αντιμετώπισης περιστατικών του εκάστοτε οργανισμού.

Όπως και από τις υπόλοιπες οντότητες του συστήματος, έτσι και με τους αισθητήρες, απαιτούμε να είναι σε θέση να προσαρμόσουν τη συμπεριφορά τους, ανάλογα με τις ανάγκες του συστήματος. Μάλιστα θα θέλαμε, η μεταβολή της συμπεριφοράς τους να είναι δυνατόν να υπαγορευτεί από ανώτερες γνωσιακές οντότητες, στο πλαίσιο εφαρμογής κάποιας γενικότερης πολιτικής του συστήματος. Επιπλέον η δυνατότητα μεταβολής της συμπεριφοράς των αισθητήρων, μας δίνει τη δυνατότητα εστίασης της προσοχής σε συγκεκριμένα γεγονότα ή διεργασίες, καθώς και τον αποτελεσματικό χειρισμό λαθών ή απρόβλεπτων καταστάσεων. Γενικά πάντως η απαίτηση μας από τους αισθητήρες είναι η υψηλή συχνότητα ελέγχου του περιβάλλοντος, με σκοπό την άμεση αντίληψη των τεκταινόμενων στο περιβάλλον και η αποδοτική διαχείριση των καταχωρητών γεγονότων.

8.1.2 Σχεδιασμός

Όπως και στο σχεδιασμό όλων των οντοτήτων του συστήματος, υπάρχει και στο σχεδιασμό των αισθητήρων ή πρόθεση ύπαρξης αυτονομίας και κατανομής του ελέγχου συμπεριφοράς στις επιμέρους οντότητες του συστήματος, αποφεύγοντας τη χρήση κάποιας συντονιστικής οντότητας, αλλά αφήνοντας το συντονισμό υλοποιηθεί, μέσω του δικτύου επικοινωνίας των οντοτήτων. Έτσι οι αισθητήρες, αποτελούνται από αυτόνομες και απλές οντότητες, των οποίων η αρχιτεκτονική ακολουθεί τις βασικές αρχές σχεδιασμού αυτόνομων οντοτήτων που έχουμε θέσει πρωθύστερα στην εργασία. Πιο ειδικά η οντότητα οι οποίες υλοποιούν τους αισθητήρες, αποτελούνται από τους καταχωρητές γεγονότων, από ένα γνωσιακό τμήμα το οποίο διαχειρίζεται και οργανώνει την πληροφορία στους καταχωρητές, και από το τμήμα λογισμικού το οποίο φροντίζει για την επικοινωνία με το γνωσιακό επίπεδο.

Η κυριότερη λειτουργία των αισθητήρων είναι η συλλογή γεγονότων, η δυνατότητα οργάνωσης της πληροφορίας αυτής ώστε να είναι διαθέσιμη από άλλες οντότητες καθώς και η μεταφορά του μηνύματος της εκδήλωσης ενός γεγονότος από ενδιάμεσες οντότητες. Η ανάγκη για εξυπηρετική προς τις άλλες οντότητες λειτουργία, είναι η γρήγορη αντίληψη των γεγονότων από τις άλλες οντότητες, η συμβάδιση του ρυθμού λειτουργίας των αισθητήρων, σύμφωνα με τον τρέχοντα κάθε φορά υπολογιστικό φόρτο του συστήματος καθώς και η διαχείριση του περιεχομένου των καταχωρητών γεγονότων, δεδομένου του πεπερασμένου της χωρητικότητας τους.

Η αποχώρηση των γεγονότων από τους καταχωρητές είναι δυνατόν να γίνει χρησιμοποιώντας κάποιον κατάλληλο αλγόριθμο (π.χ. ουρά FIFO [48], ή αλγόριθμος γήρανσης [42]), ωστόσο για βελτιστοποίηση της απόδοσης του συστήματος, μια οντότητα είναι δυνατόν να ζητήσει την αποχώρηση ενός γεγονότος από τους καταχωρητές σε περίπτωση όπου είναι ορισμένο, πως είναι η μοναδική οντότητα η οποία θα χρησιμοποιήσει το γεγονός (πχ. η οντότητα η οποία φροντίζει για την παρακολούθηση της βάσης δεδομένων του προνοσοκομειακού συστήματος επείγουσας ιατρικής περίθαλψης).

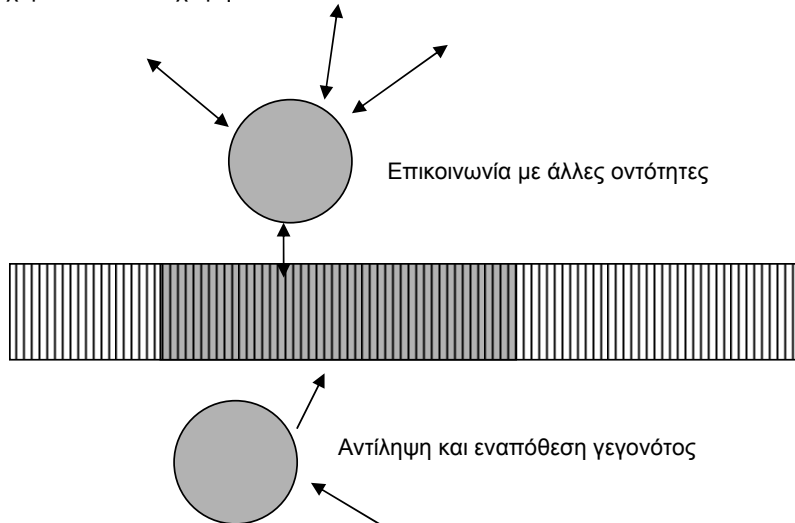
8.1.3 Υλοποίηση

Η γενική σχεδίαση των αισθητήρων [49] μας υπαγορεύει πως η αντίληψη ενός γεγονότος, ή μετατροπή του σε κατάλληλη δομή δεδομένων και ή εναπόθεση του σε καταχωρητή γεγονότων ξεκινά με την αποδοχή ενός μηνύματος από τον αισθητήρα ή στον περιοδικό έλεγχο μίας ή παραπάνω μεταβλητών του συστήματος. Και στις δύο περιπτώσεις η διαφοροποίηση έγκειται μόνο στο μηχανισμό αντίληψης και δεν προκαλεί αλλαγές στην υπόλοιπη αρχιτεκτονική του συστήματος, μιας και οι αισθητήρες παραμένουν αυτόνομες οντότητες ως προς το γνωσιακό επίπεδο το οποίο ελέγχει τους καταχωρητές γεγονότων με περιοδικό ρυθμό.

Η αντίληψη των γεγονότων από τους αισθητήρες επιτυγχάνεται με τη χρήση πολυπρογραμματιστικών νημάτων ελέγχου (threads), μιας και παραπάνω από ένα γεγονότα είναι δυνατόν να συμβούν ταυτόχρονα. Τα όποια προβλήματα είναι δυνατόν να προκύψουν από την διαχείριση κοινής μνήμης από παραπάνω από ένα νήμα ελέγχου (συνθήκες ανταγωνισμού, ταυτόχρονη εγγραφή στην ίδια θέση μνήμης από δύο νήματα ελέγχου κ.τ.λ.), αντιμετωπίζονται με αλγόριθμους σειριοποίησης διεργασιών και κλειδώματος θέσεων μνήμης [42].

Το παρακάτω σχήμα (σχήμα 8.1) εκφράζει σχηματικά την υλοποίηση και λειτουργία ενός τυπικού αισθητήρα. Οι καταχωρητές γεγονότων με τη βοήθεια νημάτων ελέγχου, περιέχουν γεγονότα ταξινομημένα με κατάλληλη διάταξη. Ο αισθητήρας εμπεριέχει ένα απλό γνωσιακό κομμάτι το οποίο ρυθμίζει την παραμονή των γεγονότων στους καταχωρητές, ενώ προαιρετικά είναι δυνατόν να παρακολουθεί τους καταχωρητές για την αντίληψη συγκεκριμένων γεγονότων ή ακολουθιών από γεγονότα και την προώθηση μηνύματος προς ανώτερη γνωσιακή οντότητα του συστήματος. Τέλος, μέρος του αισθητήρα αποτελεί το τμήμα επικοινωνίας με άλλες οντότητες, το οποίο χρησιμοποιώντας πάλι νήματα ελέγχου, είναι

δυνατόν εξυπηρετεί παραπάνω από μια οντότητες οι οποίες θέλουν να προσπελάσουν το περιεχόμενο των καταχωρήτων.



Σχήμα 8.1 Δομή και λειτουργία ενός τυπικού αισθητήρα. Μια υπό οντότητα (της οντότητας του αισθητήρα) είναι υπεύθυνη για την πρόσληψη και ταξινομημένη εναπόθεση των γεγονότων στον καταχωρητή. Μια άλλη επικοινωνεί με τις άλλες οντότητες εξυπηρετώντας αιτήσεις.

Το σύνολο των αισθητήρων του συστήματος, αποτελείται από τις εξής αισθητήριες οντότητες: αισθητήριο παρακολούθησης θέσεων κινητών μονάδων, αισθητήριο αντίληψης και παρακολούθησης εξέλιξης περιστατικών και αισθητήριο παρακολούθησης της βάσης δεδομένων του συστήματος διαχείρισης πόρων. Καταλήγοντας, πρέπει να παρατηρήσει κανείς πως τα προαναφερθέντα αισθητήρια, δεν είναι τα μόνα αισθητήρια τα οποία υπάρχουν στο σύστημα, αφού κάθε οντότητα του συστήματος, έχει κάποιο είδος αισθητήριου τόσο για την επικοινωνία με άλλες οντότητες, όσο και για την παρατήρηση της λειτουργίας κάποιας άλλης οντότητας (πχ. μια οντότητα μπορεί να παρατηρεί τη λειτουργία κάποιας άλλης η οποία επιτελεί έναν υπολογισμό, έτσι ώστε να χρησιμοποιήσει το αποτέλεσμα του υπολογισμού σε άλλη διεργασία). Ωστόσο οποιοδήποτε άλλο αισθητήριο υπάρχει στο σύστημα ακολουθεί παρόμοιο σχεδιασμό. Τα αισθητήρια στα οποία αναφερθήκαμε παραπάνω και θα περιγράψουμε με μεγαλύτερη λεπτομέρεια στη συνέχεια, αποτελούν το αισθητήριο σύστημα παρακολούθησης του περιβάλλοντος επείγουσας προνοσοκομειακής ιατρικής περίθαλψης, από το σύστημα διαχείρισης των πόρων του.

8.1.4 Αισθητήριο αντίληψης και παρακολούθησης εξέλιξης περιστατικών

Στόχος του αισθητήριου αυτού είναι η αντίληψη από το σύστημα της εκδήλωσης ενός περιστατικού καθώς και η αντίληψη επί μέρους συμβάντων τα οποία είναι δυνατόν να συμβούν κατά την εξέλιξη του περιστατικού.

8.1.4.1 Σχεδιασμός

Η πηγή των ερεθισμάτων για την αντίληψη του περιστατικού, είναι η εφαρμογή του ασυρματιστή του συστήματος διαχείρισης πόρων ή σε περίπτωση εξομοίωσης ο ίδιος ο εξομοιωτής. Ωστόσο για την παρακολούθηση της εξέλιξης των περιστατικών, το αισθητήριο παρακολουθεί, προσπελαύνοντας περιοδικά, μεταβλητές του συστήματος όπως τα βιοσήματα του ασθενή. Σε περίπτωση που κάποια μεταβλητή πάρει κάποια συγκεκριμένη τιμή ή περάσει κάποιο όριο, είναι δυνατόν να δημιουργηθεί κάποιο γεγονός, το οποίο και θα εναποτεθεί στον καταχωρητή γεγονότων του αισθητήριου. Ένα τέτοιο γεγονός είναι δυνατόν να δημιουργηθεί και από συνδυασμό μεταβολής παραπάνω από μιας μεταβλητής, όπου οι συνθήκες υπό τις

οποίες θα συμβεί κάτι τέτοιο, (είτε στην περίπτωση που το γεγονός προέρχεται από την μεταβολή μίας είτε περισσότερων μεταβλητών) ορίζονται από το χρήστη του συστήματος. Γενικά τέτοιου είδους γεγονότα θα χαρακτηρίζονται ως ιατρικά γεγονότα.

Η χρησιμότητα της δημιουργίας γεγονότων από μεταβολές μεταβλητών είναι η πρόβλεψη κρίσιμων για τον ασθενή καταστάσεων και η προληπτική ειδοποίηση του κατάλληλου ιατρικού περιστατικού ή η λήψη απόφασης για την εκκίνηση τηλεσυνδιάσκεψης, ή άλλης επικοινωνίας, μεταξύ του προσωπικού της κινητής μονάδας και άλλου ιατρικού προσωπικού. Ο χειρισμός τέτοιων γεγονότων γίνεται από εξειδικευμένη οντότητα του γνωσιακού επιπέδου (βλ. παρουσίαση γνωσιακού επιπέδου).

Μια άλλη παράμετρος την οποία λάβαμε υπ' όψιν κατά το σχεδιασμό του συγκεκριμένου αισθητήριου, είναι η κρισιμότητα των γεγονότων τα οποία αντιλαμβάνεται (δεν είναι αποδεκτό σε καμία περίπτωση το σύστημα να αγνοήσει την εκδήλωση ενός νέου επεισοδίου, λόγω υπολογιστικού φόρτου). Έτσι οι καταχωρητές γεγονότων έχουν αρκετά μεγάλο μέγεθος, ώστε να ακόμα και σε ακραίες καταστάσεις υπολογιστικού φόρτου του συστήματος, να προλαβαίνουν τα γεγονότα να γίνονται αντιληπτά από το γνωσιακό επίπεδο.

8.1.4.2 Υλοποίηση

Η υλοποίηση του προκειμένου αισθητήριου, δεν διαφέρει καθόλου από τον σχεδιασμό ενός τυπικού αισθητήριου. Δηλαδή, υπάρχει ο καταχωρητής γεγονότων και το γνωσιακό τμήμα του καταχωρητή το οποίο φροντίζει την είσοδο και έξοδο των γεγονότων με ουρά προτεραιότητας. Το αισθητήριο είναι δυνατόν να επικοινωνεί με παραπάνω από μια οντότητα ταυτόχρονα για να δώσει την απαραίτητη πληροφορία και αυτό επιτυγχάνεται με χρήση δυναμικά παραγόμενων νημάτων ελέγχου.

Αναλυτικότερα κάθε φορά που ξεκινά η εξυπηρέτηση μιας αίτησης, ένα ξεχωριστό νήμα ελέγχου αναλαμβάνει την εξυπηρέτηση της, έτσι ώστε να είναι δυνατή η εξυπηρέτηση πολλών αιτήσεων ταυτόχρονα. Δεδομένου ότι όλες οι αιτήσεις απαιτούν μόνο ανάγνωση του καταχωρητή γεγονότων, δεν υπάρχει περιορισμός συγχρονισμού των νημάτων αυτών. Ο μοναδικός περιορισμός που προκύπτει είναι στην περίπτωση της εγγραφής ενός νέου γεγονότος στον καταχωρητή, ταυτόχρονα με την ανάγνωση του καταχωρητή κατά την εξυπηρέτηση κάποιας αίτησης. Σε αυτή την περίπτωση αναστέλλεται η ανάγνωση για το ελάχιστο χρονικό διάστημα κατά το οποίο επιτελείται η εγγραφή και ακολουθεί η ανάγνωση, έτσι ώστε να λάβει υπ' όψιν της και το νέο γεγονός.

8.1.5 Αισθητήριο παρακολούθησης κινητών μονάδων

Στόχος της οντότητας λογισμικού αυτής, η οποία μπορεί να ταξινομηθεί κάτω από την κατηγορία των αισθητήριων του συστήματος μας, είναι να καταστήσει διαθέσιμη την πληροφορία των θέσεων των κινητών μονάδων του συστήματος επείγουσας προνοσοκομειακής ιατρικής περίθαλψης, σε οποιαδήποτε εφαρμογή το απαιτεί, ενώ ταυτόχρονα να απαλλάξει τους προγραμματιστές των εφαρμογών αυτών από την ανάπτυξη της επικοινωνίας με τις κινητές μονάδες. Για την επίλυση του προβλήματος λάβαμε υπ' όψιν τις ειδικές συνθήκες (χρήση κινητών τηλεφώνων και αβέβαιες δικτυακές συνδέσεις). Στη συνέχεια της παραγράφου περιγράφεται η αρχιτεκτονική και λειτουργία του εξυπηρετητή, τα πλεονεκτήματα του τρόπου επίλυσης του προβλήματος, καθώς και η σύνδεση με την υπόλοιπη αρχιτεκτονική του συστήματος.

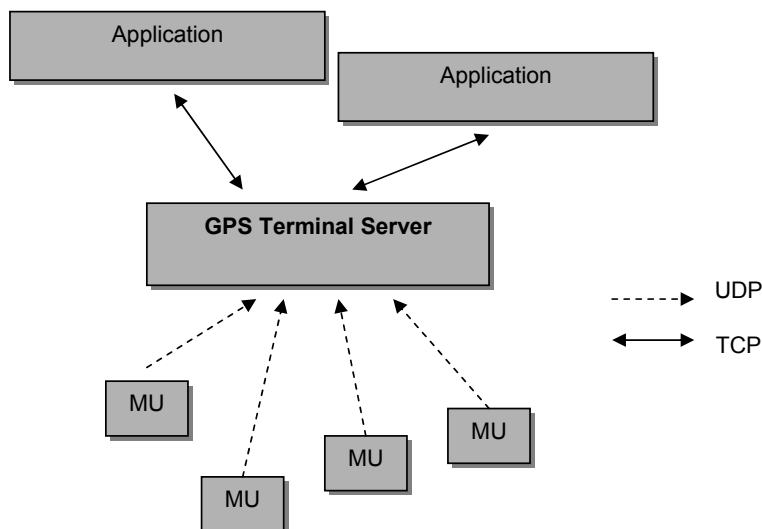
8.1.5.1 Σχεδιασμός - Αρχιτεκτονική

Το σύστημα του εξυπηρετητή αποτελείται από μια οντότητα λογισμικού η οποία λειτουργεί σε έναν υπολογιστή στον οποίο οι κινητές μονάδες αποστέλλουν τα δεδομένα της θέσης τους και από μια βιβλιοθήκη η οποία υλοποιεί την επικοινωνία μεταξύ των εφαρμογών και του εξυπηρετητή. Οι κινητές μονάδες αποστέλλουν τα δεδομένα της θέσης τους στον εξυπηρετητή, ο οποίος κρατά πάντα την τελευταία θέση της κινητής μονάδας στη μνήμη καθώς και την ώρα την οποία πραγματοποιήθηκε η μετάδοση. Υπάρχουν δύο λόγοι για τους οποίους ο εξυπηρετητής κρατά την ώρα μετάδοσης: η αβεβαιότητα των δικτυακών

συνδέσεων και το κόστος μετάδοσης του μηνύματος από κινητό τηλέφωνο. Μάλιστα όπως θα δούμε στη συνέχεια η αβεβαιότητα επιτυχούς αποστολής του μηνύματος μας αναγκάζει να χρησιμοποιήσουμε ειδικό τρόπο σύνδεσης για την αποστολή των δεδομένων από την κινητή μονάδα στον εξυπηρετητή.

Όσον αφορά την χρήση δικτύου κινητής τηλεφωνίας, χρειαζόμαστε την ώρα μετάδοσης του μηνύματος έτσι ώστε να γνωρίζουμε το εάν είναι έγκυρη η πληροφορία την οποία κατέχει ο εξυπηρετητής για τη θέση της κινητής μονάδας. Δεδομένου ότι οι κινητές μονάδες στέλνουν μηνύματα με σταθερό ρυθμό, εάν η διαφορά της τρέχουσας ώρας με την ώρα μετάδοσης του μηνύματος, είναι μεγαλύτερη από τη συχνότητα αποστολής μηνυμάτων, τότε έχουμε λόγο να υποθέσουμε πως η πληροφορία που υπάρχει στον εξυπηρετητή δεν είναι ακριβής. Από την άλλη πλευρά, λόγω του κόστους μετάδοσης είναι δυνατόν η κινητή μονάδα να μην στέλνει μηνύματα όταν παραμένει ακίνητη. Σε αυτή την περίπτωση και δεδομένου ότι, η εντολή προς αποστολή μιας κινητής μονάδας καταγράφεται στη βάση δεδομένων, μπορούμε να συμπεράνουμε εάν η κινητή μονάδα παραμένει στάσιμη, ή εάν βρίσκεται εκτός περιοχής κάλυψης από το δίκτυο κινητής τηλεφωνίας.

Κάθε εφαρμογή η οποία χρησιμοποιεί τα δεδομένα της θέσης των κινητών μονάδων και η οποία στη γενικότερη περίπτωση τρέχει σε διαφορετικό υπολογιστή από αυτό του εξυπηρετητή, χρησιμοποιεί μια βιβλιοθήκη η οποία υλοποιεί την επικοινωνία μεταξύ της εφαρμογής και του εξυπηρετητή. Οι εφαρμογές μέσω της βιβλιοθήκης είναι δυνατόν να ζητήσουν τα δεδομένα των θέσεων μίας ή παραπάνω κινητών μονάδων. Στο παρακάτω σχήμα (σχήμα 8.2) απεικονίζονται τα κανάλια επικοινωνίας τα οποία χρησιμοποιούνται για την υλοποίηση των παραπάνω.



Σχήμα 8.2 Η επικοινωνία στο αισθητήριο παρακολούθησης κινητών μονάδων.

8.1.5.2 Υλοποίηση

Για την υλοποίηση της παραπάνω αρχιτεκτονικής χρησιμοποιούνται δύο τρόποι επικοινωνίας: αυτός με τον οποίο αποστέλλει η κινητή μονάδα τη θέση της στον εξυπηρετητή και αυτός με τον οποίο μια εφαρμογή ενημερώνεται για τις θέσεις των κινητών μονάδων του συστήματος.

8.1.5.3 Επικοινωνία κινητής μονάδας με τον εξυπηρετητή

Κάθε κινητή μονάδα του συστήματος επείγουσας προνοσοκομειακής ιατρικής περίθαλψης, χρησιμοποιεί GPS [57] έτσι ώστε να υπολογίσει τη θέση της. Το λογισμικό το οποίο υπάρχει στον φορητό υπολογιστή της κινητής μονάδας, αποκωδικοποιεί την πληροφορία του GPS και

στη συνέχεια την αποστέλλει στον εξυπηρετητή, επισυνάπτοντας τον κωδικό της κινητής μονάδας, ο οποίος την χαρακτηρίζει μοναδικά και την τρέχουσα ώρα (για λόγους συγχρονισμού όλων των κινητών μονάδων η τρέχουσα υπολογίζεται από την πληροφορία του GPS).

Εξαιτίας του γεγονότος του ότι η σύνδεση δικτύου μέσω κινητού τηλεφώνου μπορεί να διακοπεί θα θέλαμε η αποστολή του μηνύματος να διαρκεί όσο το δυνατόν λιγότερο, έτσι ώστε να αυξήσουμε την πιθανότητα επιτυχούς αποστολής των δεδομένων. Για αυτό το λόγο επιλέξαμε, όσον αφορά τον τρόπο αποστολής δεδομένων από την κινητή μονάδα στον εξυπηρετητή, την αποστολή πακέτου δεδομένων χωρίς επιβεβαίωση (πρωτόκολλο UDP socket [50]). Κατά αυτόν τον τρόπο η κινητή μονάδα στέλνει πακέτα δικτύου προς τον εξυπηρετητή χωρίς να περιμένει επιβεβαίωση ότι έφτασε η πληροφορία, πράγμα το οποίο σημαίνει αποστολή πληροφορίας επιβεβαίωσης από τον εξυπηρετητή και περαιτέρω επιβεβαίωση από την κινητή μονάδα (πρωτόκολλο TCP socket [51]) και έτσι η αποστολή ενός μηνύματος διαρκεί λιγότερο. Από την άλλη πλευρά, δεδομένης αποκλειστικής χρήσης της ψηφιακής γραμμής από τον υπολογιστή της κινητής μονάδας, η απώλεια πακέτων πληροφορίας έχει μικρή πιθανότητα να συμβεί (πειραματικά δεν είχαμε καθόλου απώλεια πακέτων) και ακόμη και σε τέτοια περίπτωση δεν είναι πιθανό να διαρκέσει για πολλές συνεχόμενες μεταδώσεις του μηνύματος. Άρα η συνιθισμένη περίπτωση απώλειας πληροφορίας προβλέπεται να είναι αυτή κατά την οποία η κινητή μονάδα βρίσκεται εκτός δικτύου κινητής τηλεφωνίας.

8.1.5.4 Επικοινωνία εφαρμογών με τον εξυπηρετητή

Η επικοινωνία των εφαρμογών με τον εξυπηρετητή, δεν εμπεριέχει περιορισμούς μιας και γίνεται πάνω από σταθερή TCP / IP σύνδεση από υπολογιστή σε υπολογιστή. Οι εφαρμογές επικοινωνούν είτε με ερωτήσεις για θέση συγκεκριμένου ασθενοφόρου, είτε λαμβάνουν αναμετάδοση όλων των μηνυμάτων που λαμβάνει και ο εξυπηρετητής. Η επικοινωνία αυτή υλοποιείται μέσω πρωτοκόλλου ορισμένου από τη βιβλιοθήκη που χρησιμοποιεί η εφαρμογή για να επικοινωνήσει με τον εξυπηρετητή. Η πρόσβαση στα δεδομένα των θέσεων των κινητών μονάδων γίνεται με διαφανή τρόπο, ως προς την επικοινωνία, δηλαδή ο προγραμματιστής της εφαρμογής τα χρησιμοποιεί χωρίς να προγραμματίζει την πρόσβαση σε αυτά ή την ανανέωση τους.

8.1.5.5 Λειτουργία του εξυπηρετητή

Ο εξυπηρετητής λαμβάνει τα μηνύματα των κινητών μονάδων από μια ουρά στην οποία εναποτίθενται κατά την άφιξη τους. Τα μηνύματα αυτά είναι μικρά και η ανάγνωση τους σύντομη, και έτσι δεν απαιτείται η χρήση πολλών νημάτων ελέγχου (threads) για την ανάγνωση τους. Ωστόσο για την επικοινωνία του εξυπηρετητή με τις υπόλοιπες εφαρμογές, χρησιμοποιούνται πολλά νήματα ελέγχου, αφού η επικοινωνία εκεί μπορεί να είναι ακόμη και συνεχής, όπως στην περίπτωση που κάποια εφαρμογή λαμβάνει αναμετάδοση όλων των μηνυμάτων (στη συνέχεια θα δούμε το πως κάτι τέτοιο συμβαδίζει με την διαφάνεια της επικοινωνίας δικτύου από την εφαρμογή).

Η εφαρμογή του εξυπηρετητή διατηρεί μια δομή ευρετηρίου για τις θέσεις των κινητών μονάδων, η πρόσβαση στα οποία γίνεται με κλειδί αναζήτησης τον κωδικό της μονάδας. Οι τιμές των δεδομένων ανανεώνονται με τη λήψη μηνυμάτων, ενώ τα νήματα ελέγχου φροντίζουν να εξυπηρετούν τις αιτήσεις των εφαρμογών, ενώ σε άλλες περιπτώσεις να διατηρούν οι εφαρμογές αντίγραφο της δομής ευρετηρίου, έτσι ώστε να μην απαιτείται από την εφαρμογή να υλοποιήσει την επικοινωνία με τον εξυπηρετητή.

8.1.5.6 Πλεονεκτήματα σχεδιασμού

Δύο είναι τα κύρια πλεονεκτήματα στην επιλογή της χρήσης εξυπηρετητή για τη διάθεση των δεδομένων στις υπόλοιπες εφαρμογές: η οικονομία στη χρήση του τηλεφωνικού δικτύου και η ευκολία ολοκλήρωσης με άλλες εφαρμογές. Ωστόσο η σχεδίαση του εξυπηρετητή ως αισθητήριο του συστήματος, απαιτεί να υπάρχει η δυνατότητα προσαρμογής σε περίπτωση μεταβολής του περιβάλλοντος και άρα ανάγκη τροποποίησης της συμπεριφοράς της οντότητας λογισμικού. Έτσι η οντότητα είναι υλοποιημένη με τέτοιο τρόπο έτσι ώστε η

συμπεριφορά των νημάτων ελέγχου, να είναι επαναπρογραμματιζόμενη (πχ. μια οντότητα γνωσσιακού επιπέδου να απαιτήσει την ελάττωση του ρυθμού ενημέρωσης των εφαρμογών προς όφελος της οικονομίας υπολογιστικής ισχύος).

8.1.6 Αισθητήριο παρακολούθησης βάσης δεδομένων

Το αισθητήριο αυτό καθιστά την πληροφορία η οποία υπάρχει στη βάση δεδομένων του συστήματος διαχειρίσιμη πόρων διαθέσιμη προς τις άλλες οντότητες του γνωσσιακού επιπέδου. Η επικοινωνία της οντότητας με τη βάση δεδομένων γίνεται πάνω από δίκτυο και με περιοδικό τρόπο (πχ. μια φορά τη μέρα). Ωστόσο όσον αφορά αντικείμενα για τα οποία υπάρχει συχνή μεταβολή της κατάστασης τους, όπως οι κινητές μονάδες, των οποίων η κατάσταση (διαθέσιμες / απασχολημένες) μεταβάλλεται συχνότερα, το αισθητήριο δεν συμβουλευείται τη βάση δεδομένων, αλλά κρατά σε δική του μνήμη την τρέχουσα κατάσταση του αντικειμένου.

8.2 Γνωσιακό επίπεδο

Όπως έχει οριστεί από την αρχιτεκτονική, το γνωσιακό επίπεδο αποτελείται από αντικείμενα τα οποία αντιμετωπίζουμε ως οντότητες λόγω της επιθυμίας μας να τους αποδώσουμε μια απλή αυτόνομη συμπεριφορά. Ουσιαστικά, όπως έχει προαναφερθεί, λόγω της ομοιότητας στο σχεδιασμό όλων των οντοτήτων αυτών του συστήματος, ο χαρακτηρισμός του γνωσσιακού επιπέδου αποτελεί μια ταξινόμηση οντοτήτων οι οποίες επικοινωνούν με τα αισθητήρια του συστήματος και προσπαθούν να υλοποιήσουν στόχους οι οποίοι έχουν ανατεθεί στο σύστημα από το χρήστη, χρησιμοποιώντας τα αντικείμενα δράσης. Ωστόσο ο κυριότερος λόγος για τον προκειμένο χαρακτηρισμό, είναι πως το σύνολο των οντοτήτων αυτών, καθορίζει τη συμπεριφορά του συστήματος όπως παρατηρείται από τον εξωτερικό παρατηρητή.

8.2.1 Στόχοι

Η λειτουργία του γνωσσιακού επιπέδου, είναι η λήψη αποφάσεων καθώς και η εξαγωγή ή υπολογισμός συμπερασμάτων, σχετικά με τη βέλτιστη διαχείριση των πόρων ενός προνοσοκομειακού συστήματος επείγουσας ιατρικής περίθαλψης. Το γνωσιακό επίπεδο στη συνέχεια θα πρέπει να χρησιμοποιήσει τα αντικείμενα δράσης του συστήματος είτε για την προσπάθεια εφαρμογής των αποφάσεων, είτε για την παρουσίαση των συμπερασμάτων προς το χρήστη.

8.2.2 Σχεδίαση

Οι οντότητες οι οποίες αποτελούν το γνωσιακό επίπεδο, είναι δυνατόν να ταξινομηθούν ιεραρχικά, ωστόσο μια άλλη όψη η οποία μας δίνει μια καλύτερη άποψη για τον τρόπο απόκτησης συμπεριφοράς από το σύστημα, είναι αυτή η οποία παρουσιάζει τους δρόμους (ή κανάλια) επικοινωνίας και μεταφοράς πληροφορίας μεταξύ των οντοτήτων αυτών. Η επικοινωνία τόσο μεταξύ των οντοτήτων του γνωσσιακού επιπέδου, όσο και με τα αισθητήρια και τα αντικείμενα δράσης είναι συχνή και αποσκοπεί στην κατανομή του ελέγχου και υπολογισμού, σε απλούστερες οντότητες, όπου η κάθε μια είναι εξειδικευμένη σε κάποια όσο το δυνατόν απλή λειτουργία. Ο τρόπος με τον οποίο επικοινωνούν οι οντότητες καθορίζει και την συμπεριφορά των οντοτήτων.

Όλες οι οντότητες του γνωσσιακού επιπέδου, ακολουθούν τη σχεδίαση των αυτόνομων οντοτήτων που έχουμε περιγράψει, έχοντας έτσι η κάθε μια αισθητήρες οι οποίοι παρακολουθούν το περιβάλλον της οντότητας, το οποίο μπορεί να είναι είτε το περιβάλλον του συστήματος όπως γίνεται αυτό αντιληπτό μέσω των αισθητήρων, είτε το γνωσιακό επίπεδο και επίπεδο δράσης, το οποίο μπορεί να επικοινωνήσει με άλλες οντότητες ή με τα αντικείμενα δράσης του συστήματος. Οι οντότητες του γνωσσιακού επιπέδου μπορούν να διαχειρίζονται παραπάνω από μια διαδικασίες ταυτόχρονα, χρησιμοποιώντας τεχνικές

πολυπρογραμματισμού και συγκεκριμένα, πολλαπλά νήματα ελέγχου. Μάλιστα σε ορισμένες οντότητες οι οποίες, επιτελούν σύνθετες διεργασίες, κάθε μια από τις διαδικασίες η οποίες επιτελούν έναν υπολογισμό, είναι δυνατόν να φέρει τα γνωρίσματα σχεδίασης μιας αυτόνομης οντότητας. Στη συνέχεια θα δούμε πως διαφοροποιούνται οι οντότητες του γνωσιακού επιπέδου, ανάλογα με την εργασία την οποία έχουν να επιτελέσουν, διατηρώντας ταυτόχρονα τις βασικές ομοιότητες στο σχεδιασμό τους.

Οι κύριες οντότητες του γνωσιακού επιπέδου είναι η οντότητα αντιμετώπισης επειγόντων περιστατικών, η οντότητα διαχείρισης κινητών μονάδων, οντότητες οι οποίες αναζητούν τον καταλληλότερο πόρο (νοσοκομείο, κινητή μονάδα, ιατρικό προσωπικό κ.τ.λ.) για κάθε περιστατικό, η οντότητα εύρεσης συντομότερης χρονικά διαδρομής κ.α.

8.2.3 Δύο όψεις του γνωσιακού επιπέδου

8.2.3.1 Ιεραρχική

Ταξινομώντας ιεραρχικά τις οντότητες του γνωσιακού επιπέδου, διακρίνουμε τέσσερις κατηγορίες οντοτήτων κάτω από τις οποίες μπορούμε να τις ταξινομήσουμε: (1) οντότητες οι οποίες υλοποιούν τους βασικούς στόχους του συστήματος (2) οντότητες εξυπηρετητές (3) οντότητες οι οποίες εκτελούν μια προκαθορισμένη εργασία ή υπολογισμό, (4) οντότητες οι οποίες ελέγχουν και προσαρμόζουν την λειτουργία άλλων οντοτήτων. Κάθε μια από τις οντότητες αυτές προσπαθεί να φέρει σε πέρας μια ή παραπάνω εργασίες οι οποίες της έχουν ανατεθεί. Για να το καταφέρει είναι δυνατόν να επικοινωνήσει με κάποια άλλη οντότητα, ώστε να της ζητήσει κάποια πληροφορία (την οποία πιθανόν η άλλη οντότητα για να παραδώσει να χρειαστεί να επιτελέσει κάποιον υπολογισμό), ή να δημιουργήσει νέες. Η αρχιτεκτονική και σχεδιασμός των οντοτήτων οι οποίες αποτελούν το γνωσιακό επίπεδο, είναι κοινή με όλες τις αυτόνομες οντότητες τις οποίες έχουμε περιγράψει.

Οντότητες οι οποίες υλοποιούν στόχους: Οι οντότητες οι οποίες υλοποιούν στόχους στην πλειονότητα τους παρακολουθούν ένα η παραπάνω αισθητήρια, έτσι ώστε να μπορούν να αντιλαμβάνονται γεγονότα στα οποία θα πρέπει να αντιδράσει το σύστημα, ενώ είναι δυνατόν να χρησιμοποιήσουν τα αντικείμενα δράσης, έτσι ώστε να προσπαθήσουν να εφαρμόσουν κάποιο στόχο. Στη συνέχεια πάλι βέβαια είναι δυνατόν να χρειαστούν περαιτέρω επικοινωνία με τα αισθητήρια έτσι ώστε να επιβεβαιώσουν ότι η προσπάθεια εφαρμογής της συγκεκριμένης ενέργειας στο περιβάλλον εξελίσσεται ομαλά και εάν όχι τότε να εφαρμόσουν διορθωτικές ενέργειες, ή να καταγράψουν το λάθος.

Η κατηγορία αυτή των οντοτήτων αποτελείται από οντότητες οι οποίες επιτελούν σύνθετες διεργασίες και στη συνήθη περίπτωση παραπάνω από μια ταυτόχρονα. Κάθε οντότητα η οποία εξυπηρετεί κάποιο στόχο είναι δυνατόν να επεξεργάζεται ταυτόχρονα παραπάνω από μια διαδικασίες, ωστόσο εάν αυτές είναι αλληλεξαρτώμενες μεταξύ τους, τότε αναλαμβάνει αυτή το συντονισμό τους. Για αυτό το λόγο με την εκκίνηση κάθε μιας διεργασίας από μια τέτοια οντότητα, δημιουργείται μια νέα οντότητα η οποία έχει τον έλεγχο των υποδιεργασιών οι οποίες επιτελούνται για την εκπλήρωση της. Επιπλέον οι υπό διεργασίες αυτές υλοποιούνται από την τρίτη κατηγορία οντοτήτων που αναφέραμε παραπάνω. Ωστόσο επειδή τις πιο πολλές φορές, πολλές από τις υπό διεργασίες αλληλεξαρτώνται όσον αφορά την είσοδο και τα αποτελέσματα των υπολογισμών τους, η διαίτησία και διαχείριση καθώς και η ανάθεση ποσοστού υπολογιστικής ισχύος, γίνεται από την οντότητα η οποία έχει αναλάβει έλεγχο των υποδιεργασιών. Η τελευταία εκμεταλλεύεται το γνώρισμα των οντοτήτων της τρίτης κατηγορίας, οι οποίες μπορούν να χρονοπρογραμματιστούν, έτσι ώστε να ορίσουν τη σειρά με την οποία θα εκτελεστούν, το ποιες είναι δυνατόν να πραγματοποιηθούν ταυτόχρονα με άλλες και το ποιες θα περιμένουν το αποτέλεσμα μιας άλλης υπό διεργασίας έτσι ώστε να μπορέσουν να ξεκινήσουν τη λειτουργία τους. Στη συνέχεια θα περιγράψουμε την τεχνική με την οποία υλοποιούμε μια τέτοια δυνατότητα καθώς και τη χρησιμότητα της στη χρήση εναλλακτικών εναλλακτικών πολιτικών.

Η παρακολούθηση των αισθητηρίων του συστήματος γίνεται με περιοδικό τρόπο από ένα νήμα ελέγχου, το οποίο προσπελαίνει τους καταχωρητές γεγονότων των αντίστοιχων αισθητήρων (μετά από επικοινωνία με τις αντίστοιχες οντότητες των αισθητήρων). Όταν

υπάρξει ένα νέο γεγονός αυτό προωθείται προς την αντίστοιχη γνωσιακή οντότητα. Τις περισσότερες φορές μια οντότητα η οποία υλοποιεί ένα στόχο του συστήματος, θα ζητήσει από κάποιο αντικείμενο δράσης την εφαρμογή κάποιας εντολής. Η παρακολούθηση της εντολής αυτής, γίνεται μέσω των αισθητήρων του συστήματος και σε περίπτωση αποτυχίας η ίδια οντότητα είναι υπεύθυνη για τη λήψη αποφάσεων αντιμετώπισης της κατάστασης.

Οι οντότητες οι οποίες εκτελούν μια προκαθορισμένη διαδικασία ή υπολογισμό, δημιουργούνται από άλλες οντότητες οι οποίες θέλουν εκτελέσουν την ανατιθείσα εργασία ταυτόχρονα με άλλες. Επιπλέον, ο λόγος ύπαρξης τέτοιων οντοτήτων, είναι ο διαχωρισμός των επί μέρους διεργασιών, οι οποίες πρέπει να εκτελεστούν για την πραγματοποίηση μιας εργασίας υψηλότερου επιπέδου. Οι οντότητες αυτές παρέχουν δυνατότητες χρονοπρογραμματισμού τους, καθώς και προτεραιότητες όσον αφορά το ποσοστό της υπολογιστικής ισχύος το οποίο θα τους αντιστοιχεί, ενώ είναι δυνατόν να αντιμετωπιστούν ως απλά προγραμματιστικά αντικείμενα, γεγονός που μας επιτρέπει να τα διατάσουμε σε δομές δεδομένων (όπως βέβαια και κάθε άλλη οντότητα). Οι οντότητες του τύπου αυτού περιέχουν συνήθως τη γνώση η οποία απαιτείται για την ορθή εκτέλεση της ανατιθείσας εργασίας ή υπολογισμού.

Η αρχιτεκτονική μιας τέτοιας οντότητας έχει ως εξής: Η οντότητα αποτελείται από ένα απλό αισθητήριο με το οποίο δέχεται εντολές εκκίνησης, διακοπής ή αναστολής της εργασίας της, το κυρίως τμήμα το οποίο περιέχει τις εντολές και πληροφορία για την εκτέλεση της ανατιθείσας εργασίας και ένα απλό αντικείμενο δράσης, το οποίο πληροφορεί την οντότητα η οποία ζήτησε τη δημιουργία της προκείμενης οντότητας, για το επιτυχές πέρας της εργασίας ή για το σφάλμα που κατέστησε αδύνατη την εκπλήρωσή της. Η λειτουργία της διεργασίας μπορεί να διακοπεί και να συνεχιστεί, ενώ το ποσοστό υπολογιστικής ισχύος το οποίο αντιστοιχεί στη διεργασία, είναι ρυθμίσσιμο από την οντότητα η οποία δημιούργησε τη διεργασία. Η εκτέλεση των διεργασιών από την οντότητα, γίνεται παράλληλα με όλες τις υπόλοιπες διεργασίες του συστήματος κάνοντας χρήση νημάτων ελέγχου.

Η δυνατότητα των οντοτήτων της κατηγορίας αυτής να διατάσσονται σε δομές δεδομένων, οφείλεται στην υλοποίηση των οντοτήτων με αντικείμενα. Κατά αυτόν τον τρόπο, η οντότητα η οποία τις δημιουργεί, μπορεί είτε να τις κατατάξει π.χ. σε μια λίστα, όπου εκτελεί την κάθε μια μετά το πέρας της εκτέλεσής της προηγούμενης, ή σε κάποια πιο πολύπλοκη δομή η οποία ορίζει επιτρέπει να εκτελούνται κάποιες διεργασίες παράλληλα και άλλες όχι. Ωστόσο για την επίτευξη ενός τέτοιου μηχανισμού, γίνεται χρήση τόσο του αισθητήριου της οντότητας, το οποίο δέχεται εντολές για εκκίνηση ή διακοπή της εκτέλεσής υπό τη μορφή μηνυμάτων, όσο και της δυνατότητας της οντότητας, να πληροφορεί άλλες σχετικά με την πρόοδο της εκτέλεσής της εργασίας. Τέλος, αξίζει να σημειωθεί πως οι οντότητες αυτής της κατηγορίας παύουν να υπάρχουν μετά το τέλος της εργασίας τους, οπότε και καταστρέφονται ελευθερώνοντας τη μνήμη (ή οποιόν άλλο υπολογιστικό πόρο) χρησιμοποιούσαν.

Οντότητες εξυπηρετητές: Μια άλλη κατηγορία οντοτήτων αποτελεί αυτή η οποία παρακολουθεί με πιο παθητικό ρυθμό τις εξελίξεις στο περιβάλλον και τα μέλη της αποτελούν την κατηγορία των οντοτήτων εξυπηρετητών. Οι οντότητες αυτές κατά κύριο λόγο επιτελούν κάποιον υπολογισμό μετά από αίτηση κάποιας άλλης οντότητας. Ωστόσο είναι δυνατόν να δρουν παράλληλα πάνω σε άλλες εσωτερικές διεργασίες, προκειμένου να βελτιστοποιήσουν την απόδοσή τους. Οι οντότητες εξυπηρετητές, ανάλογα με το περιεχόμενο της εφαρμογής τους, είναι δυνατόν να εξυπηρετούν μια αίτηση τη φορά ή πολλές ταυτόχρονα, χρησιμοποιώντας τεχνικές πολυπρογραμματισμού. Σε μια οντότητα εξυπηρετητή το επίπεδο αίσθησης αποτελούν οι καταχωρητές γεγονότων στους οποίους κατακρατούνται οι εισερχόμενες αιτήσεις, το γνωσιακό επίπεδο το οποίο εξυπηρετεί την αίτηση και το επίπεδο δράσης το οποίο αναλαμβάνει να απαντήσει στην αιτούσα οντότητα.

Το γνωσιακό επίπεδο μιας οντότητας εξυπηρετητή, δημιουργεί για κάθε εισερχόμενη αίτηση, μια οντότητα της προηγούμενης κατηγορίας, στην οποία αναθέτει τον υπολογισμό. Έτσι σε μια τυχαία χρονική στιγμή μια οντότητα εξυπηρετητής είναι δυνατόν να εξυπηρετεί και ταυτόχρονα να συντονίζει έναν αριθμό από οντότητες στις οποίες έχει ανατεθεί κάποιος υπολογισμός. Μετά το πέρας του κάθε υπολογισμού το αποτέλεσμα επιστρέφεται στην αιτούσα οντότητα μέσω των αντικειμένων δράσης τα οποία έχουν αναλάβει την επικοινωνία.

Μια οντότητα εξυπηρετητής, είναι δυνατόν να εμπεριέχει μια επικουρική οντότητα η οποία να ρυθμίζει τις παραμέτρους ή την πολιτική λειτουργίας της.

Οντότητες οι οποίες ρυθμίζουν τη λειτουργία άλλων οντοτήτων: Η κατηγορία των οντοτήτων αυτών, αποτελεί μια κατηγορία επικουρικών οντοτήτων. Η λειτουργία τους συνίσταται στην παρακολούθηση της απόδοσης άλλων οντοτήτων, των οποίων η λειτουργία εξαρτάται από παραμέτρους λειτουργίας και η ρύθμιση των παραμέτρων αυτών, έτσι ώστε να επιτευχθεί βελτιστοποίηση της απόδοσης λειτουργίας σε σχέση με το συγκεκριμένο περιβάλλον λειτουργίας του συστήματος. Οι οντότητες αυτές, οι οποίες χρησιμοποιούν τεχνικές από τον τομέα της μηχανικής μάθησης [46], είναι δυνατόν να αποτελέσουν εσωτερικές διεργασίες σε άλλες οντότητες, κυρίως για λόγους διαφάνειας στο προγραμματιστικό επίπεδο, ενώ είναι δυνατόν να κάνουν χρήση του προσομοιωτή για δοκιμή προτεινόμενων πολιτικών αντί της δοκιμής κατευθείαν στο πραγματικό περιβάλλον.

Η κυριότερη τεχνική μηχανικής μάθησης η οποία χρησιμοποιείται είναι αυτή του Q - learning [34], όπου χρησιμοποιείται ένας διδιάστατος πίνακας, όπου στον ένα άξονα απαριθμούνται οι δυνατές καταστάσεις του συστήματος, ενώ στον άλλον οι δυνατές μέθοδοι αντιμετώπισης των καταστάσεων. Το περιεχόμενο του πίνακα, περιέχει τιμές οι οποίες αφορούν την αποδοτικότητα της μεθόδου και οι οποίες ενημερώνονται με κάθε χρήση της μεθόδου στη συγκεκριμένη κατάσταση. Μια οντότητα είναι δυνατόν να χρησιμοποιήσει τον προσομοιωτή, κυρίως για να αποδώσει αρχικές τιμές στον πίνακα αυτό, ή να εξετάσει μια νέα μέθοδο ή κάποια άλλη η οποία ίσως να είχε εγκαταλειφθεί στο παρελθόν. Παράδειγμα τέτοιας μεθόδου είναι δυνατόν να βρεθεί στην οντότητα διαχείρισης περιπολιών των κινητών μονάδων του συστήματος.

Στις πιο πολλές περιπτώσεις, ο πίνακας ο οποίος θα χρησιμοποιείται θα είναι αρκετά απλός, αφού αυτό που θα θέλουμε τις περισσότερες φορές να κάνουμε, είναι εύρεση του καταλληλότερου αλγορίθμου για κάποια λειτουργία του συστήματος, ή προσέγγιση των καταλληλότερων παραμέτρων για τη λειτουργία κάποιας οντότητας. Καθοριστικό ωστόσο ρόλο, παίζει η δυνατότητα αποτίμησης της απόδοσης της κάθε μεθόδου η οποία ελέγχεται, έτσι ώστε να είναι δυνατή η συμπλήρωση του πίνακα με τιμές οι οποίες εκφράζουν την καταλληλότητα της κάθε μεθόδου, ανάλογα με την κατάσταση στην οποία βρίσκεται το σύστημα.

8.2.3.2 Επικοινωνία μεταξύ των οντοτήτων

Μια άλλη όψη του γνωσιακού επιπέδου που επιδεικνύει τον τρόπο λειτουργίας του συστήματος, είναι αυτή η οποία παρουσιάζει το δίκτυο επικοινωνίας μεταξύ των οντοτήτων. Αν και στο γνωσιακό επίπεδο, δεν υπάρχει κάποιο σημείο κεντρικού ελέγχου, ο συγχρονισμός και η διαίτησία μεταξύ πολλών και αρκετές φορές ανταγωνιστικών διαδικασιών, επιτυγχάνεται με την ύπαρξη συγκεκριμένου δικτύου επικοινωνίας μεταξύ των οντοτήτων. Αναλυτικότερα, οι οντότητες προκειμένου να πραγματοποιήσουν η κάθε μια το στόχο ο οποίος της έχει ανατεθεί, χρησιμοποιούν τόσο τη βοήθεια άλλων οντοτήτων, όσο και τα αντικείμενα δράσης του συστήματος. Οι οντότητες δεν είναι πάντα ελεύθερες να χρησιμοποιήσουν όλες τις υπόλοιπες οντότητες σε οποιαδήποτε χρονική στιγμή (πχ. για την εξυπηρέτηση μιας αίτησης είναι δυνατόν να κρατείται σειρά προτεραιότητας), όπως και δεν είναι δυνατόν να χρησιμοποιηθεί ένας ήδη κατειλημμένος πόρος. Επιπλέον, κάποιες οντότητες έχουν προτεραιότητα έναντι άλλων στη χρήση οντοτήτων και πόρων του συστήματος.

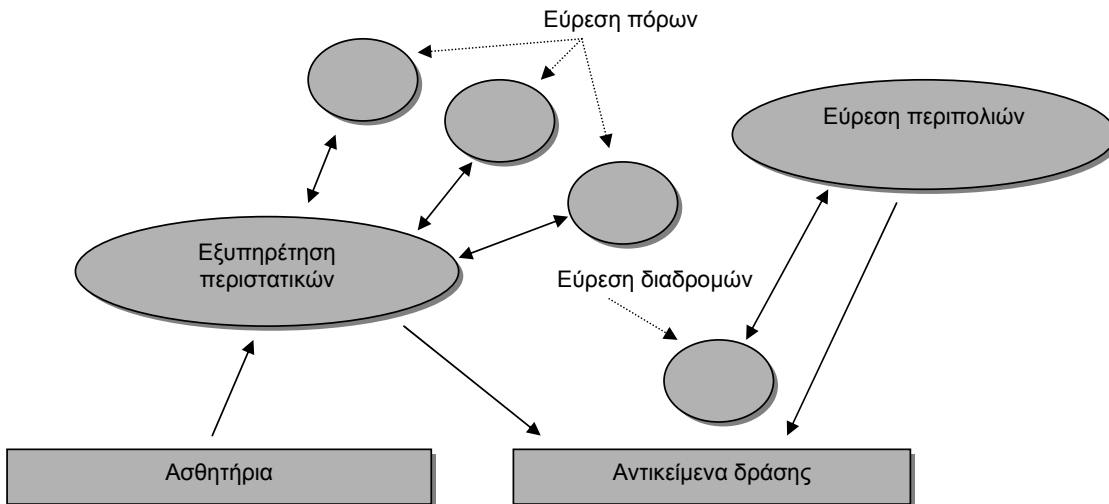
Για λόγους που έχουμε δει προηγούμενα το δίκτυο αυτό είναι προκαθορισμένο με την εκκίνηση του συστήματος, ωστόσο είναι δυνατόν πριν ξεκινήσουμε το σύστημα, να ρυθμίζουμε το δίκτυο έτσι ώστε να ανταποκρίνεται στις απαιτήσεις του εκάστοτε περιβάλλοντος και ιατρικού πρωτοκόλλου (βλ. Εφαρμογή στο προνοσοκομειακό σύστημα υγείας). Γενικά αυτό το οποίο μπορεί να ειπωθεί, είναι πως οι οντότητες φροντίζουν για την τήρηση του δικτύου επικοινωνίας, απλά ακολουθώντας τις εντολές τις οποίες έχουν για την εργασία την οποία έχουν να εκτελέσουν. Αν και η ακολουθία των εντολών αυτών είναι μεταβλητή, τις περισσότερες φορές, λόγω περιορισμών από το ιατρικό πρωτόκολλο, ή και απλά λόγω λογικής σειράς, παραμένει αμετάβλητη για την εξυπηρέτηση όλων των περιστατικών. Ωστόσο το γεγονός της δυνατότητας μεταβολής της ακολουθίας αυτής, μας

δίνει τη δυνατότητα εύκολης προσαρμογής του συστήματος μας σε διαφορετικά περιβάλλοντα.

Η κυριότερη οντότητα του συστήματος και αυτή η οποία έχει τη μεγαλύτερη προτεραιότητα έναντι των άλλων, είναι η οντότητα αντιμετώπισης επειγόντων περιστατικών. Η αντίληψη ενός γεγονότος σε αυτή την περίπτωση γίνεται με διαρκή έλεγχο του αισθητηρίου αντίληψης και παρακολούθησης εξέλιξης περιστατικών, όπου με την εκδήλωση ενός νέου επεισοδίου, δημιουργείται ένα γεγονός και η οντότητα γεννά μια οντότητα - διεργασία η οποία είναι αφιερωμένη στην αντιμετώπιση του γεγονότος. Για την διεκπεραίωση της εργασίας η οντότητα θα χρειαστεί να επικοινωνήσει με τις οντότητες εύρεσης συντομότερης διαδρομής και καταλληλότερου ιατρικού κέντρου και κινητής μονάδας και ανάλογα με την εξέλιξη του περιστατικού με την οντότητα επιλογής καταλληλότερου ιατρικού προσωπικού. Σε όλες τις προηγούμενες περιπτώσεις, τηρείται σειρά προτεραιότητας από τις αντίστοιχες οντότητες εξυπηρετητές, εκτός των περιπτώσεων περιστατικών τα οποία έχουν χαρακτηριστεί κρίσιμης σημασίας, οπότε και τα οποία εξυπηρετούνται πρώτα. Αυτό που έχει σημασία να παρατηρήσει κανείς είναι το ότι ορισμένες υποδιεργασίες της αντιμετώπισης του επεισοδίου, γίνονται με συγκεκριμένη σειρά, ενώ άλλες παράλληλα. Για παράδειγμα η πρώτη διεργασία η οποία πρέπει να γίνει είναι η επιλογή κινητή μονάδας, έτσι ώστε να ξεκινήσει η μονάδα προς τον προορισμό της. Κατά την παρακολούθηση της εξέλιξης της κίνησης της μονάδας προς την τοποθεσία του επεισοδίου, είναι δυνατόν να υπολογίζεται ο προορισμός του ασθενή και η πορεία της κινητής μονάδας προς αυτό.

Από την άλλη πλευρά η οντότητα διαχείρισης κινητών μονάδων (υπεύθυνη για περιπολίες των μη απασχολημένων κινητών μονάδων), επικοινωνεί με το αισθητήριο παρακολούθησης κινητών μονάδων, για να προσπελάσει την πληροφορία των θέσεων των κινητών μονάδων. Παρ' όλα αυτά πρέπει η μονάδα να μην είναι απασχολημένη, από την οντότητα αντιμετώπισης περιστατικών η οποία έχει προτεραιότητα στην χρήση των πόρων αυτών.

Οι οντότητες εύρεσης καταλληλότερου πόρου, ταξινομούνται κάτω από την κατηγορία των οντοτήτων εξυπηρετητών και για τη λειτουργία τους, απαιτούν επικοινωνία με τα αντίστοιχα αισθητήρια της παρακολούθησης των κινητών μονάδων και βάσης δεδομένων. Όλες οι οντότητες της κατηγορίας αυτής, εξυπηρετούν τις αιτήσεις κατά σειρά προτεραιότητας, εκτός από τις αιτήσεις οι οποίες έχουν χαρακτηριστεί ως περισσότερο επείγουσες από άλλες στις οποίες δίνεται προτεραιότητα. Σε περίπτωση όπου υπάρχουν παραπάνω από μια τέτοιες αιτήσεις ταυτόχρονα, αυτές εκτελούνται παράλληλα (εκτός από την περίπτωση επιλογής κινητής μονάδας, όπου έτσι και αλλιώς η εξυπηρέτηση της αίτησης είναι υπολογιστικά φθηνή).



Σχήμα 8.3 Επικοινωνία μεταξύ οντοτήτων

Το παραπάνω σχήμα (σχήμα 8.3) δίνει μια εικόνα μερικών από τις επικοινωνίες που λαμβάνουν χώρα στο σύστημα.

8.2.4 Παρουσίαση μηχανισμού διαχείρισης σχεδίων δράσης των οντοτήτων

Στην παράγραφο αυτή θα περιγράψουμε τον τρόπο με τον οποίο οι οντότητες ακολουθούν προκαθορισμένα σχέδια τα οποία περιγράφουν τον τρόπο λειτουργίας τους. Όπως έχουμε αναφέρει, η δυνατότητα αυτή αν και θα μπορούσε να μας χρησιμεύσει στην ανάπτυξη πιο πολύπλοκης συμπεριφοράς από τις οντότητες, σε περίπτωση που υπήρχε κάποια άλλη οντότητα η οποία να παράγανε τα σχέδια δράσης, στο σύστημα μας βρίσκει εφαρμογή στην προσαρμογή της συμπεριφοράς των οντοτήτων στο εκάστοτε ιατρικό πρωτόκολλο καθώς και στο χειρισμό λαθών. Στο τέλος της παραγράφου θα μελετήσουμε αναλυτικά ένα παράδειγμα σχεδίου δράσης, αυτό της οντότητας αντιμετώπισης περιστατικών.

Ένα σχέδιο δράσης αποτελείται από εντολές οι οποίες θα θέλαμε να εκτελεστούν με ορισμένη σειρά. Η πληροφορία η οποία ορίζει τη σειρά εκτέλεσης είναι και αυτή μέρος του σχεδίου. Οι εντολές οι οποίες πρόκειται να εκτελεστούν αποτελούνται από οντότητες διεργασίες οι οποίες αφενός είναι δυνατόν να διαταχτούν σε δομή δεδομένων, αφετέρου δέχονται εντολές για τον τρόπο εκτέλεσης τους, ενώ παράγουν και μηνύματα τα οποία περιγράφουν το τέλος ή την αποτυχία της εργασίας που τους ανατέθηκε. Επίσης το σχέδιο δράσης περιέχει τον ορισμό παραμέτρων για τις εργασίες οι οποίες απαιτούν παραμετρική είσοδο για την εκτέλεση τους, ενώ είναι δυνατόν να κρατά αποτελέσματα υπολογισμών σε τοπικές μεταβλητές. Προγραμματιστικά ένα σχέδιο δράσης υλοποιείται με ένα αντικείμενο το οποίο περιέχει τη δομή του σχεδίου και ένα ευρετήριο στο οποίο είναι δυνατόν να αποθηκεύονται και να ανακαλούνται τοπικές μεταβλητές με το όνομα τους. Τέλος, υπάρχει η δυνατότητα διακοπής της εκτέλεσης ενός σχεδίου δράσης, εν μέσο εκτέλεσης του, και η αντικατάσταση του από ένα άλλο.

Η οντότητα η οποία πρόκειται να ακολουθήσει το σχέδιο δράσης, διατηρεί δομή δεδομένων η οποία περιγράφει τη σειρά εκτέλεσης των εντολών όπως και περιγραφή των αποτελεσμάτων υπολογισμών τα οποία θα χρησιμεύσουν ως είσοδος σε άλλους. Η δομή αυτή αποτελείται από ένα δέντρο αποφάσεων του οποίου τα μέλη είναι δυνατόν να είναι απλές διεργασίες ή σύνολα διεργασιών. Ακολουθώντας το δέντρο από τη ρίζα προς τα φύλλα έχουμε κάθε φορά έναν τρέχοντα κόμβο ο οποίος μπορεί να περιέχει μια ή παραπάνω διεργασίες προς εκτέλεση. Σε κάθε βήμα οι διεργασίες οι οποίες είναι δυνατόν να εκτελεστούν παράλληλα, όμως για την πρόοδο στον επόμενο κόμβο απαιτείται η περάτωση των όλων διεργασιών του κόμβου. Η ύπαρξη κόμβων με παραπάνω από ένα παιδιά σηματοδοτεί την εξάρτηση του επόμενου βήματος από το αποτέλεσμα του προηγούμενου. Αναλυτικότερα, με αυτό τον τρόπο θέλουμε να εκφράσουμε το ότι η ροή ελέγχου είναι δυνατόν να φτάσει σε κάποιο σημείο, στο οποίο υπάρχουν παραπάνω από μια επιλογές για το ποιο θα είναι το επόμενο βήμα και η απόφαση για το ποιο θα ακολουθηθεί εξαρτάται από την έκβαση του προηγούμενου. Στην περίπτωση μας η εναλλακτικές επιλογές αφορούν κυρίως το χειρισμό λαθών τα οποία μπορεί να συμβούν κατά την εκτέλεση κάποια διεργασίας. Οι διεργασίες είναι τοποθετημένες με τέτοιο τρόπο στη δομή, έτσι ώστε οι διεργασίες των οποίων η είσοδος αποτελεί το αποτέλεσμα κάποιου άλλου υπολογισμού, να το έχουν διαθέσιμο κατά την εκκίνηση της εκτέλεσης τους.

Για την επίτευξη της επιθυμητής λειτουργίας, η οντότητα η οποία πρόκειται να ακολουθήσει το σχέδιο δράσης, ξεκινά προσπελαύνοντας τον πρώτο κόμβο του δέντρου, και δίνοντας εντολή στις διεργασίες που περιέχει να ξεκινήσουν. Η λειτουργία του νήματος ελέγχου της λειτουργίας της οντότητας αναστέλλεται, χωρίς να καταναλώνει υπολογιστικό χρόνο, έως ότου τερματιστούν όλες η διεργασίες οι οποίες ξεκίνησαν. Στη συνέχεια και ανάλογα με το αποτέλεσμα των διεργασιών το νήμα ελέγχου αποφασίζει για το ποιος θα είναι ο επόμενος κόμβος προς εκτέλεση ή εάν έχουμε φτάσει στο τέλος του σχεδίου δράσης τερματίζει τη λειτουργία της. Σε ορισμένες περιπτώσεις ολικής αποτυχίας του σχεδίου δράσης λόγω εξωγενών παραγόντων είναι δυνατόν να εκγαταληφθεί και να αντικατασταθεί από άλλο.

Επιγραμματικά το σχέδιο δράσης της οντότητας αντιμετώπισης επειγόντων περιστατικών μπορεί να περιγραφεί ως εξής:

Δεδομένης της τοποθεσίας του περιστατικού πρέπει να βρεθεί η καταλληλότερη κινητή μονάδα η οποία και θα εξυπηρετήσει το περιστατικό (όπως θα δούμε κατά την παρουσίαση της οντότητας εύρεσης κινητής μονάδας η επιλογή αυτή είναι δυνατόν να γίνεται με παραπάνω από ένα κριτήρια). Η εξέλιξη της διεργασίας αντιμετώπισης του επεισοδίου παραμένει στάσιμη, έως ότου να βρεθεί κινητή μονάδα η οποία θα εξυπηρετήσει το περιστατικό.

Αμέσως μετά την επιλογή κινητής μονάδας είναι δυνατόν να υπολογιστεί η διαδρομή από την τοποθεσία στην οποία βρίσκεται, σε αυτή στην οποία εκδηλώθηκε το περιστατικό. Η εργασία αυτή είναι ανεξάρτητη από άλλες και είναι δυνατόν να γίνει παράλληλα.

Ένα άλλο δεδομένο προς υπολογισμό είναι το ιατρικό κέντρο το οποίο θα εξυπηρετήσει το περιστατικό και άρα ο προορισμός του ασθενή. Ανάλογα με το ισχύων ιατρικό πρωτόκολλο και τα διαθέσιμα ιατρικά κέντρα και τον τύπο του περιστατικού, η επιλογή αυτή είναι δυνατόν να αποφασιστεί μονάχα με γνώμονα την τοποθεσία του επεισοδίου, ή λαμβάνοντας υπ' όψιν τον χαρακτηρισμό του περιστατικού από τον ασυρματιστή, ή ακόμα περιμένοντας ένα δεύτερο χαρακτηρισμό του περιστατικού από το προσωπικό το οποίο επιβαίνει στην κινητή μονάδα. Χρησιμοποιώντας σχέδια δράσης τα οποία παράγονται δυναμικά (ή έστω βάση κάποιου αλγορίθμου ανάλογα με το περιστατικό και τις τρέχουσες συνθήκες του συστήματος), είναι δυνατόν να διαφοροποιείται η αντιμετώπιση του κάθε περιστατικού. Από την άλλη πλευρά η κρισιμότητα τέτοιων επιλογών και η έλλειψη περιθωρίου για λανθασμένη επιλογή, μας οδηγεί στην ανάγκη οδηγιών από το χρήστη, με τη μορφή ιατρικού πρωτοκόλλου ή μεθοδολογίας αντιμετώπισης των περιστατικών. Παρατηρούμε πως ο υπολογισμός αυτός είναι δυνατόν να γίνει παράλληλα με τον πρώτο, κατά τη διάρκεια μετάβασης της κινητής μονάδας στην τοποθεσία του περιστατικού ή ακόμα όταν φτάσει η κινητή μονάδα στον πρώτο προορισμό της.

Ανάλογα με την επιλογή την οποία κάνουμε για το χρόνο υπολογισμού του ιατρικού κέντρου στο οποίο πρέπει να μεταφερθεί ο ασθενής, ο υπολογισμός της διαδρομής μεταξύ της τοποθεσίας περιστατικού και του κέντρου είναι δυνατόν να γίνει παράλληλα ή εν σειρά με τους προηγούμενους υπολογισμούς.

Μετά την άφιξη του ασθενή στο ιατρικό κέντρο, και καθ' όλα τα στάδια εξέλιξης του περιστατικού, η βάση δεδομένων ενημερώνεται και το επεισόδιο κλείνει. Για απλούστερη παρουσίαση του παραπάνω παραδείγματος, έχουν παραληφθεί λάθη τα οποία είναι δυνατόν να συμβούν (όπως πχ. βλάβη στη λειτουργία της κινητής μονάδας). Θα αναφερθούμε αναλυτικότερα σε αυτά στη συνέχεια και κατά την περιγραφή της κάθε οντότητας του γνωσιακού επιπέδου.

Στη συνέχεια θα περιγράψουμε αναλυτικότερα τις οντότητες οι οποίες αποτελούν το γνωσιακό επίπεδο.

8.2.5 Οντότητες εύρεσης καταλληλότερου πόρου

Η κατηγορία αυτή οντοτήτων του γνωσιακού επιπέδου, αποτελείται από οντότητες εξυπηρετητές οι οποίες, προσπαθούν να βρουν τον καταλληλότερο διαθέσιμο πόρο. Λόγω ομοιότητας στον τρόπο επιλογής του πόρου και άρα στον τρόπο λειτουργίας τους παρουσιάζονται όλες μαζί.

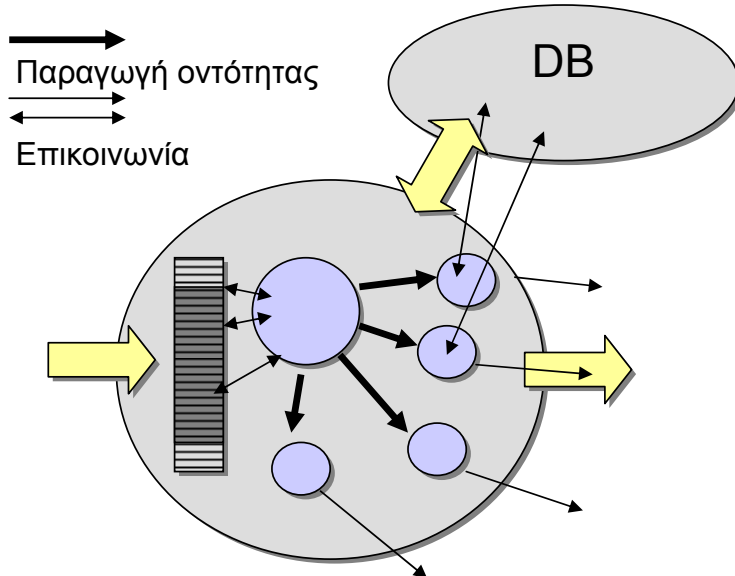
8.2.5.1 Λειτουργία

Η τυπική λειτουργία μιας οντότητας της κατηγορίας αυτής, είναι η υπόδειξη ενός πόρου κατόπιν αίτησης από κάποια άλλη οντότητα του γνωσιακού επιπέδου. Η αίτηση προς την οντότητα περιγράφει τα επιθυμητά χαρακτηριστικά του πόρου, ενώ ο πόρος ο οποίος τελικά υποδεικνύεται, είναι αυτός ο οποίος ταιριάζει καλύτερα προς την περιγραφή η οποία δόθηκε. Για την εύρεση του αναζητούμενου πόρου η οντότητα χρησιμοποιεί το αισθητήριο της βάσης δεδομένων, για την εύρεση των διαθέσιμων πόρων, τους οποίους ταξινομεί χρησιμοποιώντας μια συνάρτηση η οποία βαθμολογεί τον κάθε ένα ανάλογα με την καταλληλότητα του και

Comment [P1]:

τελικά διαλέγει αυτόν που συγκεντρώνει την υψηλότερη βαθμολογία, αφήνοντας τους υπόλοιπους ως εναλλακτικές λύσεις σε περίπτωση που το ζητά η αιτούσα οντότητα.

Αναλυτικότερα, κατά τη λειτουργία μιας τέτοιας οντότητας εξυπηρετητή, ένα νήμα ελέγχου παρακολουθεί με περιοδικό τρόπο το αισθητήριο της οντότητας. Κάθε φορά που μια νέα αίτηση αποστέλλεται στην οντότητα αυτή εναποτίθεται στον καταχωρητή γεγονότων του αισθητηρίου και τελικά γίνεται αντιληπτή από το νήμα το οποίο παρατηρεί το αισθητήριο. Με την αντίληψη μιας νέας αίτησης, το νήμα ελέγχου αυτό δημιουργεί μια νέα οντότητα, στην οποία αναθέτει τη διεργασία ανεύρεσης του καταλληλότερου πόρου ενώ αυτό συνεχίζει την παρατήρηση του αισθητηρίου για νέες εισερχόμενες αιτήσεις. Η λειτουργία μιας τέτοιας οντότητας απεικονίζεται στο σχήμα 8.4. Παρατηρούμε τη λειτουργία μιας οντότητας εύρεσης καταλληλότερου πόρου. Οι αιτήσεις άλλων οντοτήτων εισερχόμενες, εναποτίθενται με



Σχήμα 8.4 Δομή και λειτουργία μιας οντότητας εύρεσης καταλληλότερου πόρου

ταξινομημένο τρόπο στο αισθητήριο της οντότητας από νήμα ελέγχου, το οποίο δημιουργεί μία νέα οντότητα για κάθε αίτηση. Οι οντότητες οι οποίες δημιουργούνται επικοινωνούν με το αισθητήριο της βάσης δεδομένων έτσι ώστε να βρουν το κατάλληλο πόρο προς διάθεση, περιμένοντας την ελευθέρωση κάποιου πόρου, σε περίπτωση μη διαθεσιμότητας κανενός. Μετά την εύρεση του πόρου η κάθε οντότητα η οποία δημιουργήθηκε μεταβιβάζει το μήνυμα της διάθεσης του πόρου στην αντίστοιχη αιτούσα οντότητα και τερματίζει τη λειτουργία της. Στο σχήμα με μεγαλύτερα και ανοιχτόχρωμα βέλη απεικονίζεται η ροή πληροφορίας και μηνυμάτων σε γενικό επίπεδο, ενώ με μικρότερα βέλη απεικονίζονται οι ανταλλαγές μηνυμάτων μεταξύ των αντικειμένων ειδικότερα. Τα μαύρα και έντονα βέλη δείχνουν την δημιουργία μιας νέας οντότητας.

Ο μηχανισμός ο οποίος έχει αναπτυχθεί για τη διαχείριση των αιτήσεων των οντοτήτων εξυπηρετητών, επιτρέπει στις οντότητες οι οποίες έχουν αναλάβει την εργασία αναζήτησης πόρου, να εκτελούνται τόσο σε σειρά όσο και παράλληλα, γεγονός το οποίο καθορίζεται από το γνωσιακό επίπεδο της ίδιας της οντότητας. Στην περίπτωση κατά την οποία οι διεργασίες εκτελούνται σε σειρά, δηλαδή η κάθε μια αρχίζει να εκτελείται μετά το πέρας της προηγούμενης, οι αιτήσεις εναποτίθενται σε ουρά. Είναι δυνατόν, ωστόσο μια επείγουσα αίτηση να παρακάμψει την ουρά αυτή και να εκτελεστεί πιο πριν από οποιαδήποτε άλλη αίτηση, εφόσον η κρίσιμότητα της αίτησης δηλώνεται από την αιτούσα οντότητα. Εάν μάλιστα υπάρξουν παραπάνω από μια επείγουσες αιτήσεις ταυτόχρονα, αυτές εκτελούνται παράλληλα. Από την άλλη πλευρά στις οντότητες εξυπηρετητές οι οποίες εκτελούν τις αιτήσεις παράλληλα, είναι δυνατόν να ορίσουμε προτεραιότητες, όσον αφορά το ποσοστό της υπολογιστικής ισχύος το οποίο θα αντιστοιχεί σε κάθε αίτηση. Σε μια τέτοια περίπτωση και για

ευκολία χειρισμού των προτεραιοτήτων, υπάρχουν τρεις βαθμίδες προτεραιότητας οι οποίες μπορούν να ανατεθούν σε κάθε διεργασία, ανάλογα με τη δήλωση της αιτούσας οντότητας για την κρίσιμότητα της αίτησης.

Όπως αναφέραμε προηγουμένως, οι διαθέσιμοι πόροι ταξινομούνται βάση μιας συνάρτησης η οποία βαθμολογεί την καταλληλότητα του κάθε πόρου. Μια αίτηση αποτελείται από ένα σύνολο προδιαγραφών η κάθε μια από τις οποίες περιγράφει κάποιο χαρακτηριστικό του αναζητούμενου πόρου. Οι προδιαγραφές αυτές συνήθως δεν είναι της ίδιας βαρύτητας, ενώ υπάρχουν άλλες όπου η μη ικανοποίηση τους καθιστά τον πόρο μη κατάλληλο. Ένα παράδειγμα μιας τέτοιας περίπτωσης, είναι η επιλογή ιατρικού κέντρου όπου μπορεί η εγγύτητα του δίνει καλή βαθμολογία ωστόσο η έλλειψη της κατάλληλης κλινικής για την αντιμετώπιση του περιστατικού να καθιστά τον πόρο ακατάλληλο. Κατά συνέπεια χρειαζόμαστε να ταξινομήσουμε τις προδιαγραφές της αίτησης σε δύο κατηγορίες, αυτές οι οποίες πρέπει οπωσδήποτε να ικανοποιηθούν ώστε να επιλεγεί ο πόρος ως υποψήφιος και σε αυτές οι οποίες μεταξύ των υποψηφίων θα κρίνουν τον καταλληλότερο.

Η μέθοδος επιλογής του πόρου βρίσκει για κάθε προδιαγραφή της αίτησης η οποία πρέπει οπωσδήποτε να ικανοποιηθεί, ένα σύνολο πόρων από τους διαθέσιμους, όπως περιγράφονται από το αισθητήριο της βάσης δεδομένων. Στη συνέχεια υπολογίζει την τομή των συνόλων αυτών και εάν ο πληθάρθμος του συνόλου που θα προκύψει είναι μεγαλύτερος της μονάδος, τότε βαθμολογεί το κάθε στοιχείο του συνόλου βάση της συνάρτησης που έχει στη διάθεση της επιστρέφοντας αυτό με την υψηλότερη βαθμολογία, ή σε κάποιες περιπτώσεις και τα στοιχεία με τη δεύτερη και τρίτη καλύτερη βαθμολογία, ώστε να χρησιμοποιηθούν ως εναλλακτικές λύσεις, εάν αυτό απαιτείται από την αίτηση.

Η συνάρτηση βαθμολογίας υλοποιείται χρησιμοποιώντας το σταθμισμένο άθροισμα της εκπλήρωσης της κάθε προδιαγραφής ή της τιμής κάποιας μεταβλητής (πχ. χρονική εγγύτητα, πληρότητα θέσεων κ.τ.λ.). Στις οντότητες εξυπηρετητές οι οποίοι έχουν υλοποιηθεί, χρησιμοποιούμε το άθροισμα χωρίς στάθμιση των παραγόντων, τόσο διότι τις περισσότερες φορές αυτοί είναι λίγοι στον αριθμό, όσο και λόγω της έλλειψης κάποιας μελέτης η οποία να ορίζει τις τιμές βαρύτητας του κάθε παράγοντα. Παρ' όλα αυτά η δυνατότητα χρήσης σταθμισμένου αθροίσματος, διατηρείται από τις οντότητες εξυπηρετητές, τόσο για τη δυνατότητα επέκτασης του συστήματος στο μέλλον, όσο και για τη δημιουργία βιβλιοθήκης οντοτήτων γενικής χρήσεως οι οποίες μπορεί να βρουν εφαρμογή σε άλλα συστήματα.

Στη συνέχεια θα παρουσιάσουμε τις οντότητες εξυπηρετητές της κατηγορίας στην οποία αναφερόμαστε, σημειώνοντας πως ο τρόπος λειτουργίας τους είναι δυνατόν να μεταβληθεί για τις ανάγκες διαφορετικού ιατρικού πρωτοκόλλου.

8.2.5.2 Οντότητα εύρεσης ιατρικού κέντρου

Εκτός από τη διαθεσιμότητα του πόρου, η κυριότερη προδιαγραφή η οποία πρέπει να ικανοποιηθεί σε μια αίτηση αναζήτησης ιατρικού κέντρου, είναι αυτή της ύπαρξης κατάλληλης κλινικής για τον τύπο του περιστατικού (σύμφωνα με το πρωτόκολλο το οποίο ακολουθούμε). Δευτερεύον παράγοντας είναι η εγγύτητα του ιατρικού κέντρου, έτσι ώστε να μειωθεί ο οποίος απαιτείται για τη μεταφορά του ασθενή στον προορισμό του.

8.2.5.3 Οντότητα εύρεσης ιατρικού προσωπικού

Η οντότητα αυτή βρίσκει εφαρμογή κυρίως σε περιπτώσεις όπου το προσωπικό το οποίο επιβαίνει στην κινητή μονάδα χρειάζεται να επικοινωνήσει με κάποιο ιατρό ο οποίος βρίσκεται σε ιατρικό κέντρο (είτε μέσω τηλεσυνδιάσκεψης, είτε μέσω ασυρμάτου), ή για τις περιπτώσεις όπου πρέπει να ειδοποιηθεί το κατάλληλο ιατρικό προσωπικό ώστε να είναι σε ετοιμότητα για την άφιξη του περιστατικού στο ιατρικό κέντρο. Η αναζήτηση γίνεται με βάση τη διαθεσιμότητα του προσωπικού τη συγκεκριμένη χρονική στιγμή, ως πρωτεύον γνώρισμα προς αναζήτηση, και την ειδικότητα του ιατρού ως δευτερεύον.

8.2.5.4 Οντότητα εύρεσης κινητής μονάδας

Η καταλληλότητα της κινητής μονάδας, όσον αφορά τον εξοπλισμό που διαθέτει, είναι η κυρίαρχος προδιαγραφή η οποία καθορίζει την επιλογή της κινητής μονάδας, ενώ κατά δεύτερο λόγο λαμβάνεται υπ' όψιν η χρονική εγγύτητα της προς την τοποθεσία του περιστατικού. Συνήθως ο επιπλέον εξοπλισμός που μπορεί να διαθέτει μια κινητή μονάδα, είναι μια μονάδα εντατικής θεραπείας.

Ωστόσο λόγω της κρισιμότητας της απόφασης επιλογής της κινητής μονάδας, όσο και λόγω του ότι στη γενικότερη περίπτωση είναι δυνατόν να υπάρχει έλλειψη στη συγκεκριμένη κατηγορία πόρων, έχει δοθεί η δυνατότητα μερικών παραλλαγών στον τρόπο λειτουργίας της οντότητας αυτής. Έτσι ανάλογα με την κρισιμότητα του περιστατικού και δοθέντος ενός ορίου χρονικής ανοχής όσον αφορά την επιτρεπτή καθυστέρηση, είναι δυνατόν να προτιμηθεί κάποια κινητή μονάδα η οποία είναι χρονικά σε μακρινότερη απόσταση, λόγω του ότι το περιστατικό δεν απαιτεί ειδικό εξοπλισμό από την κινητή μονάδα, κάνοντας κατά αυτόν τον τρόπο οικονομία στη χρήση πόρων οι οποίοι είναι δυνατόν να απαιτηθούν στο προσεχές μέλλον.

Μια άλλη βελτιστοποίηση αφορά τις κινητές μονάδες οι οποίες είναι ήδη δεσμευμένες στην αντιμετώπιση κάποιου μη επείγοντος περιστατικού. Σε περίπτωση εκδήλωσης ενός κρίσιμου περιστατικού κοντά στην τοποθεσία της κινητής μονάδας, η οντότητα εύρεσης κινητής μονάδας είναι δυνατόν να επιλέξει αυτή τη μονάδα για την εξυπηρέτηση του περιστατικού (η δυνατότητα υποστήριξης μια τέτοιας επιλογής αναλύεται στην παρουσίαση της οντότητας αντιμετώπισης περιστατικών).

Αυτό που πρέπει να σημειωθεί είναι πως οι παραπάνω βελτιστοποιήσεις, ενεργοποιούνται μόνο σε περίπτωση επιλογής τους από το χρήστη, δηλαδή το ιατρικό προσωπικό.

8.2.6 Οντότητες υλοποίησης στόχων

Οι οντότητες οι οποίες υλοποιούν στόχους αποτελούν τις πιο σύνθετες από αυτές του γνωσιακού επιπέδου, καθώς και αυτές η συμπεριφορά των οποίων καθορίζει κατά το μεγαλύτερο λόγο αυτή του συστήματος. Η κάθε μια από τις οντότητες αυτές δρα συνεργαζόμενη με άλλες οντότητες του γνωσιακού επιπέδου, χρησιμοποιώντας αποτελέσματα υπολογισμών τους οποίους τους αναθέτει, ενώ είναι σχεδιασμένες με τέτοιο τρόπο, έτσι ώστε η συμπεριφορά τους να εξαρτάται από σχέδια δράσης τα οποία τους ανατίθενται, ενώ είναι δυνατή και η δυναμική προσθήκη νέων στόχων στο μέλλον.

Οι κυριότερες οντότητες της κατηγορίας αυτής είναι η οντότητα αντιμετώπισης περιστατικών, η οντότητα διαχείρισης κινητών μονάδων, η οντότητα η οποία διαχειρίζεται την επικοινωνία του προσωπικού της κινητής μονάδας με αυτό των ιατρικών κέντρων καθώς και αυτή η οποία διαχειρίζεται προγραμματισμένες μεταφορές ασθενών μεταξύ ιατρικών κέντρων.

8.2.6.1 Οντότητα αντιμετώπισης περιστατικών

Γενική περιγραφή

Στόχος της οντότητας αντιμετώπισης περιστατικών, είναι η παροχή εντολών προσανατολισμένων προς τη διαχείριση των πόρων του συστήματος, η εφαρμογή των οποίων αποσκοπεί στο διαχείριση των επεισοδίων με βέλτιστο τρόπο, όσον αφορά το χρόνο απόκρισης του προνοσοκομειακού συστήματος επείγουσας ιατρικής περίθαλψης και τη διάθεση των πόρων του.

Η λειτουργία την οντότητας αυτής μοιάζει αρκετά με αυτή κάποιας οντότητας εξυπηρετητή, όπου για κάθε νέο εισερχόμενο περιστατικό (το οποίο αποτελεί αίτηση εξυπηρέτησης του), δημιουργείται μια οντότητα η οποία αναλαμβάνει τη διαχείριση του. Η διαφοροποίηση με άλλες οντότητες εξυπηρετητές, έγκειται στο γεγονός του ότι το γνωσιακό επίπεδο της

οντότητας είναι δυνατόν, σε ορισμένες περιπτώσεις, να συνεχίσει να διαχειρίζεται τις οντότητες οι οποίες έχουν αναλάβει τα περιστατικά και μετά τη δημιουργία τους.

Σχεδίαση - Λειτουργία

Το αισθητήριο της οντότητας αυτής παρακολουθεί το αισθητήριο αντίληψης και παρακολουθήσης εξέλιξης περιστατικών του συστήματος για εκδήλωση νέων περιστατικών. Με την αντίληψη ενός νέου περιστατικού γίνεται προσπάθεια χαρακτηρισμού του και στη συνέχεια δημιουργείται μια οντότητα η οποία θα αναλάβει να εξυπηρετήσει το επεισόδιο, σύμφωνα με το ισχύον ιατρικό πρωτόκολλο. Η οντότητα κρατά μια δομή με τις διεργασίες αυτές οι οποίες έχουν ξεκινήσει, έτσι ώστε να είναι σε θέση να αναστείλει ή να τροποποιήσει τη λειτουργία κάποιων από αυτές εάν είναι απαραίτητο. Τα αντικείμενα δράσης της οντότητας, είναι δυνατόν να χρησιμοποιηθούν από τις διεργασίες αυτές οι οποίες έχουν δημιουργηθεί, έτσι ώστε να αποστείλουν μηνύματα ή εντολές διαχείρισης προς τους πόρους του συστήματος, ενέργεια η οποία επιτυγχάνεται προωθώντας τα προς τα αντικείμενα δράσης του ίδιου του συστήματος.

Παράλληλα η οντότητα συνεχίζει να παρατηρεί το αισθητήριο παρακολουθήσης περιστατικών για γεγονότα τα οποία αφορούν την εξέλιξη των περιστατικών, έτσι ώστε να είναι δυνατός ο έλεγχος της εφαρμογής των εντολών που έχουν δοθεί, καθώς και λήψη αποφάσεων οι οποίες βασίζονται στην εξέλιξη του επεισοδίου.

Στη συνέχεια θα δούμε ένα προς ένα τα βήματα τα οποία ακολουθεί η οντότητα προκειμένου να διαχειριστεί τα περιστατικά.

Αντίληψη - Χαρακτηρισμός επεισοδίου

Με την τεχνική της περιοδικής παρακολουθήσης των αισθητήρων του συστήματος, η οντότητα αυτή αντιλαμβάνεται την εκδήλωση ενός νέου περιστατικού σε χρόνο μικρότερο ή ίσο από την περίοδο παρακολουθήσης. Στο σύστημα μας, ο χαρακτηρισμός του επεισοδίου, γίνεται στην εφαρμογή του ασυρματιστή του προνοσοκομειακού συστήματος επείγουσας ιατρικής περίθαλψης, εφαρμογή η οποία συνεργάζεται με αυτή της διαχείρισης πόρων. Ωστόσο δεδομένης της υιοθέτησης πρωτοκόλλου χαρακτηρισμού περιστατικών μια τέτοια διαδικασία, είναι δυνατόν να γίνει και από το τμήμα διαχείρισης πόρων, δεδομένου ότι τα σημαντικότερα χαρακτηριστικά του επεισοδίου τα οποία μας ενδιαφέρουν, είναι η κρισιμότητα του περιστατικού η οποία φανερώνει το πόσο επείγον είναι αυτό καθώς και ο τύπος του περιστατικού ο οποίος υποδεικνύει τον τύπο της κινητής μονάδας ο οποίος θα πρέπει να χρησιμοποιηθεί.

Για την εξαγωγή των προηγούμενων συμπερασμάτων, χρησιμοποιήσαμε μελέτη η οποία έχει γίνει πάνω στην μοντελοποίηση ιατρικών επεισοδίων [54] και η οποία ταξινομεί τα περιστατικά σε τέσσερις κατηγορίες, χρησιμοποιώντας ως κριτήρια ταξινόμησης την κρισιμότητα και τύπο του περιστατικού. Δίνοντας μια δυαδική τιμή (αλήθεια / ψέμα) σε κάθε κριτήριο ταξινόμησης καταφέρνουμε να ταξινομήσουμε κατά απλό τρόπο τα περιστατικά. Οι τιμές αυτές για το κριτήριο κρισιμότητας του περιστατικού είναι το εάν το περιστατικό είναι επείγον ή όχι και για το κριτήριο του τύπου του περιστατικού το εάν η κινητή μονάδα η οποία θα εξυπηρετήσει το περιστατικό πρέπει να έχει μονάδα εντατικής θεραπείας ή όχι.

Ανάθεση επεισοδίου σε οντότητα εξυπηρέτησης του

Αμέσως μετά την αντίληψη και τον χαρακτηρισμό του επεισοδίου, η οντότητα αντιμετώπισης περιστατικών δημιουργεί μια νέα διεργασία / οντότητα η οποία θα αναλάβει να εξυπηρετήσει το περιστατικό. Κατά την δημιουργία της η οντότητα αυτή λαμβάνει ως παράμετρο το χαρακτηρισμό του περιστατικού καθώς και το σχέδιο δράσης το οποίο θα πρέπει να ακολουθήσει (η τεχνική ανάθεσης σχεδίων δράσης έχει περιγραφεί στη γενική παρουσίαση του γνωσιακού επιπέδου). Στη συνέχεια θα περιγράψουμε αναλυτικότερα τη λειτουργία της οντότητας αυτής η οποία θα αναλάβει τη διαχείριση του επεισοδίου.

Στο σημείο αυτό θα αναφερθούμε στην περίπτωση αλλαγής σχεδίου σε μια οντότητα η οποία ήδη εξυπηρετεί ένα περιστατικό. Μια τέτοια κατάσταση είναι δυνατόν να συναντηθεί στην

περίπτωση όπου παρουσιάζεται ένα επείγον περιστατικό το οποίο απαιτεί από την κινητή μονάδα η οποία θα το εξυπηρετήσει ειδικό εξοπλισμό και κατά την οποία δεν υπάρχει τέτοιος πόρος διαθέσιμος. Εάν υπάρχει κάποια κατάλληλη κινητή μονάδα η οποία τη συγκεκριμένη στιγμή εξυπηρετεί κάποιο άλλο, μη επείγον περιστατικό, θα μπορούσαμε να αναστείλουμε την εξυπηρέτηση του περιστατικού αυτού, στέλνοντας μια απλή κινητή μονάδα να συνεχίσει την εξυπηρέτηση του και διαθέτοντας την πρώτη στο νέο επεισόδιο το οποίο έχει προκύψει.

Δεδομένου του ότι όλες οι λειτουργίες αυτές (τόσο δηλαδή του ορισμού τέτοιων καταστάσεων, όσο και του τρόπου αντιμετώπισης τους), δεν πρόκειται να αποφασιστούν αυθαίρετα από το σύστημα, αλλά πάντα ακολουθώντας συγκεκριμένο ιατρικό πρωτόκολλο, αυτό το οποίο έχει σημασία να δούμε είναι το πως μπορούμε να υλοποιήσουμε την αλλαγή αυτή στο σχέδιο δράσης των οντοτήτων, ώστε το σύστημα να συνεχίσει να λειτουργεί αποδοτικά. Για να το επιτύχουμε αυτό θα πρέπει να επιλύσουμε δύο προβλήματα: την αναγνώριση μιας τέτοιας κατάστασης και τον τρόπο εφαρμογής της λύσης η οποία ορίζεται από τις ιατρικές οδηγίες αντιμετώπισης της. Η λύση η οποία επιλέξαμε να δώσουμε και στα δύο προβλήματα είναι η ίδια και συνίσταται στη χρήση σχεδίων δράσης.

Όσον αφορά την αναγνώριση συγκεκριμένων καταστάσεων, αυτή είναι δυνατόν να επιτευχθεί από το σχέδιο δράσης της ίδιας της οντότητας περιστατικών, όπου αυτό θα ορίζει το πότε πρέπει να συμβεί μια τέτοια διαδικασία. Συγκεκριμένα στο προηγούμενο παράδειγμα, δεν υπάρχει λόγος για ειδικό τρόπο αναγνώρισης της συγκεκριμένης κατάστασης, αφού είναι δυνατόν να περιγράψουμε στο σχέδιο δράσης το οποίο ακολουθεί η οντότητα το τι θα θέλαμε να συμβεί στην περίπτωση όπου: (i) δεν υπάρχει κατάλληλος πόρος διαθέσιμος (ii) ένας από τους υποψήφιους κατάλληλους πόρους είναι απασχολημένος στην εξυπηρέτηση άλλου μη επείγοντος περιστατικού. Οι δύο προηγούμενες συνθήκες είναι δυνατόν να περιγραφούν με if - then - else δηλώσεις τα οποία χρησιμοποιούμε για την υλοποίηση των δέντρων αποφάσεων των σχεδίων δράσης. Η παρουσίαση του προηγούμενου παραδείγματος ήταν αρκετά απλοποιημένη, μιας και μπορεί να σκεφτεί κανείς πως θα μπορούσε να υπάρχει κάποια κατάλληλη κινητή μονάδα η οποία σε πολύ μικρό χρονικό διάστημα να αναμένεται να συμπληρώσει την αποστολή της και η οποία θα μπορούσε να εξυπηρετήσει αυτή τελικά το περιστατικό. Στην πραγματικότητα, όταν υπολογίζει το σύστημα τη χρονική απόσταση μιας ήδη απασχολημένης κινητής μονάδας, λαμβάνει υπ' όψιν και τον αναμενόμενο χρόνο εκπλήρωσης της αποστολής της.

Από την άλλη πλευρά η αλλαγή του σχεδίου δράσης μιας διαδικασίας είναι πιο απλή. Αρκεί η διαδικασία να αποδεσμεύσει του πόρους τους οποίους χρησιμοποιεί (που στη συγκεκριμένη περίπτωση είναι η κινητή μονάδα) και να της ανατεθεί ένα νέο σχέδιο δράσης.

Υλοποίηση εντολών

Οι οντότητες οι οποίες έχουν αναλάβει τη διαχείριση των περιστατικών, αποστέλλουν μηνύματα στα αντικείμενα δράσης του συστήματος για την εκτέλεση τους. Η εξέλιξη των εντολών που έχουν δοθεί, συνεχίζει να παρατηρείται από τα αισθητήρια της οντότητας η οποία έχει ζητήσει την εκτέλεση τους, έτσι ώστε να επιβεβαιωθεί η επιτυχής εφαρμογή τους.

Διεργασίες αντιμετώπισης περιστατικών

Για κάθε νέο περιστατικό η οντότητα αντιμετώπισης περιστατικών, δημιουργεί μια διεργασία η οποία αναλαμβάνει να το διαχειριστεί. Η διεργασία αυτή, υλοποιείται με μια οντότητα στην οποία ανατίθεται το περιστατικό και η οποία είναι υπεύθυνη για το διαχείριση του, από την αρχή της εκδήλωσης του, έως τη μεταφορά του ασθενή στο κατάλληλο ιατρικό κέντρο. Η οντότητα παρακολουθεί μέσω του αισθητήριου της τα γεγονότα τα οποία αφορούν το συγκεκριμένο περιστατικό και δίνει εντολές για τη διαχείριση του βασισμένη στο σχέδιο δράσης το οποίο της έχει ανατεθεί. Παράλληλα παρατηρείται μέσω των αισθητηρίων του συστήματος, η εφαρμογή των εντολών και σε περίπτωση αποτυχίας της εφαρμογής τους ακολουθούνται οι εντολές οι οποίες υποδεικνύονται από το σχέδιο δράσης.

Το σχέδιο δράσης είναι δυνατόν να αλλάξει για λόγους που είδαμε στην προηγούμενη παράγραφο, όπως και επίσης είναι δυνατόν κάθε οντότητα να έχει διαφορετικό, ανάλογα με

τον τύπο του περιστατικού το οποίο εξυπηρετεί. Ένα τυπικό σχέδιο δράσης για την οντότητα αυτή είναι:

- Αντιστοίχιση της γεωγραφικά σημασιολογικής (GIS) πληροφορίας της τοποθεσίας του περιστατικού στον πλησιέστερο κόμβο του γράφου που περιγράφει το οδικό δίκτυο κυκλοφορίας.
- Επιλογή κινητής μονάδας με κριτήρια την καταλληλότητα και χρονική εγγύτητα της προς την τοποθεσία του περιστατικού. Σε περίπτωση ανεπάρκειας ή ακαταλληλότητας του πόρου, ο οποίος υποδείχθηκε από την οντότητα εύρεσης κινητής μονάδας, έλεγχος για την πιθανότητα αναστολής άλλης διεργασίας, αλλιώς αναμονή μέχρι την εύρεση του κατάλληλου πόρου.
- Υπολογισμός ταχύτερης χρονικά διαδρομής από την τοποθεσία της κινητής μονάδας σε αυτή του περιστατικού και παρακολούθηση της κίνησης της μονάδας προς τον προορισμό της.
- Εύρεση καταλληλότερου ιατρικού κέντρου στο οποίο θα προσκομιστεί ο ασθενής. Η υπό διεργασία αυτή είναι δυνατόν να εκτελείται κατά την κίνηση της μονάδας και μάλιστα με χαμηλή υπολογιστική προτεραιότητα.
- Υπολογισμός ταχύτερης χρονικά διαδρομής από την τοποθεσία του περιστατικού σε αυτή του ιατρικού κέντρου και παρακολούθηση της κίνησης της μονάδας προς τον προορισμό της.
- Εάν το περιστατικό είναι κρίσιμο εύρεση του κατάλληλου ιατρικού προσωπικού (σε σχέση με το περιστατικό) και ενημέρωση του για την αναμενόμενη ώρα άφιξης της κινητής μονάδας στο ιατρικό κέντρο.
- Ενημέρωση της βάσης δεδομένων και αρχειοθέτηση του επεισοδίου.

Σε όλα τα παραπάνω βήματα, είναι δυνατόν να υπάρξουν σφάλματα κατά την εφαρμογή της εντολής στο περιβάλλον (πχ. βλάβη κινητής μονάδας, αλλαγή προορισμού του ασθενή μετά από συνομιλία με ιατρικό προσωπικό). Από το σχέδιο δράσης απαιτείται η ύπαρξη πρόβλεψης για την αντιμετώπιση των σφαλμάτων αυτών.

8.2.6.2 Οντότητα προγραμματισμένων μεταφορών ασθενών μεταξύ ιατρικών κέντρων

Μέρος των δραστηριοτήτων ενός συστήματος προνοσοκομειακής ιατρικής περίθαλψης, αποτελεί η μεταφορά ασθενών μεταξύ ιατρικών κέντρων (πχ. για τις απαιτήσεις κάποιας ιατρικής εξέτασης ή θεραπείας). Δεδομένου ότι οι μεταφορές αυτές είναι προγραμματισμένες, θα θέλαμε να διαχειριστούμε τις κινητές μονάδες του συστήματος, ώστε να βρίσκονται εγκαίρως στο σημείο αφετηρίας. Επιπλέον η έγκαιρη γνώση της μεταφοράς του ασθενή, μας επιτρέπει να υπολογίσουμε τις ενέργειες οι οποίες πρέπει να γίνουν από πριν, σε στιγμές μειωμένου υπολογιστικού φόρτου για το σύστημα, έτσι ώστε να αποφευχθεί υπερφόρτωση του συστήματος. Πιο σημαντική όμως είναι η οικονομία στη διαχείριση των πόρων των κινητών μονάδων, η οποία μπορεί να επιτευχθεί αναθέτοντας στις μεταφορές αυτές κινητές μονάδες, οι οποίες δεν απαιτούνται από άλλα περιστατικά και οι οποίες είναι δυνατόν να βρίσκονται αρκετά μακριά, ώστε να μην προτιμούνταν εάν ο υπολογισμός εύρεσης κινητής μονάδας γινόταν, λίγο πριν την μεταφορά του ασθενούς.

Συνήθως η μεταφορά ενός ασθενούς από ένα ιατρικό κέντρο σε κάποιο άλλο δεν αποτελεί επείγον περιστατικό (στις περιπτώσεις που αποτελεί επείγον αντιμετωπίζεται από την οντότητα αντιμετώπισης περιστατικών), οπότε και μπορούμε να είμαστε λίγο πιο ελαστικοί όσον αφορά τον υπολογισμό της συντομότερης διαδρομής. Ο συμβιβασμός αυτός υιοθετείται, έτσι ώστε να είμαστε σε θέση να υπολογίσουμε τη διαδρομή μεταξύ των δύο ιατρικών κέντρων, βασιζόμενοι στις μέσες τιμές κυκλοφοριακού φόρτου, ώστε να μην είμαστε αναγκασμένοι να τον εκτελέσουμε λίγο πριν τη στιγμή της μεταφοράς.

Ο υπολογισμός της επιλογής κινητής μονάδας γίνεται αρκετό χρονικό διάστημα πιο πριν την μεταφορά του ασθενή, έτσι ώστε να υπάρχει χρόνος για τη μεταβίβαση της κινητής μονάδας από το σημείο το οποίο βρίσκεται τη συγκεκριμένη στιγμή στο ιατρικό κέντρο όπου βρίσκεται ο ασθενής. Ο χρόνος αυτός ορίζεται ως ο χρόνος ο οποίος απαιτείται για τη κάλυψη της μεγαλύτερης δυνατής απόστασης μεταξύ δύο κόμβων του γράφου δεδομένων των μέσων τιμών κυκλοφορίας (η οντότητα υπολογισμού διαδρομών είναι σε θέση να απαντήσει το ποιο είναι οι δύο ποιο απομακρυσμένοι χρονικά κόμβοι).

Καταλήγοντας, πρέπει να σημειωθεί πως η ανάλυση του συγκεκριμένου προβλήματος, αυτό δηλαδή της μεταφοράς ασθενών, δε βασίζεται σε κάποια μελέτη ή πρωτόκολλο, κάνοντας έτσι πιθανή την ύπαρξη καλύτερου σχεδίου δράσης από το προτεινόμενο. Η ανάπτυξη ενός καλύτερου τρόπου αντιμετώπισης του συγκεκριμένου προβλήματος, ίσως να έπρεπε να αποτελέσει μελλοντικό στόχο της ανάπτυξης του συστήματος.

8.2.6.3 Οντότητα επικοινωνιών κινητής μονάδας

Η λειτουργία της οντότητας αυτής είναι επικουρική προς αυτή της οντότητας αντιμετώπισης περιστατικών και η λειτουργία της πολλές φορές είναι προαιρετική. Στόχος της είναι η διαχείριση των επικοινωνιών οι οποίες μπορεί να λάβουν χώρα μεταξύ του προσωπικού της κινητής μονάδας και άλλου ιατρικού προσωπικού το οποίο βρίσκεται σε κάποιο ιατρικό κέντρο. Η οντότητα είναι δυνατόν να προσπαθήσει να καταστήσει δυνατή την επικοινωνία μεταξύ του ιατρικού προσωπικού και αυτού της κινητής μονάδας, οπότε και είναι στην κρίση του προσωπικού της κινητής μονάδας το εάν θα τη χρησιμοποιήσει ή όχι, ή να προσπαθήσει να υλοποιήσει την επικοινωνία μετά από απαίτηση του επιβαίνοντος προσωπικού. Για την επικοινωνία είναι δυνατόν να χρησιμοποιηθεί είτε τηλεσυνδιάσκεψη μέσω του φορητού υπολογιστή, είτε η ασύρματη επικοινωνία που διαθέτει η κινητή μονάδα και ίσως να προτιμάται από το προσωπικό της κινητής μονάδας. Μέρος της λειτουργίας της οντότητας αυτής είναι και η εύρεση του κατάλληλου προσωπικού το οποίο θα μετέχει στην επικοινωνία.

Για να λειτουργήσει η οντότητα πρέπει να έχουν οριστεί συγκεκριμένα ιατρικά γεγονότα ή ακολουθίες γεγονότων τα οποία να αναγνωρίζει η οντότητα, η οποία παρακολουθεί τα ιατρικά γεγονότα τα οποία λαμβάνουν χώρα κατά τη μεταβίβαση του ασθενή στο ιατρικό κέντρο, μέσω του αισθητηρίου της. Η παρακολούθηση των γεγονότων αυτών από το αισθητήριο της οντότητας γίνεται με περιοδική προσπέλαση του καταχωρητή γεγονότων του αισθητηρίου παρακολούθησης της εξέλιξης γεγονότων του συστήματος και επιλογή αυτών τα οποία αφορούν το συγκεκριμένο περιστατικό.

Κατά την εκκίνηση του περιστατικού, ή σε οποία άλλη χρονική στιγμή χαρακτηρίζεται το περιστατικό, η οντότητα κάνει μια ερώτηση στο αισθητήριο της βάσης δεδομένων, έτσι ώστε να βρει το κατάλληλο και διαθέσιμο προσωπικό το οποίο θα μπορούσε να μετέχει στην επικοινωνία. Εάν ο χαρακτηρισμός του περιστατικού αλλάξει, πχ. κατά την επί τόπου διάγνωση, η ερώτηση αυτή επαναλαμβάνεται με τα νέα δεδομένα. Όταν συμβεί κάποιο γεγονός το οποίο προτρέπει, σύμφωνα με τους κανόνες με τους οποίους έχουν δοθεί στην οντότητα και επιβεβαιωθεί και από τις δύο πλευρές η διάθεση ύπαρξης επικοινωνίας, ή όταν ζητηθεί από το επιβαίνων στην κινητή μονάδα προσωπικό, ειδοποιούνται (αυτό βέβαια δουλεύει στην εξομοίωση, δεν είναι ακόμη ακριβές με ποιο τρόπο θα ειδοποιούνται στην πραγματικότητα), ώστε να μετέχουν στην επικοινωνία.

Ενδιαφέρουσα εξέλιξη της οντότητας αυτής θα ήταν η ανάπτυξη μεθόδου μηχανικής μάθησης η οποία από σύνολο παραδειγμάτων ιατρικών γεγονότων και αιτήσεων επικοινωνίας, να έβγαζε συμπεράσματα για το πότε είναι επιθυμητή η ύπαρξη επικοινωνίας.

8.2.6.4 Οντότητα διαχείρισης κινητών μονάδων

Κύριος στόχος της λειτουργίας της οντότητας διαχείρισης κινητών μονάδων, είναι η διευθέτηση τους στο οδικό δίκτυο σε τέτοιες θέσεις ώστε να είναι δυνατή η γρήγορη πρόσβαση τους σε περιοχές υψηλής συχνότητας εμφάνισης περιστατικών και άρα τελικά μείωση του χρόνου απόκρισης του συστήματος. Παράλληλα δευτερεύων στόχος της οντότητας, είναι η κάλυψη άλλων αναγκών διαχείρισης κινητών μονάδων, όπως πχ. η εγκατάσταση μιας κινητής μονάδας σε κάθε νοσοκομείο ή η κάλυψη ειδικών γεγονότων. Η διαχείριση των κινητών μονάδων, γίνεται πάντα σε συνεργασία με την οντότητα αντιμετώπισης περιστατικών και μεταφοράς ασθενών, οι οποίες έχουν μεγαλύτερη προτεραιότητα στη δέσμευση και χρήση των πόρων των κινητών μονάδων.

Για την αποτελεσματική λειτουργία της οντότητας, απαιτείται η εύρεση των περιοχών υψηλής εμφάνισης περιστατικών και άρα η ανάπτυξη αλγόριθμου ο οποίος βάση συνόλου παραδειγμάτων τα οποία έχουν συμβεί να βρίσκει τις περιοχές που μας ενδιαφέρουν.

Μάλιστα οι περιοχές αυτές είναι δυνατόν να αλλάζουν κατά χρονικά διαστήματα ή εποχές του χρόνου, κάνοντας έτσι δυσκολότερη την εκτίμησή τους. Ο αλγόριθμος ο οποίος αναπτύχθηκε, βασίζεται σε συνδυασμό τεχνικών της περιοχής της μηχανικής μάθησης και περιγράφεται σε επόμενη ενότητα.

Χρησιμοποιώντας τον αλγόριθμο που αναπτύχθηκε, έχουμε ως δεδομένο κάποιες περιοχές στις οποίες είναι πιθανότερο να εμφανιστεί κάποιο περιστατικό από ότι σε άλλες. Η οντότητα διαχείρισης κινητών μονάδων, θα φροντίσει, έτσι ώστε να δώσει εντολή προς τις διαθέσιμες προς περιπολία κινητές μονάδες να μεταβούν στις συγκεκριμένες περιοχές (η σε περίπτωση εκπαιδευτικής χρήσης του συστήματος να υποδείξει στο χρήστη τις περιοχές αυτές). Η επιλογή των κινητών μονάδων, είναι δυνατόν να γίνει με κριτήρια τα οποία καθορίζονται από τις ανάγκες του εκάστοτε συστήματος επείγουσας προνοσοκομειακής ιατρικής περίθαλψης, με την έννοια του ότι μπορεί να τίθενται περιορισμοί όσον αφορά το μέγιστο επιτρεπτό αριθμό κινητών μονάδων που μπορεί να βρίσκονται σε περιπολία ή τη διατήρηση αριθμού κινητών μονάδων κοντά σε ιατρικά κέντρα για άλλες ανάγκες.

Μια περιοχή υψηλής συχνότητας παρουσίασης περιστατικών, είναι δυνατόν να οριστεί ως ένα σύνολο κόμβων του γράφου ο οποίος αναπαριστά το οδικό δίκτυο. Πιο συγκεκριμένα το σύνολο των κόμβων αυτών μαζί με τις ακμές οι οποίες του συνδέουν, αποτελεί ένα συνδεδεμένο υπογράφο του αρχικού γράφου. Έτσι μας μένει να βρούμε ποιος είναι ο καταλληλότερος τρόπος κίνησης μιας κινητής μονάδας, στο συγκεκριμένο υπογράφο, έτσι ώστε να ελαχιστοποιείται ο χρόνος πρόσβασης σε οποιοδήποτε κόμβο του. Για να το κάνουμε αυτό είναι δυνατόν να χρησιμοποιήσουμε έναν αριθμό από τεχνικές οι οποίες αναπτύχθηκαν για το σκοπό αυτό και οι οποίες περιγράφονται στη συνέχεια.

Τυχαία κίνηση από κόμβο σε κόμβο: Επιλέγουμε με τυχαίο τρόπο κάθε φορά, ένα κόμβο στον οποίο μεταβαίνει η κινητή μονάδα. Παραλλαγή αυτής της μεθόδου, είναι η αριθμηση (έστω και με τυχαίο τρόπο), των κόμβων του υπογράφου και η διατήρηση σταθερής περιπολίας ακολουθώντας τους κόμβους με αύξουσα σειρά. Το πλεονέκτημα της παραλλαγής είναι ότι δεν απαιτείται ο εκ νέου κάθε φορά υπολογισμός των διαδρομών που πρέπει να ακολουθήσει η κινητή μονάδα που επιτελεί την περιπολία.

Εύρεση κέντρου βάρους του υπογράφου: Υπολογίζοντας το κυρτό περίβλημα των κόμβων του υπογράφου, είναι δυνατόν να βρούμε το «κέντρο βάρους», του πολυγώνου το οποίο σχηματίζεται. Επιλέγοντας ένα ποσοστό (έστω 10%) των κόμβων του υπογράφου στο οποίο θέλουμε να γίνεται η περιπολία, μπορούμε να βρούμε τους κόμβους οι οποίοι απέχουν λιγότερο από το κέντρο βάρους και να επιτελείται η περιπολία μεταξύ των κόμβων αυτών. Η μέθοδος αυτή έχει το μειονέκτημα του ότι είναι δυνατόν να βρεθούν γράφοι στους οποίους το γεωμετρικό κέντρο βάρους να μην ισαπέχει (ούτε κατά προσέγγιση) από τους κόμβους του γράφου.

Υπολογισμός κόμβων οι οποίοι ισαπέχουν γεωμετρικά από τους υπόλοιπους: Εφαρμόζοντας τη μέθοδο αυτή, υπολογίζουμε το καρτεσιανό γινόμενο των αποστάσεων (όσον αφορά την ευκλείδεια απόσταση) όλων των κόμβων μεταξύ τους και στη συνέχεια βρίσκουμε το μέσο όρο των τιμών αυτών. Επιλέγοντας ένα ποσοστό των κόμβων στο οποίο θέλουμε να λαμβάνει χώρα η περιπολία, βρίσκουμε τους κόμβους των οποίων ο μέσος όρος της απόστασης είναι πιο κοντά στο μέσο όρο που υπολογίσαμε προηγούμενα και θέτουμε την περιπολία συμβαίνει μεταξύ των κόμβων αυτών.

Υπολογισμός κόμβων οι οποίοι ισαπέχουν χρονικά από τους υπόλοιπους: Εφαρμόζοντας τη μέθοδο αυτή, υπολογίζουμε το καρτεσιανό γινόμενο των αποστάσεων (όσον αφορά τα βάρη των ακμών) όλων των κόμβων μεταξύ τους και στη συνέχεια βρίσκουμε το μέσο όρο των τιμών αυτών. Επιλέγοντας πάλι ένα ποσοστό των κόμβων στο οποίο θέλουμε να λαμβάνει χώρα η περιπολία, βρίσκουμε τους κόμβους των οποίων ο μέσος όρος της απόστασης είναι πιο κοντά στο μέσο όρο που υπολογίσαμε προηγούμενα και θέτουμε την περιπολία συμβαίνει μεταξύ των κόμβων αυτών.

Η οντότητα θα επιλέξει τη μέθοδο την οποία θα εφαρμόσει σε κάθε περιοχή χρησιμοποιώντας μια οντότητα μάθησης πάνω σε τυχαία δεδομένα τα οποία θα δοκιμάσει στον προσομοιωτή. Αναλυτικότερα χρησιμοποιώντας ως μέτρο απόδοσης της κάθε μεθόδου, το χρόνο ο οποίος

απαιτείται για να φτάσει η κινητή μονάδα από το σημείο στο οποίο βρίσκεται στον σημείο του επεισοδίου, η οντότητα μάθησης θα επιλέξει την πειραματικά καταλληλότερη μέθοδο για τη συγκεκριμένη περιοχή.

Μια άλλη διάσταση του προβλήματος, λαμβάνει υπ' όψιν το κόστος συνεχούς κίνησης της κινητής μονάδας σε περιπολία. Σε αυτή την περίπτωση είναι δυνατόν να επιλέξουμε τον κόμβο του οποίου ο μέσος όρος αποστάσεων από τους άλλους κόμβους, είναι πιο κοντά στο μέσο όρο του καρτεσιανού γινομένου όλων των αποστάσεων και τοποθετούμε την κινητή μονάδα σε αυτόν.

8.2.6.5 Οντότητα εύρεσης χρονικά συντομότερης διαδρομής

Η οντότητα εύρεσης συντομότερης διαδρομής ανήκει στην κατηγορία των οντοτήτων εξυπηρετητών. Τυπική λειτουργία της αποτελεί η εύρεση της συντομότερης διαδρομής μεταξύ δύο κόμβων του γράφου ο οποίος περιγράφει το οδικό δίκτυο κατόπιν αίτησης άλλης οντότητας του γνωσιακού επιπέδου.

Όπως σε όλες οντότητες εξυπηρετητές, οι εισερχόμενες αιτήσεις γίνονται αντιληπτές από το αισθητήριο της οντότητας, η οποία για κάθε αίτηση δημιουργεί μια διεργασία η οποία αναλαμβάνει την εξυπηρέτηση της. Οι αιτήσεις εξυπηρετούνται παράλληλα, όμως στις επείγουσες αιτήσεις δίνεται μεγαλύτερο ποσοστό υπολογιστικού χρόνου.

Αρχικά για την εύρεση συντομότερης διαδρομής μεταξύ δύο κόμβων του γράφου είχε υιοθετηθεί η λύση του εκ των προτέρων υπολογισμού της συντομότερης διαδρομής, όλων δυνατών συνδυασμών μεταξύ των κόμβων του γράφου (δηλαδή του καρτεσιανού γινόμενου) με τον αλγόριθμο του Dijkstra [48], και στη συνέχεια η φύλαξη τους σε κατάλληλη δομή. Στη συνέχεια για κάθε εισερχόμενη αίτηση αναζητούσαμε την αντίστοιχη διαδρομή στη δομή μας και την επιστρέψαμε. Η λύση αυτή είχε τα εξής προβλήματα:

- Για πολύ μεγάλους γράφους χρειαζόμασταν μεγάλη ποσότητα μνήμης τόσο για την εκτέλεση του αλγορίθμου όσο και για την αποθήκευση των αποτελεσμάτων.
- Στη γενική περίπτωση θα θέλαμε να χρησιμοποιούμε βάρη με μεταβλητό βάρος, έτσι ώστε να περιγράψουμε καλύτερα το φόρτο λόγω κυκλοφορία που έχει κάποια διαδρομή.

Η οντότητα εύρεσης συντομότερων διαδρομών χρησιμοποιεί μια αφηρημένη περιγραφή του γράφου έτσι ώστε να μην λαμβάνει υπ' όψιν όλους τους κόμβους του γράφου κατά την εφαρμογή του αλγορίθμου εύρεσης συντομότερης διαδρομής. Κάτι τέτοιο γίνεται εφικτό με την παρατήρηση χαρακτηριστικών του συγκεκριμένου γράφου, όπως πχ. δενδρικές δομές. Η εύρεση της διαδρομής γίνεται συνδυάζοντας τον αλγόριθμο του Dijkstra καθώς και άλλες τεχνικές οικονομικότερης μετάβασης μεταξύ δύο κόμβων ενός δέντρου.

8.3 Αντικείμενα δράσης

Η ενότητα αυτή περιγράφει το σχεδιασμό και λειτουργία των αντικειμένων δράσης του συστήματος. Τα αντικείμενα δράσης του συστήματος αποτελούν το μέσο με το οποίο το σύστημα δρα πάνω στο περιβάλλον εκτελώντας εντολές προερχόμενες από τις οντότητες του γνωσιακού επιπέδου.

Τα αντικείμενα δράσης του συστήματος αποτελούνται από τις κινητές μονάδες, το ιατρικό προσωπικό και τα ιατρικά κέντρα στα οποία μεταβιβάζονται οι ασθενείς (αν και τα τελευταία παίζουν κυρίως παθητικό ρόλο και ταξινομούνται ταυτόχρονα κάτω από του πόρους του συστήματος). Ως αντικείμενα δράσης του συστήματος μπορούν να χαρακτηριστούν και τα αντικείμενα που υλοποιούν τις επικοινωνίες του συστήματος με τον πραγματικό κόσμο και άλλα πληροφοριακά συστήματα (όπως η βάση δεδομένων του συστήματος επείγουσας προνοσοκομειακής ιατρικής περίθαλψης). Αποτελούν αυτόνομες οντότητες με την ίδια έννοια όπως έχουμε περιγράψει προηγούμενα, ωστόσο παρουσιάζουν μια ιδιαιτερότητα στη συγκεκριμένη εφαρμογή, αφού δεν άπτονται του άμεσου ελέγχου του συστήματος, αφού μάλιστα αποτελούνται και από ανθρώπινο προσωπικό.

Γενικά τα αντικείμενα δράσης συλλέγουν εντολές οι οποίες προέρχονται από το γνωσιακό επίπεδο και προσπαθούν να τις εκτελέσουν. Η εξέλιξη της προσπάθειας εκτέλεσης της εντολής παρατηρείται από τα αισθητήρια του συστήματος, και σε περίπτωση αποτυχίας αποστέλλονται διορθωτικές εντολές.

Η αντιμετώπιση των οντοτήτων δράσης από τις υπόλοιπες (ουσιαστικά αυτές του γνωσιακού επιπέδου), είναι παρόμοια με αυτή όλων των υπολοίπων οντοτήτων. Οι οντότητες του γνωσιακού επιπέδου μεταδίδουν εντολές στις οντότητες δράσης, τις οποίες αντιλαμβάνονται αυτές και τις μεταβιβάζουν στο γνωσιακό επίπεδο του. Οι εντολές αυτές, αποτελούν ουσιαστικά το σχέδιο δράσης των οντοτήτων αυτών.

Μια τυπική οντότητα δράσης, έχει ένα αισθητήριο το οποίο χρησιμεύει στην αντίληψη νέων εισερχόμενων εντολών από το γνωσιακό επίπεδο του συστήματος. Το γνωσιακό επίπεδο μιας οντότητας δράσης αποτελείται από έναν αποταμιευτή εντολών όπου εναποτίθενται οι εισερχόμενες εντολές. Το δρων τμήμα μια οντότητας δράσης, είναι το ίδιο το υλικό αντικείμενο το οποίο εκτελεί τις εντολές. Φυσικά σε μια τέτοια μοντελοποίηση δεν εμπίπτουν όλα τα αντικείμενα δράσης του συστήματος (όπως φυσικά οι ανθρώπινοι πόροι), αντιμετωπίζονται όμως έτσι από το γνωσιακό επίπεδο, στην προσπάθεια επικοινωνίας του με αυτά.

Όπως είπαμε οι οντότητες δράσης δέχονται εντολές τις οποίες εναποθέτουν στο γνωσιακό τους επίπεδο και τις εκτελούν σειριακά. Ωστόσο είναι δυνατόν το γνωσιακό επίπεδο του συστήματος, να δώσει εντολή μεγαλύτερης προτεραιότητας η οποία να εκτελεστεί πρώτα, η να διαγράψει εντολές οι οποίες είχαν δοθεί και δεν έχουν εκτελεστεί ακόμα (μια τέτοια περίπτωση είναι η ανακατεύθυνση μια εμπλεκόμενης σε περιστατικό κινητής μονάδας για να παραλάβει ένα πιο επείγον περιστατικό).

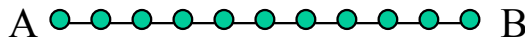
Κατά κανόνα τα αντικείμενα δράσης δεν εκτελούν ενέργειες ταυτόχρονα, ενώ είναι δύσκολο να μοντελοποιηθούν αφού είναι αντικείμενα του πραγματικού κόσμου. Σε περίπτωση επικουρικής χρήσης του συστήματος δεν υπάρχουν στη ουσία αντικείμενα δράσης, αλλά οι εντολές του γνωσιακού επιπέδου, γνωστοποιούνται σε αυτόν μέσω της διεπιφάνειας χρήσης.

9 Μεθοδολογία απλοποίησης γράφου και εύρεσης συντομότερων διαδρομών σε αυτόν

9.1 Εισαγωγή

Η ενότητα αυτή περιγράφει την εργασία η οποία έχει γίνει προκειμένου να υπολογίζουμε διαδρομές για τις κινητές μονάδες, οι οποίες εξυπηρετούν περιστατικά σε ένα σύστημα επείγουσας προνοσοκομειακής ιατρικής περίθαλψης. Στόχος μας είναι η επίτευξη της ικανότητας του συστήματος να αποκριθεί σε αιτήσεις για εύρεση συντομότερων διαδρομών σε ένα μεγάλο γράφο, ο οποίος αναπαριστά το οδικό δίκτυο, του οποίου τα βάρη των ακμών είναι μεταβλητά μιας εξαρτώνται και από άλλους παράγοντες εκτός από την απόσταση, όπως η πυκνότητα κυκλοφορίας. Η προσπάθεια η οποία έγινε, επικεντρώνεται σε προεπεξεργασία του γράφου, προκειμένου να βρούμε ιδιότητες του, τις οποίες θα χρησιμοποιήσουμε όταν υπολογίζουμε τις ζητούμενες διαδρομές. Στη συνέχεια θα παρουσιάσουμε τη μεθοδολογία την οποία αναπτύξαμε προκειμένου να χειριζόμαστε υπολογισμό συντομότερων διαδρομών, σε μεγάλους γράφους, οι οποίοι μοντελοποιούν δρόμους και όπου τα βάρη των ακμών του γράφου, είναι μεταβλητά.

Θα περιγράψουμε πρώτα τη φύση των γράφων με τους οποίους θα δουλέψουμε, οι οποίοι όπως αναφέραμε μοντελοποιούν οδικά δίκτυα. Ακόμα και εάν υπάρχει ένα μόνος δρόμος μεταξύ της τοποθεσίας A και της B, ο γράφος που θα χρησιμοποιούμε δεν θα έχει μια απλή ακμή η οποία ενώνει τον κόμβο A με τον B. Αντίθετα, ο γράφος θα έχει ένα αριθμό από κόμβους μεταξύ A και B προκειμένου να περιγράψουμε τις διαφορετικές τοποθεσίες από τις οποίες περνά ο δρόμος (βλ Σχήμα 9.1). Αυτό εξάλλου, είναι μια επιθυμητή ιδιότητα του γράφου, δεδομένου ότι θα θέλαμε να είμαστε ακριβείς σχετικά με την τοποθεσία στην οποία συνέβη το περιστατικό. Επιπλέον θέλουμε να διατηρήσουμε τη δομή αυτή λόγω των γεωγραφικής (GIS) πληροφορίας η οποία είναι συνδεδεμένη με κάθε κόμβο και η οποία στην ουσία είναι αυτή η οποία καθιστά εφικτή την εύρεση του αντίστοιχου κόμβου, από το όνομα της τοποθεσίας.



Σχήμα 9.1 Η θέση A και B ενώνονται με ένα δρόμο ο οποίος διασχίζει αριθμό θέσεων, μοντελοποιημένοι ως κόμβοι μεταξύ των αντίστοιχων κόμβων A και B.

Μια πρώτη προσέγγιση στη λύση του προβλήματος ήταν να προϋπολογίσουμε το καρτεσιανό γινόμενο των ζευγών των κόμβων του γράφου και στη συνέχεια να υπολογίσουμε, τις αντίστοιχες διαδρομές μεταξύ των ζευγών κόμβων αυτών. Κατά αυτό τον τρόπο θα μπορούσαμε να κρατήσουμε τις διαδρομές αυτές σε κάποια δομή δεδομένων και να βρίσκουμε την κατάλληλη κάθε φορά που εισερχόταν μια νέα αίτηση. Ακόμη όμως και εάν υπολογίζαμε το μέσο βάρος όλων των ακμών του γράφου για συγκεκριμένες περιόδους του χρόνου, ή της ημέρας (καταναλώνοντας μεγάλες ποσότητες αποθηκευτικού χώρου στη δευτερεύουσα μνήμη), η λύση θα ήταν πάλι μη αποδεκτή. Ο λόγος είναι ότι εάν για κάποιο απρόβλεπτο λόγο, ένας δρόμος είναι αποκλεισμένος από το κυκλοφοριακό δίκτυο, ή ένας άλλος έχει ασυνήθιστα ελαφρά κυκλοφορία, τότε θα υπολογίζαμε λάθος μονοπάτι στην πρώτη περίπτωση, ενώ θα χάναμε την ευκαιρία να υπολογίσουμε ένα πολύ γρήγορο μονοπάτι στη δεύτερη.

Οι μέθοδοι οι οποίες προτείνονται, δεν βελτιστοποιούν τον υπολογισμό του συντομότερου μονοπατιού για κάθε γράφο, αλλά αποδίδουν καλύτερα όταν χρησιμοποιούνται σε μεγάλους γράφους οι οποίοι περιγράφουν οδικά δίκτυα, κυρίως διότι βασίζονται στη δομή τους. Η πρώτη απλοποίηση επικεντρώνεται στην απλοποίηση ακμών σαν και αυτή που είδαμε στο σχήμα ένα, και στον υπολογισμό της συντομότερης διαδρομής αφότου έχει γίνει η απλοποίηση. Η δεύτερη προσπάθει, να βρει απομονωμένες περιοχές σε ένα γράφο, δημιουργώντας έτσι ένα ευρετήριο περιοχών του γράφου. Είναι σημαντικό να πούμε πως ο απλοποιημένος γράφος υπολογίζεται μόνο μια φορά, δεδομένου ότι δεν επηρεάζεται από

μεταβολές στα βάρη των ακμών του γράφου. Θα δούμε επίσης πως ο υπολογισμός συντομότερης διαδρομής μεταξύ δύο κόμβων, με τη χρήση του απλοποιημένου γράφου, είναι διεργασία η οποία αποτελείται από παραπάνω από μια υπό διεργασίες. Οι χρήσεις αυτή της παρατήρησης είναι ότι:

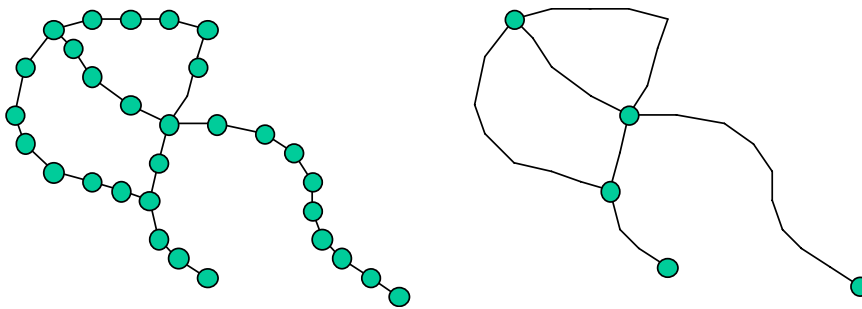
- Μπορούμε να υπολογίζουμε πρώτα το πιο επείγον τμήμα της διαδρομής και να το στέλνουμε στην κινητή μονάδα η οποία θα είναι σε θέση να αρχίσει να κινείται προς τον προορισμό της όσο υπολογίζεται το υπόλοιπο τμήμα της διαδρομής
- Τα τμήματα αυτά είναι δυνατόν να υπολογιστούν παράλληλα επιταχύνοντας έτσι την όλη διαδικασία εύρεσης της διαδρομής μεταξύ δύο σημείων

9.2 Πρώτη μέθοδος

Σε αυτή την παράγραφο θα περιγράψουμε την πρώτη μέθοδο απλοποίησης γράφου που χρησιμοποιούμε και τη χρήση του αλγορίθμου του Dijkstra προκειμένου να υπολογίσουμε τις ζητούμενες διαδρομές στο γράφο, μετά την απλοποίηση.

9.2.1 Απλοποίηση γράφου

Στόχος του αλγορίθμου είναι η αντικατάσταση τμημάτων του γράφου με απλούστερα, έτσι ώστε όταν υπολογίζουμε το συντομότερο μονοπάτι από έναν κόμβο σε έναν άλλο, να δουλεύουμε με ένα γράφο λιγότερων κόμβων από τον αρχικό, επιταχύνοντας έτσι τη διαδικασία υπολογισμού. Ο αλγόριθμος βασίζεται στην παρατήρηση του ότι μπορούμε να αντικαταστήσουμε ένα μονοπάτι από συνδεδεμένους κόμβους με μία απλή ακμή, εάν δεν υπάρχουν "σταυροδρόμια" στο μονοπάτι αυτό (όπως συμβαίνει στο σχήμα 1) χωρίς συνεπείς στα μονοπάτια τα οποία διαλέγουμε. Ένα τέτοιο παράδειγμα απλοποίησης φαίνεται στο σχήμα 9.2.



Σχήμα 9.2 Το επιθυμητό αποτέλεσμα της απλοποίησης ενός γράφου με την πρώτη μεθοδολογία

Συγκεκριμένα μπορούμε να ορίσουμε τέτοια μονοπάτια ως υπογράφοι του αρχικού οι οποίοι έχουν συγκεκριμένες ιδιότητες. Ένας υπογράφος γ του γράφου Γ , ο οποίος περιέχει $N = \{v_0, v_1, v_2, \dots, v_N\}$, κόμβους και $M = \{e_0, e_1, e_2, \dots, e_N\}$ ακμές, θα είναι ένας από τους υπογράφους προς απλοποίηση εάν:

- οι κόμβοι v_k , με $k = 1, 2, \dots, N-1$ έχουν ακριβώς δύο γειτονικούς κόμβους
- οι κόμβοι v_0 και v_N έχουν τουλάχιστον ένα γειτονικό κόμβο
- η ακμή e_k συνδέει τους κόμβους v_{k-1} και v_k

Θα θέλαμε να αντικαταστήσουμε τον υπογράφο γ με δύο κόμβους και μια ακμή η οποία συνδέει τους κόμβους αυτούς και έχει βάρος, ίσο με το άθροισμα των βαρών των ακμών του γράφου γ .

Δεδομένου ενός αρχικού κόμβου και μιας από τις ακμές του (δηλαδή μια ακμή του γράφου της οποίας το ένα άκρο είναι ο δοθείς κόμβος) ο αλγόριθμος θα δημιουργήσει ένα νέο γράφο του οποίου οι κόμβοι αποτελούνται από μόνο κόμβους οι οποίοι αντιστοιχούν στα σταυροδρόμια του αρχικού γράφου, ή κόμβους ή οποίοι έχουν μόνο ένα γείτονα. Για να αποφύγουμε να επιλέξουμε ως κόμβο εκκίνησης του αλγορίθμου, κάποιον ο οποίος δεν αντιστοιχεί σε σταυροδρόμι ή ακραίο κόμβο του γράφου (δηλαδή κόμβο ο οποίος έχει μόνο ένα γείτονα), θα επιλέξουμε κάποιον από του κόμβους του γράφου ο οποίος έχει παραπάνω από δύο γείτονες. Από τον κόμβο εκκίνησης τον οποίο επιλέξαμε και μια ακμή του θα ακολουθήσουμε το μονοπάτι έως ότου, είτε βρούμε ένα κόμβο ο οποίος έχει παραπάνω από δύο γείτονες οπότε και θα καλέσουμε αναδρομικά τον αλγόριθμο για κάθε ακμή την οποία δεν έχουμε ήδη επισκεφτεί, είτε βρούμε ένα κόμβο με ένα μόνο γείτονα οπότε και θα τερματίσουμε την τρέχουσα αναδρομή. Σε ψευδοκώδικα ο αλγόριθμος μπορεί να περιγραφεί ως εξής:

```

Graph execute()
{
    let v be any graph vertex with more than two neighbors
    let e be one of v's edges
    simplicateGraph(v,e);
}

void simplicateGraph(Vertex v, Edge e)
{
    Branch branch = followBranch(v,e);
    brances.addElement(branch);

    ShapeNode nextNode = branch.end;

    for (all of unmarked nextNode's edges)
        simplicateGraph(nextNode, next unmarked edge);
}

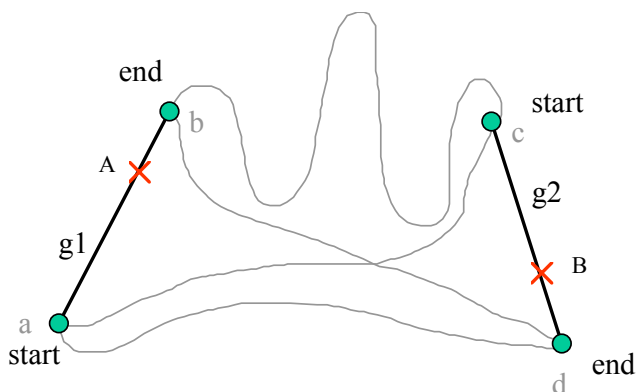
```

Στην υπορουτίνα execute() επιλέγουμε πρώτα το αρχικό κόμβο v ο οποίος είναι ένα κόμβος με παραπάνω από δύο γείτονες και ονομάζουμε e μια από τις ακμές οι οποίες πρόσκεινται σε αυτόν. Η αναδρομική υπορουτίνα simplicateGraph() θα ακολουθήσει το μονοπάτι που υπονοείται από την ακμή e μέχρι να βρει είτε ένα σταυροδρόμι είτε κάποιον ακραίο κόμβο. Ο υπογράφος (ο οποίος είναι στην ουσία ένα μονοπάτι) αποθηκεύεται στο αντικείμενο branch το οποίο περιέχει δύο δείκτες οι οποίοι δείχνουν προς την αρχή και τέλος του μονοπατιού αντίστοιχα. Η υπορουτίνα simplicateGraph καλεί αναδρομικά τον εαυτό της για κάθε ακμή του αρχικού γράφου την οποία δεν έχει επισκεφτεί, μεγαλώνοντας τον παραγόμενο γράφο, έως ότου όλες οι ακμές του αρχικού έχουν επισκεφτεί.

9.2.2 Υπολογισμός συντομότερης διαδρομής

Θα θέλαμε τώρα να υπολογίσουμε τη συντομότερη διαδρομή μεταξύ δύο κόμβων του γράφου, έχοντας υπ' όψιν τον απλοποιημένο γράφο τον οποίο δημιουργήσαμε, προκειμένου να επιτύχουμε ταχύτερο υπολογισμό.

Στη γενική περίπτωση ο αρχικός και ο τελικός κόμβος του μονοπατιού το οποίο θέλουμε να υπολογίσουμε, θα εμπεριέχονται σε δύο διαφορετικές ακμές (πχ. g1 και g2) του απλοποιημένου γράφου, οι οποίες είναι συνδεδεμένες με δύο διαφορετικά, άγνωστα όμως ακόμη, μονοπάτια. Θα δείξουμε αργότερα πως οι υπόλοιπες περιπτώσεις αποτελούν ειδικές περιπτώσεις αυτής. Και οι δύο ακμές αυτές θα έχουν από έναν αρχικό και ένα τελικό κόμβο. Ονομάζοντας A τον αρχικό κόμβο και B τον τελικό του μονοπατιού προς υπολογισμό, μπορούμε να δημιουργήσουμε μια αναπαράσταση της γενικής περίπτωσης η οποία και απεικονίζεται στο σχήμα 9.3. Ο αναγνώστης μπορεί να παρατηρήσει στο σχήμα τις δύο ακμές g1 και g2, σχεδιασμένες με έντονη γραμμή και τις διαδρομές οι οποίες τις συνδέουν με αχνά σχεδιασμένη γραμμή.



Σχήμα 9.3 Αναπαράσταση της γενικής περίπτωσης

Όταν αναφερόμαστε στους κόμβους A και B του απλοποιημένου γράφου, η μόνη πληροφορία για την τοποθεσία τους που έχουμε, είναι το ότι ανήκουν σε μια απλοποιημένη ακμή. Δεδομένης της απλοποίησης του γράφου το πρόβλημα το οποίο έχουμε να λύσουμε είναι αυτό της εύρεσης της συντομότερης διαδρομής, από κάποιο κόμβο της απλοποιημένης ακμής g1, σε κάποιον της απλοποιημένης ακμής g2. Η ζητούμενη συντομότερη διαδρομή από το A στο B θα χρησιμοποιεί ένα από τα μονοπάτια (τα οποία δεν έχουν υπολογιστεί προς το παρόν), (g1.start,g2.start), (g1.start,g2.end), (g1.end,g2.start), (g1.end,g2.end), προκειμένου να μεταβεί από την g1 στην g2. Το ποιο θα είναι το συντομότερο μονοπάτι δεν εξαρτάται μόνο από τα βάρη των προηγούμενων τεσσάρων διαδρομών, αλλά επίσης και από τα βάρη των μονοπατιών: (Ab), (Aa), (cB), (dB).

Το μονοπάτι το οποίο θα θέλαμε να υπολογίσουμε, αποτελείται από τρία τμήματα. Το πρώτο χρησιμοποιείται για να βρούμε μια διαδρομή από τον κόμβο A σε έναν από τους κόμβους του οποίους ενώνει η g1. Το δεύτερο θα είναι ένα μονοπάτι το οποίο ενώνει την g1 με την g2, και στην προκειμένη περίπτωση θα είναι είτε το (bc) είτε το (ad). Το τρίτο και τελευταίο τμήμα, θα ενώνει ένα από τους δύο κόμβους της g2, με τον κόμβο B. Ο υπολογισμός της συντομότερης διαδρομής μεταξύ του A και B στηρίζεται στην επιλογή του φθηνότερου (με την έννοια του βάρους των ακμών) συνδυασμού των τριών αυτών τμημάτων.

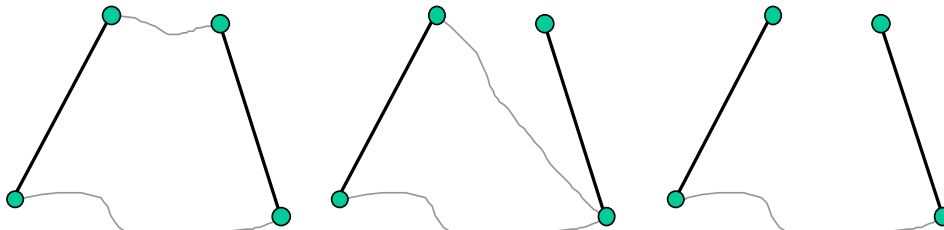
Το πρώτο και τρίτο τμήμα της ζητούμενης από το A στο B διαδρομής υπολογίζεται σε γραμμικό χρόνο, χρησιμοποιώντας τον μη απλοποιημένο γράφο, ακολουθώντας τις ακμές, έως ότου φτάσουμε σε κόμβο ο οποίος είναι είτε ο αρχικός είτε ο τελικός της απλοποιημένης ακμής.

Το δεύτερο τμήμα του μονοπατιού, πρέπει να υπολογιστεί βρίσκοντας το μονοπάτι το οποίο ενώνει την g1 με τη g2. Όπως δείξαμε αυτό θα είναι ένα από τα εξής: (g1.start,g2.start), (g1.start,g2.end), (g1.end,g2.start), (g1.end,g2.end). Προκειμένου να υπολογίσουμε τα τέσσερα αυτά μονοπάτια θα πρέπει να υπολογίσουμε τις τέσσερις συντομότερες διαδρομές οι οποίες ενώνουν τα άκρα των g1 και g2. Για να το κάνουμε αυτό θα χρησιμοποιήσουμε τον αλγόριθμο του Dijkstra, ο οποίος είναι ο ταχύτερος αλγόριθμος υπολογισμού συντομότερων διαδρομών (από κόμβο σε κόμβο), ο οποίος έχει ένα γνώρισμα το οποίο μας βοηθά να μειώσουμε τις εκτελέσεις του από τέσσερις σε δύο. Το γνώρισμα αυτό είναι το ότι δεδομένου ενός κόμβου του γράφου ο αλγόριθμος υπολογίζει τις συντομότερες διαδρομές από όλους τους κόμβους προς αυτόν. Προκειμένου να βρούμε τα τέσσερα αυτά μονοπάτια θα εκτελέσουμε τον αλγόριθμο του Dijkstra, μόνο δύο φορές, για τους κόμβους g1.start και g1.end. Έπειτα θα μπορούμε να υπολογίσουμε σε γραμμικό χρόνο τις ζητούμενες διαδρομές.

Χρησιμοποιώντας την παραπάνω λογική, μπορούμε να καταλήξουμε στο ότι το συντομότερο μονοπάτι μεταξύ των A και B θα είναι ένα από τα εξής: [A - g1.start - g2.start - B], [A - g1.start - g2.end - B], [A - g1.end - g2.start - B], [A - g1.end - g2.end - B]. Το ζητούμενο μονοπάτι είναι αυτό με το μικρότερο βάρος.

Το κόστος του υπολογισμού των παραπάνω τεσσάρων διαδρομών θα είναι $O(N_{\text{κόμβοι_στην_g1}})$ για τον υπολογισμό του πρώτου τμήματος της διαδρομής και $O(N_{\text{κόμβοι_στην_g2}})$ για τον υπολογισμό του τελευταίου. Το κόστος για τον υπολογισμό των τεσσάρων διαδρομών μεταξύ $g1$ και $g2$, θα είναι το κόστος της εκτέλεσης δύο φορών του αλγόριθμου του Dijkstra για τον απλοποιημένο γράφο, το οποίο είναι $2^0(N^2)$, όπου N ο αριθμός των κόμβων του απλοποιημένου γράφου. Έτσι η τάξη του κόστους του υπολογισμού από το A στο B χρησιμοποιώντας τον απλοποιημένο γράφο, είναι $O(N_{\text{κόμβοι_στην_g1}}) + 2^0(N^2) + O(N_{\text{κόμβοι_στην_g2}}) = 2^0(N^2)$, σε αντίθεση με το $O(N_{\text{κόμβοι_στον_αρχικό_γράφο}}^2)$ το οποίο είναι το κόστος εκτέλεσης του αλγορίθμου του Dijkstra στον αρχικό γράφο. Όπως αναφέραμε και αρχικά η ωφέλεια στο κόστος υπολογισμού βασίζεται ισχυρά στη φύση του γράφου τον οποίο απλοποιούμε.

Θα δείξουμε τώρα ότι οποιαδήποτε άλλη περίπτωση εκτός της γενικής (η οποία σκιαγραφείται στο σχήμα 3), αποτελεί ειδική περίπτωση της. Εκτός από την τριμμένη περίπτωση όπου τα A και B , είναι κόμβοι της ίδιας απλοποιημένης ακμής, οπότε και έχουμε μοναδικό μονοπάτι το οποίο συνδέει τα A και B και το οποίο μπορούμε να υπολογίσουμε σε χρόνο τάξης $O(N_{\text{κόμβοι_στη_μη_απλοποιημένη_ακμή}})$, το επόμενο σχήμα (σχήμα 9.5) δείχνει πως σε όλες τις υπόλοιπες περιπτώσεις μπορούμε να αναθέσουμε άπειρο κόστος στις μη υπάρχουσες ακμές (σε σχέση με αυτές που υπάρχουν στη γενική περίπτωση), υπολογίζοντας έτσι το επιθυμητό αποτέλεσμα με τον ίδιο τρόπο.



Σχήμα 9.5 Ειδικές περιπτώσεις, στις οποίες αναθέτουμε άπειρο κόστος στις μη υπάρχουσες ακμές.

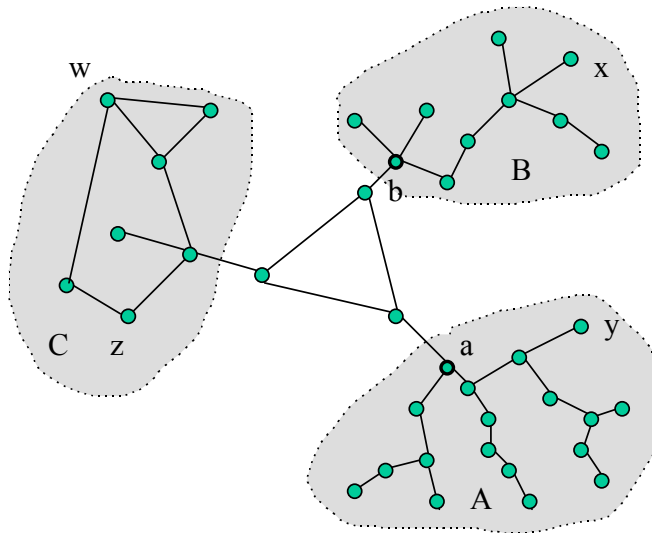
9.3 Δεύτερη μέθοδος

Σε αυτή την παράγραφο, θα περιγράψουμε τη δεύτερη μέθοδο απλοποίησης την οποία χρησιμοποιούμε και τη χρήση του αλγορίθμου του Dijkstra προκειμένου να υπολογίσουμε συντομότερες διαδρομές στον αρχικό γράφο δεδομένου του γράφου ο οποίος προέκυψε από τη απλοποίηση. Θα υποθέσουμε πως ο γράφος ο οποίος χρησιμοποιούμε ως αρχικό είναι ένας γράφος απλοποιημένος από την πρώτη μέθοδο, αν και οι δύο μέθοδοι είναι δυνατόν να χρησιμοποιηθούν ανεξάρτητα. Ωστόσο, η εύρεση της συντομότερης διαδρομής, θα αποδίδει μονοπάτι με κόμβους του μη επεξεργασμένου γράφου και έτσι δεν θα χρειάζεται να χρησιμοποιήσουμε στη συνέχεια τη μέθοδο εύρεσης διαδρομής η οποία περιγράφηκε στην προηγούμενη παράγραφο.

9.3.1 Απλοποίηση γράφου

Στόχος του αλγορίθμου, είναι να εντοπίσει υπογράφοις του αρχικού οι οποίοι παρουσιάζουν μορφή δέντρου. Υπάρχουν τρία πλεονεκτήματα σε αυτό: Πρώτο είναι το ότι όταν υπολογίσουμε το συντομότερο μονοπάτι μεταξύ δύο κόμβων οι οποίοι ανήκουν στον ίδιο υπογράφο, όχι μόνο θα μπορούμε να χρησιμοποιήσουμε αλγόριθμο ταχύτερο από αυτό του Dijkstra, αλλά θα δουλεύουμε και με μικρότερο αριθμό κόμβων. Δεύτερον δεδομένων δύο κόμβων εκ των οποίων ο ένας ή και οι δύο ανήκουν σε υπογράφο με δενδρική δομή, θα είμαστε εκ των προτέρων σίγουροι για ένα ή δύο τμήματα αντίστοιχα της διαδρομής μεταξύ τους, δεδομένου ότι υπάρχει ένα μόνο μονοπάτι από ένα φύλλο του δέντρου προς την ρίζα. Τέλος ο υπολογισμός οποιουδήποτε μονοπατιού, είναι δυνατόν να επιταχυνθεί, δεδομένου ότι ο αλγόριθμος υπολογισμού της συντομότερης διαδρομής, δε χρειάζεται να λάβει υπ' όψιν περιοχές του γράφου οι οποίες δε συσχετίζονται με το προς υπολογισμό μονοπάτι, και άρα θα τρέχει σε υποσύνολο κόμβων του αρχικού γράφου. Το παρακάτω σχήμα (Σχήμα 9.6) απεικονίζει τα παραπάνω. Το παράδειγμα μπορεί να περιγραφεί ως εξής: Στο γράφο, δύο

υπογράφοι δενδρικής μορφής, έχουν εντοπιστεί (οι A και B). Το να βρούμε το συντομότερο μονοπάτι μεταξύ των κόμβων x και y εξαρτάται από το να βρούμε τη συντομότερη διαδρομή μεταξύ των a και b, δεδομένου ότι τα μονοπάτια από το x στο b και από το y στο a, είναι μοναδικά. Επίσης το να βρούμε το συντομότερο μονοπάτι από το z στο w, δεν εξαρτάται από τους κόμβους του A και B τους οποίους μπορούμε να αγνοήσουμε στον υπολογισμό. Στο τρίτο μέρος της παραγράφου, θα δείξουμε πως παρόλο που η περιοχή C δεν είναι δενδρικής μορφής, δεν χρειάζεται να λάβουμε υπ' όψιν εξωτερικούς κόμβους εκτός του C, όπως όταν υπολογίζουμε το μονοπάτι από το w στο x, δεν χρειάζεται να λάβουμε υπ' όψιν τους κόμβους του A.



Σχήμα 9.6 Παράδειγμα εφαρμογής της δεύτερης μεθόδου απλοποίησης ενός γράφου

Όπως αναφέραμε και πιο πριν το βασικό σημείο της μεθοδολογίας μας, είναι το να εντοπίσουμε υπογράφους του αρχικού γράφου, οι οποίοι παρουσιάζουν δενδρική μορφή. Σε αυτή μας την προσπάθεια θα χρειαστούμε ένα κριτήριο το οποίο να μας λει εάν ένας γράφος είναι δενδρικής μορφής ή όχι. Το κριτήριο αυτό βασίζεται στην κατά βάθος απαρίθμηση ενός γράφου, προκειμένου να ταξινομήσουμε τις ακμές του σε δενδρικές και μη, ταξινομώντας τις οποίες επισκεφτήκαμε παραπάνω από μια φορά ως δενδρικές. Εάν μετά από αυτή την ταξινόμηση ένα γράφος έχει μόνο δενδρικές ακμές, τότε αποτελεί δέντρο. Η κατά βάθος απαρίθμηση απαιτεί χρόνο τάξης $O(V + E)$, όπου V είναι ο αριθμός των κόμβων και E ο αριθμός των ακμών του γράφου. Επίσης θα χρειαστούμε ένα κριτήριο το οποίο θα μας χρησιμεύσει στο να αποφανθούμε για το εάν μια ακμή διαχωρίζει ένα γράφο σε δύο ξένους υπογράφους (με την έννοια του ότι οι δύο υπογράφοι δεν θα έχουν κοινούς κόμβους ή ακμές). Το κριτήριο αυτό υλοποιείται βρίσκοντας τους δύο υπογράφους χρησιμοποιώντας κατά πλάτος απαρίθμηση σε χρόνο $O(V + E)$ και υπολογίζοντας την τομή τους. Εάν αυτή ισούται με το κενό σύνολο, τότε οι δύο γράφοι δεν έχουν κοινά μέλη και άρα μπορούν να θεωρηθούν ξένοι, ενώ η ακμή η οποία τους διαχωρίζει θα αναφέρεται από εμάς ως *κρίσιμη* ακμή.

Η εκτέλεση του αλγορίθμου θα παράγει έναν νέο γράφο, στον οποίο οι υπογράφοι οι οποίοι παρουσιάζουν δενδρική μορφή, θα έχουν αντικατασταθεί από κόμβους. Ο αντίστοιχος υπογράφος, θα είναι αντιστοιχισμένος σε κάθε τέτοιο κόμβο, προκειμένου να μπορούμε να βρούμε γρήγορα το εάν ένας κόμβος είναι μέλος ενός δενδρικού υπογράφου ή όχι. Ο αλγόριθμος μπορεί να περιγραφεί ως εξής:

```
void createAbstract(Edge e)
{
    mark e as visited
    if (e is not critical)
```

```

    add e with the vertices that it connects, to the under computation graph
    recurse for each unmarked edge of both vertices
    return;

since e is critical, let L1 and L2 be the names of the sub graphs it separates the graph into
add e with it the vertices that it connects to the under computation graph
if (neither L1 nor L2 are trees)
    and recurse for each unmarked edge of both vertices
else if (one of L1 or L2 is a tree but the other is not)
    substitute the tree structured sub graph with a vertex
    add lg with it the vertices that it connects to the under computation graph
    recurse for each unmarked edge of the vertex contained
    in the non tree structured sub graph
else if (if both L1 and L2 are trees)
    substitute the tree structured sub graphs with a vertices
    add lg with it the vertices that it connects to the under computation graph
}

```

Κάποιος μπορεί να παρατηρήσει πως ο αλγόριθμος, μπορεί να βρει δέντρα τα να περιέχονται σε άλλα, προκαλώντας δυσκολίες στο να βρούμε ποιο δέντρο είναι υποδέντρο ποιανού άλλου, σπαταλώντας ταυτόχρονα τη μνήμη του υπολογιστή, όμως στην πραγματικότητα αυτό δε θα συμβεί, εάν αρχίσουμε την αναδρομή μας από μια μη κρίσιμη ακμή. Τότε ο αλγόριθμος θα προσεγγίζει κάθε δέντρο "από έξω", δηλαδή θα επισκεφτεί πρώτα την ακμή η οποία ενώνει τη ρίζα του δέντρου με τον υπόλοιπο γράφο και μετά τις υπόλοιπες.

9.3.2 Υπολογισμός συντομότερης διαδρομής

Προκειμένου να εξηγήσουμε τη βασική ιδέα της μεθοδολογίας μας ευκολότερα, θα ονομάσουμε ως *κλειστό υπογράφο*, ενός γράφου, έναν υπογράφο του αρχικού ο οποίος είναι συνδεδεμένος με τον υπόλοιπο γράφο με μια μόνο ακμή, και την ακμή αυτή ως *συνδετική ακμή* του υπογράφου (μπορεί να παρατηρήσει κανείς πως η συνδετική ακμή είναι μια κρίσιμη ακμή). Η μεθοδολογία την οποία χρησιμοποιούμε, βασίζεται στην παρατήρηση του ότι εάν η αρχή (ή τέλος) του προς υπολογισμό μονοπατιού, είναι μέλος ενός κλειστού υπογράφου και αυτός ο υπογράφος αποτελεί δέντρο, τότε αρκεί να υπολογίσουμε το τμήμα της διαδρομής από το τέλος (ή αρχή) προς την συνδετική ακμή του κλειστού υπογράφου χρησιμοποιώντας τον αλγόριθμο του Dijkstra και το υπόλοιπο σε γραμμικό χρόνο. Ταυτόχρονα ο υπολογισμός της συντομότερης διαδρομής γίνεται στον απλοποιημένο γράφο, γεγονός το οποίο επιταχύνει τον υπολογισμό, αφού εκτελείται σε μικρότερο αριθμό κόμβων.

Ο αλγόριθμος ο οποίος χρησιμοποιείται για τον υπολογισμό της συντομότερης διαδρομής μεταξύ δύο κόμβων του αρχικού γράφου έχει ως εξής:

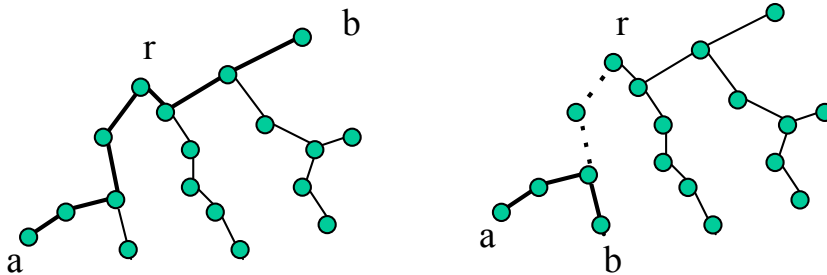
```

if (start vertex is member of a tree)
    let A be the tree root
else
    let A be the start vertex
if (end vertex is member of a tree)
    let B be the tree root
else
    let B be the start vertex
// in any case A and B will be vertices of the simplified graph
if (A and B are in different trees)
    compute the shortest path between A and B using Dijkstra
else
    compute the shortest path between A and B in linear time

```

Σε περίπτωση όπου οι κόμβοι A και B συνυπάρχουν στο ίδιο δέντρο, μπορούμε να βρούμε το συντομότερο μονοπάτι σε γραμμικό χρόνο (όσον αφορά του κόμβους του δέντρου), χρησιμοποιώντας την ακόλουθη μέθοδο: Δεδομένου του ότι κάθε κόμβος ενός δέντρου δείχνει

προς τον πατρικό του κόμβο, βρίσκουμε το μονοπάτι από τον κόμβο αυτό προς η ρίζα του δέντρου. Βρίσκοντας τις δύο αυτές διαδρομές (μια για κάθε κόμβο), και ενώνοντας αυτές, βρίσκουμε ένα μονοπάτι που ενώνει τους δύο κόμβους, το οποίο όμως είναι δυνατόν να μην είναι το βέλτιστο, δεδομένου ότι οι δύο κόμβοι μπορεί να συνυπάρχουν στο ίδιο υποδέντρο του αρχικού. Σε αυτή την περίπτωση όμως, το τελευταίο τμήμα των διαδρομών αυτών, θα είναι το ίδιο (συγκεκριμένα θα είναι ίδιο από το σημείο όπου εμφανίζεται η ρίζα του υποδέντρου στο οποίο ανήκουν οι κόμβοι και έπειτα). Διαγράφοντας το κοινό τμήμα των δύο διαδρομών και παίρνοντας την ένωση τους, βρίσκουμε το βέλτιστο μονοπάτι. Το παρακάτω σχήμα (Σχήμα 9.7), απεικονίζει το σκεπτικό μας. Στο πρώτο σχήμα η συντομότερη διαδρομή μεταξύ a και b μπορεί να υπολογιστεί απλά ενώνοντας τις διαδρομές κάθε κόμβου προς τη ρίζα, δεδομένου του ότι οι δύο κόμβοι δεν ανήκουν στο ίδιο υποδέντρο του αρχικού και τα δύο μονοπάτια δεν θα έχουν κοινούς κόμβους (εκτός από τη ρίζα). Στο δεύτερο σχήμα, τα a και b ανήκουν στο ίδιο υποδέντρο και έχουν κοινό τμήμα (σημειωμένο με τη διακεκομμένη γραμμή). Το μονοπάτι μεταξύ a και b υπολογίζεται διαγράφοντας το κοινό τμήμα από τις διαδρομές και ενώνοντας τα υπόλοιπα.



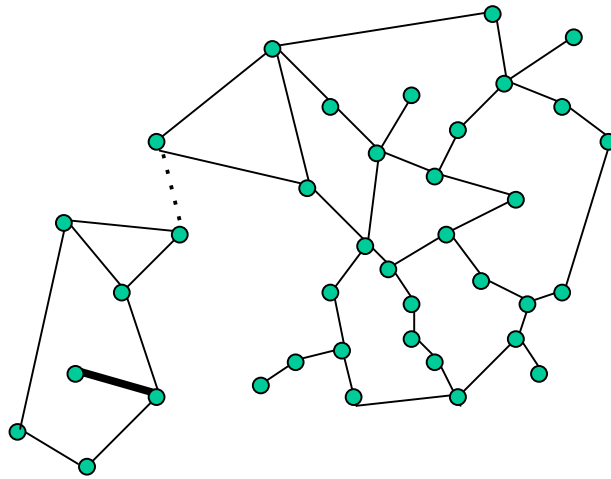
Σχήμα 9.7 Εύρεση διαδρομής σε υπό γράφο δενδρικής μορφής.

9.3.3 Βελτιστοποίηση μεθόδου

Ένα γεγονός το οποίο δεν τονίστηκε στην προηγούμενη παράγραφο, είναι το ότι κατά τον υπολογισμό ενός μονοπατιού μεταξύ δύο κόμβων, δε λαμβάνουμε υπ' όψιν κόμβους οι οποίοι ανήκουν σε άσχετες περιοχές. Οι περιοχές αυτές είναι αυτές τις οποίες δεν θα διέσχιζε το συντομότερο μονοπάτι ανεξάρτητα από τα βάρη των ακμών του (δεδομένου βέβαια του ότι αυτά είναι θετικά). Για παράδειγμα στο σχήμα 6, κατά τον υπολογισμό της συντομότερης διαδρομής από το w στο x , δεν υπάρχει λόγος να υπολογίσουμε τον υπογράφο A και εάν το κάναμε αυτό, απλά θα χειροτερεύαμε το βέλτιστο μονοπάτι. Οι κόμβοι του A δε λαμβάνονται υπ' όψιν όταν υπολογίζουμε ένα τέτοιο μονοπάτι, δεδομένου του ότι ο A έχει αντικατασταθεί στον απλό γράφο με ένα κόμβο.

Είναι δυνατόν να απλοποιήσουμε περαιτέρω το γράφο ο οποίος παρήχθη από τον αλγόριθμο, ο οποίος περιγράφηκε, προκειμένου να μειώσουμε ακόμα παραπάνω το αριθμό των κόμβων τους οποίους χρησιμοποιεί ο αλγόριθμος. Στόχος της βελτιστοποίησης αυτής είναι η εύρεση κλειστών υπογράφων του αρχικού και να του απλοποιήσουμε αντικαθιστώντας τους με έναν κόμβο. Όπως είδαμε προηγούμενα μπορούμε να εντοπίσουμε κλειστούς υπογράφους, χρησιμοποιώντας το κριτήριο της κρίσιμης ακμής.

Όμως αντικαθιστώντας απλά κάθε κλειστό υπογράφο του γράφου, είναι δυνατόν να καταλήξουμε σε αποτυχία, διότι μερικοί υπογράφοι, είναι δυνατόν να έχουν είτε πάρα πολλούς, είτε πάρα πολύ λίγους κόμβους (στο σχήμα 9.8 βλέπουμε μια τέτοια περίπτωση). Χρειαζόμαστε ένα κριτήριο το οποίο να απλοποιεί ή όχι ένα κλειστό υπογράφο δεδομένου του αριθμού των κόμβων του. Γενικά είναι δυνατόν να δημιουργήσουμε ευρετήρια πάνω σε ευρύτερες περιοχές ενός χάρτη (πχ. μια τέτοια περιοχή θα μπορούσε να είναι το οδικό δίκτυο μιας πόλης), ακόμη και εάν οι περιοχές τις οποίες αντικαθιστούμε δεν είναι συνδεδεμένες στον υπόλοιπο γράφο με μόνο μια ακμή. Τα ζητήματα αυτά αποτελούν μελλοντική εργασία η οποία προτίθεται να γίνει, όπου είναι δυνατόν να χρησιμοποιηθούν μέθοδοι μηχανικής μάθησης, προκειμένου να βρούμε τον καταλληλότερο διαχωρισμό ενός γράφου.



Σχήμα 9.8 Η δύο ακμές οι οποίες απεικονίζονται τόσο με την έντονη όσο και με τη διακεκομμένη γραμμή, αποτελούν κρίσιμες ακμές. Ωστόσο διασπώντας το γράφο στην έντονη ακμή δεν επιταχύνουμε τον υπολογισμό.

9.4 Σύνδεση με το υπόλοιπο σύστημα

Οι μεθοδολογίες οι οποίες αναφέρθηκαν αποτελούν τμήμα του γνωστικού επιπέδου της οντότητας εύρεσης συντομότερων διαδρομών. Το σύστημα απλοποιεί το γράφο του οδικού δικτύου κατά την εκκίνηση του και στη συνέχεια δημιουργεί μια νέα οντότητα η οποία υπολογίζει τη ζητούμενη κάθε φορά διαδρομή, για κάθε αίτηση.

Λόγω της φύσης του υπολογισμού της συντομότερης διαδρομής ο οποίος γίνεται σε τμήματα, είναι δυνατόν να υπολογίσουμε το αρχικό τμήμα της διαδρομής πρώτα και έπειτα τα υπόλοιπα, προκειμένου να επιτύχουμε καλύτερο χρόνο απόκρισης του συστήματος. Κάτι τέτοιο μας είναι χρήσιμο προκειμένου να μεταδώσουμε το πρώτο τμήμα αυτό της διαδρομής προς την κινητή μονάδα η οποία να αρχίσει να ακολουθεί το πρώτο τμήμα της διαδρομής, όσο η οντότητα υπολογίζει τα υπόλοιπα τα οποία και μεταδίδονται προς αυτή κατά τη διάρκεια κίνησης της μονάδας προς το τέλος του πρώτου τμήματος της διαδρομής.

Οι περιπτώσεις όπου είναι δυνατόν να κάνουμε μια τέτοια ενέργεια, είναι αυτές στις οποίες τμήμα της διαδρομής ανήκει σε απλοποιημένο τμήμα του γράφου. Έτσι σε περίπτωση όπου ο αρχικός κόμβος ανήκει σε δενδρικό υπογράφο, μεταδίδουμε πρώτα το τμήμα της διαδρομής το οποίο συνδέει τον υπογράφο με τη ρίζα του δέντρου, ενώ σε άλλη περίπτωση αυτό που συνδέει τον αρχικό κόμβο με τη ρίζα του δέντρου στο οποίο ανήκει ο κόμβος προορισμού. Αντίστοιχα ενεργούμε και σε περίπτωση όπου τα απλοποιημένα τμήματα του γράφου δεν είναι αναγκαστικά δενδρικής μορφής, αλλά κλειστοί υπογράφοι.

10 Αλγόριθμος εντοπισμού περιοχών υψηλής συχνότητας εμφάνισης περιστατικών

Σε αυτό το τμήμα, παρουσιάζεται η εργασία η οποία έγινε πάνω στον εντοπισμό γεωγραφικών υποπεριοχών του οδικού δικτύου οι οποίες παρουσιάζουν υψηλή συχνότητα εμφάνισης επειγόντων περιστατικών. Στόχος της εργασίας είναι ο εντοπισμός των περιοχών αυτών και στη συνέχεια η εκμετάλλευση της γνώσης αυτής για τη μείωση του χρόνου απόκρισης του συστήματος. Στη συνέχεια θα εξετάσουμε τη μέθοδο η οποία σχεδιάστηκε για τον εντοπισμό των περιοχών αυτών και η οποία βασίζεται σε τεχνικές προερχόμενες από την περιοχή της μηχανικής μάθησης και συγκεκριμένα στους αλγόριθμους CobWeb και ID3 [46].

10.1 Αλγόριθμος

Ο αλγόριθμος ο οποίος πρόκειται να χρησιμοποιηθεί, προσπαθεί να χαρακτηρίσει κάθε κόμβο του γράφου, του οδικού δικτύου, με κάποιο όνομα το οποίο εκφράζει τον αριθμό και τον τύπο των περιστατικών τα οποία συνέβησαν σε αυτόν. Στη συνέχεια χρησιμοποιώντας τα ονόματα αυτά καθώς και το ελάχιστο συνδετικό δέντρο του γράφου, προσπαθεί να βρει ομάδες από κόμβους με ίδια ονόματα και άρα παρόμοιες συχνότητες εμφάνισης περιστατικών. Ο τρόπος με τον οποίο πραγματοποιείται μια τέτοια διαδικασία, είναι ο επαναληπτικός διαχωρισμός του ελάχιστου συνδετικού δέντρου, μέχρις ότου φτάσουμε σε σημείο να έχουμε ομογενή τμήματα του γράφου, δηλαδή ομάδες κόμβων με ίδια ονόματα. Από την όλη διαδικασία παράγεται ένα δέντρο ταξινόμησης των κόμβων του γράφου. Τέλος παρουσιάζουμε μια μέθοδο βελτιστοποίησης του αποτελέσματος του αλγορίθμου, η οποία αποκόπτει κλάδους του δέντρου ταξινόμησης.

10.2 Γενική τοποθέτηση προβλήματος

Τα περιστατικά τα οποία είναι δυνατόν να συμβούν στο περιβάλλον επείγουσας προνοσοκομειακής ιατρικής περίθαλψης, είναι δυνατόν να διαφέρουν τόσο όσον αφορά την κρισιμότητα τους, όσο και τον εξοπλισμό ο οποίος απαιτείται από την κινητή μονάδα, για την ορθότερη αντιμετώπιση τους (στην πραγματικότητα είναι δυνατόν να διαφέρουν κατά πολλούς άλλους παράγοντες, ωστόσο αυτοί οι δύο μας ενδιαφέρουν στην παρούσα περίπτωση). Στη συγκεκριμένη εργασία, καθώς και άλλη μελέτη η οποία έχει γίνει [54], τα περιστατικά έχουν ταξινομηθεί με γνώμονες την κρισιμότητα τους και τον απαιτούμενο εξοπλισμό και σε κάθε κατηγορία περιστατικού έχει αποδοθεί και κάποιο όνομα, το οποίο στη συγκεκριμένη περίπτωση είναι ο αύξων αριθμός της κατηγορίας (η συνάρτηση ταξινόμησης των περιστατικών $Name(x)$ φαίνεται στον παρακάτω πίνακα). Φυσικά, εάν ήταν απαραίτητο θα μπορούσαμε να έχουμε μεγαλύτερη διαβάθμιση και στους δύο γνώμονες ταξινόμησης, υιοθετώντας για παράδειγμα την κλάση των ημιεπείγοντων περιστατικών ή αναγνωρίζοντας περιστατικά τα οποία χρειάζονται τμήμα μόνο του ειδικού εξοπλισμού. Επίσης θα ήταν δυνατό να παρουσιάσουμε και έναν τρίτο γνώμονα ταξινόμησης πχ. το αν το περιστατικό είναι καρδιολογικής φύσεως ή όχι κ.τ.λ. Σε κάποια τέτοια περίπτωση η ονομασία των κλάσεων θα γινόταν πάλι με τον ίδιο τρόπο, δηλαδή δίνοντας με αυτόματο τρόπο μοναδικές ονομασίες σε κάθε κλάση. Η ονομασία των κλάσεων με τη μέθοδο της απαρίθμησης μας δίνει τη δυνατότητα, τόσο να έχουμε μη πεπερασμένο αριθμό διαβαθμίσεων (πριν από τη μελέτη των περιστατικών), όσο και να αποφασίζουμε για τον αριθμό των διαβαθμίσεων με αυτόματο τρόπο [46].

$Name(x), x \in Episodes$	Μη επείγον	Επείγον
Δεν χρειάζεται κινητή εντατική μονάδα θεραπείας	1	2
Χρειάζεται κινητή εντατική μονάδα θεραπείας	3	4

10.3 Χαρακτηρισμός παραδειγμάτων

Δεδομένης της αναπαράστασης του οδικού δικτύου με χρήση γεωγραφικού γράφου, το πρόβλημα μας είναι δυνατόν να επαναδιατυπωθεί σε αυτό της ομαδοποίησης των κόμβων του γράφου, σε ομάδες με κοινά χαρακτηριστικά τη γεωγραφική τους θέση και τον τύπο του περιστατικού. Κατά συνέπεια θα χρειαστούμε να χαρακτηρίσουμε κάθε κόμβο του γράφου μας ανάλογα με τα περιστατικά τα οποία συνέβησαν σε αυτόν. Έχοντας ως σύνολο τιμών τα ονόματα των κλάσεων, θα χαρακτηρίσουμε κάθε κόμβο με μια τιμή την οποία θα ονομάσουμε *αριθμητικό βάρος (numeric weight)* και η οποία θα εκφράζει την βαρύτητα του κόμβου σε σχέση με το είδος και τον αριθμό των ατυχημάτων που συνέβησαν στην γεωγραφική περιοχή του κόμβου. Έτσι, για να υπολογίσουμε το αριθμητικό βάρος του κόμβου η θα χρησιμοποιήσουμε το άθροισμα:

$$\sum_{i \in \text{EpisodesAt}(n)} = \text{Name}(i) / \text{card}(\text{EpisodesAt}(n)),$$

όπου $\text{card}()$, η συνάρτηση πληθαρισμού συνόλων.

Στη συνέχεια θα χρειαστεί να διακριτοποιήσουμε τις τιμές αυτές έτσι ώστε να καταλήγουμε πάντοτε σε ένα πεπερασμένο αριθμό χαρακτηρισμών για κάθε κόμβο. Για να το επιτύχουμε αυτό, θα χρειαστούμε ένα τρόπο μοναδικής απεικόνισης κάθε συνόλου τιμών σε ένα πεδίο τιμών. Χρησιμοποιούμε λοιπόν μια συνάρτηση διακριτοποίησης, ακριβώς για να επιτύχουμε τη μοναδικότητα αυτή (σε μια συνάρτηση κάθε τιμή του πεδίου τιμών αντιστοιχίζεται με ένα και μόνο στοιχείο του συνόλου τιμών). Η έξοδος της συνάρτησης για κάθε κόμβο του γράφου, θα είναι μια τιμή η οποία θα ονομάζεται *ονομαστικό βάρος (Nominal Weight)* του κόμβου.

Ως συνάρτηση διακριτοποίησης προσπαθήσαμε να βρούμε μια συνάρτηση η οποία αφενός διαχωρίζει τα παραδείγματα σε τόσες ομάδες, όσα και τα είδη των περιστατικών, αφετέρου είναι σε θέση να τα ταξινομήσει σε διατεταγμένη σειρά, όσον αφορά την κλάση του περιστατικού, το κάθε ένα από αυτά. Η συνάρτηση η οποία χρησιμοποιούμε για τον υπολογισμό του ονομαστικού βάρους για κάθε κόμβο του γράφου x , θα είναι:

$$\text{NominalWeight} = 2 \wedge (\text{round}(\text{NumericWeight}(x)))$$

Χρησιμοποιώντας τη στρογγυλοποίηση αριθμών ως μέσο για τη διακριτοποίηση των αριθμητικών τιμών αυτών θα καταλήξουμε σε τέσσερις δυνατούς χαρακτηρισμούς για κάθε κόμβο στον οποίο έχει συμβεί έστω και ένα περιστατικό, και έναν πέμπτο ο οποίος θα αφορά τους κόμβους στους οποίους δεν έχει συμβεί κανένα περιστατικό.

Εάν θα θέλαμε να κατατάξουμε τις περιοχές δίνοντας μικρή σημασία στην ποικιλομορφία των περιστατικών (δηλαδή εάν μας ενδιέφερε να βρούμε τις περιοχές στις οποίες συνέβησαν τα περισσότερα περιστατικά, χωρίς να θέλουμε να τις κατατάξουμε ανάλογα με την κλάση των περιστατικών που συνέβησαν), θα επιλέγαμε αντί της βάσης 2 τη μονάδα, δημιουργώντας έτσι τη σταθερή συνάρτηση. Στο άλλο άκρο, μεγαλώνοντας το βαθμό της βάσης (πχ. αντικαθιστώντας το 2 με 2^k , όπου k θετικός ακέραιος), αυξάνουμε τη διαφορά βαρύτητας η οποία υπάρχει μεταξύ των περιστατικών, όσο αυξάνουμε και το συντελεστή k .

Ωστόσο ένας άλλος τρόπος με τον οποίο θα μπορούσαμε να χρησιμοποιήσουμε τον αλγόριθμο, θα ήταν να τον τρέξουμε περιορίζοντας τα παραδείγματα μας σε κάθε εκτέλεση, σε μια κατηγορία παραδειγμάτων. Έτσι για εκτελώντας τον αλγόριθμο στη δικιά μας περίπτωση τέσσερις φορές, είναι δυνατόν έχουμε ένα σύνολο περιοχών για κάθε είδος περιστατικού. Η διαφορά στην χρήση της μιας η της άλλης μεθόδου χρήσης του αλγορίθμου, έγκειται στο εάν μας ενδιαφέρει να χειριστούμε τα διαφορετικά περιστατικά με ξεχωριστές ενέργειες για την αντιμετώπισή τους, ή ως περιοχές πρωτεύουσας, δευτερεύουσας κ.τ.λ. σημασίας. Συγκεκριμένα όπως μπορεί να παρατηρήσει κανείς, οι ονομασίες των κλάσεων των περιστατικών (δηλαδή οι αριθμοί 1, 2, 3, 4) αντιστοιχίζονται με την κρίσιμότητα της κλάσης περιστατικών. Δηλαδή με την αριθμητικά μικρότερη ονομασία έχει χαρακτηριστεί η κλάση των περιστατικών τα οποία ούτε επείγοντα είναι, ούτε απαιτούν ιδιαίτερο εξοπλισμό, ενώ με την αριθμητικά μεγαλύτερη ονομασία έχουν χαρακτηριστεί τα πιο κρίσιμα περιστατικά. Κατά αυτόν τον τρόπο, θα είμαστε σε θέση να χρησιμοποιήσουμε τον αλγόριθμο κατά δύο διαφορετικούς τρόπους:

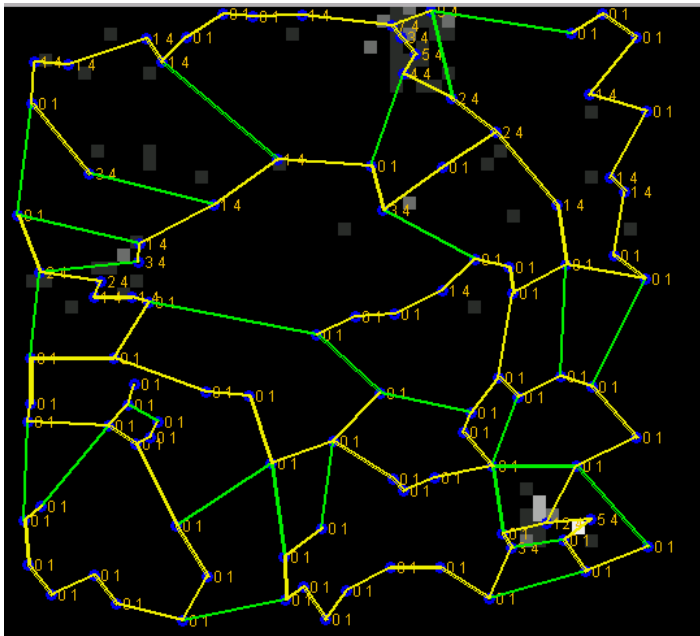
Περιορίζοντας κάθε φορά τα παραδείγματα μας, σε μια κατηγορία παραδειγμάτων και εκτελώντας τον αλγόριθμο μια φορά για κάθε κατηγορία. Κατά αυτόν τον τρόπο, καταλήγουμε σε, στη συγκεκριμένη περίπτωση, τέσσερα σύνολα το καθένα από τα οποία περιέχει ομάδες κόμβων και κάθε ένα από τα οποία περιέχει τις περιοχές οι οποίες παρουσιάζουν υψηλή συχνότητα περιστατικών της ίδιας κλάσης. Προφανώς τα τέσσερα σύνολα, αυτά είναι δυνατόν να περιέχουν κόμβους από κοινού αφού μια γεωγραφική περιοχή είναι δυνατόν να παρουσιάζει υψηλή συχνότητα εμφάνισης περιστατικών, από άνω της μιας κλάσης.

Εάν τρέξουμε τον αλγόριθμο για το σύνολο των παραδειγμάτων, τότε το αποτέλεσμα που θα πάρουμε θα αποτελείται από ένα σύνολο το οποίο θα περιέχει ομάδες κόμβων του γράφου και το ονομαστικό βάρος της κάθε ομάδας θα περιγράφει την κρισιμότητα της περιοχής (δεδομένου ότι έχουμε ονομάσει τις κλάσεις των παραδειγμάτων με αύξουσα σειρά κρισιμότητας). Ο λόγος για αυτό είναι πως είναι δυνατόν να υπάρξουν κόμβοι οι οποίοι παρουσιάζουν υψηλή συχνότητα εμφάνισης περιστατικών σε παραπάνω από μια κλάσεις, με συνέπεια το αριθμητικό τους βάρος να είναι ο σταθμισμένος μέσος όρος των αριθμητικών βαρών που θα προέκυπταν εάν υπολογίζαμε τα αριθμητικά βάρη του κόμβου τέσσερις φορές, μια για κάθε κλάση γεγονότων. Η χρησιμότητα αυτού του τρόπου χρήσης του αλγορίθμου, θα ήταν ίσως μια πιο γενική ταξινόμηση των περιοχών, από λιγότερο σε περισσότερο κρίσιμες.

10.4 Υπολογισμός ελάχιστου συνδετικού δέντρου

Έστω ένα υποσύνολο των ακμών ενός γράφου. Αν το υποσύνολο των ακμών αυτό, συνδέει όλους τους κόμβους του γράφου, τότε μπορούμε να το ονομάσουμε ως ένα συνδετικό υπογράφο. Εάν περιορίσουμε τους συνδετικούς υπογράφους ενός γράφου, σε αυτούς οι οποίοι είναι ακυκλικοί και από αυτούς διαλέξουμε αυτόν με το ελάχιστο άθροισμα βαρών, τότε θα έχουμε ένα δένδρο το οποίο ονομάζεται *ελάχιστο συνδετικό δένδρο*. (Σημείωση: πάντως ένας γράφος είναι δυνατόν να έχει παραπάνω από ένα ελάχιστα συνδετικά δέντρα). Για τον υπολογισμό του ελάχιστου συνδετικού δέντρου χρησιμοποιήσαμε τον αλγόριθμο του Kruskal [48].

Το ελάχιστο συνδετικό δένδρο ενός γράφου, ενώνει τους n κόμβους του γράφου με $n-1$ ακμές. Εάν αφαιρέσουμε μια ακμή από αυτό, τότε προκύπτουν και πάλι δύο δέντρα έστω τα Δ_1 και Δ_2 . Τα δύο αυτά δέντρα θα είναι και αυτά ελάχιστα και συνδετικά για τους δύο υπογράφους, του αρχικού, τους οποίους ορίζουν. Τα παρακάτω σχήμα (σχήμα 10.1) δείχνει μια οθόνη του συστήματος όπου έχει υπολογιστεί το ελάχιστο συνδετικό δένδρο του γράφου, και έχουν αποδοθεί ονομαστικά βάρη στους κόμβους. Με φωτεινότερο χρώμα έχουν σχεδιαστεί οι ακμές οι οποίες ανήκουν στο ελάχιστο συνδετικό δένδρο του γράφου, ενώ οι γκριζες τετραγωνικές περιοχές οι οποίες έχουν χρωματισθεί στο υπόβαθρο, εκφράζουν το πλήθος των ατυχημάτων τα οποία συνέβησαν στην κάθε μια (οι περιοχές με το φωτεινότερο χρώμα, είναι αυτές στις οποίες έχουν συμβεί τα περισσότερα περιστατικά, ενώ οι μαύρες αυτές στις οποίες δεν έχει συμβεί κανένα). Δίπλα σε κάθε κόμβο αναγράφεται ο αριθμός των περιστατικών ο οποίος του έχει ανατεθεί, καθώς και το ονομαστικό του βάρος.



Σχήμα 10.1 Με φωτεινό χρώμα απεικονίζονται οι ακμές του ελάχιστου συνδετικού δέντρου, ενώ διπλά σε κάθε κόμβο ο αριθμός των περιστατικών που έχουν συμβεί σε αυτόν καθώς και το ονομαστικό του βάρος.

10.5 Επαναληπτικός διαχωρισμός του ελάχιστου συνδετικού δέντρου

Κάθε φορά που διασπάμε στα δύο το ελάχιστο συνδετικό δέντρο ενός γράφου, σχηματίζονται δύο νέα δέντρα. Συνεχίζοντας αναδρομικά τη διαδικασία αυτή για κάθε νέο δέντρο το οποίο δημιουργείται, δηλαδή επιλέγοντας μια ακμή του κάθε δέντρου την οποία αφαιρούμε με αποτέλεσμα να σχηματιστούν δύο νέα δέντρα, είναι δυνατόν να δημιουργήσουμε ένα δέντρο το οποίο ακριβώς περιγράφει τις διασπάσεις οι οποίες έγιναν στο γράφο. Κάθε κόμβος του δέντρου αντιστοιχεί σε ένα σύνολο κόμβων του γράφου, το οποίο μάλιστα είναι ένα από τα δύο δέντρα στα οποία είχε διασπαστεί το αρχικό δέντρο. Προφανώς κάθε κόμβος θα έχει δύο παιδιά, τα οποία είναι τα δύο υποδέντρα στα οποία διασπάστηκε. Θα αναφερόμαστε στο δέντρο το οποίο περιγράφει τις διασπάσεις του ελάχιστου συνδετικού δέντρου ενός γράφου, ως *δέντρο ταξινόμησης (classification tree)*.

Σύμφωνα με όσα έχουμε πει προηγουμένως, θα διασπάσουμε επαναληπτικά το ελάχιστο συνδετικό δέντρο του γράφου, δημιουργώντας το δέντρο ταξινόμησης. Σε κάθε βήμα της αναδρομής, θα επιλέγουμε μια ακμή του ελάχιστου συνδετικού δέντρου και θα διασπάμε το δέντρο σε δύο, βάση αυτής της ακμής. Ο αλγόριθμος θα συνεχίζει αναδρομικά για τους δύο υπογράφους οι οποίοι προέκυψαν. Ο σχεδιασμός του αλγόριθμου θα πρέπει να γίνει με τέτοιο τρόπο έτσι ώστε αφενός να επιλέγεται η κατάλληλη ακμή προς διάσπαση, αφετέρου να βρεθεί η κατάλληλη συνθήκη τερματισμού της αναδρομής.

Όσον αφορά την εύρεση της καταλληλότερης ακμής θα χρειαστούμε να δανειστούμε την έννοια της εντροπίας, όσον αφορά την οργάνωση της πληροφορίας, από την θεωρία πληροφοριών (όπου η εντροπία δίνεται από τον τύπο $E = -\sum_x p(x) \ln[p(x)]$ όπου $p(x)$, είναι η πιθανότητα της κατάστασης x , και για $p(x) = 0$ ορίζουμε $\ln[p(x)] = 0$) [46]. Δεδομένου ότι στόχος μας είναι να διαχωρίσουμε το συνδετικό δέντρο του γράφου, σε ομάδες κόμβων του, οι οποίες έχουν κοινά χαρακτηριστικά (στην περίπτωση μας ίδια ονόματα), θα επιλέγουμε την ακμή αυτή η οποία μειώνει την εντροπία της οργάνωσης της πληροφορίας του γράφου, ή αλλιώς, θα μας δίνει δύο υπογράφους στους οποίους η πληροφορία των ονομάτων των κόμβων είναι περισσότερο, ή ίδια οργανωμένη από αυτή του αρχικού. Έτσι για την εύρεση της καταλληλότερης προς διάσπαση ακμής του γράφου υπολογίζουμε για κάθε ακμή την αριθμητική ποσότητα *Category Utility* η οποία εκφράζει τη μείωση της εντροπίας την οποία θα επιφέρει η διάσπαση του συνδετικού δέντρου, εάν αφαιρέσουμε τη συγκεκριμένη ακμή. Τελικά σε κάθε βήμα της αναδρομής επιλέγουμε προς διάσπαση, την ακμή η οποία παρουσιάζει τη μέγιστη τιμή όσον αφορά τον υπολογισμό της ποσότητας *Category Utility* η οποία υπολογίζεται από τον τύπο:

$$CU(G_1, G_2) = \frac{\sum_{c=1}^2 \frac{|G_c|}{n} \left[\sum_{i=1}^{|NominalWeightNodeValues|} \left(\frac{g_i, G_c}{|G_c|} \right)^2 - \sum_{i=1}^{|NominalWeightNodeValues|} \left(\frac{g_i}{n} \right)^2 \right]}{2}$$

Όπου n ο ολικός αριθμός παραδειγμάτων στους δύο υπογράφοι G_1, G_2
 $|G_c|$ ο αριθμός των παραδειγμάτων στο γράφο G_c
 g_i μια τιμή ονομαστικού βάρους (η οποία φυσικά ανήκει στο διακριτοποιημένο σύνολο τιμών)
 g_i, G_c ο αριθμός των φορών που συναντάται το g_i στο G_c

Από την άλλη πλευρά το κριτήριο τερματισμού της αναδρομής πρέπει να επιλεγεί κατά τέτοιο τρόπο έτσι ώστε η αναδρομή να σταματά όταν περαιτέρω διάσπαση του ελάχιστου συνδεδετικού δέντρου δεν επιφέρει μείωση της εντροπίας οργάνωσης της πληροφορίας των κόμβων του γράφου. Άρα θα χρειαστούμε να υπολογίσουμε την εντροπία μετά την υποτιθέμενη διάσπαση και εάν αυτή μειώνεται τότε συνεχίζουμε τη διάσπαση για αυτό τον κόμβο, αλλιώς σταματάμε και ο συγκεκριμένος κόμβος αποτελεί πλέον φύλλο του δέντρου ταξινόμησης, η αλλιώς μια από τις ομάδες κόμβων τις οποίες προσπαθούμε να υπολογίσουμε. Ο τρόπος με τον οποίο θα υπολογίσουμε την μείωση της εντροπίας είναι με το να συγκρίνουμε το Category Utility του συγκεκριμένου υπογράφου, με αυτό των δύο υπογράφων στους οποίους θα διασπαστεί εάν επιλέξουμε να κάνουμε τη διάσπαση, χρησιμοποιώντας το κριτήριο που προαναφέραμε.

10.6 Βελτιστοποίηση (κλάδεμα) του δέντρου ταξινόμησης

Κατά την εφαρμογή του αλγορίθμου συναντήσαμε το πρόβλημα που στη θεωρία της μηχανικής μάθησης αναφέρεται ως overfitting [46], παρατηρώντας ότι ο αλγόριθμος συνέχιζε να διασπά περιοχές του ελάχιστου συνδεδετικού δέντρου, για τις οποίες θα περιμέναμε ότι θα τερμάτιζε.

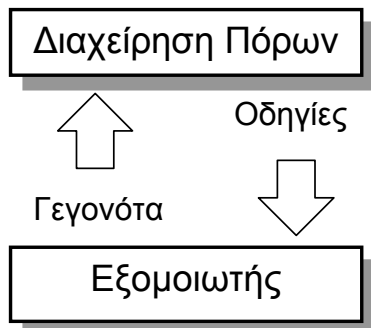
Στην εργασία αυτή, υλοποιήσαμε μια μέθοδο βελτιστοποίησης (tree pruning) του δέντρου ταξινόμησης. Η μέθοδος συνίσταται στον υπολογισμό την ομογένειας ενός υπογράφου, ο οποίος αντιστοιχεί σε ένα κόμβο του δέντρου ταξινόμησης. Αναλυτικότερα, ως ομογένεια ενός γράφου (του οποίου οι κόμβοι έχουν χαρακτηριστεί από κάποιο Nominal Weight), ορίζουμε το ποσοστό της πολυπληθέστερης υποομάδας κόμβων (βάση του Nominal Weight του κόμβου), στο πλήθος των κόμβων του γράφου. Για παράδειγμα εάν ένας γράφος έχει 100 κόμβους από τους οποίους οι 97 έχουν Nominal Weight ίσο με 1, 2 έχουν Nominal Weight ίσο με 2 και ένας ίσο με 3, θα λέμε πως αυτό ο γράφος, έχει ομογένεια 97%.

Δεδομένου του ότι κάθε κόμβος του ελάχιστου συνδεδετικού αντιστοιχεί και σε ένα υπογράφο του αρχικού γράφου, η μέθοδος υπολογίζει για κάθε κόμβο (αρχίζοντας από τη ρίζα του δέντρου), την ομογένεια του γράφου ο οποίος αντιστοιχεί στον κόμβο και αποκόπτει τα παιδιά του κόμβου, εάν η ομογένεια είναι υψηλότερη από κάποιο κατώφλι το οποίο έχει ορίσει ο χρήστης.

11 Περιβάλλον Εφαρμογής

11.1 Εισαγωγή

Μιας και δεν είναι δυνατόν να δοκιμάσουμε τη λειτουργία του συστήματος διαχείρισης πόρων κατευθείαν στο πραγματικό περιβάλλον επείγουσας προνοσοκομειακής ιατρικής περίθαλψης (τόσο για λόγους ασφάλειας, όσο και διότι θα θέλαμε να δοκιμάσουμε το σύστημα διαχείρισης πόρων σε πολλές διαφορετικές καταστάσεις), απαιτείται η δημιουργία ενός εξομοιωτή ο οποίος θα προσομοιώνει το περιβάλλον. Ο εξομοιωτής θα προσομοιώνει τη συμπεριφορά των οντοτήτων του συστήματος δημιουργώντας έτσι μια πλατφόρμα πάνω στην οποία θα μπορούμε να δοκιμάσουμε το σύστημα διαχείρισης πόρων. Επίσης χρησιμοποιώντας τον εξομοιωτή θα μας είναι δυνατό να δοκιμάσουμε το σύστημα διαχείρισης πόρων σε καταστάσεις οι οποίες δεν προκύπτουν συχνά, όπως και σε συστήματα επείγουσας προνοσοκομειακής ιατρικής περίθαλψης άλλων κρατών. Το παρακάτω σχήμα απεικονίζει τον τρόπο λειτουργίας του συστήματος εξομοίωσης σε συνεργασία με αυτό της διαχείρισης πόρων. Ο εξομοιωτής προσομοιώνει τη συμπεριφορά των αντικειμένων της εξομοίωσης και ενημερώνει το τμήμα διαχείρισης πόρων για τα γεγονότα που συμβαίνουν. Το σύστημα διαχείρισης πόρων παρακολουθεί το περιβάλλον και τα γεγονότα και δίνει οδηγίες για τη διαχείριση πόρων (βλ σχήμα 11.1).



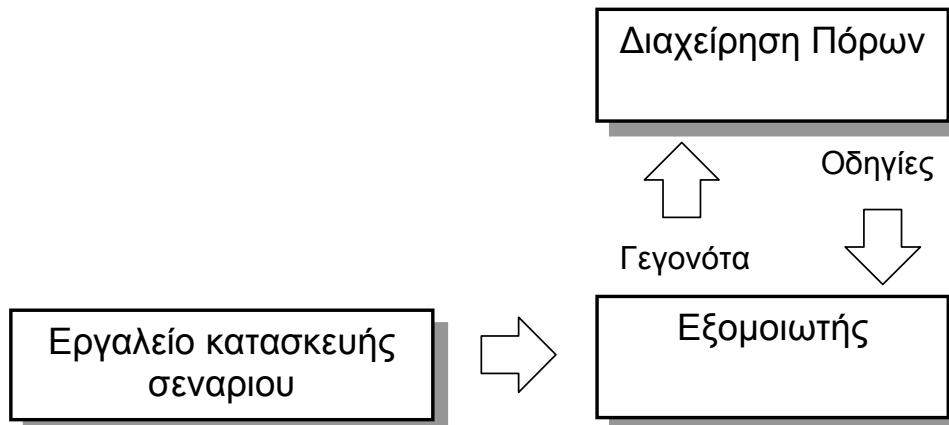
Σχήμα 11.1 Συνεργασία εξομοιωτή με το σύστημα διαχείρισης πόρων.

Θα θέλαμε ο εξομοιωτής που θα δημιουργήσουμε να είναι ένα όσο το δυνατόν ευέλικτο εργαλείο, ενώ επίσης θα θέλαμε να μπορούμε να προσομοιώσουμε πληθώρα περιβαλλόντων σε αυτόν. Μάλιστα, απαίτηση του σχεδιασμού του εξομοιωτή είναι και το να είναι δυνατόν να μπορεί να προσομοιώσει πολλές δυνατές καταστάσεις - δηλαδή να είναι δυνατόν να ξεκινήσει την προσομοίωση από πολλές δυνατές αρχικές καταστάσεις. Κάτι τέτοιο είναι εφικτό εάν ο προσομοιωτής παίρνει ως παράμετρο το περιβάλλον προσομοίωσης, την αρχική κατάσταση του περιβάλλοντος καθώς και τα κάποια τα γεγονότα τα οποία θέλει να προσομοιώσει (δηλαδή τις κλήσεις επειγόντων περιστατικών και άλλα ιατρικά συμβάντα).

Για την απλοποίηση των παραπάνω ο εξομοιωτής θα πρέπει να παίρνει ως είσοδο ένα αρχείο το οποίο θα περιέχει όλες τις σχετιζόμενες παραμέτρους σχετιζόμενες με την εξομοίωση, το περιβάλλον και τα αντικείμενα της εξομοίωσης καθώς και τα προγραμματισμένα γεγονότα που θέλουμε να προκύψουν. Το αρχείο αυτό περιγράφει τόσο την αρχική κατάσταση, όσο και κάποια γεγονότα τα οποία θα συμβούν κατά την εξέλιξη της εξομοίωσης και στη συνέχεια θα το ονομάζουμε αρχείο "σεναρίου". Για την απλούστερη κατασκευή του αρχείου σεναρίου ο χρήστης θα έχει στη διάθεση του ένα εργαλείο κατασκευής τέτοιων αρχείων, το οποίο είναι δυνατόν να αποτελεί ξεχωριστή εφαρμογή ή μέρος του εξομοιωτή, ανάλογα με τις ανάγκες των χρηστών.

Κατά την εκκίνηση της εξομοίωσης ο εξομοιωτής θα διαβάζει το περιεχόμενο του αρχείου σεναρίου, θα δημιουργεί το περιβάλλον προσομοίωσης και στη συνέχεια θα αρχίζει η λειτουργία της προσομοίωσης. Κατά τη διάρκεια της προσομοίωσης, ο εξομοιωτής, θα ενημερώνει το τμήμα διαχείρισης πόρων για κάθε γεγονός το οποίο λαμβάνει χώρα (π.χ.

αποδοχή σήματος το οποίο περιέχει το στίγμα κινητής μονάδας, επείγον περιστατικό, ιατρικό συμβάν, καιρικές συνθήκες κ.α.). Το τμήμα διαχείρισης πόρων θα αντιλαμβάνεται τα γεγονότα αυτά μέσω αισθητήρων και στη συνέχεια θα δίνει οδηγίες για τη διαχείριση των πόρων του συστήματος (βλ σχήμα 11.2).



Σχήμα 11.2 Στο σχήμα παρουσιάζεται η συνεργασία των εφαρμογών του εξομοιωτή, μονάδας διαχείρισης πόρων και εργαλείου κατασκευής σεναρίων.

11.2 Εργαλείο κατασκευής αρχείων σεναρίων

11.2.1 Σκοπός

Σκοπός του εργαλείου αυτού είναι η δημιουργία αρχείων σεναρίου τα οποία θα χρησιμοποιούνται ως είσοδος στον εξομοιωτή. Ο χρήστης θα μπορεί να σχεδιάσει το γράφο που περιγράφει το οδικό δίκτυο, συμβουλευόμενος ίσως το χάρτη του περιβάλλοντος και να χρονοπρογραμματίσει γεγονότα επειγόντων περιστατικών ή άλλα ιατρικά συμβάντα τα οποία θα ήθελε να εξομοιώσει. Επίσης ο χρήστης είναι δυνατόν να ζητά στατιστικά στοιχεία για το περιβάλλον το οποίο δημιουργεί, η ακόμα να ζητά από το εργαλείο να του κατασκευάσει αυτόματα ένα περιβάλλον σύμφωνα με υποδείξεις του.

11.2.2 Δομή αρχείου σεναρίου

Το αρχείο το οποίο περιγράφει το σενάριο εξομοίωσης περιέχει τις εξής πληροφορίες:

- Ένα κατευθυνόμενο γράφο ο οποίος θα περιγράφει το οδικό δίκτυο του περιβάλλοντος εξομοίωσης
- Περιγραφή των κινητών μονάδων και των χαρακτηριστικών τους
- Περιγραφή όλων των υπόλοιπων οντοτήτων και των χαρακτηριστικών τους (νοσοκομεία, ιατρικά κέντρα)
- Περιγραφή των προς εξομοίωση γεγονότων επειγόντων περιστατικών
- Περιγραφή των προς εξομοίωση ιατρικών συμβάντων
- Εικόνα του χάρτη του περιβάλλοντος εξομοίωσης (προαιρετικό)

11.2.3 Λειτουργικότητα εργαλείου κατασκευής αρχείων σεναρίου

Οι λειτουργίες του εργαλείου κατασκευής σεναρίων είναι οι εξής:

- Εισαγωγή εικόνας χάρτη και κατασκευή γράφου οδικού δικτύου
- Χρονοπρογραμματισμός γεγονότων
- Περιγραφή οντοτήτων εξομοίωσης
- Στατιστική παρουσίαση περιβάλλοντος

11.2.3.1 Εισαγωγή εικόνας χάρτη και κατασκευή γράφου οδικού δικτύου

Το εργαλείο δίνει τη δυνατότητα στο χρήστη να κατασκευάσει το γράφο ο οποίος αναπαριστά το οδικό δίκτυο της εξομοίωσης. Ο σχεδιασμός του γράφου γίνεται με γραφικό και φιλικό προς το χρήστη τρόπο ενώ η σχεδίαση του είναι δυνατόν να υποβοηθηθεί και από την παρουσία της εικόνας του χάρτη του περιβάλλοντος εξομοίωσης. Ο σχεδιαζόμενος γράφος είναι κατευθυνόμενος έτσι ώστε να μπορεί να αναπαρασταθούν σε αυτόν οι δρόμοι μονής και διπλής κατεύθυνσης.

Οι ακμές του κόμβου έχουν βάρη όπου η αρχική τους τιμή είναι η απόσταση των κόμβων που ενώνουν. Ωστόσο μιας και στην εφαρμογή μας θα ασχοληθούμε με χρονικές αποστάσεις, ο χρήστης είναι δυνατόν να καθορίσει από αυτό το σημείο το βάρος του κόμβου.

Επίσης ο χρήστης είναι δυνατόν να εισάγει ένα αρχείο τεχνικού σχεδίου (τύπου DXF), το οποίο να περιγράφει το γράφο του οδικού δικτύου, του οποίου το περιεχόμενο θα μετατραπεί αυτόματα σε γράφο. Η λειτουργία αυτή είναι αρκετά σημαντική, διότι κατά αυτό τον τρόπο αυτοματοποιείται η διαδικασία κατασκευής του χάρτη ενώ ο γεωγραφικός γράφος ο οποίος θα κατασκευαστεί, θα είναι ακριβής (δεδομένου ότι οι συντεταγμένες του αρχείου τεχνικού σχεδίου θα έχουν τις σωστές τιμές).

11.2.3.2 Χρονοπρογραμματισμός γεγονότων

Χρησιμοποιώντας το εργαλείο κατασκευής σεναρίων ο χρήστης χρονοπρογραμματίζει τα επείγοντα περιστατικά και ιατρικά συμβάντα τα οποία θέλει να λάβουν χώρα κατά τη διάρκεια της προσομοίωσης. Ο χρήστης ορίζει το χρόνο και τα άλλα χαρακτηριστικά του γεγονότος, ενώ είναι δυνατόν να έχει και μια γραφική απεικόνιση των γεγονότων τα οποία έχουν χρονοπρογραμματιστεί.

11.2.3.3 Περιγραφή οντοτήτων εξομοίωσης

Παράλληλα ή μετά τον σχεδιασμό του χάρτη οδικού δικτύου, ο χρήστης μπορεί να ορίζει τις οντότητες οι οποίες θα λάβουν μέρος στην προσομοίωση. Οι οντότητες αυτές μπορεί να είναι είτε κινητές μονάδες, είτε άλλες οντότητες της εξομοίωσης, όπως νοσοκομεία ή ιατρικά κέντρα. Ο ορισμός της οντότητας γίνεται επιλέγοντας την τοποθεσία στην οποία θα βρίσκεται αρχικά η οντότητα, ενώ στη συνέχεια ο χρήστης είναι δυνατόν να τροποποιήσει τα χαρακτηριστικά της κάθε οντότητας (ο επεξεργαστής αρχείων σεναρίου δίνει κάποιες αρχικές τιμές για κάθε οντότητα, έτσι ώστε ο ορισμός των οντοτήτων να είναι πιο φιλικός προς το χρήστη).

Τα χαρακτηριστικά των οντοτήτων όσον αφορά τις κινητές μονάδες, μπορεί να αφορούν τον εξοπλισμό ή τύπο τους, ενώ είναι δυνατόν να προσδιοριστεί και η κατάσταση τους την αρχική στιγμή της εξομοίωσης. Όσον αφορά τις υπόλοιπες οντότητες ο χρήστης μπορεί να ορίσει τις κλινικές σε κάθε νοσοκομείο και των αριθμό κλινών σε κάθε μια από αυτές, τη διαθεσιμότητα κλινών σε κάθε μια κλινική, το αν το νοσοκομείο ή ιατρικό κέντρο εφημερεύει, τον εξοπλισμό του νοσοκομείου ή ιατρικού κέντρου κ.α.

11.2.3.4 Στατιστική παρουσίαση περιβάλλοντος

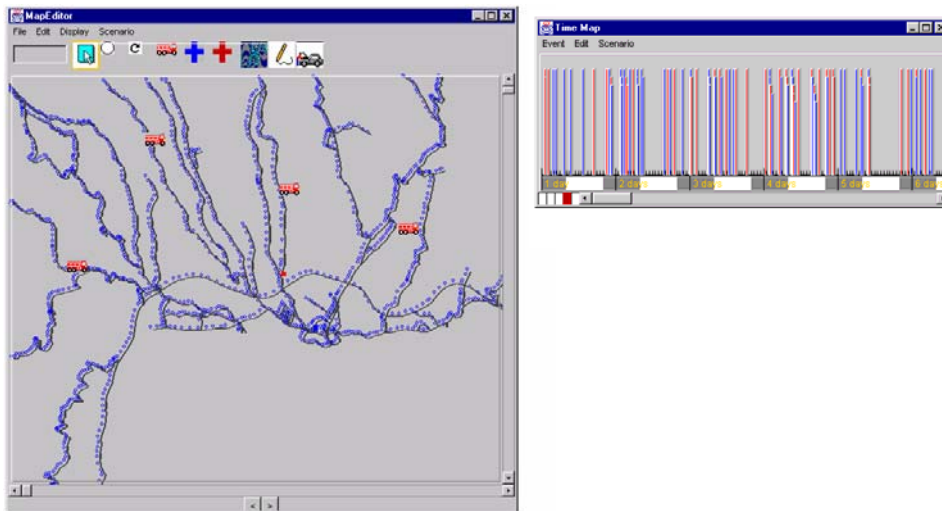
Κατά τη διάρκεια ή μετά το σχεδιασμό του περιβάλλοντος, ο χρήστης είναι δυνατόν να ζητήσει στατιστικά στοιχεία για το περιβάλλον το οποίο σχεδιάζει. Έτσι, είναι δυνατόν να ενημερώνεται για τη διαθεσιμότητα των νοσοκομείων σε μια περιοχή κατά το σχεδιασμό της, ή την κατάσταση των κινητών μονάδων κτλ.

11.2.3.5 Αυτοματισμοί

Ο επεξεργαστής αρχείων σεναρίου δίνει στο χρήστη τη δυνατότητα, να ζητήσει τον αυτόματο σχεδιασμό ενός χάρτη ή την αυτόματη τοποθέτηση γεγονότων ή οντοτήτων στο περιβάλλον κάνοντας έτσι ευκολότερο τον σχεδιασμό ενός περιβάλλοντος. Η αυτόματη σχεδίαση του χάρτη, τοποθέτηση των οντοτήτων και ορισμός γεγονότων, είναι δυνατόν να γίνει είτε με τυχαίο τρόπο είτε ακολουθώντας υποδείξεις του χρήστη (π.χ. αυτόματος ορισμός 100 γεγονότων, όπου το 75% να είναι επείγοντα περιστατικά). Ακόμη ο χρήστης μπορεί να χρησιμοποιήσει τη λειτουργικότητα της αυτόματης ανεύρεσης κόμβων από την εικόνα του χάρτη, ενώ υπάρχει πρόβλεψη για την είσοδο πληροφορίας από GIS, όταν αυτή θα μας είναι διαθέσιμη.

11.2.4 Υλοποίηση

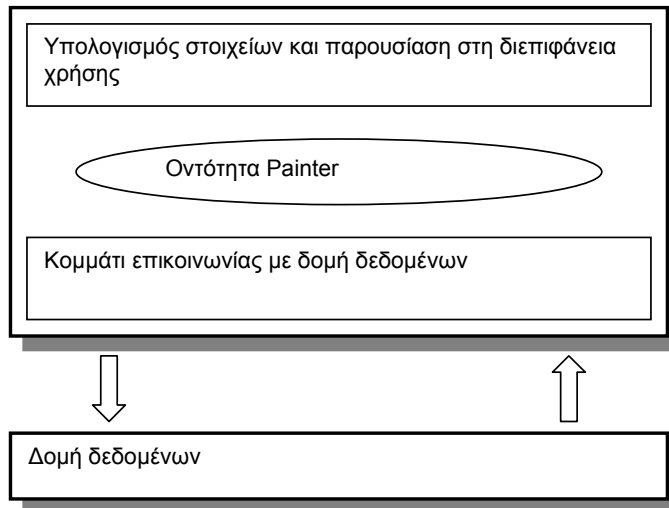
Ο επεξεργαστής αρχείων σεναρίου κρατά σε δομές το γράφο, τη δομή των χρονοπρογραμματιζόμενων γεγονότων, καθώς και όλες τις άλλες πληροφορίες που απαιτούνται για τη δημιουργία του σεναρίου. Η διεπιφάνεια χρήσης χρησιμοποιεί αντικείμενα για την παρουσίαση των δομών αυτών, καθώς και για την τροποποίηση τους σύμφωνα με τα γεγονότα διεπιφάνειας χρήσης τα οποία προέρχονται από το χρήστη. Τα παρακάτω σχήματα (σχήμα 11.3) παρουσιάζουν ένα στάδιο της κατασκευής ενός σεναρίου. Στην πρώτη οθόνη που παρουσιάζεται ο χρήστης έχει κατασκευάσει το γράφο ο οποίος περιγράφει το οδικό δίκτυο



Σχήμα 11.3 Οθόνες του εργαλείου κατασκευής σεναρίων

και έχει τοποθετήσει τέσσερις κινητές μονάδες στα επιθυμητά σημεία. Στη δεύτερη οθόνη παρουσιάζεται μια αναπαράσταση στο χρόνο των γεγονότων τα οποία έχουν προγραμματιστεί να λάβουν μέρος στην εξομίωση.

Για την εύκολη επέκταση της εφαρμογής, καθώς και για τη δυνατότητα επαναχρησιμοποίησης οντοτήτων τόσο στον εξομοιωτή όσο και στη μονάδα διαχείρισης πόρων, τα αντικείμενα είναι όσο το δυνατόν ανεξάρτητα μεταξύ τους, ενώ πολλές φορές περικλείουν ένα ή παραπάνω νήματα ελέγχου (threads of control) τα οποία καθορίζουν τη συμπεριφορά της οντότητας. Για παράδειγμα το αντικείμενο που παρουσιάζει στο χρήστη τα στατιστικά στοιχεία όσον αφορά το περιβάλλον (δρόμοι, κινητές μονάδες, νοσοκομεία, κ.τ.λ.) είναι σχεδιασμένο όπως φαίνεται στο σχήμα (σχήμα 11.4):



Σχήμα 11.4 Δομή ενός αντικειμένου της διεπιφάνειας χρήσης του εργαλείου κατασκευής σεναρίων

Η διεπιφάνεια χρήσης χρησιμοποιεί το αντικείμενο Visualizator για την γραφική αναπαράσταση σε αναπαράσταση των στατιστικών στοιχείων. Το αντικείμενο Visualizator κληρονομεί από την κλάση Component, η οποία είναι αφηρημένη κλάση και περιγράφει γραφικά αντικείμενα. Η κλάση Visualizator περιέχει μεθόδους για τον υπολογισμό των στατιστικών στοιχείων, δομές διεπιφάνειας χρήσης (παράθυρα, πίνακες κ.τ.λ) καθώς και ένα νήμα ελέγχου, το οποίο αντιστοιχίζεται με το αντικείμενο Visualizator και προκαλεί την επανασχεδίαση του κάθε ένα συγκεκριμένο χρονικό διάστημα.

Τα πλεονεκτήματα σχεδιασμού που προαναφέραμε έχουν ως εξής:

(1) επεκτασιμότητα: ο κατασκευαστής της κλάσης Visualizator απαιτεί μόνο μια αναφορά στη δομή του χάρτη για την δημιουργία του αντικειμένου. Έτσι το αντικείμενο που παρουσιάζει τα στατιστικά στοιχεία, είναι δυνατόν να δημιουργηθεί μετά τη δημιουργία της δομής της οποίας την στατιστική πληροφορία υπολογίζει και παρουσιάζει. Παρόμοια ο προγραμματιστής είναι δυνατόν να προσθέσει νέες οντότητες με νέα λειτουργικότητα, κατά πιθανή επέκταση της εφαρμογής στο μέλλον.

(2) επαναχρησιμοποίηση οντοτήτων: η συνάρτηση η οποία σχεδιάζει την πληροφορία στην διεπιφάνεια χρήσης, πραγματοποιεί και τους απαραίτητους υπολογισμούς για τη εξαγωγή των στατιστικών στοιχείων. Αφού η κλάση Visualizator κληρονομεί από την κλάση Component (η οποία περιέχει την αφηρημένη μέθοδο paint() - υπεύθυνη για το σχεδιασμό στην οθόνη του γραφικού αντικειμένου). Η κλάση Painter απαιτεί για την κατασκευή της την αναφορά σε ένα αντικείμενο της κλάσης Component (η κλάση Visualizator είναι τέτοιου τύπου αφού κληρονομεί από την κλάση Component). Προφανώς, η κλάση Painter μπορεί να χρησιμοποιηθεί και για την περιοδική επανασχεδίαση κάποιου άλλου γραφικού οποιουδήποτε αντικειμένου, αφού όλα τα αντικείμενα διεπιφάνειας χρήσης κληρονομούν από την αφηρημένη κλάση Component.

11.3 Εξομοιωτής

11.3.1 Σκοπός

Ο σκοπός της δημιουργίας του εξομοιωτή περιβάλλοντος, είναι η προσομοίωση του περιβάλλοντος επείγουσας προνοσοκομειακής ιατρικής περίθαλψης, για τον έλεγχο της ορθότητας λειτουργίας της μονάδας διαχείρισης πόρων. Δευτερεύοντες σκοποί της δημιουργίας του εξομοιωτή, είναι η εξαγωγή γνώσης για το περιβάλλον (λειτουργία η οποία

γίνεται σε συνεργασία με τη μονάδα διαχείρισης πόρων), καθώς και η επικουρία στην εκπαίδευση του προσωπικού επείγουσας προνοσοκομειακής ιατρικής περίθαλψης.

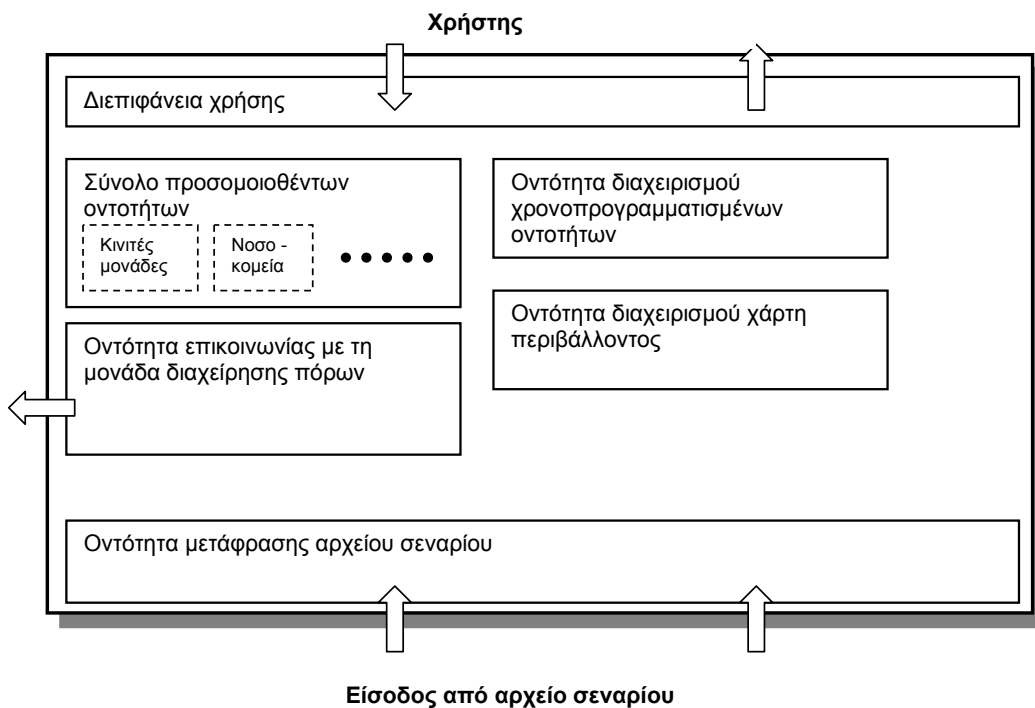
11.3.2 Λειτουργικότητα

Ο εξομοιωτής προσομοιώνει τη συμπεριφορά των οντοτήτων του περιβάλλοντος επείγουσας προνοσοκομειακής ιατρικής περίθαλψης. Σε αυτό συμπεριλαμβάνονται και έμμεσα σχετιζόμενες οντότητες όπως το οδικό δίκτυο, νοσοκομεία και ιατρικά κέντρα. Ο εξομοιωτής προσομοιώνει την εξέλιξη γεγονότων επειγόντων περιστατικών και ιατρικών συμβάντων καθώς και άλλα γεγονότα που είναι δυνατόν να συμβούν στο περιβάλλον και σχετίζονται με τη διαχείριση πόρων του συστήματος. Η συμπεριφορά των οντοτήτων και τα γεγονότα μεταφράζονται σε μηνύματα τα οποία μεταβιβάζονται στη μονάδα διαχείρισης πόρων, υλοποιώντας κατά αυτόν τον τρόπο, μέρος των αισθητηρίων της. Επιπλέον η εξομοιωτής δέχεται μηνύματα από τη μονάδα διαχείρισης πόρων, τα οποία περιέχουν οδηγίες για τη συμπεριφορά των οντοτήτων του συστήματος. Τέλος ο εξομοιωτής μπορεί να χρησιμοποιηθεί για τη γραφική απεικόνιση της θέσης, κίνησης και χαρακτηριστικών των οντοτήτων της εξομοίωσης, καθώς και για την συλλογή στατιστικών στοιχείων τα οποία αφορούν το περιβάλλον και τις οντότητες εξομοίωσης.

11.3.3 Αρχιτεκτονική

11.3.3.1 Γενική επισκόπηση

Η αρχιτεκτονική του εξομοιωτή αποτελείται από τα εξής τμήματα όπως φαίνεται και το σχήμα παρακάτω (σχήμα 11.5):



Σχήμα 11.5 Αρχιτεκτονική του εξομοιωτή

- Διεπιφάνεια χρήσης
- Οντότητα διαχείρισης χρονοπρογραμματισθέντων γεγονότων
- Σύνολο προσομοιοθέντων οντοτήτων
- Οντότητα χάρτη περιβάλλοντος
- Οντότητα επικοινωνίας με τη μονάδα διαχείρισης πόρων
- Οντότητα μετάφρασης αρχείου σεναρίου

Κατά την εκκίνηση της προσομοίωσης ο εξομοιωτής χρησιμοποιεί την οντότητα μετάφρασης αρχείου σεναρίου για την αποκωδικοποίηση του αρχείου και στη συνέχεια δημιουργεί τις οντότητες του χάρτη περιβάλλοντος και των υπόλοιπων αντικείμενων της προσομοίωσης (κινητές μονάδες, νοσοκομεία κ.τ.λ.). Επίσης μετά την ανάγνωση του αρχείου σεναρίου ο εξομοιωτής αρχικοποιεί την οντότητα διαχείρισης χρονοπρογραμματισθέντων γεγονότων.

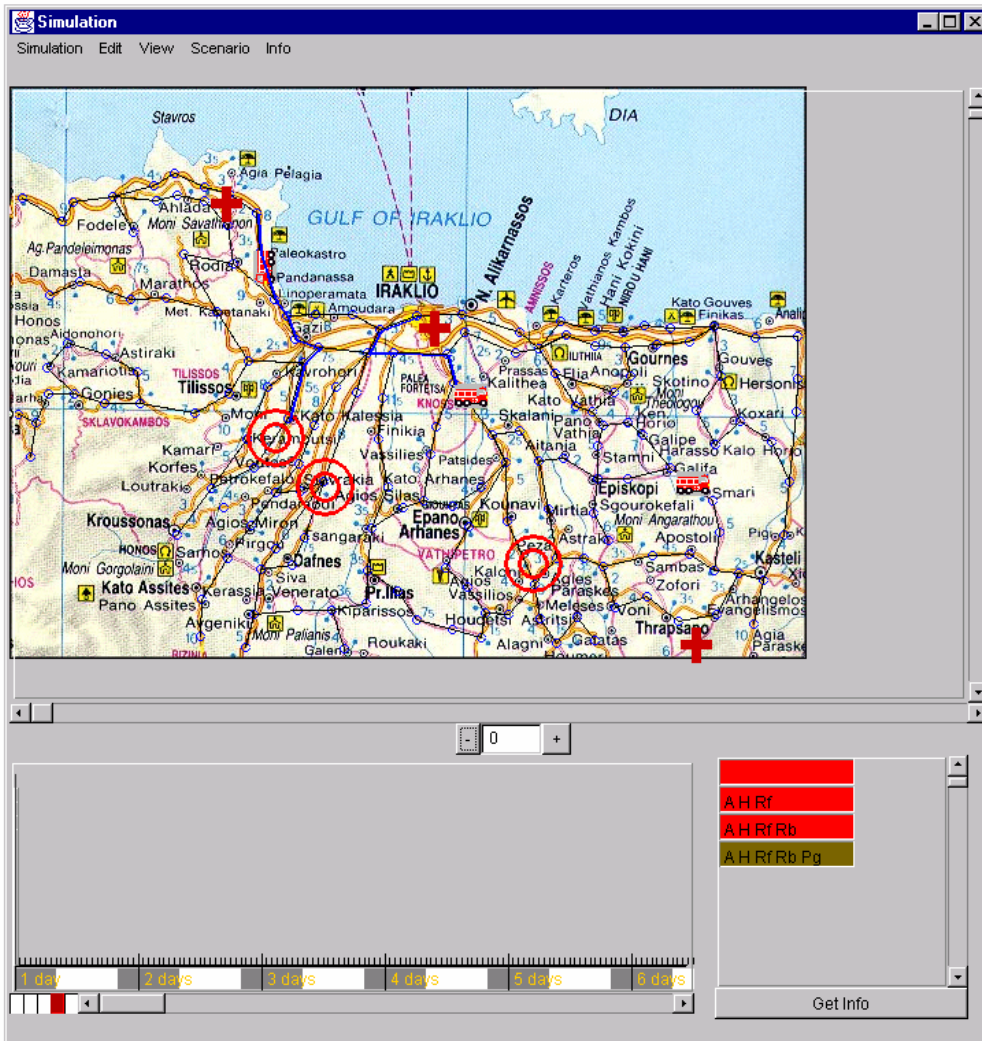
Κατά την εξέλιξη της προσομοίωσης, οι οντότητες που μετέχουν σε αυτή δημιουργούν (με τη συμπεριφορά τους) ορισμένα γεγονότα. Τα γεγονότα αυτά αποτελούν την αναφορά της συμπεριφοράς τους, η οποία χρησιμοποιείται από την μονάδα διαχείρισης πόρων για την παρακολούθηση του περιβάλλοντος εξομοίωσης. Παράλληλα η οντότητα διαχείρισης χρονοπρογραμματισθέντων γεγονότων εξομοιώνει την εξέλιξη των γεγονότων που έχουν χρονοπρογραμματιστεί από το χρήστη.

Τα μηνύματα τα οποία περιέχουν τη συμπεριφορά των οντοτήτων καθώς και τα γεγονότα τα οποία προσομοιώνονται από την οντότητα διαχείρισης χρονοπρογραμματισθέντων γεγονότων διαβιβάζονται στην οντότητα επικοινωνίας με τη μονάδα διαχείρισης πόρων, έτσι ώστε να λάβει γνώση των γεγονότων αυτών. Η οντότητα επικοινωνίας με τη μονάδα διαχείρισης πόρων, μπορεί να λάβει μηνύματα από αυτή τα οποία περιέχουν οδηγίες για ενέργειες που πρέπει να εκτελεστούν από αντικείμενα του περιβάλλοντος επείγουσας προνοσοκομειακής ιατρικής περιθαλψής, και τα θα διαβιβάσει στα αναφερόμενα αντικείμενα.

Το σύνολο των οντοτήτων οι οποίες λαμβάνουν μέρος στην προσομοίωση είναι οργανωμένο σε ένα αντικείμενο το οποίο μπορεί να κρατήσει πολλαπλά ευρετήρια πάνω σε αυτά. Έτσι αυτή τη στιγμή υπάρχουν οργανωμένα σε κατά τύπους ομάδες, για την γρηγορότερη αναφορά σε αυτά. Για παράδειγμα για την ανεύρεση του καταλληλότερου νοσοκομειακού κέντρου για ένα ατύχημα, η αναζήτηση θα γίνει μόνο στις δομές των νοσοκομείων. Παρόμοια ευρετήρια, τα οποία μπορεί να δημιουργούν και λεπτομερέστερες κατηγορίες είναι δυνατόν να δημιουργηθούν, ενώ είναι δυνατή και η δυναμική τους τροποποίηση κατά την εξέλιξη της προσομοίωσης.

Τέλος η οντότητα της διεπιφάνειας χρήσης, παρακολουθεί της δομές της προσομοίωσης και της αναπαριστά με γραφικό τρόπο στην οθόνη του υπολογιστή. Επιπλέον, μπορεί να δώσει στατιστικά ή άλλα στοιχεία στο χρήστη, τα οποία αφορούν τις οντότητες, τα γεγονότα ή την εξέλιξη της προσομοίωσης. Το παρακάτω σχήμα παρουσιάζει μια τυπική οθόνη του προσομοιωτή. Στο σχήμα 11.6 παρατηρεί κανείς το χάρτη και οδικό δίκτυο καθώς και τις κινητές μονάδες από τις οποίες τρεις βρίσκονται σε κίνηση. Οι τροχιές των κινητών μονάδων προς τους προορισμούς τους αποτυπώνονται στο χάρτη με έντονη γραμμή. Τα σημεία των περιστατικών φαίνονται πάνω στο χάρτη με τους έντονους ομόκεντρους κύκλους. Στο κάτω αριστερά τμήμα της οθόνης υπάρχει χώρος για χρονική αναπαράσταση των γεγονότων τα οποία λαμβάνουν χώρα κατά τη διάρκεια της προσομοίωσης, ενώ στο κάτω δεξιό τμήμα της φαίνονται τα περιστατικά τα οποία έχουν γίνει αντιληπτά από το σύστημα. Το χρώμα καθώς και οι συντομογραφίες πάνω στην απεικόνιση των περιστατικών εκφράζουν την τρέχουσα κατάσταση του περιστατικού. Ο χρήστης μπορεί να επιλέξει κάποιο από τα περιστατικά και να ζητήσει από το σύστημα πλήρη αναφορά των ενεργειών οι οποίες έχουν γίνει καθώς και τον χρόνο ο οποίος απαιτήθηκε για τον υπολογισμό της κάθε απόφασης. Επιπλέον το σύστημα στην αναφορά αιτιολογεί τους λόγους για την επιλογή της κάθε απόφασης (πχ. το αίτιο της επιλογής μιας κινητής μονάδας είναι η καταλληλότητα και χρονική εγγύτητα του). Τέλος ο προσομοιωτής είναι δυνατόν να επιτελέσει και κάποιες στοιχειώδεις λειτουργίες ενός γεωγραφικού πληροφοριακού συστήματος, αφού επιλέγοντας οντότητες του περιβάλλοντος

μέσω της διεπιφάνειας χρήσης, μπορεί να προσπελάσει πληροφορία που κρατά το σύστημα για αυτές και οι οποίες προέρχονται από τη βάση δεδομένων του προνοσοκομειακού συστήματος επείγουσας ιατρικής (όπως έχουμε δει σε προηγούμενη ενότητα το σύστημα αποκτά την πληροφορία αυτή μέσω των αισθητηρίων αντικειμένων του).



Σχήμα 11.6 Οθόνη του προσομοιωτή

12 Μελλοντική εργασία

Εδώ περιγράφουμε εργασία η οποία θα μπορούμε να βελτιώσει τόσο την απόδοση του συστήματος, όσο την καλύτερη κατανόηση των αρχών της τεχνολογίας αυτόνομων οντοτήτων με σκοπό τη σχεδίαση καλύτερων συστημάτων.

12.1 Χρήση γλώσσας επικοινωνίας πρακτόρων μεταξύ των οντοτήτων

Χρησιμοποιώντας κάποια γλώσσα επικοινωνίας πρακτόρων για την επικοινωνία των οντοτήτων μεταξύ τους, όσο και του ίδιου του συστήματος με άλλα είναι δυνατόν να επιτύχουμε τα εξής:

- Χρήση άλλων βέλτιστων οντοτήτων για ορισμένες εργασίες του συστήματος. Για παράδειγμα είναι δυνατόν να υπάρχει ένας πράκτορας τρίτου κατασκευαστή, ο οποίος ανάλογα με τα χαρακτηριστικά ενός επεισοδίου να μπορεί να κάνει μια τυπική διάγνωση (η να αντιστοιχίσει το περιστατικό με άλλα παρόμοια), σε πραγματικό χρόνο, και έτσι να είναι σε θέση να επιλέξει με καλύτερο τρόπο τους κατάλληλους πόρους. Σε περίπτωση όπου η απόδοση της οντότητας αυτής είναι καλύτερη από τη δικιά μας θα προτιμούσαμε να χρησιμοποιήσουμε αυτή.
- Χρήση του συστήματος από άλλα. Είναι δυνατόν κάποιος άλλος πράκτορας να θέλει να χρησιμοποιήσει υπηρεσίες τις οποίες προσφέρει το σύστημα (πχ. εύρεση συντομότερης διαδρομής) ή απλά γνώσεις ή πληροφορίες (πχ. μια οντότητα η οποία ψάχνει για χάρτη του οδικού δικτύου μιας περιοχής). Πιο σημαντικό όμως για το πλαίσιο της εφαρμογής μας είναι η δυνατότητα συνεργασίας του συστήματος μας με άλλα συστήματα διαχείρισης πόρων ή συστήματα ιατρικής πληροφορικής. Έτσι στην πρώτη περίπτωση θα μπορούσε το σύστημα επείγουσας προνοσοκομειακής ιατρικής περίθαλψης να συνεργαστεί πχ. με τρίτους φορείς υγείας ή άλλους πχ. αστυνομία. Στη δεύτερη περίπτωση θα ήταν δυνατή αφενός η συνεργασία του συστήματος με άλλα συστήματα υγείας για την ανταλλαγή γνώσης ή η συνεργασία του συστήματος μας με άλλα (πχ. τηλεματικής, εργαστηριακά κ.τ.λ.) για την παροχή υπηρεσιών καλύτερης ποιότητας.

12.2 Παραγωγή σχεδίων δράσης

Προς το παρόν τα σχέδια δράσης των οντοτήτων δημιουργούνται στατικά από τον προγραμματιστή. Θα ήταν ενδιαφέρον να δοκιμάζαμε την βελτιστοποίηση των σχεδίων δράσης προσπαθώντας να τα παράγουμε με δυναμικό τρόπο από αντίστοιχες γνωσιακές οντότητες. Ακόμη η δυναμική παραγωγή σχεδίων δράσης θα μπορούσε να χρησιμεύσει στην καλύτερη διαχείριση των πόρων, έτσι ώστε να υπάρχει καλύτερος προγραμματισμός της λειτουργίας των κινητών μονάδων και διαχείριση τους λαμβάνοντας υπ' όψιν περισσότερες παραμέτρους.

Μια άλλη, και πιο απλή, διάσταση θα ήταν ο καθορισμός του σχεδίου δράσης της κάθε οντότητας μέσω γραφικής διεπιφάνειας χρήσης. Ανάλογα με την ελευθερία επιλογών που θα θέλαμε να δώσουμε στο χρήστη, η εφαρμογή που θα υλοποιούσε την επεξεργασία των σχεδίων δράσης, θα μπορούσε να παράγει σχέδια δράσης οδηγώντας το χρήστη μέσω συγκεκριμένων ερωτήσεων ή να επιτρέπει στον ίδιο το χρήστη να σχεδιάσει το διάγραμμα ροής.

12.3 Σχεδίαση οντότητας ελέγχου σημάτων κυκλοφορίας

Είναι δυνατόν να υλοποιήσουμε μια οντότητα η οποία δεδομένου του ότι το σύστημα μας έχει πρόσβαση στα ηλεκτρικά σήματα κυκλοφορίας (σηματοδότες), θα μπορεί να τα ελέγχει για την ταχύτερη διέλευση των κινητών μονάδων τα οποία εξυπηρετούν επείγοντα περιστατικά.

Η λογική της οντότητα θα βασίζεται σε μια πρόβλεψη όσον αφορά το άμεσο μέλλον για το που θα βρίσκεται η οντότητα μέσα στα επόμενα λεπτά ώστε να θέσουμε το σηματοδότη για εκείνη τη χρονική περίοδο σε τέτοια κατάσταση ώστε να είναι ελεύθερη η διέλευση του ασθενοφόρου. Ωστόσο ο αλγόριθμος θα πρέπει να λαμβάνει υπ' όψιν τις συνέπειες που θα έχει μια τέτοια ενέργεια και στο υπόλοιπο δίκτυο σηματοδότησης.

Η υλοποίηση και χρήση μιας τέτοιας οντότητας θα είχε περισσότερο νόημα σε άλλα συστήματα επείγουσας προνοσοκομειακής ιατρικής περίθαλψης τα οποία συνεργάζονται με αρμόδιους φορείς για την πραγματοποίηση τέτοιων ενεργειών.

12.4 Βελτίωση αλγορίθμου ευρετηριασμού του χάρτη του οδικού δικτύου

Όπως έχει διατυπωθεί και στην παρουσίαση των μεθοδολογιών ευρέσεως συντομότερης διαδρομής θα θέλαμε να δημιουργήσουμε ευρετήριο περιοχών του γράφου που περιγράφει το οδικό δίκτυο ταχύτερη εύρεση των συντομότερων διαδρομών. Πιθανός τρόπος εύρεσης των ευρετηριαζόμενων περιοχών να είναι κάποιος αλγόριθμος μηχανικής μάθησης, ο οποίος θα προσαρμόζει την ευρετηρίαση πάνω στο συγκεκριμένο γράφο ή άλλος γεωμετρική φύσεων που να βελτιστοποιεί τη διαδικασία αυτή.

Επιπλέον κάτι άλλο που θα μπορούσε να γίνει θα ήταν η ανάπτυξη οντότητας η οποία με πιθανοκρατικό τρόπο και βασισμένη σε στατιστικά δεδομένα να έκανε εκτιμήσεις για το φόρτο των ακμών του γράφου στο άμεσο μέλλον. Κατά αυτό τον τρόπο θα λαμβανόταν υπ' όψιν ο φόρτος που θα έχουν οι δρόμοι τη στιγμή που πρόκειται να τους διασχίσει η κινητή μονάδα και όχι αυτός που έχουν τη στιγμή υπολογισμού της συντομότερης διαδρομής.

13 Συμπεράσματα

Στην παράγραφο αυτή θα παρουσιάσουμε συμπεράσματα τα οποία αφορούν τη σχεδίαση λογισμικού με τη χρήση αυτόνομων οντοτήτων και τα οποία προέκυψαν τόσο από τη μελέτη της σχετικής βιβλιογραφίας όσο και από τη διαδικασία σχεδιασμού και υλοποίησης του συστήματος.

Το βασικό συμπέρασμα το οποίο εξήχθηκε από αυτή την εργασία είναι το ότι η σύνθετη και αυτόνομη συμπεριφορά είναι αποτέλεσμα του συγκερασμού της επικοινωνίας και δράσης πολλών απλούστερων, αλλά εξειδικευμένων οντοτήτων. Η συμπεριφορά που από τον εξωτερικό παρατηρητή είναι δυνατόν να χαρακτηριστεί ως ευφυής είναι δυνατόν να συντίθεται από πολύ απλές ενέργειες. Για την πραγματοποίηση σύνθετης συμπεριφοράς, όμως φαίνεται πως απαιτείται και δίκτυο επικοινωνίας οντοτήτων μεγάλης πολυπλοκότητας αποτελούμενο από οντότητες στοιχειώδους ευφυΐας. Η παρατηρούμενη αυτόνομη και ευφυής συμπεριφορά είναι χαρακτηριστικό το οποίο αναδεικνύεται από τη σύνθεση της συμπεριφοράς όλων των στοιχειωδών οντοτήτων και μπορεί να χαρακτηριστεί ως ολιστικό γνώρισμα του συστήματος.

Τόσο προσαρμοστικότητα όσο και η ποικιλομορφία αντιδράσεων και ενεργειών, βασίζεται στην ανάπτυξη του συστήματος από πολλές απλές οντότητες αντί των λίγων και σύνθετων. Μια μονολιθική οντότητα έχει απόλυτα προβλέψιμη συμπεριφορά, ενώ χρησιμοποιώντας ένα σύνολο οντοτήτων, μπορεί κανείς να συνδυάσει τις συμπεριφορές τους παράγοντας μεγαλύτερο εύρος δυνατών ενεργειών. Παράλληλα το μεγάλο πλήθος και απλότητα των οντοτήτων (fine grain) είναι δυνατόν να κάνει το παραγόμενο σύστημα ευκολότερα προσαρμόσιμο μιας και ένα πολύπλοκο σύστημα έχει και περισσότερες προϋποθέσεις προσαρμογής. Δύο παραδείγματα ίσως να βοηθήσουν στην καλύτερη κατανόηση. Όσον αφορά την ποικιλομορφία στη συμπεριφορά θα δανειστούμε ένα παράδειγμα από τη ρομποτική, όπου όσο μεγαλύτερος ο βαθμός ελευθερίας ενός βραχίονα (πράγμα που σημαίνει ότι θα αποτελείται και από περισσότερα τμήματα), τόσο περισσότερες είναι οι δυνατές κινήσεις οι οποίες μπορεί να επιτύχει. Όσον αφορά την προσαρμοστικότητα μπορεί να αναλογιστεί κανείς το παράδειγμα ενός εργάτη ο οποίος προσπαθεί να εξομαλύνει μια επιφάνεια η οποία έχει μια εσοχή. Προσπαθώντας να λύσει το πρόβλημα κλείνοντας την εσοχή με μια πέτρα θα πρέπει να ψάξει πάρα πολύ ώστε να βρει την κατάλληλη και ίσως και να μην τα καταφέρει ποτέ, ωστόσο χρησιμοποιώντας κάποιο άλλο υλικό πχ. άμμο θα καταφέρει καλύτερο αποτέλεσμα.

Για την ανάπτυξη πραγματικά ευφυούς συμπεριφοράς ή τουλάχιστο συμπεριφοράς η οποία να έχει ευφυή χαρακτηριστικά δεν απαιτείται μόνο ενημέρωση των εσωτερικών γνωσιακών δομών των οντοτήτων, αλλά προσαρμογή και του τρόπου επικοινωνίας των οντοτήτων. Επίσης η εξειδίκευση της κάθε οντότητας παίζει πολύ σημαντικό ρόλο στη βελτίωση της απόδοσης των οντοτήτων, όμως από την άλλη πλευρά πρέπει να υπάρχουν κάποιες ομοιότητες τόσο στη σχεδίαση των αυτόνομων οντοτήτων όσο και στον τρόπο εξέλιξης τους. Κατά αυτό τον τρόπο, ίσως να γίνει δυνατή η χρήση συμπερασμάτων τα οποία εξήχθηκαν από κάποιο πεδίο γνώσης σε άλλα.

Η προσαρμογή τόσο των αισθητηρίων όσο και των υπόλοιπων οντοτήτων στην εκάστοτε τρέχουσα κατάσταση καθώς και η εστίαση της προσοχής πρέπει να είναι μεταξύ των επιθυμητών χαρακτηριστικών ενός ευφυούς συστήματος, έτσι ώστε να έχει συμπεριφορά ανάλογη τόσο της κατάστασης του περιβάλλοντος όσο και της εσωτερικής κατάστασης στην οποία βρίσκεται.

Ως γενικό συμπέρασμα μπορεί να πει κανείς πως παρατηρώντας και αντιγράφοντας μηχανισμούς που μπορεί να βρει στη φύση είναι δυνατόν να υλοποιήσει οντότητες με ευφυέστερη συμπεριφορά. Για να κάνει κάτι τέτοιο όμως η ίδιος οι οντότητες θα πρέπει να απαρτίζονται από στοιχειώδη δομικά στοιχεία, διατεταγμένα με τέτοιο τρόπο όμως ώστε να εξειδικεύουν της συμπεριφορά της κάθε οντότητας. Η ομοιότητα στη σχεδίαση και η τήρηση κοινών αρχών λειτουργίας και επικοινωνίας, κάνει και ευκολότερο το συνδυασμό και επικοινωνία των επί μέρους τμημάτων μεταξύ τους. Επιπλέον απαιτείται μεγάλος αριθμός και πολυπλοκότητα διασύνδεσης και επικοινωνίας μεταξύ των οντοτήτων ώστε να υπάρχει έστω

και θεωρητικά η έννοια της αυτόνομης συμπεριφοράς (σε ένα σύστημα που τρέχει σε έναν υπολογιστή είναι δύσκολο να υπάρξει μη προβλέψιμη διεργασία).

Όσον αφορά τη σχεδίαση του τρόπου εξαγωγής συμπερασμάτων και επιλογής ενέργειας αν και οι αλγοριθμικές προσεγγίσεις συνήθως δίνουν αυστηρά σωστό αποτέλεσμα σε κάποιο συγκεκριμένο πρόβλημα, απαιτούν έμφυτη γνώση που δίδεται στο σύστημα από τον προγραμματιστή, ενώ παρατηρώντας κανείς την ευφυή συμπεριφορά έμβιων οντοτήτων, βλέπει πως η μάθηση σε αυτές τις περιπτώσεις είναι ευρύτερης εμβέλειας, όσον αφορά τα διαφορετικά πεδία γνώσης στα οποία μπορεί να εφαρμοστεί. Το ίδιο ισχύει και για το δίκτυο επικοινωνίας μεταξύ των οντοτήτων το οποίο ίσως και να μην έπρεπε να είναι τόσο αυστηρά καθορισμένο δίνοντας έτσι δυνατότητα πολλαπλών τρόπων χρήσεως του καθώς και τη δυνατότητα εξέλιξης του.

Από την άλλη πλευρά όμως, ο στόχος είναι να δημιουργήσουμε συστήματα (η αλλιώς εργακεία) τα οποία αφενός να κάνουν χωρίς σφάλματα τις εργασίες οι οποίες τους ανατίθενται, πράγμα το οποίο δεν συμβαίνει σε οντότητες φυσικής νοημοσύνης, και μάλιστα με συγκεκριμένο τρόπο. Εξάλλου σκοπός της τεχνολογίας των αυτόνομων πρακτόρων είναι η περάτωση κάποιας συγκεκριμένης εργασίας η οποία τους ανατίθεται. Καταλαβαίνουμε πως ακόμα και με τη δυναμική παραγωγή σχεδίων δράσης, και ειδικά εάν αυτή λειτουργεί με αλγοριθμικό τρόπο, η συμπεριφορά του συστήματος θα είναι πάντα προβλέψιμη. Ωστόσο πάλι δεν μπορεί να πει κανείς πως η συμπεριφορά ενός έμβιου όντος δεν είναι (αν δει κανείς τη νοητική εργασία ως αποτέλεσμα ενός δικτύου νευρικών κυττάρων), και κάτι τέτοιο μπορεί να παρατηρηθεί σε πολύ απλές μορφές ζωής. Σίγουρα πάντως συστήματα τα οποία αποτελούνται από αυτόνομες οντότητες είναι δυνατό να προσαρμοστούν και να παρουσιάσουν ευφυέστερη συμπεριφορά από άλλα μονολιθικά συστήματα. Αυτό μάλλον είναι κιόλας αυτό που μας ενδιαφέρει.

Βιβλιογραφία

- [1] Goscinski A., "Distributed Operating Systems, The Logical Design", Addison Wesley, 1991.
- [2] Davies P., "God and the New Physics", Simon & Schuster, 1984, ISBN: 0671528068
- [3] Deneubourg J. L., Goss S., Franks N., Sendova – Franks A., Detrain C., Chretien L., "The Dynamics of Collective Sorting: Robot – like Ants and Ant – like Robots", In: From Animals to Animats, Proceedings of the First International Conference on the Simulation of Adaptive Behavior, edited by Meyer J.-A. & Wilson S. W., MIT Press / Bradford Books, 1991
- [4] Deneubourg J. L., Theraulaz G. & Beckers R., "Swarm – Made Architectures", In: Toward a Practice of Autonomous Systems, Proceedings of the First European Conference on Artificial Life, edited by F. J. Varela & P. Bourguine, MIT Press / Bradford Books, 1992.
- [5] Foner L., "What is an agent anyway? A Sociological Case Study", In: The Proceedings of the First International Conference on Autonomous Agents (AA '97)
- [6] Genesereth M. R. & Ketchpel S. P., "Software Agents", Communications of the ACM, v.37, n.7, July 1994, p.48-53.
- [7]: B. Hayes-Roth. "An Architecture for Adaptive Intelligent Systems". Artificial Intelligence: Special Issue on Agents and Interactivity, 72(1995): 329-365, 1995.
- [8] Kaelbling and Rosenchein, 1990 "Action and Planning in Embedded System" Robotics and Autonomous Systems, Vol 6, No 1-2, pg. 35-48; 1990
- [9] Firby R. 1987 "An Investigation into Reactive Planning in Complex Domains". In Sixth National Conference on Artificial Intelligence; 1987
- [10] Ernman L. et al. "The Hearsay-II speech-understanding System: Integrating knowledge to resolve uncertainty". Computing Surveys, 12, 213-253, 1980
- [11] Brooks R.A., "Intelligence without Reason", Computers and Thought lecture, Proceedings of IJCAI-91, Sidney Australia, 1991.
- [12] Sucjman L.A., "Plans and Situated Actions: The Problem of Human – Machine Communication", Cambridge University Press, 1991.
- [13] Horswill I, "Characterizing Adaptation by Constraint", In: Toward a Practice of Autonomous Systems, Proceedings of the First European Conference on Artificial Life, edited by F.J. Varela & P.Bourguine. MIT Press / Bradford Books, 1992.
- [14] Maes P. & Kozierok R., "Learning Interface Agents", Proceedings of AAAI-93, the Eleventh National Conference on Artificial Intelligence, MIT Press, 1993.
- [15] Brooks R.A., "Artificial Life and Real Robots", In: : Toward a Practice of Autonomous Systems, Proceedings of the First European Conference on Artificial Life, edited by Meyer J.-A & Wilson S. W., MIT Press / Bradford Books, 1991.
- [16] Resnick M., "Beyond the Centralized Mindset: Explorations in Massively Parallel Microworlds", PhD Thesis, MIT Media – Laboratory Epistemology and Learning Group, 1992.
- [17] Agre P. E. "The Dynamic Structure of Everyday Life", Cambridge University Press, 1991.
- [18] Mayes P., "Situated Agents Can Have Goals", In: Designing Autonomous Agents: Theory and Practice from Biology to Engineering and Back, edited by P.Mayes, MIT Press / Bradford Books, 1990.

- [19] Malone T. W., Fikes R.E., Grant K. R., & Howard M.T., Enterprise: "A Market – Like Task Scheduler for Distributed Computing Environments", In: The Ecology of Computation, edited by B. Huberman, North – Holland, 1988.
- [20] Brooks R. A., "A Robust Layered Control System for a Mobile Robot", IEEE Journal of Robotics and Automation, RA – 2, April 1986.
- [21] Tyrrell T., "Computational Mechanisms for Action Selection", PhD Thesis, King's College, Cambridge, 1989.
- [22] Chapman D., "Vision, Instruction and Action", MIT Press, 1992.
- [23] Beer R. Chiel H. & Sterling L., "A Biological Perspective on Autonomous Agent Design", In: Designing Autonomous Agents: Theory and Practice from Biology to Engineering and Back, edited by P. Mayes, MIT Press / Bradford Books, 1990.
- [24] Connell J., "Minimalist Mobile Robotics", Academic Press, 1990.
- [25] Kaelbling L. P. & Rosenschein S. "Action and Planning in Embedded Agents", In: Designing Autonomous Agents: Theory and Practice from Biology to Engineering and Back, edited by P. Mayes, MIT Press / Bradford Books, 1990.
- [26] Nilsson N. J., "Toward Agent Programs with Circuit Semantics", Department of Computer Science, Report number STAN-CS-92-1412, Stanford University, January 1992.
- [27] Maes P., "A Bottom – Up Mechanism for Behavior – Selection in an Artificial Creature", In: From Animals to Animats, Proceedings of the First International Conference on the Simulation of Adaptive Behavior, edited by Meyer J.-A. & Wilson S. W., MIT Press / Bradford Books, 1991
- [28] Travers M., "Animal Construction Kits", In: Artificial Life, edited by Langton C., Addison Wesley, 1989.
- [29] Blumberg B., "Action Selection in Hamsterdam: Lessons from Ethology", Submitted to the Third International Conference on the Simulation of Adaptive Behavior, Brighton, August 1994.
- [30] Rossenblat J., & Payton D., "A Fine – Grained Alternative to the Subsumption Architecture for Mobile Robot Control", In: Proceedings of the International Joint Conference on Neural Networks, IJCNN, Washington DC, June 1989.
- [31] Sutton R. S., "Integrated Architectures for Learning, Planning and Reacting based on Approximating Dynamic Programming", Proceedings of the Seventh International Conference in Machine Learning, 1990
- [32] Sutton R. S., "Reinforcement Learning Architectures for Animats", In: From Animals to Animats, Proceedings of the First International Conference on the Simulation of Adaptive Behavior, edited by Meyer J.-A. & Wilson S. W., MIT Press / Bradford Books, 1991
- [33] Kaelbling L. P., "Learning in an Embedded Systems", MIT Press, 1993.
- [34] Watkins C., "Learning from Delayed Rewards", PhD Thesis, King's College, Cambridge, 1989.
- [35] Booker L., "Classifier Systems that Learn Internal World Models", Machine Learning Journal, Volume 1, Number 2, 3, 1988.
- [36] Holland J. H., "Escaping Brittleness: the Possibilities of General – Purpose Learning Algorithms applied to Parallel Rule – Based Systems", In: Machine Learning, an Artificial Intelligence Approach, Volume II, edited by R. S. Michalski, J. G., Carbonell & T. M. Mitchell, Morgan Kaufmann, 1986.
- [37] Wilson S. W., "Knowledge Growth in an Artificial Animal", In: Proceeding of the First International Conference on Generic Algorithms and their Applications, edited by Greffenstein, Lawrence Erlbaum Associates, 1985.
- [38] Drescher G. L. "Made – Up Minds: A Constructivist Approach to Artificial Intelligence", MIT Press, 1991.
- [39] Maes P., "Learning Behavior Networks from Experience", In: Toward a Practice of Autonomous Systems, Proceedings of the First European Conference on Artificial Life, edited by F.J. Varela & P. Bourgine. MIT Press / Bradford Books, 1992.

- [40] Payton D. W., Keirse D., Krozel J. & Rosenblatt K., "Do Whatever Works: A Robust Approach to Fault – Tolerant Autonomous Control", Journal of Applied Intelligence, Vol. 3, 1992.
- [41] Franklin S., "Agent Oriented Programming", Artificial Intelligence, vol 60, 51-92, 1993
- [42] Tanebaum A. S., "Modern Operating Systems" Vol 1,2 Hanser – Prentice Hall London 1994
- [43] Wooldrige M., Jennigs N. R., "Intelligent Agnets: Theory and Practice", Knowledge Engineering Review, vol. 10(2), 115-152, 1995
- [44] Welch B. B., "Practical Programming in Tcl & Tk", Prentice Hall, 1997, ISBN: 0136168302
- [45] Snodgrass R., "The Interface Description Language: Definition and Use", Computer Science Press, North Carolina 1989
- [46] Michalski S. R., "Machine Learning : An Artificial Intelligence Approach", Springer Verlag, 1984, ISBN: 0387132988
- [47] Laplante A. P., "Real-Time Systems Design and Analysis : An Engineer's Handbook", Published by IEEE, 1997, ISBN: 0780334000
- [48] Cormen H. T., Leiserson E. C., Rivest L. R., "Introduction to Algorithms", MIT Electrical Engineering and Computer Science Series, MIT Press, 1990, ISBN: 0262031418
- [49] Hayes-Roth, Barbara, "Architectural Foundations for Real -Time Performance in Intelligent Agents", The Journal of Real -Time Systems, Vol. 2, 1990. Available from: Open literature
- [50] Brown K., Singh S, "M-UDP: UDP for Mobile Networks", In: ACM CCR, 1996.
- [51] Davidson J., "An Introduction to TCP/IP", Springer Verlag, 1988, ISBN: 038796651X
- [52] Jump N. D., "AutoCAD Programming", Computer Graphics Technology and Management Series, 1989, ISBN 0830693939
- [53] Stroustrup B., "The C++ Programming Language", Addison-Wesley, 1997, ISBN: 0201889544
- [54] Medical Priority Consultants, "Advanced Medical Priority Dispatch System", Medical Priority Consultants Inc., 1990
- [55] Shumaker M. T., Madsen A. D., "AutoCAD and Its Applications", Goodheart - Willcox Co. 1991
- [56] Bradshaw M. J., "Software Agents", MIT Press, 1997
- [57] Leick A., "GPS Satellite Surveing", John Willey & Sons 1995
- [58] Kosko B., "The New Science of Fuzzy Logic", Hyperion Books, 1993

Παράρτημα Α Πειραματικά αποτελέσματα

Στην ενότητα αυτή θα παρουσιάσουμε πειραματικά αποτελέσματα τα οποία παρουσιάζουν τη λειτουργία τμημάτων (ή οντοτήτων) του συστήματος το οποίο αναπτύχθηκε. Θα δούμε τη λειτουργία του αλγόριθμου απλοποίησης γράφων και την απόδοση του αλγόριθμου εύρεσης συντομότερης διαδρομής βάση του απλοποιημένου γράφου. Επίσης θα παρουσιάσουμε τη λειτουργία εύρεσης περιοχών υψηλής συχνότητας εμφάνισης περιστατικών και αλγόριθμο εύρεσης περιπολιών σε αυτές.

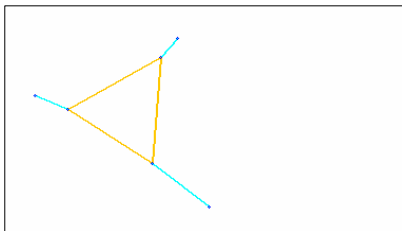
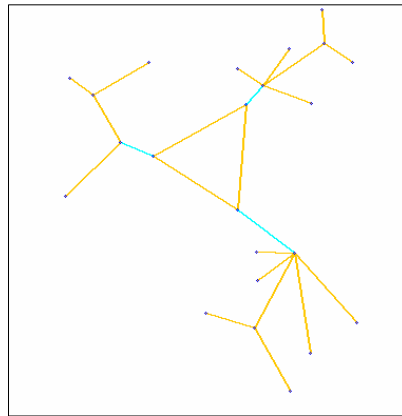
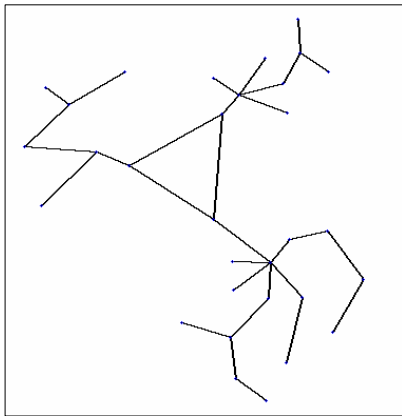
Απλοποίηση γράφου

Θα παρουσιάσουμε δύο παραδείγματα. Στο πρώτο θα δούμε την εφαρμογή του αλγόριθμου σε ένα μικρό γράφο, ενώ στο δεύτερο παράδειγμα σε γράφο το οποίο περιγράφει το οδικό δίκτυο του νομού του Ηρακλείου της Κρήτης.

Παράδειγμα 1

Τα επόμενα σχήματα (σχήμα Α1) παρουσιάζουν με τη σειρά εμφάνισης:

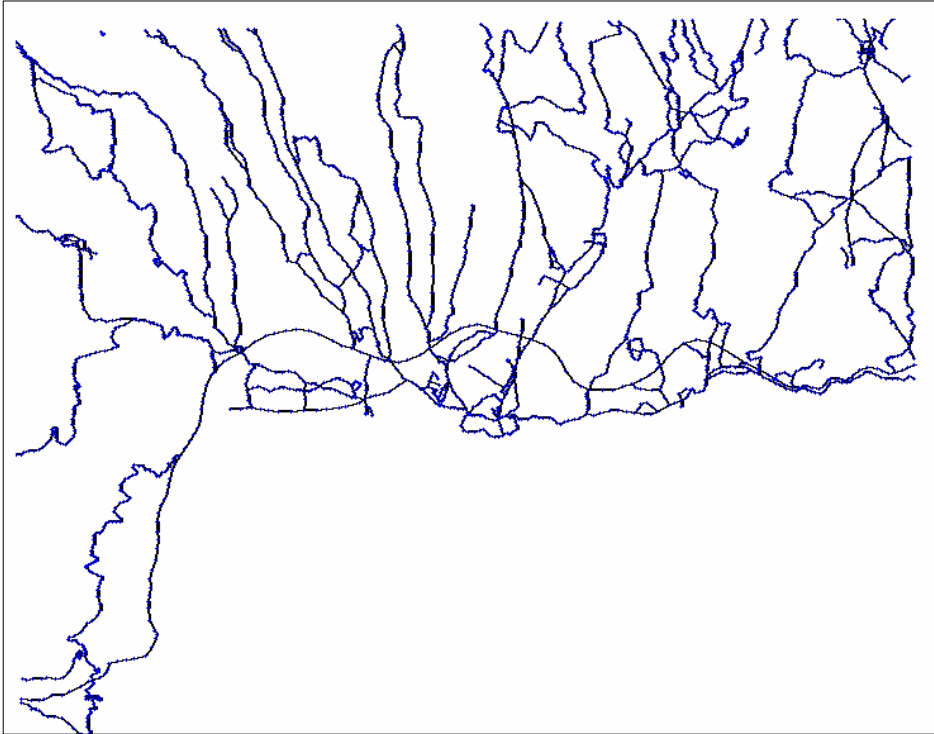
1. το πραγματικό γράφο
2. την απλοποίηση του σε πρώτο επίπεδο ενώνοντας συνεχείς ακμές σε μια
3. τον αφηρημένο γράφο ο οποίος προκύπτει από τον προηγούμενο απαλείφοντας απομονωμένους υπογράφους του οι οποίοι εμφανίζουν δενδρική μορφή



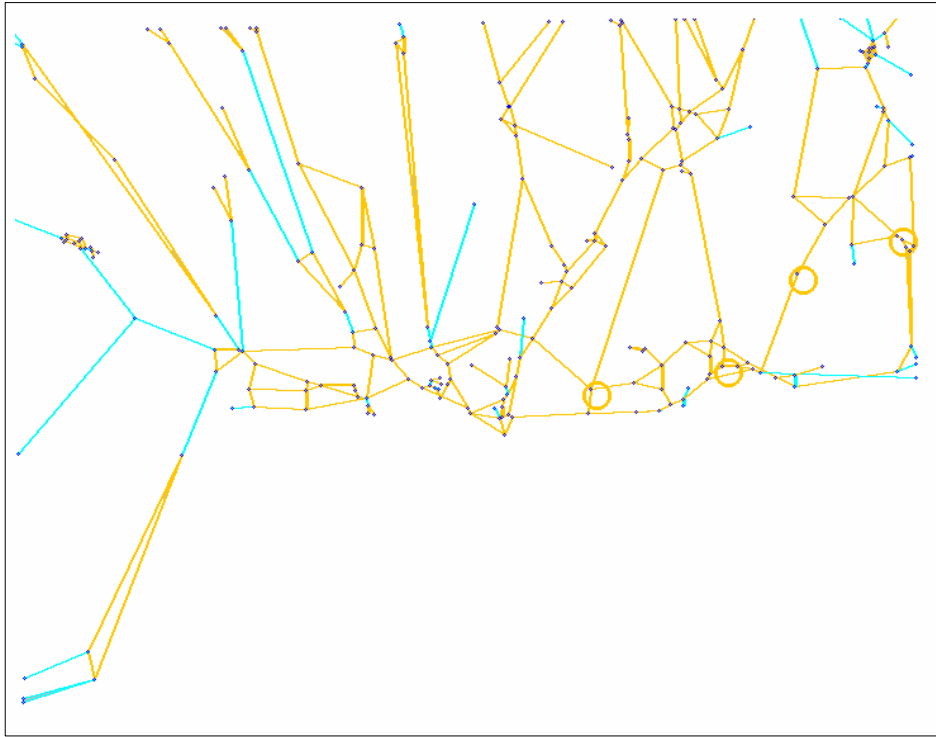
Σχήμα Α1 Παράδειγμα απλοποίησης γράφου

Παράδειγμα 2

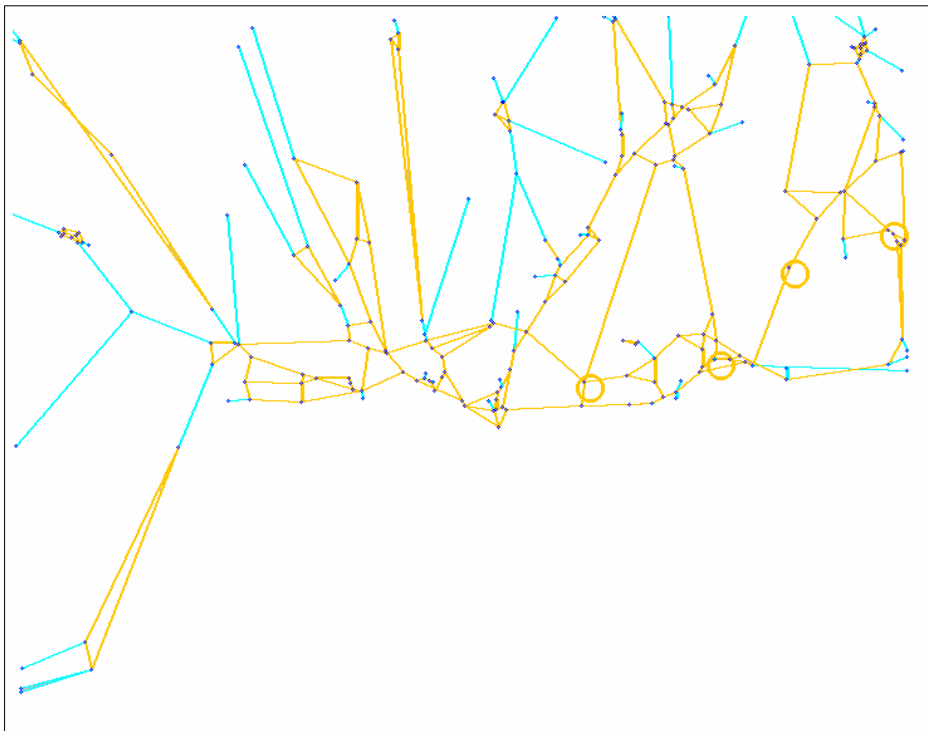
Με όμοια σειρά εμφανίζονται οι εικόνες (βλ. σχήμα A2, A3, A4, A4). Το τελευταίο σχήμα παρουσιάζει πραγματικό και απλοποιημένο γράφο ταυτόχρονα. Ο παρόν γράφος απεικονίζει το οδικό δίκτυο του νομού Ηρακλείου Κρήτης.



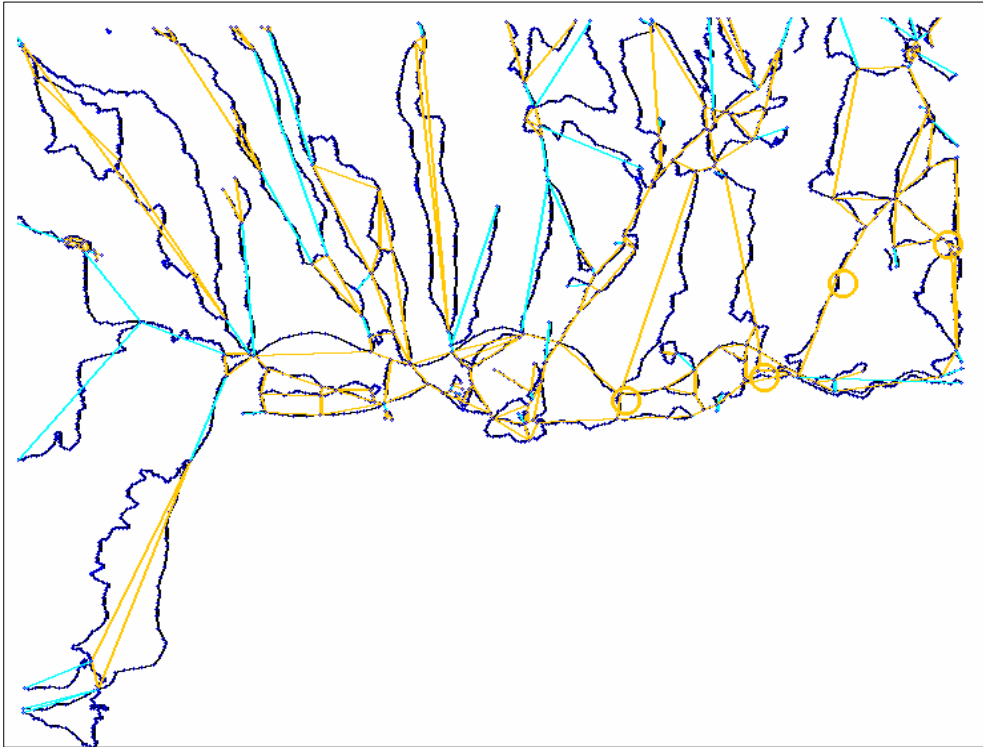
Σχήμα A2 Αρχικός γράφος



Σχήμα A3 Ο γράφος μετά την πρώτου βαθμού απλοποίηση του.



Σχήμα A4 Ο γράφος μετά την δευτέρου βαθμού απλοποίηση του



Σχήμα A5 Αρχικός και απλοποιημένος γράφος μαζί.

Εύρεση συντομότερης διαδρομής

Θα μελετήσουμε τρία παραδείγματα της απόδοσης της μεθοδολογίας εύρεσης συντομότερης διαδρομής που χρησιμοποιεί η αντίστοιχη οντότητα του συστήματος. Θα δούμε πρώτα τα μεγέθη των γράφων πάνω στα οποία βασίζονται τα παραδείγματα, έτσι ώστε να κατανοήσουμε τα πειραματικά αποτελέσματα. Θα μετράμε την περιεκτικότητα ενός γράφου σε δέντρα υπολογίζοντας το ποσοστό των κόμβων του πρώτου επιπέδου απλοποίησης γράφου προς οι οποίοι ανήκουν σε δέντρα προς τον ολικό αριθμό κόμβων του.

Γράφος 1:

Ο γράφος αποτελείται από 1536 κόμβους και 1558 ακμές. Ο πρώτου επιπέδου απλοποίησης γράφος 106 κόμβους και 128 ακμές αντίστοιχα, ενώ ο μεγαλύτερου επιπέδου αφαίρεσης γράφος 82 κόμβους και 104 ακμές. Η περιεκτικότητα του γράφου σε δέντρα είναι 46%.

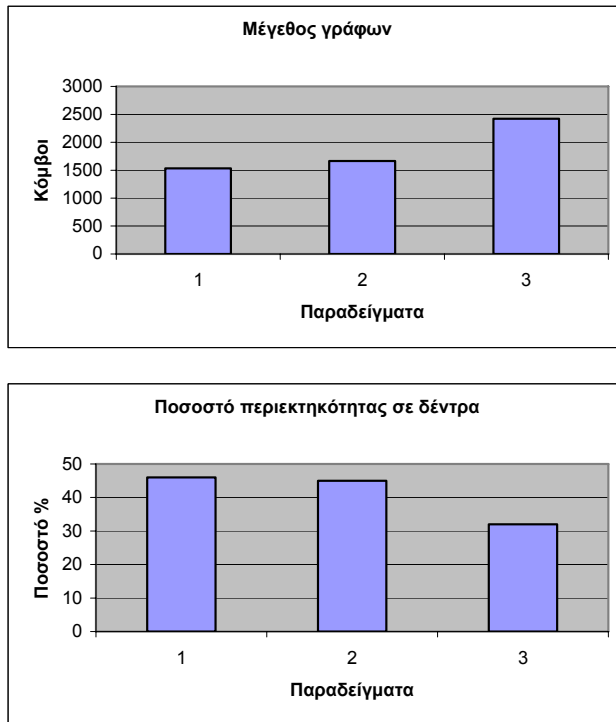
Γράφος 2:

Ο γράφος αποτελείται από 1667 κόμβους και 1697 ακμές. Ο πρώτου επιπέδου απλοποίησης γράφος 124 κόμβους και 155 ακμές αντίστοιχα, ενώ ο μεγαλύτερου επιπέδου αφαίρεσης γράφος 96 κόμβους και 127 ακμές. Η περιεκτικότητα του γράφου σε δέντρα είναι 45%.

Γράφος 3:

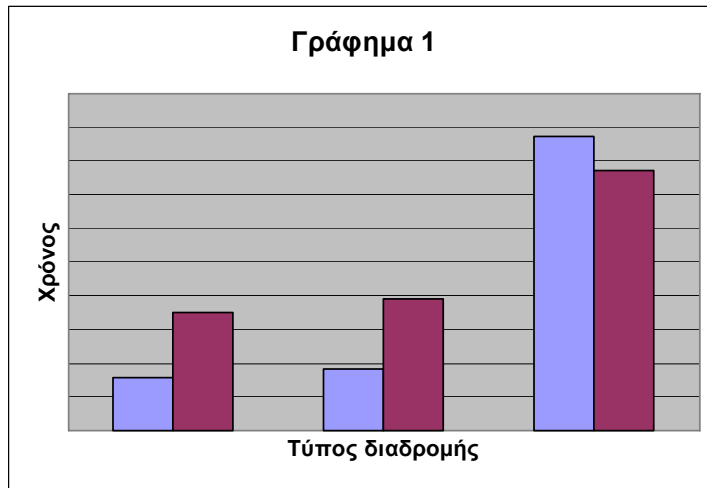
Ο γράφος αποτελείται από 2421 κόμβους και 2497 ακμές. Ο πρώτου επιπέδου απλοποίησης γράφος 249 κόμβους και 328 ακμές αντίστοιχα, ενώ ο μεγαλύτερου επιπέδου αφαίρεσης γράφος 222 κόμβους και 301 ακμές. Η περιεκτικότητα του γράφου σε δέντρα είναι 32%.

Παρατηρούμε πως στα τρία αυτά παραδείγματα αυξάνουμε το μέγεθος του γράφου και μειώνουμε το ποσοστό περιεκτικότητας του γράφου σε δέντρα. Τα δύο παρακάτω γραφήματα παρουσιάζουν τις δύο αυτές ποσότητες (βλ. σχήμα Α6).



Σχήμα Α6 Δύο γραφήματα τα οποία παρουσιάζουν συγκριτικά το μέγεθος των γράφων των τριών παραδειγμάτων και το ποσοστό περιεκτικότητας τους σε δέντρα.

Λόγω της σταδιακής μείωσης της περιεκτικότητας των γράφων σε δέντρα περιμένουμε πως η απλοποίηση του γράφου θα παίζει όλο και μικρότερο ρόλο στην επιτάχυνση του υπολογισμού, ενώ είναι δυνατόν να τον επιβραδύνει κιόλας. Το Γράφημα 1 παρουσιάζει τη σύγκριση αυτή. Με φωτεινότερο χρώμα παρουσιάζονται οι χρόνοι της μεθόδου μας και με λιγότερο φωτεινό αυτοί της μεθόδου του Dijkstra.

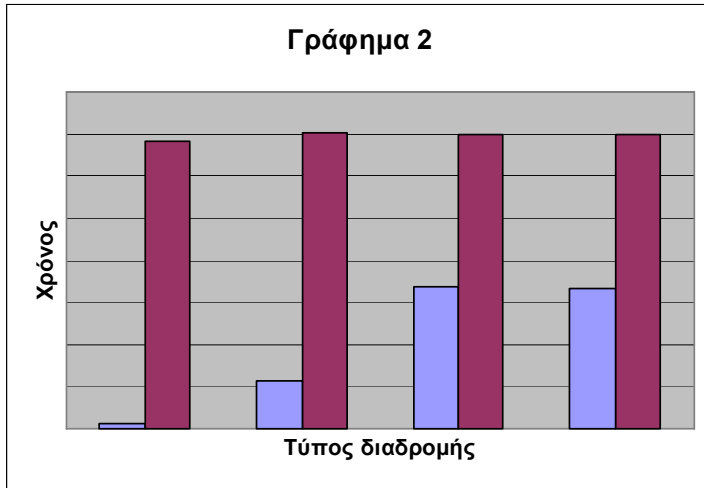


Γράφημα 1 Σύγκριση χρόνων εύρεσης συντομότερης διαδρομής με χρήση απλοποιημένου γράφου και με τον αλγόριθμο του Dijkstra.

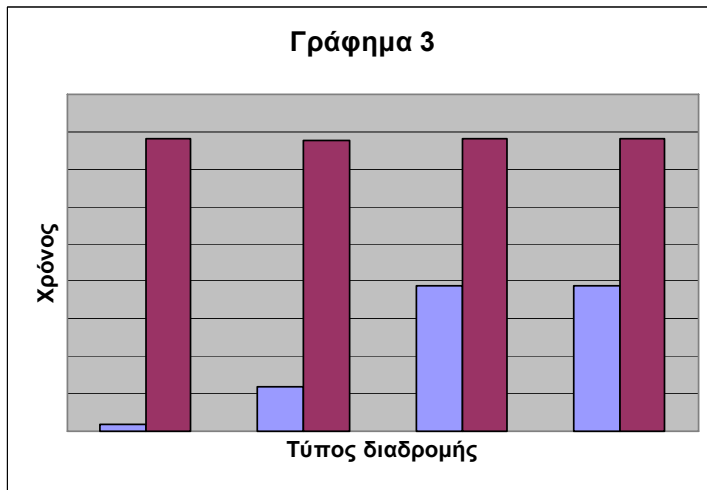
Παρατηρούμε πως ενώ στα δύο πρώτα παραδείγματα υπάρχει όφελος από τη χρήση της μεθόδου, στο τελευταίο λόγω της χαμηλής περιεκτικότητας θα πρέπει να προτιμήσουμε τον αλγόριθμο του Dijkstra. Το Γράφημα 1 παρουσιάζει συγκριτικά της διαφορά των δύο μεθόδων για όλων των τύπων τις διαδρομές που αιτήθηκαν στα παραδείγματα. Θα δούμε τη σύγκριση των ταχυτήτων υπολογισμού, διαχωρίζοντας τις αιτούμενες διαδρομες ανάλογα με τη θέση του αρχικού και τελικού σημείου της διαδρομής. Οι κατηγορίες είναι:

- Αρχικός και τελικός κόμβος βρίσκονται στο ίδιο δέντρο
- Αρχικός και τελικός κόμβος βρίσκονται σε διαφορετικά δέντρα
- Ο ένας από τους δύο κόμβους βρίσκεται σε δέντρο ενώ ο άλλος όχι
- Κανένας από τους δύο δεν βρίσκεται σε δέντρο

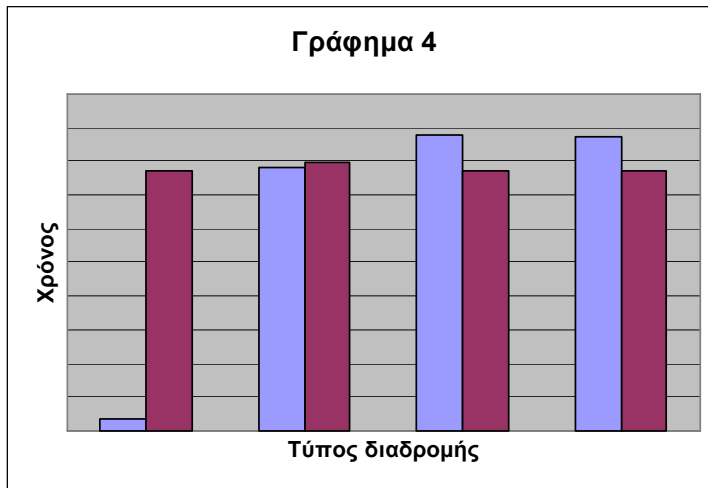
Τα επόμενα γραφήματα παρουσιάζουν την σύγκριση.



Γράφημα 2 Σύγκριση μεθόδων εύρεσης συντομότερης διαδρομής με χρήση απλοποιημένου γράφου και του αλγορίθμου του Dijkstra στο πρώτο παράδειγμα.



Γράφημα 3 Σύγκριση μεθόδων εύρεσης συντομότερης διαδρομής με χρήση απλοποιημένου γράφου και του αλγορίθμου του Dijkstra στο δεύτερο παράδειγμα.



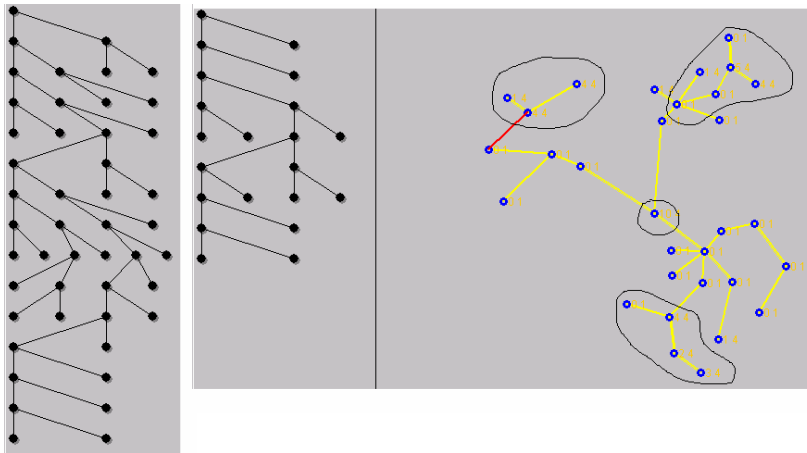
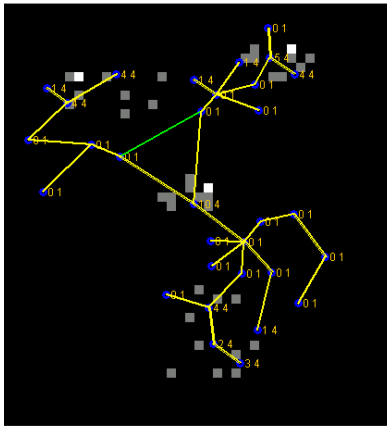
Γράφημα 4 Σύγκριση μεθόδων εύρεσης συντομότερης διαδρομής με χρήση απλοποιημένου γράφου και του αλγορίθμου του Dijkstra στο τρίτο παράδειγμα.

Παρατηρούμε πως στην πρώτη περίπτωση όπου και οι δύο κόμβοι βρίσκονται στο ίδιο δέντρο η εύρεση γίνεται σε μικρό χρονικό διάστημα λόγω της γραμμικής τάξης χρόνου μεθοδολογίας. Αμέσως πιο γρήγορη μέθοδος είναι αυτή στην οποία και οι δύο κόμβοι βρίσκονται σε κάποιο δέντρο, αφού ο αλγόριθμος του Dijkstra εφαρμόζεται στον αφηρημένο γράφο (αυτόν δηλαδή στον οποίο έχουν αντικατασταθεί οι δενδρικής μορφής υπογράφοι με κόμβους). Οι πιο αργές μέθοδοι είναι αυτές στις οποίες κάποιος κόμβος δεν βρίσκεται σε τέτοιο υπογράφο, αφού τότε χρειάζεται να χρησιμοποιήσουμε τον αλγόριθμο του Dijkstra στον πρώτου βαθμού απλοποίησης γράφο.

Εύρεση περιοχών υψηλής συχνότητας εμφάνισης περιστατικών

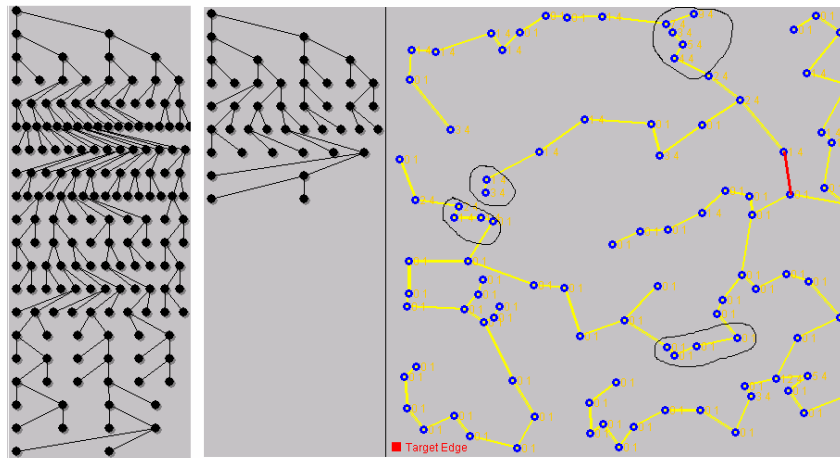
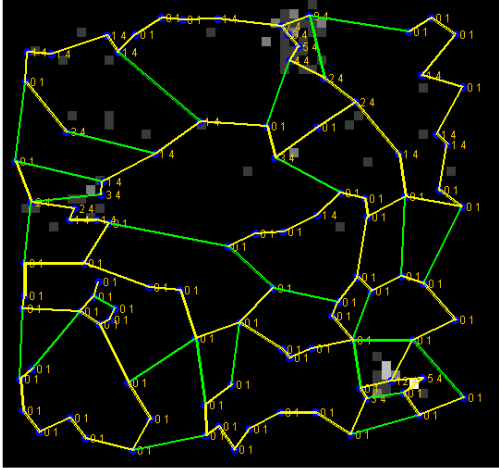
Θα παραθέσουμε δύο παραδείγματα τα οποία παρουσιάζουν την εφαρμογή του αλγορίθμου επαναληπτικής τμηματοποίησης ενός γράφου προκειμένου να υπολογίσουμε τις ζητούμενες περιοχές. Στα παραδείγματα που θα δούμε απεικονίζουμε τα γεγονότα με στον υπόβαθρο (background) της εικόνας του γράφου με τον εξής τρόπο: με φωτεινότερο χρώμα έχουν σχεδιαστεί οι ακμές οι οποίες ανήκουν στο ελάχιστο συνδετικό δέντρο του γράφου, ενώ οι γκριζες τετραγωνικές περιοχές οι οποίες έχουν χρωματισθεί στο υπόβαθρο, εκφράζουν το πλήθος των ατυχημάτων τα οποία συνέβησαν στην κάθε μια (η περιοχή με το φωτεινότερο χρώμα, είναι αυτές στις οποίες έχουν συμβεί τα περισσότερα περιστατικά, ενώ οι μαύρες αυτές στις οποίες δεν έχει συμβεί κανένα). Δίπλα σε κάθε κόμβο αναγράφονται ο αριθμός των περιστατικών ο οποίος τους έχει ανατεθεί, καθώς και το ονομαστικό τους βάρος. Στη συνέχεια θα παρουσιάσουμε το αρχικό δέντρο ταξινόμησης το οποίο υπολογίστηκε και τέλος το βελτιστοποιημένο μαζί με το γράφο στον οποίο έχουν σημειωθεί οι υπολογισθείσες περιοχές.

Παράδειγμα 1



Σχήμα A7 Στο πρώτο πλαίσιο παρατηρούμε τον αρχικό γράφο και την απεικόνιση των γεγονότων με γκρι χρώμα στο υπόβαθρο. Το αμέσως επόμενο πλαίσιο παρουσιάζει το δέντρο ταξινόμησης το οποίο παράγεται. Το τελευταίο πλαίσιο παρουσιάζει το βελτιστοποιημένο δέντρο ταξινόμησης καθώς και τις περιοχές οι οποίες τελικά επιλέχθηκαν.

Παράδειγμα 2



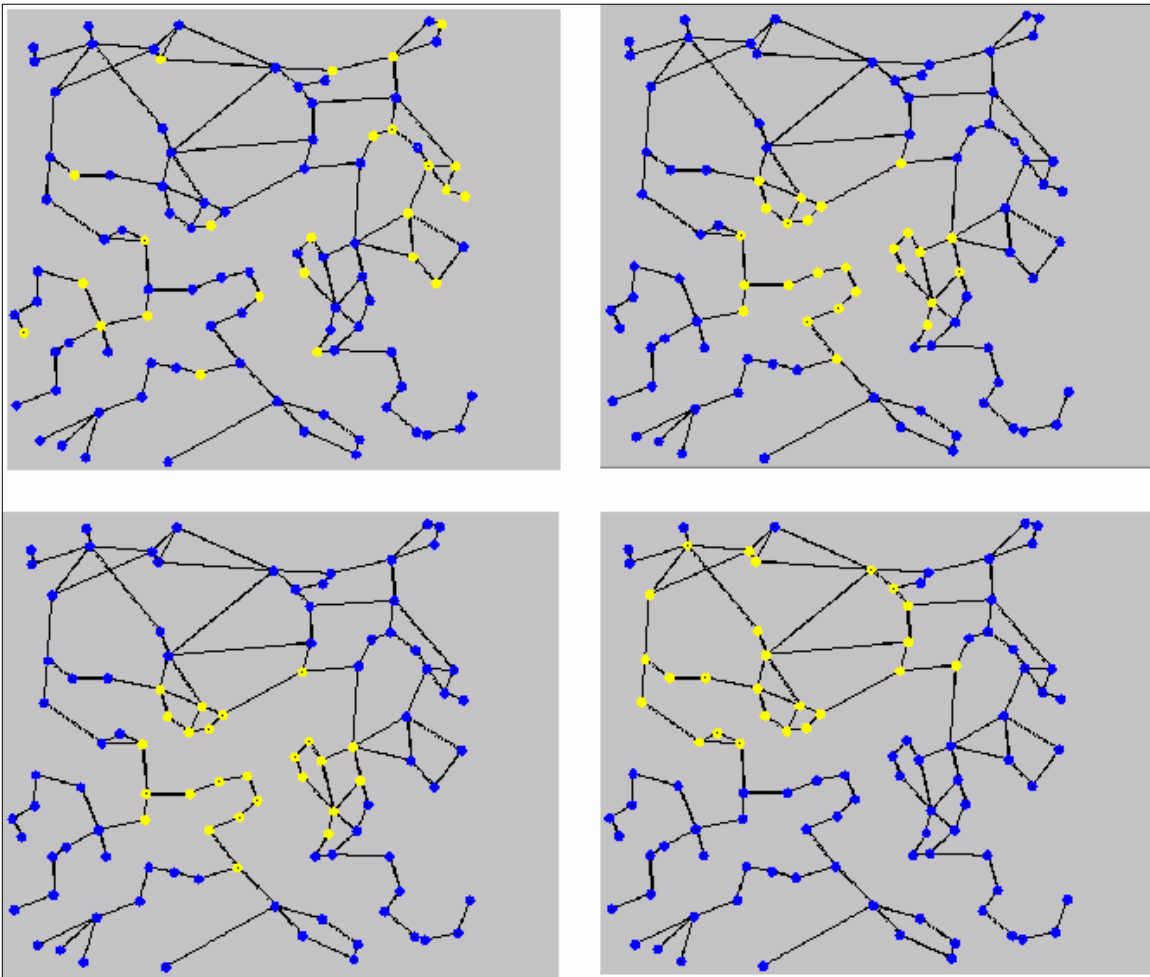
Σχήμα A8 Στο πρώτο πλαίσιο παρατηρούμε τον αρχικό γράφο και την απεικόνιση των γεγονότων με γκρι χρώμα στο υπόβαθρο. Το αμέσως επόμενο πλαίσιο παρουσιάζει το δέντρο ταξινόμησης το οποίο παράγεται. Το τελευταίο πλαίσιο παρουσιάζει το βελτιστοποιημένο δέντρο ταξινόμησης καθώς και τις περιοχές οι οποίες τελικά επιλέχθηκαν.

Υπολογισμός περιπολιών

Η παράγραφος αυτή παρουσιάζει τον υπολογισμό περιπολιών σε υπογράφους με τις τέσσερις μεθοδολογίες τις οποίες έχουμε αναφέρει στην αντίστοιχη ενότητα. Στο παράδειγμα (βλ σχήμα A9) που παρουσιάζουμε φαίνεται μια εικόνα του υπογράφου, και στη συνέχεια οι εξής περιπολίες (με ωρολογιακή φορά και ξεκινώντας από την πάνω αριστερά εικόνα):

- τυχαία
- κοντινότερα σημεία στο κέντρο μάζας
- σημεία τα οποία απέχουν (ευκλείδεια απόσταση) λιγότερο από τα υπόλοιπα
- σημεία τα οποία απέχουν (απόσταση με βάρη ακμών) λιγότερο από τα υπόλοιπα

Η κινητή μονάδα λαμβάνει την περιπολία αφού αυτή σχηματιστεί από τα σημεία με αλγόριθμο εύρεσης συντομότερης διαδρομής μεταξύ συνόλου σημείων (Travelling salesman) [48].



Σχήμα Α9 Υπολογισμός περιτολίας. Με φωτεινότερο χρώμα εμφανίζονται οι κόμβοι του γράφου οι οποίοι έχουν επιλεγεί για περιτολία

