ΜΟΝΤΕΛΟΠΟΙΗΣΗ ΝΟΗΤΙΚΩΝ ΔΙΑΔΙΚΑΣΙΩΝ ΚΑΙ ΑΝΘΡΩΠΙΝΩΝ ΛΑΘΩΝ ΓΙΑ ΤΟΝ ΣΧΕΔΙΑΣΜΟ ΚΑΙ ΑΣΦΑΛΗ ΔΙΑΧΕΙΡΙΣΗ ΡΟΩΝ ΕΡΓΑΣΙΑΣ

Γιαννόπουλος Νικόλαος

Πολυτεχνείο Κρήτης

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΑΡΑΓΩΓΗΣ & ΔΙΟΙΚΗΣΗΣ

Μία εργασία που παρουσιάστηκε στο Πολυτεχνείο Κρήτης για την εκπλήρωση των απαιτήσεων απόκτησης Μεταπτυχιακού Διπλώματος στο τμήμα Μηχανικών Παραγωγής και Διοίκησης

Χανιά, 2012

Αφιέρωση,

Στην οικογένειά μου

ΠΕΡΙΛΗΨΗ

Τα σύγχρονα συστήματα διαχείρισης ροών εργασίας ασχολούνται με την ανάπτυξη μεθόδων της Επιχειρησιακής Έρευνας για την κατανομή καθηκόντων, εργαλείων και πόρων στους εργαζόμενους ενός συστήματος παραγωγής. Παρόλο που έχει αναπτυχθεί πληθώρα μεθόδων και εφαρμογών λογισμικού με μεγάλο εύρος χρήσης, η μοντελοποίηση του ανθρωπίνου παράγονται είναι ανεπαρκής. Στις περισσότερες μεθόδους, η ανθρώπινη απόδοση μοντελοποιείται ως μία στοχαστική μεταβλητή χωρίς να παρέχονται επεξηγήσεις για το πώς οι νοητικές διαδικασίες (π.χ., μνήμη, εκτίμηση σπουδαιότητας, αντίληψη, λήψη αποφάσεων) μπορούν να επηρεάσουν τις τιμές της μεταβλητής που αναφέρεται ως «ανθρώπινος παράγοντας». Αυτό οδηγεί σε μια σειρά απλοποιημένων υποθέσεων για την ανθρώπινη συμπεριφορά και σε αδυναμία βελτίωσης της συνολικής απόδοσης του συστήματος ροών εργασίας.

Για παράδειγμα, οι σύγχρονες μέθοδοι μελέτης ροών εργασίας δεν μπορούν να υλοποιήσουν εργασίες που γίνονται από τον ίδιο εργαζόμενο αλλά μόνο από διαφορετικούς. Αυτό οφείλεται στην αδυναμία ανάπτυξης ενός μοντέλου «νοητικού φόρτου» που να εξετάζει τον φόρτο εργασίας σε συνάρτηση με τον αριθμό των εργασιών, το είδος των εργασιών και τις ικανότητες του εργαζομένου. Επίσης, δεν είναι δυνατόν να προβλεφθούν τα ανθρώπινα λάθη που προκύπτουν από την αύξηση του νοητικού φόρτου πέραν ενός ορίου. Η «Εργονομία» μπορεί να παίξει ένα σημαντικό ρόλο στον σχεδιασμό συστημάτων ροών εργασίας με την ανάπτυξη μοντέλων εκτίμησης της σπουδαιότητας των εργασιών και μοντέλων του νοητικού φόρτου εργασίας.

Οι σύγχρονες μέθοδοι μελέτης ροών εργασίας επίσης δεν είναι επαρκείς στην μοντελοποίηση εκτάκτων περιπτώσεων όπου η «ενδεικνυόμενη» σειρά εκτέλεσης εργασιών πρέπει να αλλάξει ανάλογα με τις περιστάσεις ή τα λάθη που έγιναν. Για παράδειγμα, μερικές εργασίες μπορεί να διακοπούν προσωρινά ή η σειρά προτεραιότητάς τους μπορεί να αλλάξει. Υπάρχει λοιπόν κίνδυνος ο

εργαζόμενος να «ξεχάσει» να εκτελέσει κάποιες εργασίες. Χωρίς ένα μοντέλο «ανθρώπινης μνήμης» δεν μπορούμε να εξετάσουμε εργασίες που μπορεί να παραληφθούν ή να εκτελεσθούν καθυστερημένα. Υπάρχει ανάγκη επιλογής μοντέλων μνήμης από την Εργονομία ώστε αυτά να ενσωματωθούν στο συνολικό σύστημα ροών εργασίας.

Ο στόχος της παρούσας μεταπτυχιακής διατριβής είναι να αναπτύξει μια σειρά από μοντέλα «νοητικών διαδικασιών» και «ανθρωπίνων λαθών» για την ενσωμάτωση του ανθρώπινου παράγοντα στον σχεδιασμό συστημάτων ροής εργασιών. Η υλοποίηση των μοντέλων αυτών θα γίνει με την χρήση τεχνικών που μπορούν να μοντελοποιήσουν τον «άνθρωπο» και την «μηχανή» με παραπλήσιους όρους. Προτείνεται λοιπόν η χρήση τεχνικών από το πεδίο «μηχανών πεπερασμένων θέσεων» (Finite State Machines) και πιο συγκεκριμένα θα εξετασθεί η δυνατότητα μοντελοποίησης με την χρήση «Έγχρωμων Δικτύων Πέτρι» (Coloured Petri Nets). Τα «Έγχρωμα Δίκτυα Πέτρι» επιτρέπουν τόσο την προσομοίωση των συστημάτων ροών εργασίας όσο και την ανάλυση των δομικών και δυναμικών ιδιοτήτων τους (Formal Analysis and Verification).

Η μεταπτυχιακή διατριβή περιλαμβάνει δύο στάδια:

- Το πρώτο στάδιο εξετάζει την χρήση των «Έγχρωμων Δικτύων Πέτρι» στην ενσωμάτωση ροών εργασίας και νοητικών διαδικασιών.
- Το δεύτερο στάδιο εφαρμόζει και προσομοιώνει την προτεινόμενη μεθοδολογία σε ένα πειραματικό βιομηχανικό καθήκον που θα περιλαμβάνει τον συντονισμό μίας ομάδας εργαζομένων για την περάτωση ενός συνόλου προκαθορισμένων εργασιών που ενέχουν τον κίνδυνο σοβαρών απωλειών.

Για την υλοποίηση θα χρησιμοποιηθεί το λογισμικό πακέτο CPN Tools και η γλώσσα προγραμματισμού Standard ML. Πέραν της προσομοίωσης, η χρήση των Έγχρωμων Δικτύων Πέτρι θα επιτρέψει επιπλέον και την ανάλυση των δομικών και δυναμικών ιδιοτήτων των συστημάτων ροών εργασίας.

ΕΥΧΑΡΙΣΤΙΕΣ

Θα ήθελα να εκφράσω τις ιδιαίτερες ευχαριστίες μου στον επιβλέποντα καθηγητή μου κ. Θωμά Κοντογιάννη για την επίβλεψη και καθοδήγησή του στην εκπόνηση αυτής της μεταπτυχιακής διατριβής.

Θα ήθελα επίσης να ευχαριστήσω τους καθηγητές κκ. Βασίλη Μουστάκη και Νικόλαο Ματσατσίνη για το χρόνο που αφιέρωσαν στην ανάγνωση του κειμένου και για τις εποικοδομητικές παρατηρήσεις τους.

Τέλος οφείλω να πω ένα μεγάλο ευχαριστώ στον αδελφικό μου φίλο Στάθη Παπανικολάου για την παρότρυνση και υποστήριξή του όλο αυτό το διάστημα.

Πίνακας περιεχομένων

ΠΕΡΙΛΗΨΗ 3
Κατάλογος Εικόνων 9
Κατάλογος Πινάκων 9
Κεφάλαιο 1: Εισαγωγή 11
Αναγκαιότητα 11
Στόχοι της Εργασίας 11
Δομή της Εργασίας 13
Κεφάλαιο 2: Σχετική Έρευνα 15
Εισαγωγή 15
Ένα κοινό πρότυπο λειτουργικών προβλημάτων 15
Αντιστοίχιση των εργασιακών απαιτήσεων με τις ανθρώπινες ικανότητες 16
Ταξινόμηση σφαλμάτων 17
Εργαλεία μοντελοποίησης εργασιών 17
Εργαλεία που συνδυάζουν μοντέλα εργασιών και γνωστικά μοντέλα 17
Εργαλεία με την χρήση δικτύων Πέτρι 18
Ανακεφαλαίωση 19
Κεφάλαιο 3: Τεχνολογική Βάση 20
Εισαγωγή 20
Δίκτυα Πέτρι 20
Έγχρωμα Δίκτυα Πέτρι 23
Standard ML 24
CPN Tools 25
Ανακεφαλαίωση 26
Κεφάλαιο 4: Αρχιτεκτονική 27
Εισαγωγή 27
Περιγραφή Αρχιτεκτονικής 27
Ανακεφαλαίωση 32
Κεφάλαιο 5: Μοντελοποίηση Νοητικών Διαδικασιών 34
Εισαγωγή 34
Διαχείριση της Προτεραιότητας των Εργασιών 35

Διαχείριση Μνήμης 38
Οργάνωση Εργασιών 39
Ανακεφαλαίωση 40
Κεφάλαιο 6: Η Μοντελοποίηση με τα Έγχρωμα Δίκτυα Πέτρι 42
Εισαγωγή 42
Βασικές Έννοιες 42
Το Θεμελιώδες Επίπεδο Του Δικτύου 46
Το Μοντέλο Εργασιών 47
Εκτελεστικός Μηχανισμός 57
Μοντέλο Νοητικών Διαδικασιών Εργαζομένων 63
Παρακολούθηση εργασιών στο Μοντέλο Νοητικών Διαδικασιών Εργαζομένων 71
Εκτέλεση εργασιών στο Μοντέλο Νοητικών Διαδικασιών Εργαζομένων 71
Οι Τύποι Δεδομένων που Χρησιμοποιούνται στο Δίκτυο 73
Οι Συναρτήσεις (Functions) που Χρησιμοποιούνται στο Δίκτυο 77
Ανακεφαλαίωση 81
Κεφάλαιο 7: Ανάλυση Ιδιοτήτων Δικτύων Πέτρι 83
Εισαγωγή 83
Ανάλυση Χώρου Καταστάσεων 83
Περιοριστικότητα (boundedness) Δικτύου 85
Home Σήμανση 85
Ιδιότητες Ζωτικότητας (liveness) 85
Αντιστρεψιμότητα (reversibility) 86
Άπειρες μεταβάσεις (fairness) 87
Ανακεφαλαίωση 87
Κεφάλαιο 8: Προσομοίωση 88
Εισαγωγή 88
Αρχική Κατάσταση Δικτύου 89
Σενάριο 1: Εκτέλεση Προσομοίωσης 94
Προσομοίωση 1 95
Προσομοίωση 2 96
Προσομοίωση 3 98
Σενάριο 2: Εκτέλεση Προσομοίωσης 100
Προσομοίωση 4 100

Προσομοίωση 5 102
Προσομοίωση 6 104
Ανακεφαλαίωση 107
Κεφάλαιο 9: Συμπεράσματα 113
ΠΑΡΑΡΤΗΜΑ 118
Οι αρχικές τιμές των δεδομένων του δικτύου 118
Συνάρτηση «AssignTasks» και οι βοηθητικές της συναρτήσεις 119
Συνάρτηση «PrioritizeTasks» και οι βοηθητικές της συναρτήσεις 120
Συναρτήσεις αναζήτησης 121
Συναρτήσεις ενημέρωσης και οι βοηθητικές τους συναρτήσεις 122
Διάφορες συναρτήσεις και οι βοηθητικές τους συναρτήσεις 125
ΒΙΒΛΙΟΓΡΑΦΙΑ 127

Κατάλογος Εικόνων

Εικόνα 1: Προτεινόμενη Αρχιτεκτονική 28
Εικόνα 2: Διάγραμμα ροής προτεινόμενης αρχιτεκτονικής 30
Εικόνα 3: Δείγμα Γράφου Δικτύου (Net Graph) 44
Εικόνα 4: Παράδειγμα Βασικών Εννοιών Δικτύων Πέτρι 45
Εικόνα 5: Το Θεμελιώδες Επίπεδο του Δικτύου 47
Εικόνα 6: Το Μοντέλο Εργασιών (CPN Tools) 48
Εικόνα 7: Το Μοντέλο Εργασιών (αρχιτεκτονική) 49
Εικόνα 8: Το Πρότυπο Εργασιών (CPN Tools) 53
Εικόνα 9: Το Πρότυπο Εργασιών (αρχιτεκτονική) 54
Εικόνα 10: Ο Εκτελεστικός Μηχανισμός (CPN Tools) 58
Εικόνα 11: Ο Εκτελεστικός Μηχανισμός (αρχιτεκτονική) 58
Εικόνα 12: Το Μοντέλο των Νοητικών Διαδικασιών των Εργαζομένων (CPN Tools) 63
Εικόνα 13: Το Μοντέλο των Νοητικών Διαδικασιών των Εργαζομένων (αρχιτεκτονική) 63
Εικόνα 14: Update Global Communication Places (CPN Tools) 68
Εικόνα 15: Update Global Communication Places (αρχιτεκτονική) 69

Κατάλογος Πινάκων

Πίνακας 1: Πίνακας Θέσεων Μοντέλου Εργασιών 51
Πίνακας 2: Πίνακας Μεταβάσεων Μοντέλου Εργασιών 51
Πίνακας 3: Πίνακας Θέσεων Προτύπου Εργασιών 55
Πίνακας 4: Πίνακας Μεταβάσεων Προτύπου Εργασιών 56
Πίνακας 5: Πίνακας Θέσεων Εκτελεστικού Μηχανισμού 59
Πίνακας 6: Πίνακας Μεταβάσεων Εκτελεστικού Μηχανισμού 61
Πίνακας 7: Πίνακας Θέσεων Μοντέλου Νοητικών Διαδικασιών Εργαζομένων 66
Πίνακας 8: Πίνακας Μεταβάσεων Μοντέλου Νοητικών Διαδικασιών Εργαζομένων 67
Πίνακας 9: Πίνακας Θέσεων Υποσυστήματος "Updated Global Communication Places" 70
Πίνακας 10: Πίνακας Μεταβάσεων Υποσυστήματος "Updated Global Communication Places" 70
Πίνακας 11: Οι Τύποι Δεδομένων που Χρησιμοποιούνται στο Δίκτυο 76
Πίνακας 12: Οι Συναρτήσεις που Χρησιμοποιούνται στο Δίκτυο 81
Πίνακας 13: Αρχική Κατάσταση Εργαζομένων 90
Πίνακας 14: Χαρακτηριστικά Εργαζομένων Α’ 90
Πίνακας 15: Χαρακτηριστικά Εργαζομένων Β’ 91
Πίνακας 16: Χαρακτηριστικά Γνωρίσματα της Κατάστασης κάθε Εργασίας 91
Πίνακας 17: Σενάριο 1: Φόρτος κάθε εργαζομένου από τις εργασίες 92
Πίνακας 18: Σενάριο 2: Φόρτος κάθε εργαζομένου από τις εργασίες 92

Πίνακας 19: Σενάριο 1: Επιπλέον Χαρακτηριστικά Εργασιών 93
Πίνακας 20: Οι συνέπειες διακοπής των εργασιών και οι πιθανότητες εμφάνισής τους..... 93
Πίνακας 21: Προσομοίωση 1 (χρόνοι 0-72) 95
Πίνακας 22: Προσομοίωση 1 (χρόνοι 72-146) 95
Πίνακας 23: Προσομοίωση 1 (χρόνοι 146-231) 96
Πίνακας 24: Προσομοίωση 2 (χρόνοι 0-102) 96
Πίνακας 25: Προσομοίωση 2 (χρόνοι 102-171) 97
Πίνακας 26: Προσομοίωση 2 (χρόνοι 171-255) 97
Πίνακας 27: Προσομοίωση 3 (χρόνοι 0-105) 98
Πίνακας 28: Προσομοίωση 3 (χρόνοι 105-182) 98
Πίνακας 29: Προσομοίωση 3 (χρόνοι 182-342) 99
Πίνακας 30: Προσομοίωση 4 (χρόνοι 0-91) 100
Πίνακας 31: Προσομοίωση 4 (χρόνοι 91-155) 100
Πίνακας 32: Προσομοίωση 4 (χρόνοι 155-237) 101
Πίνακας 33: Προσομοίωση 4 (χρόνοι 237-286) 102
Πίνακας 34: Προσομοίωση 5 (χρόνοι 0-67) 102
Πίνακας 35: Προσομοίωση 5 (χρόνοι 67-129) 103
Πίνακας 36: Προσομοίωση 5 (χρόνοι 129-219) 103
Πίνακας 37: Προσομοίωση 5 (χρόνοι 219-261) 104
Πίνακας 38: Προσομοίωση 6 (χρόνοι 0-61) 104
Πίνακας 39: Προσομοίωση 6 (χρόνοι 61-132) 105
Πίνακας 40: Προσομοίωση 6 (χρόνοι 132-221) 106
Πίνακας 41: Προσομοίωση 6 (χρόνοι 221-271) 106
Πίνακας 42: Συγκεντρωτικά αποτελέσματα 1^{ου} σεναρίου Α' 107
Πίνακας 43: Συγκεντρωτικά αποτελέσματα 1^{ου} σεναρίου Β' 108
Πίνακας 44: Συγκεντρωτικά αποτελέσματα 2^{ου} σεναρίου Α' 109
Πίνακας 45: Συγκεντρωτικά αποτελέσματα 2^{ου} σεναρίου Β'..... 111
Πίνακας 46: Συγκεντρωτικά Εργαζομένων..... 111

Κεφάλαιο 1: Εισαγωγή

Αναγκαιότητα

Οι αλλαγές στις απαιτήσεις των εργασιών λόγω απρόβλεπτων γεγονότων και τεχνολογικών αλλαγών μπορούν να προκαλέσουν μεταβολές στον σχεδιασμό των έργων όπως τροποποιήσεις στις διαδικασίες που ακολουθούνται για την περάτωση των έργων και στην κατανομή των επιμέρους εργασιών. Η αποτυχία της προσαρμογής σε αυτές τις μεταβολές στον σχεδιασμό των έργων μπορεί να οδηγήσει σε ανθρώπινα λάθη τα οποία μπορούν να προκαλέσουν σοβαρές συνέπειες στην ασφαλή λειτουργία των συστημάτων. Οι υπάρχουσες τεχνικές για την μοντελοποίηση των εργασιών αδυνατούν να μοντελοποιήσουν επαρκώς τον τρόπο με τον οποίο τα δίκτυα των εργασιών προσαρμόζονται στις μεταβαλλόμενες εργασιακές συνθήκες και απαιτήσεις. Για τον λόγο αυτό, υπάρχει η ανάγκη ενσωμάτωσης των δικτύων εργασιών με μοντέλα που να περιγράφουν τις νοητικές διαδικασίες των εργαζομένων, τα οποία να μπορούν να εκφράζουν τον τρόπο με τον οποίο οι εργαζόμενοι επεξεργάζονται τις πληροφορίες, λαμβάνουν αποφάσεις, διαχειρίζονται εργασίες που έχουν ανασταλεί, ή αντιμετωπίζουν τα λάθη που μπορούν να προκύψουν κατά την εκτέλεση των εργασιών.

Στόχοι της Εργασίας

Σκοπός της παρούσας διατριβής είναι η ανάπτυξη ενός εργαλείου, χρησιμοποιώντας τα «Έγχρωμα Δίκτυα Πέτρι», για την δημιουργία δικτύων εργασιών και την μοντελοποίηση συμπεριφορών βασισμένων σε

συγκεκριμένους κανόνες στα πλαίσια εργασιακών περιβαλλόντων τα οποία χαρακτηρίζονται από συχνές μεταβολές στις εργασιακές τους διαδικασίες καθώς και από μεταβολές στην κατανομή των εργασιών αυτών. Στα πλαίσια αυτά, προσδιορίζεται ένα δίκτυο από εργασίες το οποίο μοντελοποιεί την ροή των πληροφοριών, την χρήση των πόρων και τον έλεγχο των εργασιών. Αυτό το μοντέλο εργασιών ενσωματώνεται με ένα μοντέλο των νοητικών διαδικασιών των εργαζομένων το οποίο εξετάζει στρατηγικές με σκοπό την τροποποίηση των προτεραιοτήτων των στόχων, την αντιμετώπιση εργασιών που έχουν αναβληθεί, την διαχείριση του φόρτου εργασίας και την κατανομή των διαθέσιμων πόρων στο ανθρώπινο δυναμικό. Με τον τρόπο αυτό, το μοντέλο των νοητικών διαδικασιών των εργαζομένων μπορεί να χρησιμοποιηθεί στην προσαρμογή ενός τυπικού δικτύου εργασιών στις μεταβαλλόμενες συνθήκες ενός εργασιακού περιβάλλοντος. Το μοντέλο των νοητικών διαδικασιών των εργαζομένων αποτελείται από ένα υποσυστήματα επιλογής εργασιών που χρειάζονται προσοχή, ένα υποσύστημα διαχείρισης της μνήμης που περιέχει τις υπό αναστολή εργασίες και ένα υποσύστημα οργάνωσης των εργασιών. Στο εργαλείο αυτό, υπολογίζονται και γίνεται χρήση διαφόρων παραγόντων διαμόρφωσης της απόδοσης (Performance Shaping Factors), όπως ο φόρτος εργασίας, η κούραση, και η πνευματική κόπωση, ώστε να λαμβάνεται υπόψη το ευρύτερο εργασιακό πλαίσιο (π.χ. ανταγωνιστικές δραστηριότητες, λάθη, εργασίες υπό αναστολή). Οι διαφορετικοί πιθανοί τύποι ανθρωπίνων λαθών μπορούν να μοντελοποιηθούν για τον προσδιορισμό συμπεριφορών που υπακούουν σε συγκεκριμένους κανόνες, το οποίο είναι απαραίτητο σε κατανεμημένα περιβάλλοντα εργασίας.

Επιπλέον έχει αναπτυχθεί ένα εργαλείο προσομοίωσης με σκοπό την διαχείριση του μεγάλου όγκου δεδομένων που απαιτούνται για την πραγματοποίηση μίας εκτενούς ανάλυσης των εργασιών σε κατανεμημένα εργασιακά περιβάλλοντα και για την εκτέλεση αρκετών συνδυασμών εργασιακών συνθηκών οι οποίες μπορούν να οδηγήσουν σε ανθρώπινα λάθη

λόγω ανεπαρκών στρατηγικών διαχείρισης του εργασιακού φόρτου και της κατανομής των εργασιών. Με τον τρόπο αυτό, ο σχεδιασμός υφιστάμενων και νέων εργασιών μπορεί να επαληθευτεί με όρους στρατηγικών που ορίζονται στο μοντέλο των νοητικών διαδικασιών των εργαζομένων και σε τυπικές λειτουργικές διαδικασίες. Η αναλύσεις που παρέχονται μέσω αυτών των προσομοιώσεων καθώς και οι επίσημες τεχνικές ανάλυσης μπορούν να εφαρμοστούν για την επεξεργασία εργασιών ελέγχου για την επαλήθευση διαδικασιών εργασιών, στρατηγικών διαχείρισης φόρτου εργασίας και σχεδίων κατανομής εργασιών ανταποκρινόμενες σε τεχνολογικές αλλαγές και άγνωστα γεγονότα.

Δομή της Εργασίας

Το κεφάλαιο 2 περιλαμβάνει την σχετική με την παρούσα διατριβή έρευνα παρουσιάζοντας ένα κοινό πρότυπο λειτουργικών προβλημάτων, τις προσπάθειες για αντιστοίχιση των εργασιακών απαιτήσεων με τις ανθρώπινες ικανότητες καθώς και της ταξινόμησης σφαλμάτων και επιπλέον εργαλείων για την μοντελοποίηση εργασιών, εργαλείων που συνδυάζουν μοντέλα εργασιών και γνωστικά μοντέλα και εργαλείων με την χρήση δικτύων Πέτρι. Το κεφάλαιο 3 παρουσιάζει την τεχνολογική βάση της παρούσας διατριβής και πιο συγκεκριμένα τα δίκτυα Πέτρι, τα έγχρωμα δίκτυα Πέτρι, την γλώσσα προγραμματισμού Standard ML και το εργαλείο μοντελοποίησης CPN Tools. Το κεφάλαιο 4 περιλαμβάνει την περιγραφή της αρχιτεκτονικής του υλοποιημένου προτεινόμενου δικτύου. Το κεφάλαιο 5 περιλαμβάνει μία εκτενής αναφορά στην μοντελοποίηση γνωστικών διεργασιών και πιο συγκεκριμένα περιγράφει την διαδικασία εκτίμησης της προτεραιότητας εργασιών, της διαχείρισης μνήμης και της οργάνωσης εργασιών. Το κεφάλαιο 6 περιλαμβάνει την λεπτομερή παρουσίαση του υλοποιημένου δικτύου και των επιμέρους τμημάτων του και πιο συγκεκριμένα την παρουσίαση βασικών εννοιών, του Μοντέλου Εργασιών, του Εκτελεστικού Μηχανισμού, του

Γνωστικού Μοντέλου Χρηστών και των τύπων δεδομένων και των συναρτήσεων που χρησιμοποιούνται στο δίκτυο. Στο κεφάλαιο 7 περιλαμβάνεται η ανάλυση των ιδιοτήτων των δικτύων Πέτρι όπως αυτές εφαρμόζονται στην υλοποίηση της διατριβής. Οι υπό εξέταση ιδιότητες είναι η περιοριστικότητα (boundedness), η ύπαρξη home σήμανσης, οι ιδιότητες ζωτικότητας (liveness), η αντιστρεψιμότητα (reversibility) και η ύπαρξη απείρων μεταβάσεων (fairness). Το κεφάλαιο 8 περιλαμβάνει την παρουσίαση πέντε επιλεγμένων δοκιμαστικών προσομοιώσεων καθώς και την περιγραφή της αρχικής κατάστασης του δικτύου. Τέλος, στο κεφάλαιο 9 γίνεται η ανακεφαλαίωση και η ανάπτυξη των συμπερασμάτων της διατριβής. Στο Παράρτημα υπάρχει ο κώδικας των υλοποιημένων συναρτήσεων που χρησιμοποιούνται στο δίκτυο.

Κεφάλαιο 2: Σχετική Έρευνα

Εισαγωγή

Το κεφάλαιο περιλαμβάνει σύντομες περιγραφές των σχετικών με την παρούσα διατριβή ερευνών στην βιβλιογραφία [1].

Ένα κοινό πρότυπο λειτουργικών προβλημάτων

Υπάρχουν αρκετές αξιολογήσεις στην βιβλιογραφία σχετικές με παρ' ολίγον αποτυχίες εργασιών και αναλύσεις περιστατικών κατά την διάρκεια ελέγχου διαδικασιών σύμφωνα με τις οποίες έχει αναγνωριστεί ένα κοινό πρότυπο λειτουργικών προβλημάτων τα οποία συσχετίζονται με τον τρόπο με τον οποίο οι άνθρωποι προσαρμόζονται στις μεταβολές στον σχεδιασμό εργασιών που προκαλούνται από απρόβλεπτα γεγονότα και τεχνολογικές αλλαγές [2][3][4][5]. Τα λειτουργικά προβλήματα είναι πιθανό να περιλαμβάνουν μηχανικές αστοχίες, μη διαθεσιμότητα των εργαλείων, χρήση νέου ή άγνωστου εξοπλισμού, αντιμετώπιση υψηλού επιπέδου εργασιακού φόρτου, πραγματοποίηση των ίδιων εργασιών με μικρότερο αριθμό εργαζομένων κτλ. Σε αυτές τις περιπτώσεις, οι εργαζόμενοι θα πρέπει να τροποποιήσουν τις προτεραιότητες των στόχων τους, την εκτέλεση των εργασιών τους, την κατανομή των εργασιών και την χρήση των εργαλείων ή των διαθέσιμων πόρων. Αυτές οι τροποποιήσεις μπορεί αρχικά να οδηγήσουν σε ανεπιτυχείς ενέργειες καθώς πολλές φορές εκτελούνται κάτω από στενά χρονικά πλαίσια και συχνές διακοπές. Η ικανότητα του σχεδιασμού των εργασιών να προσαρμόζεται σε μεταβολές των συνηθισμένων πρακτικών και να επανέρχεται

από προηγούμενες αποτυχημένες ενέργειες δεν επηρεάζει μόνο την παραγωγικότητα αλλά και την ασφάλεια του συστήματος.

Αντιστοίχιση των εργασιακών απαιτήσεων με τις ανθρώπινες ικανότητες

Η αντιστοίχιση των εργασιακών απαιτήσεων με τις ανθρώπινες ικανότητες έχει υπάρξει το αντικείμενο της ανάλυσης εργασιών για πολλές δεκαετίες και μία πληθώρα από τεχνικές έχει αναπτυχθεί στα πλαίσια της βιομηχανικής εργονομίας [6][7]. Οι τεχνικές ανάλυσης εργασιών έχουν χρησιμοποιηθεί με επιτυχία για την περιγραφή των στόχων των εργαζομένων, την κατανομή και τον προγραμματισμό των εργασιών με σκοπό την βελτιστοποίηση της αλληλεπίδρασης των ανθρώπων με τις μηχανές. Ωστόσο, η ανάλυση των εργασιών περιγράφει τον τρόπο με τον οποίο η εκάστοτε εργασία πρέπει να πραγματοποιηθεί κάτω από ένα πλαίσιο από σαφείς και προκαθορισμένες συνθήκες του εργασιακού περιβάλλοντος. Κάτω από αυτές τις συνθήκες, προκύπτουν δυσκολίες για περιπτώσεις όπου τεχνολογικές αλλαγές και απρόσμενα γεγονότα απαιτούν τροποποιήσεις της συνηθισμένης εργασιακής ρουτίνας. Οι συνέπειες των τροποποιήσεων αυτών, όπως είναι οι μεταβολές στις εργασιακές ρουτίνες και οι αλλαγές στην κατανομή των εργασιών, συνήθως εξετάζονται σε κάποιο επόμενο στάδιο με την χρήση συμπληρωματικών αναλύσεων των ανθρωπίνων λαθών. Ένας ακόμα παράγοντας που πρέπει να εξετασθεί και αφορά την παραδοσιακή ανάλυση των εργασιών είναι ο σχεδιασμός νέων εργασιών για τις οποίες δεν υπάρχουν επαρκείς πληροφορίες σχετικά με το επίπεδο στελέχωσης και τον απαιτούμενο προγραμματισμό εργασιών και πόρων. Φαίνεται πως χρειάζονται νέες τεχνικές οι οποίες θα εξετάζουν με τρόπο συστηματικό τις μεταβολές των υπαρχουσών εργασιών καθώς και να μοντελοποιούν τις απαιτήσεις για τον σχεδιασμό νέων εργασιών.

Ταξινόμηση σφαλμάτων

Μία ταξινόμηση σφαλμάτων [3] έχει χρησιμοποιηθεί για την μοντελοποίηση των συνεπειών των ανθρωπίνων σφαλμάτων και της προοπτικής ανάκαμψης από τα σφάλματα αυτά.

Εργαλεία μοντελοποίησης εργασιών

Οι έως σήμερα προσεγγίσεις για την μοντελοποίηση των εργασιών είχαν την τάση να επικεντρώνονται κυρίως σε αναπαραστάσεις σε Η/Υ των εργασιών και των πλάνων αυτών, χωρίς να επεξεργάζονται τις νοητικές στρατηγικές που χρησιμοποιούνταν από τους εργαζομένους. Εργαλεία μοντελοποίησης εργασιών όπως το MicroSaint [8] και το GOMS [9], παρέχουν μία χρήσιμη βάση για την εξέταση των συνεπειών αρκετών πολιτικών της κατανομής των εργασιών και του προγραμματισμού του σχεδιασμού των διαδικασιών, ωστόσο δεν ενσωματώνουν μοντέλα χρηστών που να υποδεικνύουν τον τρόπο με τον οποίο οι εργαζόμενοι επεξεργάζονται τις πληροφορίες, λαμβάνουν αποφάσεις, ή αντιμετωπίζουν τις υπό αναστολή εργασίες και τα σφάλματα. Στα πλαίσια της αλληλεπίδρασης ανθρώπου-μηχανής, εργαλεία μοντελοποίησης εργασιών όπως το ConcurTaskTrees [10], το TOBOLA [11], το TOOD [12] και το DIANE+ [13] έχουν αναπτυχθεί για τον σχεδιασμό των διεπαφών των χρηστών ωστόσο παρέχουν περιορισμένες διευκολύνσεις για την μοντελοποίηση των νοητικών στρατηγικών των χρηστών.

Εργαλεία που συνδυάζουν μοντέλα εργασιών και γνωστικά μοντέλα

Στα πλαίσια της αλληλεπίδρασης μεταξύ ανθρώπου-μηχανής, όπως περιγράφεται στο [14], ενσωματώθηκε ένα εργαλείο μοντελοποίησης εργασιών (IMPRINT) με ένα ευρέως διαδεδομένο νοητικό μοντέλο της ανθρώπινης μνήμης και μάθησης (ACT-R). Στον τομέα της συντήρησης αεροσκαφών, όπως περιγράφεται στο [15], αναπτύχθηκε ένα εργαλείο προσομοίωσης το οποίο ενσωμάτωνε την μοντελοποίηση εργασιών με ένα μοντέλο εργαζομένων

το οποίο υπολόγιζε τους παράγοντες διαμόρφωσης της απόδοσης δυναμικά. Υπήρξαν επιπλέον προσπάθειες στον τομέα της ανθρώπινης αξιοπιστίας οι οποίες επικεντρωνόντουσαν σε επίσημα μοντέλα ανθρώπινης συμπεριφοράς τα οποία προέβλεπαν τα ανθρώπινα λάθη στο πλαίσιο του ελέγχου της εναέριας κυκλοφορίας [16]. Τέλος, ένα εργαλείο γενικής χρήσης αναπτύχθηκε από τους [17], το οποίο ενσωμάτωνε την θεωρεία ελέγχου της αντίληψης (Perceptual Control Theory) με δίκτυα εργασιών για ένα ευρύ φάσμα εφαρμογών.

Εργαλεία με την χρήση δικτύων Πέτρι

Τα δίκτυα Πέτρι έχουν χρησιμοποιηθεί σε αρκετές εφαρμογές ως μία κοινή τεχνική για την μοντελοποίηση εργασιών, εργαζομένων και τεχνικών συστημάτων. Χαρακτηριστικό παράδειγμα είναι οι προτάσεις [18] και [19] στις οποίες ενσωματώνεται μία επίσημη αναπαράσταση ενός συστήματος με ένα μοντέλο ανθρώπινου επεξεργαστή (Human Processor Model) [20]. Σε άλλες προτάσεις έχει χρησιμοποιηθεί η μοντελοποίηση με τα δίκτυα Πέτρι για την μελέτη των νοητικών μοντέλων των χρηστών και των προσδοκιών τους κατά την διάρκεια μίας αλληλεπίδρασης [21][22]. Στα πιο πρόσφατα χρόνια, τα δίκτυα Πέτρι προσέφεραν την βάση για την ανάλυση των ανθρώπινων αλληλεπιδράσεων με τα πολυσύνθετα βιομηχανικά συστήματα. Στον τομέα της πυρηνικής ενέργειας, για παράδειγμα, τα δίκτυα Πέτρι χρησιμοποιήθηκαν για την ανάπτυξη ενός μοντέλου εξομοιωτή των νοητικών διαδικασιών του τρόπου με τον οποίο οι εργαζόμενοι μπορούν να διαγνώσουν ακολουθίες από ατυχήματα [23]. Στην έρευνα που περιέχεται στο [24], παρουσιάζονται χρήσιμα συμπεράσματα της προσαρμογής μοντέλων πλοήγησης αεροσκαφών βασισμένα στα δίκτυα Πέτρι με την χρήση συλλογιστικής η οποία βασιζόταν στην εκάστοτε περίπτωση. Στην μελέτη που περιέχεται στο [25], περιγράφονται οι διάφορες πτυχές αξιολόγησης των καταστάσεων όσον αφορά τα συστήματα ελέγχου εναέριας κυκλοφορίας χρησιμοποιώντας δυναμικά στοχαστικά δίκτυα Πέτρι. Τέλος, στον τομέα άμυνας [26] χρησιμοποιήθηκαν

έγχρωμα δίκτυα Πέτρι για την ενσωμάτωση της διαδικασίας λήψης αποφάσεων και του επιχειρησιακού σχεδιασμού σε διοικητικές και ελεγκτικές διαδικασίες.

Ανακεφαλαίωση

Τα οφέλη της ενσωμάτωσης των μοντέλων εργασιών με τα μοντέλα των νοητικών διαδικασιών των χρηστών είναι προφανή. Καταρχάς, ο τρόπος με τον οποίο οι εργαζόμενοι προσαρμόζονται στις μεταβολές των εργασιακών διαδικασιών, οι οποίες επιβάλλονται από τεχνολογικές αλλαγές, μπορεί να εξεταστεί καλύτερα με τα μοντέλα των νοητικών διαδικασιών των χρηστών τα οποία απευθύνονται στην διαχείριση εργασιών που χρειάζονται προσοχή και στην λήψη αποφάσεων με βάση τις μεταβαλλόμενες εργασιακές συνθήκες, τις προτεραιότητες αναφορικά με τους στόχους, και τον προγραμματισμό των απαιτήσεων. Επιπλέον, η αντιμετώπιση νέων γεγονότων, διακοπών εργασιών και αποτυχημένων προσπαθειών απαιτεί μοντέλα της ανθρώπινης μνήμης του τρόπου με τον οποίο οι άνθρωποι διατηρούν στην μνήμη τους τις δραστηριότητές τους και την πρόοδό τους. Ακόμα, τα μοντέλα των νοητικών διαδικασιών των χρηστών παρέχουν μία καλή βάση για τον υπολογισμό των δυναμικών επιπτώσεων των παραγόντων διαμόρφωσης της απόδοσης (Performance Shaping Factors) της ανθρώπινης αξιοπιστίας. Τέλος, οι επιπτώσεις των ανθρωπίνων σφαλμάτων μπορούν να κατανοηθούν καλύτερα όταν τα μοντέλα των νοητικών διαδικασιών των χρηστών καθορίζουν τον τρόπο με τον οποίο αυτά τα σφάλματα προέκυψαν εξ αρχής.

Αναγνωρίζοντας τα παραπάνω οφέλη, έχει παρατηρηθεί τα τελευταία χρόνια ένα διαρκώς αυξανόμενο ενδιαφέρον για εργαλεία τα οποία ενσωματώνουν μοντέλα εργασιών με μοντέλα των νοητικών διαδικασιών των εργαζομένων.

Κεφάλαιο 3: Τεχνολογική Βάση

Εισαγωγή

Το κεφάλαιο περιλαμβάνει την περιγραφή των βασικών τεχνολογιών που χρησιμοποιήθηκαν για την υλοποίηση της παρούσας διατριβής. Περιλαμβάνονται οι παρουσιάσεις των δικτύων Πέτρι και των έγχρωμων δικτύων Πέτρι [42], της γλώσσας προγραμματισμού Standard ML [44], και του γραφικού εργαλείου CPN Tools [47].

Δίκτυα Πέτρι

Ένα δίκτυο Πέτρι (επίσης γνωστό ως ένα δίκτυο θέσεων/μεταβάσεων) είναι μία από τις πολλές γλώσσες μαθηματικής μοντελοποίησης για την περιγραφή κατανεμημένων συστημάτων. Ένα δίκτυο Πέτρι είναι ένας κατευθυνόμενος γράφος (διπλής κατεύθυνσης), στον οποίο οι κόμβοι αντιπροσωπεύουν τις μεταβάσεις (δηλαδή γεγονότα που μπορεί να συμβούν, που αναπαριστώνται από ορθογώνια σχήματα) και τις θέσεις (δηλαδή καταστάσεις, που αναπαριστώνται από κύκλους). Οι κατευθυνόμενες ακμές περιγράφουν ποιες θέσεις είναι προϋποθέσεις ή/και συνέπειες για ποιες μεταβάσεις (το οποίο καθορίζεται από βέλη). Ορισμένες πηγές [43] αναφέρουν ότι τα δίκτυα Πέτρι εφευρέθηκαν τον Αύγουστο του 1939 από τον Carl Adam Petri στην ηλικία των 13 ετών με σκοπό την περιγραφή των χημικών διεργασιών.

Όπως και τα πρότυπα της βιομηχανίας, όπως είναι τα UML διαγράμματα δραστηριοτήτων (activity diagrams), η BPMN (Business Process Model and Notation) και τα EPCs (Event-driven Process Chains), τα δίκτυα Πέτρι

προσφέρουν μια γραφική αναπαράσταση για διαδικασίες αποτελούμενες από στάδια που περιλαμβάνουν την επιλογή, την επανάληψη, και την ταυτόχρονη εκτέλεση. Σε αντίθεση με αυτά τα πρότυπα, τα δίκτυα Πέτρι έχουν ένα ακριβές μαθηματικό προσδιορισμό της σημασιολογίας της εκτέλεσής τους, με μία άρτια ανεπτυγμένη μαθηματική θεωρία για την ανάλυση των διαδικασιών.

Ένα δίκτυο Πέτρι αποτελείται από θέσεις (places), μεταβάσεις (transitions), και ακμές (arcs). Οι ακμές ξεκινούν από μια θέση και καταλήγουν σε μία μετάβαση ή το αντίστροφο, ωστόσο δεν μπορούν να υπάρξουν ποτέ μεταξύ δύο θέσεων ή μεταξύ δύο μεταβάσεων. Οι θέσεις από τις οποίες μία ακμή ξεκινάει και καταλήγει σε μία μετάβαση ονομάζονται θέσεις εισόδου της μετάβασης. Οι θέσεις στις οποίες οι ακμές καταλήγουν από μία μετάβαση ονομάζονται θέσεις εξόδου της μετάβασης.

Γραφικά, οι θέσεις σε ένα δίκτυο Πέτρι μπορεί να περιέχουν ένα διακριτό αριθμό από μάρκες (tokens). Κάθε διανομή των μαρκών στις θέσεις του δικτύου αναπαριστά μία διαμόρφωση (configuration) του δικτύου που ονομάζεται σήμανση (marking). Κατά μία αφηρημένη έννοια, σχετικά με ένα διάγραμμα δικτύου Πέτρι, μία μετάβαση ενός δικτύου Πέτρι μπορεί να εκτελεστεί, όταν υπάρχουν επαρκείς μάρκες στην αρχή όλων των ακμών εισόδου. Όταν η μετάβαση εκτελεστεί, καταναλώνει αυτές τις μάρκες, και τοποθετεί μάρκες στο τέλος όλων των ακμών εξόδου. Κάθε εκτέλεση είναι ατομική, δηλαδή ένα μοναδικό μη διακοπτόμενο βήμα.

Η εκτέλεση των δικτύων Πέτρι είναι μη ντετερμινιστική, δηλαδή, όταν πολλαπλές μεταβάσεις είναι ενεργοποιημένες την ίδια στιγμή, οποιαδήποτε από αυτές μπορεί να εκτελεστεί. Εάν μία μετάβαση είναι ενεργοποιημένη, μπορεί να εκτελεστεί, αλλά δεν είναι υποχρεωτικό να συμβεί αυτό.

Δεδομένου ότι η εκτέλεση των μεταβάσεων είναι μη ντετερμινιστική διαδικασία, και πολλαπλές μάρκες μπορούν να υπάρχουν οπουδήποτε στο δίκτυο (ακόμα και στην ίδια θέση), τα δίκτυα Πέτρι είναι κατάλληλα για τη

μοντελοποίηση της ταυτόχρονης συμπεριφοράς των κατανεμημένων συστημάτων.

Στην παρούσα διατριβή η προτεινόμενη προσέγγιση μοντελοποίησης είναι βασισμένη σε μία αναπαράσταση, με την χρήση των δικτύων Πέτρι, των ανθρώπινων δραστηριοτήτων, εργαλείων, και οργανωτικών ρόλων. Τα δίκτυα Πέτρι μπορούν να προσδιοριστούν είτε σε γραφική μορφή είτε με την μορφή μαθηματικών εξισώσεων. Η γραφική μορφή των δικτύων Πέτρι βοηθάει στην απεικόνιση μεταβάσεων καταστάσεων στις νοητικές διεργασίες στις οποίες μπορούν να συνδυαστούν τόσο η σειριακή όσο και η παράλληλη επεξεργασία. Παρόλο που το μεγαλύτερο κομμάτι της μοντελοποίησης μπορεί να γίνει με την χρήση γράφων δικτύων Πέτρι, ο μαθηματικός προσδιορισμός παρέχει τα θεμέλια για την χρησιμοποίηση μίας ποικιλίας από επίσημες τεχνικές ανάλυσης. Οι τεχνικές αυτές μπορούν να υλοποιηθούν σε πακέτα λογισμικού ώστε να επιτρέπουν την διερεύνηση αρκετών δομικών και δυναμικών ιδιοτήτων των δικτύων Πέτρι. Οι επίσημες τεχνικές ανάλυσης χρησιμοποιούν γράφους προσβασιμότητας (reachability graphs) και γραμμική άλγεβρα ώστε να είναι σε θέση να εξετάζουν ένα δίκτυο εργασιών περιέχει αδιέξοδα (deadlocks), ατέρμονες βρόγχους (livelocks), εργασίες οι οποίες δεν συνεισφέρουν στον στόχο, «νεκρές» εργασίες οι οποίες δεν εκτελούνται ποτέ, και εργασίες οι οποίες έχουν ως αποτέλεσμα την έλλειψη συγχρονισμού. Ο γράφος προσβασιμότητας μπορεί να χρησιμοποιηθεί για την διερεύνηση ορισμένων συγκεκριμένων ιδιοτήτων συμπεριφοράς (π.χ. ζωτικότητα (liveness)) και την αναγνώριση διαδρομών εργασιών οι οποίες μπορούν να οδηγήσουν σε επικίνδυνες καταστάσεις [27]. Η γραμμική άλγεβρα δίνει την δυνατότητα στους αναλυτές να αποδεικνύουν ορισμένες δομικές ιδιότητες χρησιμοποιώντας τις δομές των θέσεων (places) και των μεταβάσεων (transitions). Για παράδειγμα, οι δομές των θέσεων μπορούν να αποδείξουν ότι μία εργασία μπορεί να βρεθεί σε ένα συγκεκριμένο πλήθος καταστάσεων ή πως ορισμένα συγκεκριμένα εργαλεία χρησιμοποιούνται καθ' όλη την διάρκεια της εργασίας. Από την άλλη

πλευρά, οι δομές των μεταβάσεων μπορούν να χρησιμοποιηθούν για την διερεύνηση του τρόπου με τον οποίο ένα σύστημα μπορεί να αποσυντεθεί σε υποσυστήματα τα οποία πραγματοποιούν συγκεκριμένες λειτουργίες, όπως, για παράδειγμα, μία αυτόνομη κυκλική επεξεργασία δεδομένων και προϊόντων. Τέλος, τα δίκτυα Πέτρι παρέχουν μία κοινή γλώσσα για την μοντελοποίηση τόσο των τεχνικών συστημάτων όσο και του ανθρώπινου στοιχείου. Με τον τρόπο αυτό, η έρευνα που πραγματοποιείται στα πλαίσια της επίσημης ανάλυσης της κατασκευής συστημάτων μπορεί να διερευνηθεί μέσω της μοντελοποίησης των συστημάτων ανθρώπου-μηχανής.

Έγχρωμα Δίκτυα Πέτρι

Το προτεινόμενο πλαίσιο μοντελοποίησης βασίζεται στα έγχρωμα δίκτυα Πέτρι [28][29][46]. Το μεγάλο βήμα στα δίκτυα Πέτρι, αυτό που πραγματικά τα μετατρέπει από ένα θεωρητικό πλαίσιο σε ένα σοβαρό εργαλείο που χρησιμοποιείται, μεταξύ άλλων, από προγραμματιστές πρωτοκόλλων, είναι η επέκτασή τους σε έγχρωμα δίκτυα Πέτρι. Η χρήση του όρου "έγχρωμα" αποτελεί ακυριολεξία. Η αρχική ιδέα ήταν η ανάθεση χρωμάτων σε μάρκες, και η χρήση κατάλληλων εκφράσεων με βάση τα χρώματα αυτά στα στοιχεία του δικτύου. Ωστόσο η μελέτη των μοντέλων ανάλυσης γρήγορα έδειξε ότι οι ερευνητές μπορούσαν να προχωρήσουν πολύ περισσότερο από αυτό χωρίς να χάνουν τις βασικές ιδιότητες ανάλυσης που είναι τόσο πολύτιμες. Στην πλήρη ανάπτυξη των έγχρωμων δικτύων Πέτρι, μπορεί κάποιος να συσχετίσει ουσιαστικά αυθαίρετες συλλογές δεδομένων με μάρκες δικτύων Πέτρι, εφόσον χρησιμοποιεί ένα ισχυρό σύστημα τύπων, και τηρώντας τους απαραίτητους περιορισμούς στις εκφράσεις που χρησιμοποιούνται στο δίκτυο. Κατά συνέπεια, τα χρώματα γίνονται τύποι δεδομένων, που περιγράφουν τα είδη των δεδομένων που μπορούν να μεταφέρονται από τις μάρκες και τα είδη των μαρκών που μπορούν να βρίσκονται σε μια θέση στο δίκτυο. Αυτή είναι η

θεμελιώδης ιδέα των έγχρωμων δικτύων Πέτρι: οι μάρκες έχουν τύπους, και κάθε τύπος μίας μάρκας έχει κάποια δεδομένα που σχετίζονται με αυτό.

Τα έγχρωμα δίκτυα Πέτρι προσθέτουν μια συλλογή από επεκτάσεις στα υπόλοιπα στοιχεία του δικτύου ώστε να επωφεληθούν από τις μάρκες που διακρίνονται από κάποιον τύπο και έχουν κάποια συγκεκριμένη τιμή. Σε κάθε θέση στο δίκτυο έχει ανατεθεί επίσης ένας τύπος δεδομένων, και μπορεί μόνο να κρατήσει μάρκες του ιδίου τύπου. Οι εισερχόμενες ακμές μίας μετάβασης μπορούν να έχουν προϋποθέσεις: η μετάβαση ενεργοποιείται μόνο όταν κάποιο σύνολο από μάρκες από τις θέσεις εισόδου της ικανοποιήσει το σύνολο των προϋποθέσεων για τις εισερχόμενες ακμές του. Οι προϋποθέσεις για τις εισερχόμενες ακμές μιας μετάβασης μπορούν να αναφέρονται στις τιμές άλλων εισερχόμενων ακμών της ίδιας μετάβασης. Για παράδειγμα, οι προϋποθέσεις μπορεί να απαιτήσουν ότι οι τιμές δύο μαρκών που προέρχονται από διαφορετικές εισερχόμενες ακμές να ταιριάζουν. Οι ακμές που εξέρχονται από μία μετάβαση μπορεί να έχουν εκφράσεις που καθορίζουν τον τρόπο με τον οποίο υπολογίζονται οι τιμές των μαρκών που παράγονται από τη μετάβαση. Όταν εκτελείται μία μετάβαση με επιτυχία, οι εκφράσεις στις εξερχόμενες ακμές της αξιολογούνται για την παραγωγή νέων μαρκών ώστε να τροφοδοτηθεί η θέση στο τέλος της ακμής. Τα στοιχεία αυτά αποτελούν μία μόνο μικρή εισαγωγή των εννοιών που χρησιμοποιούνται από τα έγχρωμα δίκτυα Πέτρι και η περαιτέρω εμβάθυνση στις έννοιες αυτές είναι εκτός του πλαισίου της παρούσας διατριβής και κατά συνέπεια δεν θα γίνει περαιτέρω ανάλυση των ιδιοτήτων αυτών.

Standard ML

Η Standard ML (SML) είναι μια γενικής χρήσης, ορθά δομημένη, λειτουργική γλώσσα προγραμματισμού η οποία παρέχει έλεγχο των τύπων δεδομένων κατά την διάρκεια της μεταγλώττισης. Είναι δημοφιλής μεταξύ των

προγραμματιστών μεταφραστών (compilers) και των ερευνητών των γλωσσών προγραμματισμού, καθώς και στην ανάπτυξη των αποδείξεων θεωρημάτων.

Η γλώσσα είναι μια σύγχρονη απόγονος της γλώσσας προγραμματισμού ML που χρησιμοποιούταν στο έργο LCF (Logic for Computable Functions) για την απόδειξη θεωρημάτων. Η Standard ML ξεχωρίζει μεταξύ των ευρέως διαδεδομένων γλωσσών καθώς έχει αυστηρές προδιαγραφές, με την μορφή κανόνων γραψίματος και λειτουργικών σημασιολογικών δομών όπως ορίζονται στο «The Definition of Standard ML» (1990) και αναθεωρήθηκε και απλοποιήθηκε ως «The Definition of Standard ML (Revised)» το 1997 [45]. Στα πλαίσια της παρούσας διατριβής έγινε ευρεία χρήση της γλώσσας προγραμματισμού Standard ML.

CPN Tools

Το εργαλείο CPN Tools είναι ένα εργαλείο για την επεξεργασία, την προσομοίωση, και την ανάλυση των έγχρωμων δικτύων Πέτρι. Το εργαλείο διαθέτει λειτουργίες όπως έλεγχο σύνταξης και παραγωγή κώδικα, οι οποίες λαμβάνουν χώρα ενώ το δίκτυο βρίσκεται υπό κατασκευή. Ένας γρήγορος προσομοιωτής χειρίζεται αποτελεσματικά χρονικά και μη δίκτυα. Επιπλέον μπορούν να παραχθούν πλήρεις ή/και ημιτελείς καταστάσεις χώρου (state spaces) και να αναλυθούν, καθώς και να παραχθεί μία πρότυπη αναφορά των καταστάσεων χώρου η οποία περιέχει πληροφορίες για το δίκτυο, όπως είναι οι ιδιότητες της περιοριστικότητας (boundedness) και της ζωτικότητας (liveness) οι οποίες θα παρουσιαστούν στο κεφάλαιο 7 το οποίο περιλαμβάνει την ανάλυση των ιδιοτήτων του υλοποιημένου δικτύου Πέτρι. Η έκδοση του εργαλείου που χρησιμοποιήθηκε στην παρούσα διατριβή είναι η 3.0.3 (Φεβρουάριος 2011).

Ανακεφαλαίωση

Στο κεφάλαιο έγινε μία επιγραμματική περιγραφή των τεχνολογικών πλαισίων, των γλωσσών προγραμματισμού και των εργαλείων που χρησιμοποιήθηκαν για την υλοποίηση της παρούσας διατριβής.

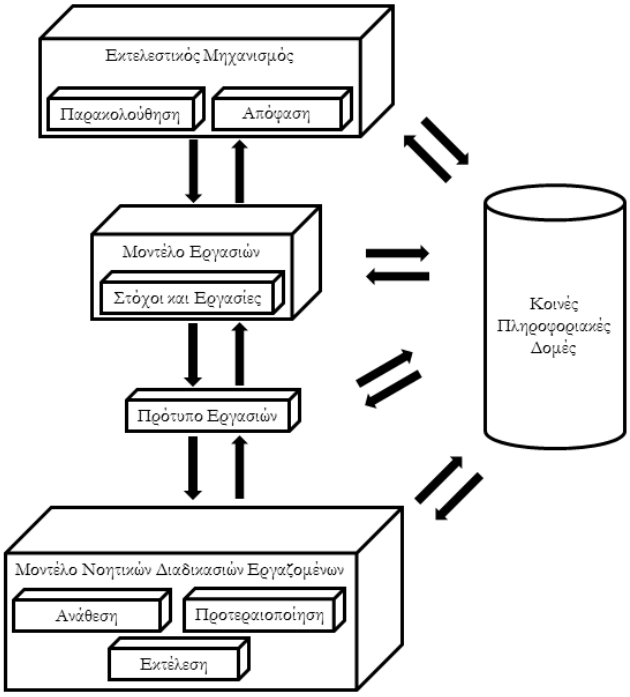
Κεφάλαιο 4: Αρχιτεκτονική

Εισαγωγή

Το κεφάλαιο περιλαμβάνει την περιγραφή του προτεινόμενου πλαισίου για την ενσωμάτωση δικτύων εργασιών με μοντέλα των νοητικών διαδικασιών των εργαζομένων το οποίο παρουσιάζει την ροή των αποφάσεων και των δεδομένων καθώς και τις αλληλεπιδράσεις μεταξύ των συστατικών του μερών. Για την υλοποίηση του προτεινόμενου πλαισίου θα χρησιμοποιηθούν τα έγχρωμα δίκτυα Πέτρι.

Περιγραφή Αρχιτεκτονικής

Το προτεινόμενο πλαίσιο αποτελείται από ένα μοντέλο εργασιών το οποίο προσδιορίζει την οργάνωση των εργασιών που απαιτούνται για την επίτευξη ορισμένων συγκεκριμένων στόχων και από ένα μοντέλο των νοητικών διαδικασιών των εργαζομένων το οποίο προσδιορίζει τον τρόπο με τον οποίο οι εργαζόμενοι επιλέγουν και δίνουν προτεραιότητα σε εργασίες, τον τρόπο με τον οποίο κατανέμουν τις εργασίες σύμφωνα με τις προτιμήσεις τους και τις απαιτήσεις της δουλειάς τους, και τον τρόπο με τον οποίο διατηρούν στην μνήμη τους τις υπό αναστολή ή αποτυχημένες εργασίες. Επιπλέον, χρησιμοποιείται ένας εκτελεστικός μηχανισμός ο οποίος παρακολουθεί τα εξωτερικά γεγονότα και λαμβάνει αποφάσεις σχετικά με τις αλλαγές στην επεξεργασία των εργασιών. Το δίκτυο εργασιών, το μοντέλο των νοητικών διαδικασιών των εργαζομένων και ο εκτελεστικός μηχανισμός εκτελούνται παράλληλα (Εικόνα 1).



Εικόνα 1: Προτεινόμενη Αρχιτεκτονική

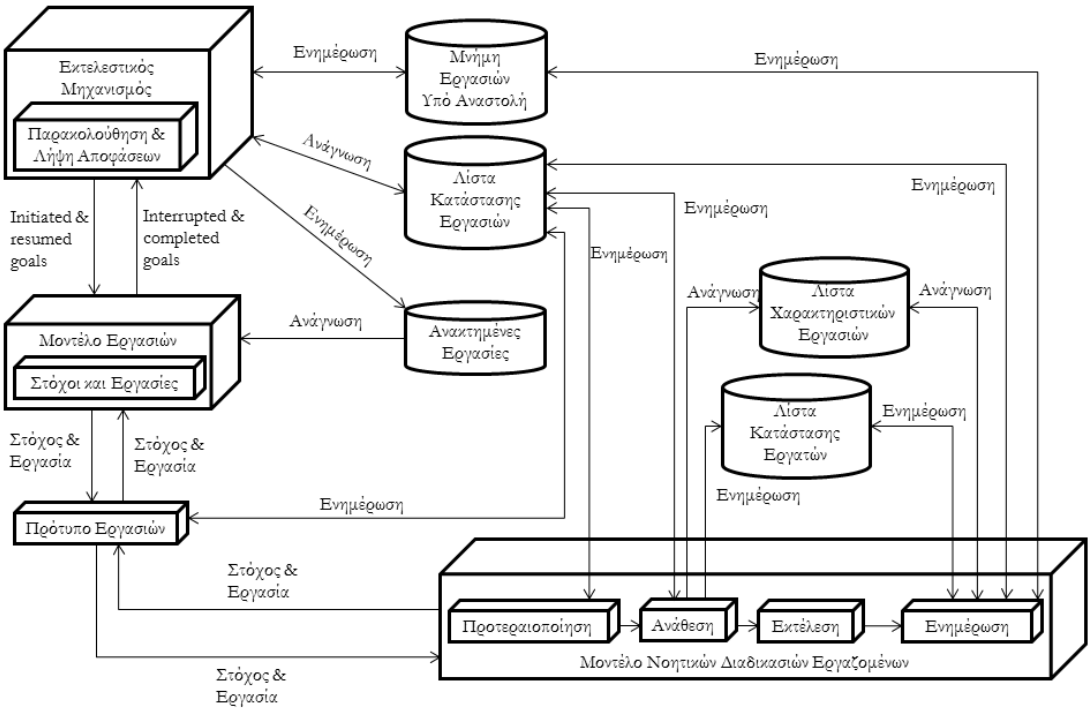
Ένα από τα θεμελιώδη δομικά συστατικά του προτεινόμενου πλαισίου είναι η εργασίες (tasks), δηλαδή τα δομικά συστατικά των λειτουργιών που απαιτούνται για την επίτευξη κάποιων συγκεκριμένων στόχων. Οι εργασίες μπορούν να προγραμματιστούν και να οργανωθούν με διαφορετικούς τρόπους για να σχηματίσουν υψηλότερου επιπέδου λειτουργίες και επιπλέον μπορούν να εξυπηρετηθούν από διαφορετικούς εργαζόμενους και υλικούς πόρους. Οι εργασίες χαρακτηρίζονται από συγκεκριμένες ιδιότητες που συσχετίζονται με την χρονική τους σειρά, όπως προϋποθέσεις, διάρκεια των εργασιών, προτεραιότητες των εργασιών και εάν οι εργαζόμενοι είναι οι βασικοί ή βοηθητικοί εργαζόμενοι. Υπάρχουν επιπλέον επιπρόσθετες εργασιακές ιδιότητες οι οποίες συσχετίζονται με συγκεκριμένες πτυχές του μοντέλου των νοητικών διαδικασιών των εργαζομένων. Για παράδειγμα, οι εργασίες αξιολογούνται με βάση τον βαθμό στον οποίο βασίζονται σε νοητικά κανάλια, μηχανικά κανάλια, ή κανάλια αντίληψης με σκοπό τον υπολογισμό του φόρτου εργασίας των εργαζομένων στις περιπτώσεις που απαιτείται ο διαχωρισμός της προσοχής των εργαζομένων μεταξύ πολλαπλών εργασιών. Άλλες ιδιότητες των

εργασιών συσχετίζονται με εξωτερικές ουρές εργασιών οι οποίες υπενθυμίζουν στους εργαζόμενους να ξαναρχίζουν τις υπό αναστολή εργασίες οι οποίες περιέχονται στο υποσύστημα της μνήμης του εκτελεστικού μηχανισμού. Κατά συνέπεια, ένας μεγάλος όγκος δεδομένων απαιτείται καθώς διαφορετικές ιδιότητες των εργασιών συσχετίζονται με διαφορετικές πτυχές του δικτύου εργασιών και του μοντέλου των νοητικών διαδικασιών των χρηστών.

Το μοντέλο εργασιών είναι ένα δίκτυο ή οργανισμός εργασιών, ο οποίος καθορίζει την ακολουθία των εργασιών όπως αυτή περιγράφεται στις λειτουργικές διαδικασίες. Εναλλακτικές ακολουθίες μπορούν να δοκιμαστούν στο μοντέλο εργασιών ώστε να βρεθούν οι πιο αποδοτικές ακολουθίες, ή να εξεταστούν οι συνέπειες των ανθρώπινων λαθών κατά τον σχεδιασμό των λειτουργιών. Οι πληροφορίες από το μοντέλο εργασιών μεταφέρονται στο μοντέλο των νοητικών διαδικασιών των εργαζομένων μέσω ενός προτύπου εργασιών το οποίο μετατρέπει την γραφική μορφή μίας εργασίας σε μία μάρκα (token) η οποία εισέρχεται στο μοντέλο των νοητικών διαδικασιών των εργαζομένων για περεταίρω επεξεργασία (Εικόνα 1). Τα δίκτυα Πέτρι επιβάλλουν ένα πλαίσιο τοπικής επικοινωνίας από σημείο σε σημείο, το οποίο σε μερικές περιπτώσεις περιορίζει την διαχείριση των δυναμικών αλλαγών των δεδομένων και των εργασιών. Για τον λόγο αυτό, έχει οριστεί ένα μικρός αριθμός από κοινές πληροφοριακές δομές ώστε να επιτυγχάνεται μίας μορφής καθολικής επικοινωνίας μεταξύ των υποσυστημάτων που έχουν οριστεί μέσα στην υλοποίηση του δικτύου.

Το μοντέλο των νοητικών διαδικασιών των εργαζομένων αποτελείται από τρία υποσυστήματα για την αντιμετώπιση των ζητημάτων της επιλογής εργασιών, της ανάθεσης εργαζομένων σε εργασίες και της εκτέλεσης των εργασιών. Το υποσύστημα της παρακολούθησης και ανάκλησης των υπό αναστολή εργασιών βρίσκεται στον εκτελεστικό μηχανισμό του δικτύου. Τα ανθρώπινα λάθη μοντελοποιούνται είτε στο στάδιο του σχεδιασμού των ακολουθιών στο μοντέλο εργασιών, είτε στο στάδιο της εκτέλεσης των εργασιών στο μοντέλο

των νοητικών διαδικασιών των εργαζομένων. Οι συνέπειες των λαθών αυτών προσδιορίζονται στο υποσύστημα το οποίο είναι υπεύθυνο για την εκτέλεση των εργασιών και διαδίδονται (έχουν ισχύ) σε όλο το δίκτυο εργασιών. Οι αλληλεπιδράσεις μεταξύ του δικτύου εργασιών, του μοντέλου των νοητικών διαδικασιών των εργαζομένων και του εκτελεστικού μηχανισμού παρουσιάζονται καλύτερα στο διάγραμμα ροής στην Εικόνα 2. Τα μονοπάτια που φαίνονται στο διάγραμμα ροής περιγράφουν τα δεδομένα και τις αποφάσεις που λαμβάνονται στα πλαίσια της προτεινόμενης αρχιτεκτονικής με πιο κατανοητό τρόπο. Κάθε υποσύστημα (κύβος) που περιέχεται στο διάγραμμα αντιστοιχεί σε ένα μικρότερο μοντέλο έγχρωμου δικτύου Πέτρι το οποίο μπορεί να συλλέγει δεδομένα, να λαμβάνει αποφάσεις και να παράγει εξόδους. Οι κύλινδροι αναπαριστούν «βάσεις δεδομένων» οι οποίες περιέχουν δυναμικές πληροφορίες αναφορικά με την κατάσταση των εργασιών και την λειτουργία του συστήματος ευρύτερα οι οποίες είναι προσβάσιμες καθολικά σε όλο το δίκτυο.



Εικόνα 2: Διάγραμμα ροής προτεινόμενης αρχιτεκτονικής

Ο εκτελεστικός μηχανισμός εκκινεί το δίκτυο εργασιών το οποίο στέλνει νέες εργασίες προς το μοντέλο των νοητικών διαδικασιών των εργαζομένων σύμφωνα με ένα προσχεδιασμένο και συγκεκριμένο πλάνο ή ακολουθία εργασιών. Η προκαθορισμένη ακολουθία από εργασίες η οποία πρέπει να πραγματοποιηθεί για την επίτευξη των στόχων περιέχεται μέσα στο μοντέλο εργασιών. Κάθε εργασία λογίζεται ως μία ξεχωριστή μονάδα η οποία μπορεί να παραληφθεί ή να προωθηθεί για επεξεργασία προς το μοντέλο των νοητικών διαδικασιών των εργαζομένων. Όλες οι εργασίες έχουν συγκεκριμένη δομή η οποία περιλαμβάνεται στο πρότυπο εργασιών. Μέσα στο πρότυπο εργασιών περιέχεται το μοντέλο των νοητικών διαδικασιών των εργαζομένων το οποίο συνεχίζει την επεξεργασία και ολοκληρώνει την εκτέλεση των εργασιών. Στα πλαίσια του μοντέλου των νοητικών διαδικασιών των εργαζομένων, αρχικά, σχηματίζεται μία λίστα από υποψήφιες εργασίες η οποία μπορεί να περιλαμβάνει και εργασίες που ήταν υπό αναστολή και έχουν πάλι ενεργοποιηθεί. Οι υποψήφιες εργασίες προτεραιοποιούνται με βάση έναν συγκεκριμένο αλγόριθμο, ο οποίος υπολογίζει έναν δείκτη προτεραιότητας για κάθε υποψήφια εργασία όπως περιγράφεται στο κεφάλαιο 5, σχηματίζοντας μία λίστα από ταξινομημένες εργασίες και στην συνέχεια επιλέγεται η σημαντικότερη εργασία από την λίστα αυτή και προωθείται προς ανάθεση. Οι υποψήφιες εργασίες που δεν επιλέχτηκαν, παραμένουν σε αναμονή και σε επόμενο βήμα επανεξετάζονται. Η ανάθεση των εργασιών σε εργαζομένους πραγματοποιείται σύμφωνα με συγκεκριμένους κανόνες κατανομής και τις προτιμήσεις των εργαζομένων όπως αυτές προσδιορίζονται σε ένα κατάλληλο μοντέλο εργασιακής οργάνωσης. Η επιλογή του καταλληλότερου εργαζομένου για την ανάθεση και περάτωση μίας εργασίας πραγματοποιείται με βάση έναν αλγόριθμο ο οποίος υπολογίζει έναν δείκτη ανάθεσης εργασίας για κάθε υποψήφιο εργαζόμενο όπως περιγράφεται στο κεφάλαιο 5. Η παράλληλη επεξεργασία των εργασιών επηρεάζεται από τον φόρτο εργασίας των εργαζομένων όπως υπολογίζεται από έναν αλγόριθμο κατά την διάρκεια

ανάθεσης των εργασιών στους εργαζομένους. Η περιγραφή του αλγορίθμου περιλαμβάνεται στο κεφάλαιο 5. Έχουν οριστεί τέσσερις στρατηγικές διαχείρισης του φόρτου εργασίας για την αντιμετώπιση της κατανομής των εργασιών. Τέλος, οι επεξεργασμένες εργασίες εκτελούνται στο μοντέλο των νοητικών διαδικασιών των εργαζομένων από τους εργαζόμενους στους οποίους έχουν ανατεθεί.

Η έξοδος του μοντέλου των νοητικών διαδικασιών των εργαζομένων, η οποία περιλαμβάνει τόσο τις ολοκληρωμένες εργασίες όσο και αυτές που διακόπηκαν από ανθρώπινα λάθη ή εκείνες που ανεστάλησαν για διάφορους λόγους, στέλνεται σε κατάλληλες δομές δεδομένων οι οποίες λειτουργούν ως βάσεις δεδομένων και διατηρούν εκτός των άλλων πληροφορίες για την κατάσταση όλων των εργασιών. Πολλά προβλήματα σχετικά με την απόδοση μπορούν να μοντελοποιηθούν συμπεριλαμβανομένων των παραλείψεων, της έλλειψης συγχρονισμού, των διακοπών, και των καθυστερήσεων. Δίνεται η δυνατότητα της μοντελοποίησης λαθών ως λάθη σχεδιασμού στο μοντέλο εργασιών. Τα λάθη κατά την διάρκεια της λειτουργίας του δικτύου μπορούν να έχουν επιπτώσεις στην απόδοση των εργασιών και μπορούν να προκαλέσουν παρενέργειες και σε άλλες εργασίες οι οποίες βρίσκονται υπό εκτέλεση ή πρόκειται να εκτελεστούν και βρίσκονται στην ίδια ακολουθία με την προβληματική εργασία. Ο εκτελεστικός μηχανισμός ελέγχει το περιβάλλον εργασίας και επιτρέπει ή αποτρέπει την επανεκκίνηση εργασιών οι οποίες βρίσκονται υπό αναστολή ή έχουν διακοπεί.

Τέλος, όλα τα επιμέρους υποσυστήματα της αρχιτεκτονικής επικοινωνούν όταν χρειάζεται με τις δομές δεδομένων που είναι διαθέσιμες καθολικά σε όλο το δίκτυο και εάν κρίνεται απαραίτητο τις ενημερώνουν κατάλληλα.

Ανακεφαλαίωση

Στο κεφάλαιο περιγράφηκε το προτεινόμενο αρχιτεκτονικό πλαίσιο το οποίο στοχεύει στην ενσωμάτωση των δικτύων εργασιών με μοντέλα των νοητικών

διαδικασιών των εργαζομένων. Το πλαίσιο αυτό περιλαμβάνει την ροή των αποφάσεων και των δεδομένων καθώς και τις αλληλεπιδράσεις μεταξύ των συστατικών του μερών.

Κεφάλαιο 5: Μοντελοποίηση Νοητικών Διαδικασιών

Εισαγωγή

Τα τελευταία χρόνια, έχει υπάρξει μία διαρκώς αυξανόμενη προσπάθεια για την ανάπτυξη υπολογιστικών μοντέλων της ανθρώπινης απόδοσης όπως χαρακτηριστικά συνοψίζεται σε αρκετές αναφορές στην βιβλιογραφία [30][31][32]. Ορισμένες μελέτες επικεντρώνονται στις λεπτομερείς αναπαραστάσεις της ανθρώπινης νόησης, ενώ άλλες δίνουν έμφαση στην αλληλεπίδραση μεταξύ της νόησης και του εργασιακού περιβάλλοντος. Αυτή η χρονικά εξαρτημένη αλληλεπίδραση μεταξύ της νόησης και της εργασίας είναι πολύ σημαντική για την μελέτη της ανθρώπινης αξιοπιστίας σε σύνθετα περιβάλλοντα εργασίας. Στα πλαίσια της παρούσας διατριβής, η επιλογή των μοντέλων εκτίμησης της σπουδαιότητας των εργασιών και της μνήμης στο εργασιακό περιβάλλον βασίστηκε στο βαθμό στον οποίο τα υποψήφια μοντέλα έκαναν αναφορά στο πλαίσιο εργασίας (π.χ. στις τρέχουσες εργασίες, ανταγωνιζόμενες εργασίες, εργασίες υπό αναστολή, εργασίες που έχουν αποτύχει) και περιλάμβαναν χαρακτηριστικά των εργασιών τα οποία θα μπορούσαν να τροποποιηθούν ως τμήμα της διαδικασίας επανασχεδιασμού των διαδικασιών. Ορισμένα από τα επιλεγμένα μοντέλα εκτίμησης της σπουδαιότητας των εργασιών και της μνήμης απλοποιήθηκαν για να διευκολύνουν την ποσοτικοποίηση των αλληλεπιδράσεων μεταξύ ανθρώπου-μηχανής με τρόπο κατάλληλο για την γλώσσα μοντελοποίησης. Τα δίκτυα Πέτρι είναι ένα εργαλείο το οποίο έχει χρησιμοποιηθεί κυρίως για την μοντελοποίηση τεχνικών συστημάτων, ωστόσο, υπάρχει το ενδεχόμενο της

διερεύνησης της χρήσης τους για την μοντελοποίηση της ανθρώπινης απόδοσης [25][33][26].

Το μοντέλο των νοητικών διαδικασιών των εργαζομένων περιλαμβάνει ένα υποσύστημα οργάνωσης της εργασίας με σκοπό την ανάθεση εργασιών στους εργαζομένους σύμφωνα με τις δυνατότητές τους, τις προτιμήσεις τους και τους κανόνες κατανομής του συστήματος. Η παρακολούθηση των εργασιών και η λήψη αποφάσεων είναι δύο νοητικές διαδικασίες οι οποίες δεν έχουν επαρκώς αναπτυχθεί στην παρούσα διατριβή. Η αιτία είναι πως επειδή δεν βασίζονται σε επίσημα μοντέλα νόησης, δεν συμπεριλήφθησαν στο μοντέλο των νοητικών διαδικασιών των εργαζομένων. Ωστόσο, λήφθησαν υπόψη στον εκτελεστικό μηχανισμό ο οποίος λειτουργεί παράλληλα με τα υπόλοιπα υποσυστήματα.

Διαχείριση της Προτεραιότητας των Εργασιών

Ο σκοπός του υποσυστήματος διαχείρισης της προτεραιότητας των εργασιών είναι ο προγραμματισμός της εκτέλεσης των εργασιών λαμβάνοντας υπόψη συγκεκριμένες προτεραιότητες, περιορισμούς των εργασιών και παρεμβολές καθ' όλη την διάρκεια λειτουργίας του συστήματος. Το υποσύστημα αυτό καθορίζει εάν μία εργασία εκτελείται κατόπιν εντολής, έχει διακοπεί, έχει ξαναρχίσει, ή έχει αναβληθεί. Το δίκτυο εργασιών παράγει νέες εργασίες οι οποίες μπορεί να ανταγωνίζονται για προσοχή με άλλες εργασίες οι οποίες εκτελούνται ήδη ή με εργασίες οι οποίες προσωρινά βρίσκονται υπό αναστολή. Όταν μία νέα εργασία φτάσει στο μοντέλο των νοητικών διαδικασιών των εργαζομένων, δημιουργείται μία λίστα από υποψήφιες εργασίες οι οποίες περιέχουν τόσο τις νέες εργασίες όσο και τις εργασίες οι οποίες ήταν υπό αναστολή και ξανάρχισαν. Η εκτίμηση της προτεραιότητας των εργασιών είναι μία δυναμική διαδικασία η οποία λαμβάνει υπόψη αρκετές προτεραιότητες σχετικά με την επιλογή και βαθμονόμηση των εργασιών ανάλογα με το πόσο σημαντικές είναι. Παρόλο που οι εργασίες έχουν τις δικές τους προεπιλεγμένες προτεραιότητες, το σύστημα που είναι υπεύθυνο για την εκτίμηση της

προτεραιότητάς τους υπολογίζει ένα δείκτη προτεραιότητας (priority index) ο οποίος λαμβάνει υπόψη το πλαίσιο εργασίας σε οποιαδήποτε χρονική στιγμή (π.χ. άλλες ανταγωνιστικές εργασίες).

Ο παρακάτω αλγόριθμος έχει προσαρμοστεί από την δουλειά που περιγράφεται στο [34] και στο [35].

$$P = IC + \sum_{i=1}^{n}(p_i * c_i) \quad (1)$$

όπου,

- P είναι ο δείκτης προτεραιότητας (Priority Index) για κάποια νέα ή υπό αναστολή εργασία,
- IC είναι το κόστος διακοπής (Interruption Cost) το οποίο συσχετίζεται με την διακοπή ή αναβολή της εργασίας,
- i = 1…n, είναι οι πιθανές επιπτώσεις ως αποτέλεσμα της μετάθεσης της εργασίας,
- p_i είναι η πιθανότητα να συμβεί η i συνέπεια και,
- c_i είναι το κόστος της i συνέπειας.

Το κόστος διακοπής (όρος IC της εξίσωσης 1) είναι μία αριθμητική τιμή η οποία ενσωματώνει το κόστος της διακοπής ή αναβολής μίας εργασίας. Συσχετίζεται με την διατήρηση των εργαλείων και των πόρων σε κατάλληλες καταστάσεις κατά την διάρκεια του κενού χρονικού διαστήματος της διακοπής και την επανεγκατάσταση των προϋποθέσεων με σκοπό την επανεκτίμηση της εργασίας με κάποιο επόμενο στάδιο. Ο δεύτερος όρος της εξίσωσης (1) αναφέρεται στο ρίσκο του να συμβεί μία δυσμενής συνέπεια ως αποτέλεσμα της διακοπής ή αναβολής μίας εργασίας. Η ιδέα μίας προθεσμίας, στο τέλος της οποίας μπορεί να συμβούν ανεπιθύμητες επιπτώσεις είναι ένα κεντρικό χαρακτηριστικό σε όλες τις προσεγγίσεις προτεραιοποίησης [35]. Η αναβολή και η μετάθεση μίας εργασίας σε μετέπειτα χρονικό διάστημα μπορεί να συσχετίζεται με αρκετές προθεσμίες, ή επιπτώσεις, οι οποίες να έχουν διαφορετικές πιθανότητες να συμβούν. Η πιθανότητα αυτή αποτελεί συνάρτηση

αρκετών παραγόντων συμπεριλαμβανομένων, (1) του χρόνου που απομένει μέχρι να ακολουθήσει μία συνέπεια, (2) της διάρκειας της εργασίας, (3) του πλήθους των γεγονότων κατά τα οποία μία εργασία έχει διακοπεί ή ανασταλεί στο παρελθόν, και (4) των χαρακτηριστικών των τρεχουσών ανταγωνιστικών εργασιών. Ο τελευταίος παράγοντας είναι ο πιο δύσκολος να υπολογιστεί επειδή απαιτεί να προβλεφθούν ποιες ανταγωνιστικές εργασίες μπορεί να υπάρχουν κατά την διάρκεια μίας δεύτερης ή τρίτης απόπειρας επιλογής μίας εργασίας η οποία να έχει αναβληθεί στο παρελθόν. Η ικανότητα αυτή της πρόβλεψης πιθανών ανταγωνιστικών εργασιών στο μέλλον δεν συμπεριλαμβάνεται στη παρούσα διατριβή η οποία λαμβάνει υπόψη τις ανταγωνιστικές εργασίες στην τρέχουσα προσπάθεια μόνο. Το γινόμενο της πιθανότητας και του κόστους αποτελεί το ρίσκο μίας συνέπειας το οποίο αθροίζεται με τις υπόλοιπες συνέπειες. Το άθροισμα αυτό προστίθεται στο κόστος διακοπής με σκοπό τον υπολογισμό ενός δείκτη προτεραιότητας (priority index) για μία νέα ή υπό αναστολή εργασία. Ο μηχανισμός επιλογής της πιο σημαντικής εργασίας επιλέγει την εργασία με την υψηλότερο δείκτη προτεραιότητας η οποία στην συνέχεια προστίθεται στην λίστα των εργασιών που περιμένουν περαιτέρω επεξεργασία. Οι εργασίες με τον ίδιο βαθμό προτεραιότητας προγραμματίζονται σύμφωνα με την αρχική τους προγραμματισμένη ώρα έναρξης ή ανάλογα με την διάρκειά τους (επιλέγεται η πιο σύντομη πρώτα).

Ο προγραμματισμός των εργασιών επηρεάζεται από τις στρατηγικές τις οποίες οι εργαζόμενοι χρησιμοποιούν για να διαχειριστούν τον φόρτο εργασίας τους σε διάφορες χρονικές στιγμές. Στα πλαίσια της παρούσας διατριβής, περιλαμβάνονται τέσσερις στρατηγικές οι οποίες ενεργοποιούνται όταν ο φόρτος εργασίας ενός εργαζομένου υπερβαίνει ένα προκαθορισμένο κατώφλι (όριο) [36]:

(1) Δίνεται ένα πέναλτι, υπό την μορφή της μειωμένης ακρίβειας ή ταχύτητας εκτέλεσης της εργασίας.

(2) Η νέα εργασία δεν ξεκινάει από κανένα εργαζόμενο το οποίο αναφέρεται και ως αναβολή ή παράλειψη της εργασίας.

(3) Η νέα εργασία πραγματοποιείται από τον εργαζόμενο σειριακά και όχι παράλληλα με τις υπόλοιπες υπό εξέλιξη εργασίες.

(4) Η νέα εργασία δίνεται σε κάποιον άλλον εργαζόμενο ο οποίος είναι λιγότερο απασχολημένος.

Η επιλογή μίας από τις παραπάνω στρατηγικές, όποτε αυτό κρίνεται απαραίτητο, γίνεται με τρόπο τυχαίο. Επιπλέον στρατηγικές για την διαχείριση του φόρτου εργασίας των εργαζομένων μπορούν να προσδιοριστούν συνδυάζοντας τις τέσσερις προαναφερθείσες βασικές στρατηγικές. Για τον λόγο αυτό, ο προγραμματισμός των εργασιών προσδιορίζεται από το στιγμιαίο φόρτο εργασίας των εργαζομένων και την επιλεγμένη στρατηγική διαχείρισης του φόρτου εργασίας.

Διαχείριση Μνήμης

Μία προσωρινή ουρά από εργασίες παράγεται και ρυθμίζεται σύμφωνα με ένα μοντέλο της ανθρώπινης μνήμης με σκοπό την διαχείριση ενός μεγάλου αριθμού από νέες υποψήφιες εργασίες, υπό αναστολή εργασίες, εργασίες οι οποίες αναμένουν επεξεργασία (δηλαδή η λίστα με τις επιλεγμένες εργασίες) και από εργασίες υπό εκτέλεση (δηλαδή η λίστα με τις ενεργές εργασίες). Η έννοια του επιπέδου ενεργοποίησης (activation level) μίας υπό αναστολή εργασίας περιγράφεται στο [49] και [50] και ουσιαστικά αναφέρεται στο επίπεδο ετοιμότητας της υπό αναστολή εργασίας ώστε να ενεργοποιηθεί ξανά και να ετοιμαστεί προς επεξεργασία. Όταν το επίπεδο ενεργοποίησης υπερβεί ένα προκαθορισμένο όριο τότε η εργασία θεωρείται έτοιμη να ενεργοποιηθεί ξανά. Το μοντέλο της μνήμης της παρούσας διατριβής θεωρεί ως δεδομένο πως όλες οι υπό αναστολή εργασίες έχουν το ίδιο επίπεδο ενεργοποίησης. Κατά συνέπεια, όταν μία εργασία τίθεται υπό αναστολή, μεταβαίνει στην αντίστοιχη δομή (μνήμη) του δικτύου η οποία είναι υπεύθυνη για την

καταγραφή των υπό αναστολή εργασιών, και στην συνέχεια, όταν ολοκληρωθεί η εκτέλεση (επιτυχής ή ανεπιτυχής) του συνόλου των εργασιών που συνθέτουν έναν στόχο (goal) τότε ανακαλούνται από την μνήμη οι εργασίες του συγκεκριμένου στόχου οι οποίες βρίσκονται υπό αναστολή και διοχετεύονται εκ νέου στο δίκτυο. Το αποτέλεσμα της εκτέλεσής τους μπορεί να είναι είτε πετυχημένο είτε αποτυχημένο είτε να ανασταλούν εκ νέου. Στην τρίτη περίπτωση, οι υπό αναστολή εργασίες μεταφέρονται πάλι στην μνήμη που καταγράφει τις υπό αναστολή εργασίες και με την ολοκλήρωση του στόχου επαναλαμβάνεται η παραπάνω διαδικασία.

Οργάνωση Εργασιών

Για να μοντελοποιηθεί η απόδοση ενός συνόλου από εργαζόμενους κρίνεται απαραίτητη η ύπαρξη ενός συστήματος οργάνωσης των εργασιών το οποίο να προσδιορίζει τον τρόπο με τον οποίο η εργασίες κατανέμονται στους εργαζόμενους σύμφωνα με ορισμένους συγκεκριμένους κανόνες κατανομής, τις δυνατότητες των εργαζομένων και τα επίπεδα του εργασιακού φόρτου κάθε εργαζόμενου. Οι κανόνες κατανομής μπορεί να είναι εξειδικευμένοι για ένα συγκεκριμένο υποσύνολο εργασιών (για παράδειγμα δύο εργασίες θα πρέπει να εκτελεστούν από τον ίδιο εργαζόμενο), ή μπορεί να αφορούν γενικές στρατηγικές κατανομής εργασιών. Για παράδειγμα, ένας γενικός κανόνας μπορεί να ορίζει πως ένας εργαζόμενος δεν επιτρέπεται να εκτελεί παραπάνω από τέσσερις εργασίες για το χρονικό διάστημα στο οποίο κάποιος άλλος εργαζόμενος δεν θα εκτελούσε καμία εργασία. Για τον λόγο αυτό, είναι πιθανό να υπάρχει ένας βαθμός συμμόρφωσης (compliance score) για κάθε εργαζόμενο ο οποίος να περιγράφει το μέγεθος στο οποίο η απόδοσή του συμμορφώνεται με ένα τοπικό ή γενικό κανόνα. Οι δυνατότητες των εργαζομένων λαμβάνονται υπόψη ενσωματώνοντας την ακρίβεια και ταχύτητα εκτέλεσης μίας εργασίας από τον εργαζόμενο σε έναν αλγόριθμο ο οποίος υπολογίζει ένα δείκτη ανάθεσης (assignment score) για κάθε εργαζόμενο για

εκείνο το χρονικό διάστημα στο οποίο θα πρέπει να ληφθεί μία απόφαση σχετικά με τον τρόπο με τον οποίο θα πρέπει μία εργασία να κατανεμηθεί.

Ο παρακάτω αλγόριθμος προτείνεται για τον υπολογισμό του δείκτη ανάθεσης για κάθε υποψήφιο εργαζόμενο σε κάποια χρονική στιγμή:

$$R = P * (A/D) \qquad (2)$$

όπου,

- R είναι ο δείκτης ανάθεσης εργασίας για έναν εργαζόμενο,
- P είναι η προτεραιότητα μίας συγκεκριμένης εργασίας,
- A είναι η ακρίβεια εκτέλεσης της εργασίας από τον εργαζόμενο, και,
- D είναι η διάρκεια εκτέλεσης της εργασίας από τον εργαζόμενο.

Οι εργασίες μπορεί να έχουν συγκεκριμένες τιμές προτεραιότητας οι οποίες να αλλάζουν με την πάροδο του χρόνου ανάλογα με το πλαίσιο εργασίας. Ωστόσο, η εξίσωση (2) χρησιμοποιεί μία απλούστερη έκφραση για την προτεραιότητα των εργασιών λαμβάνοντας υπόψη μόνο την προεπιλεγμένη τιμή της προτεραιότητας κάθε εργασίας (δηλαδή το κόστος της διακοπής της εργασίας όπως φαίνεται στην εξίσωση (1)). Ο αλγόριθμος της εξίσωσης (2) επιλέγει τον εργαζόμενο με τον υψηλότερο δείκτη ανάθεσης εργασίας ως τον πιο κατάλληλο υποψήφιο για να εκτελέσει την συγκεκριμένη εργασία. Εάν ο επιλεγμένος εργαζόμενος είχε υψηλό φόρτο εργασίας, τότε επιλέγεται μία από τις τέσσερις λειτουργικές στρατηγικές όπως περιγράφηκαν στην ενότητα «Διαχείριση της Προτεραιότητας των Εργασιών».

Ανακεφαλαίωση

Το κεφάλαιο επικεντρώθηκε στις θεωρητικές δομές που έχουν υιοθετηθεί για την αναπαράσταση της ανθρώπινης απόδοσης στα πλαίσια των υπαρχουσών προσεγγίσεων μοντελοποίησης των νοητικών διαδικασιών. Παρουσιάστηκαν τρία υποσυστήματα: (1) ένα υποσύστημα διαχείρισης της προτεραιότητας των εργασιών το οποίο υπολογίζει και χρησιμοποιεί ένα δείκτη προτεραιότητας,

(2) ένα υποσύστημα διαχείρισης της μνήμης και (3) ένα υποσύστημα οργάνωσης εργασιών το οποίο υπολογίζει και χρησιμοποιεί έναν δείκτη ανάθεσης εργασίας.

Κεφάλαιο 6: Η Μοντελοποίηση με τα Έγχρωμα Δίκτυα Πέτρι

Εισαγωγή

Το κεφάλαιο περιλαμβάνει την περιγραφή της υλοποίησης του προτεινόμενου πλαισίου της διατριβής χρησιμοποιώντας τα έγχρωμα δίκτυα Πέτρι και το λογισμικό πακέτο CPN Tools.

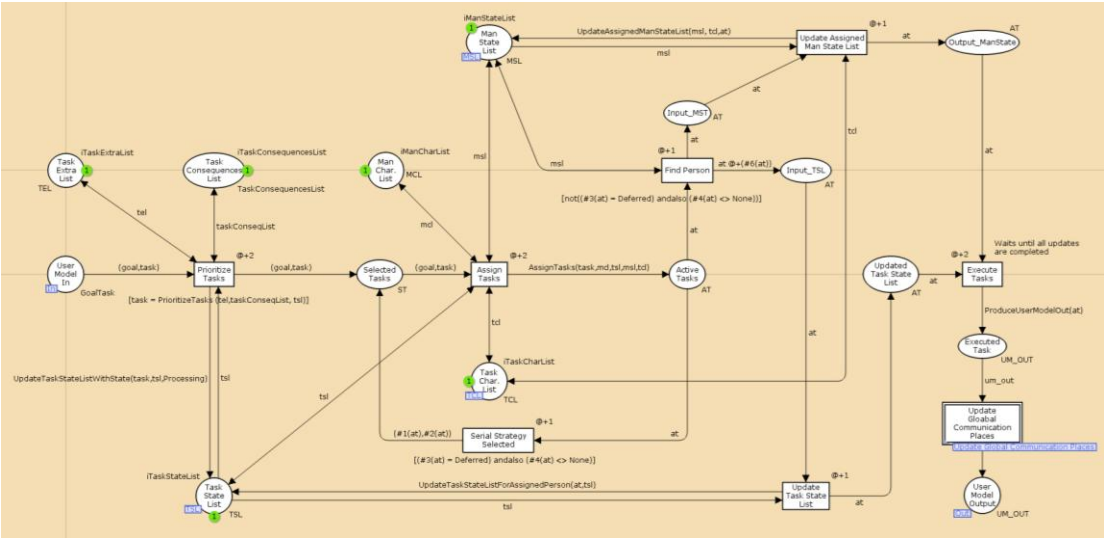
Βασικές Έννοιες

Τα έγχρωμα δίκτυα Πέτρι [28][29] έχουν επιλεγεί ως υποψήφια για την μοντελοποίηση στην παρούσα διατριβή καθώς παρέχουν διευκολύνσεις για την δημιουργία ιεραρχικών και χρονικών περιγραφών, την μετάδοση «πληροφοριακών δομών» (tokens), την προσομοίωση των εργασιών, και λεπτομερείς αναλύσεις μέσω των γράφων προσβασιμότητας (reachability) ή συμβάντων (occurrence). Η ανάλυση των εργασιών μπορεί να πραγματοποιηθεί με την αποσύνθεση των εργασιών σε ιεραρχίες λειτουργιών και πλάνων με τρόπο παρόμοιο με την ιεραρχική ανάλυση των εργασιών (hierarchical task analysis [37]). Για τον σκοπό αυτό, έχει αναπτυχθεί μία ταξινόμηση σχεδίων που αφορά περιγραφές ανθρωπίνων σχεδίων και ακολουθιών εργασιών οι οποίες περιγραφές βασίζονται στα δίκτυα Πέτρι [38]. Οι δομικές και δυναμικές ιδιότητες του δικτύου εργασιών εξετάζονται με την χρήση των γράφων συμβάντων (occurrence graphs). Η λειτουργία προσομοίωσης του εργαλείου CPN Tools χρησιμοποιείται για την συλλογή δεδομένων απόδοσης, όπως είναι ο χρόνος για την εκτέλεση μίας εργασίας, ο χρόνος κατά τον οποίο εργαζόμενοι και μηχανές δεν εξυπηρετούν κάποια

εργασία, η απόδοση των προϊόντων, και ο φόρτος εργασίας των εργαζομένων. Επιπλέον, η προσομοίωση των εργασιών επιτρέπει στους αναλυτές να εξερευνούν μία ποικιλία από ερωτήματα σχετικά με πτυχές της υλοποίησης των πλάνων. Ορισμένες από αυτές τις ερωτήσεις συμπεριλαμβάνουν τα ακόλουθα: υπάρχουν αρκετοί πόροι (εργαζόμενοι και εργαλεία) για την επίτευξη ενός πλάνου; Πόσο διάστημα θα χρειαστεί για την επίτευξη ενός πλάνου; Υπάρχουν περίοδοι όπου κάποιοι εργαζόμενοι υπερφορτώνονται ή περιμένουν χωρίς να έχουν εργασία; Ποια αδιέξοδα θα πρέπει να αποφευχθούν για να εξασφαλιστεί πως κάποιος στόχος θα επιτευχθεί; Ποιες λειτουργίες θα πρέπει να εκτελεστούν παράλληλα για να επιταχυνθεί μία εργασία χωρίς να δημιουργούνται αδιέξοδα; Αυτές οι ερωτήσεις αποτέλεσαν τον οδηγό για την παρούσα πρόταση για την μοντελοποίηση των λειτουργιών των εργασιών με την χρήση των δικτύων Πέτρι.

Τυπικά, η δομή ενός δικτύου Πέτρι είναι ένας γράφος διπλής κατεύθυνσης της μορφής $G = (P, T, A)$, όπου $P = \{p_1, p_2, p_3,\ldots, p_n\}$ είναι ένα πεπερασμένο σύνολο από θέσεις (places), $T = \{t_1, t_2, t_3,\ldots, t_n\}$ είναι ένα πεπερασμένο σύνολο από μεταβάσεις (transitions), και $A = (P \times T) \cup (T \times P)$ είναι ένα σύνολο από κατευθυνόμενες ακμές (arcs). Το σύνολο των θέσεων εισόδου μίας μετάβασης (t) δίνεται από το $I(t) = \{p \mid (p,t) \in A\}$ και το σύνολο των θέσεων εξόδου δίνεται από το $O(t) = \{p \mid (t,p) \in A\}$. Ένας γράφος ενός δικτύου Πέτρι αποτελείται από τα ακόλουθα θεμελιώδη στοιχεία:

- *Ένα γράφο δικτύου (net graph).* Αποτελείται από κόμβους (nodes), δηλαδή θέσεις (places) και μεταβάσεις (transitions), και από ακμές (arcs) οι οποίες συνδέουν τις θέσεις με τις μεταβάσεις (Εικόνα 3).

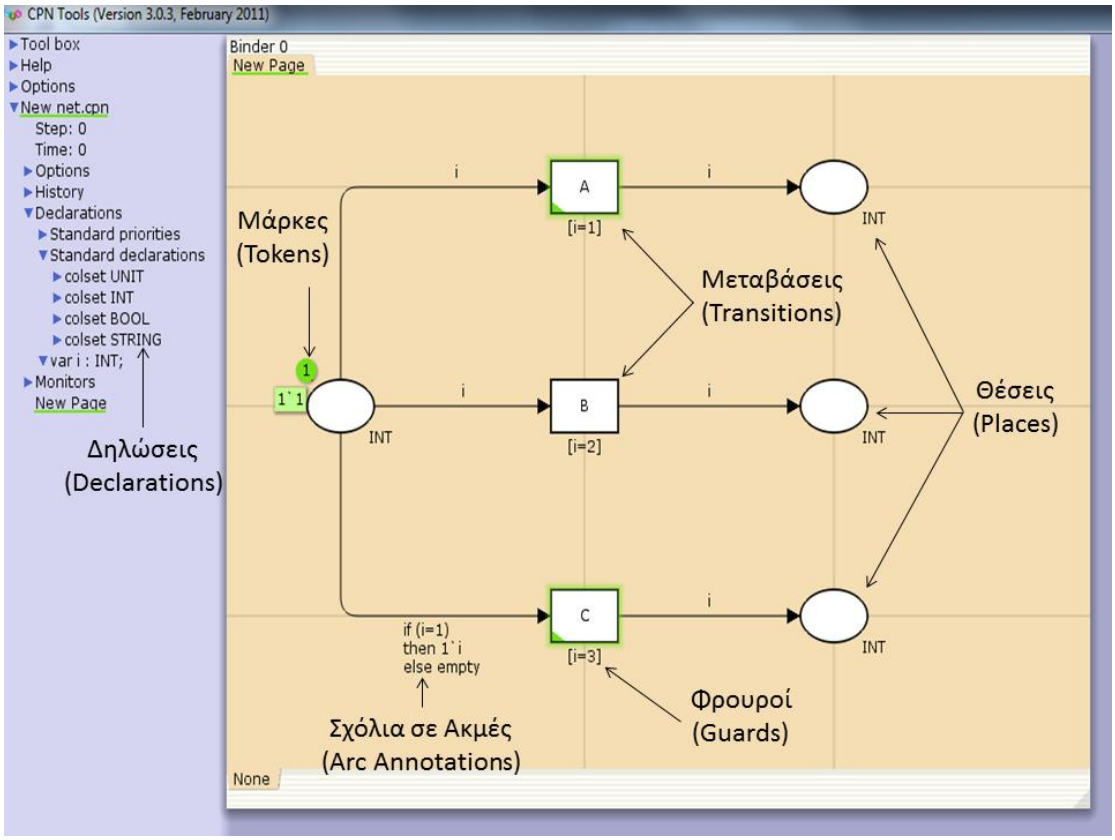


Εικόνα 3: Δείγμα Γράφου Δικτύου (Net Graph)

- *Θέσεις (places).* Είναι παθητικά στοιχεία του δικτύου τα οποία μπορούν να περιγράφουν αρχικές και τελικές καταστάσεις των εργασιών, προϋποθέσεις, και buffers για να διατηρούν εργαλεία και πόρους. Οι θέσεις αναπαριστώνται ως κύκλοι ή οβάλ σχήματα.
- *Μάρκες (tokens).* Είναι στοιχεία που μεταφέρουν δεδομένα τα οποία υπάρχουν μέσα στις θέσεις και μετακινούνται μέσα στο δίκτυο. Κάθε θέση μπορεί να περιέχει αρκετές μάρκες του ίδιου τύπου δεδομένων οι οποίες συνήθως αναπαριστώνται ως μικροί και συμπαγείς κύκλοι.
- *Σχόλια σε ακμές (arc annotations).* Οι ακμές μπορούν να περιέχουν επιγραφές με εκφράσεις οι οποίες μπορούν να περιλαμβάνουν σταθερές, μεταβλητές και συναρτήσεις. Οι εκφράσεις αξιολογούνται αντικαθιστώντας τις μεταβλητές με τις τιμές τους. Οι επιγραφές πάνω στις μεταβάσεις (φρουροί - guards) αξιολογούνται ως boolean τιμές.
- *Μεταβάσεις (transitions).* Είναι τα ενεργά στοιχεία του δικτύου τα οποία μπορούν να χρησιμοποιηθούν για την αναπαράσταση εργασιών, την μεταφορά πόρων, την δρομολόγηση εργασιών και την επεξεργασία γεγονότων. Υπάρχουν πολλά λογισμικά πακέτα τα οποία επιτρέπουν στους σχεδιαστές να γράφουν κώδικα μέσα στις μεταβάσεις ώστε να γίνεται

δυνατός ο έλεγχος της ροής των μαρκών και η αξιολόγηση των σχολίων πάνω στις ακμές.

- *Δηλώσεις (declarations)*. Αναφέρονται στους τύπους δεδομένων οι οποίοι περιέχονται στις θέσεις, τους τύπους των μεταβλητών, και τους προσδιορισμούς αρκετών συναρτήσεων.



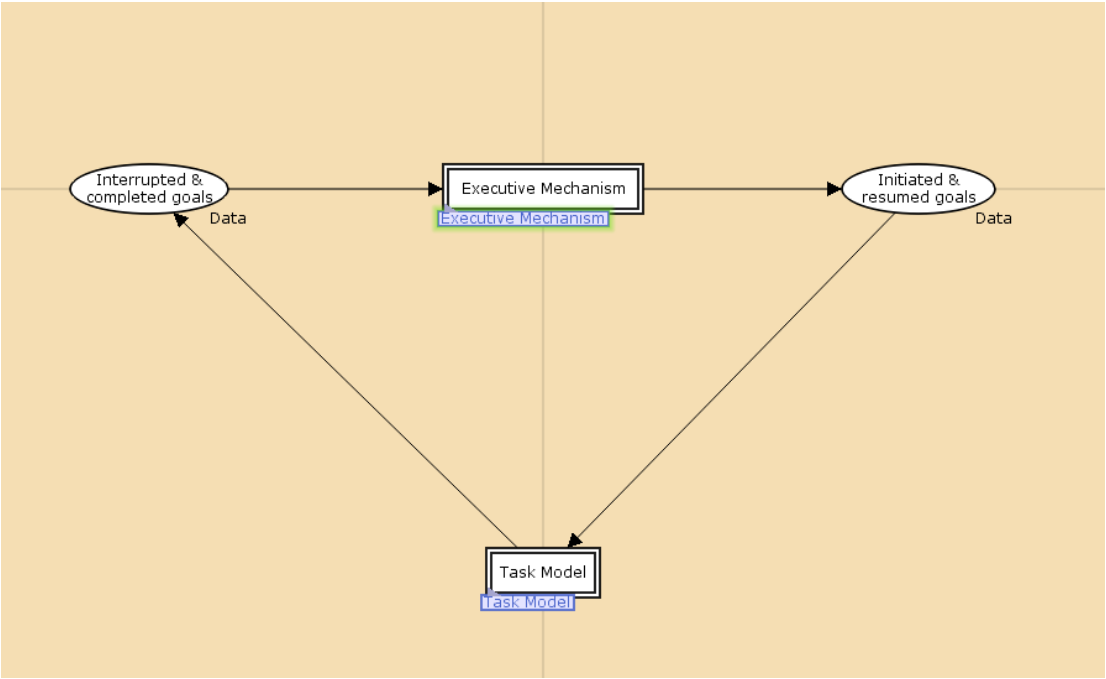
Εικόνα 4: Παράδειγμα Βασικών Εννοιών Δικτύων Πέτρι

Το μοντέλο ενός συστήματος με την χρήση έγχρωμων δικτύων Πέτρι περιγράφει τις καταστάσεις στις οποίες το σύστημα μπορεί να βρεθεί και τις μεταβάσεις μεταξύ αυτών των καταστάσεων. Η κατάσταση του συστήματος αντιστοιχεί στην κατανομή των μαρκών σε όλες τις θέσεις σε οποιαδήποτε χρονική στιγμή. Όταν μία μετάβαση ενεργοποιείται ή εκτελείται (fired), οι μάρκες καταναλώνονται στις θέσεις εισόδου και νέες μάρκες δημιουργούνται στις θέσεις εξόδου, σύμφωνα με τους κανόνες που προσδιορίζονται στους κώδικες των ακμών ή των μεταβάσεων. Ένα μοντέλο έγχρωμου δικτύου Πέτρι έχει μία αρχική κατάσταση και μπορεί να έχει αρκετές τελικές καταστάσεις οι

οποίες αναπαριστούν τις επιθυμητές και μη επιθυμητές εξόδους του μοντέλου. Μπορεί να υπάρχουν αρκετές εκατοντάδες ή χιλιάδες μονοπάτια μεταξύ της αρχικής κατάστασης και των τελικών καταστάσεων και αυτό το σύνολο των μονοπατιών αποτελεί τον χώρο των καταστάσεων (state space) του μοντέλου του έγχρωμου δικτύου Πέτρι. Το λογισμικό εργαλείο CPN Tools περιέχει μεθόδους ανάλυσης οι οποίες μπορούν να χρησιμοποιηθούν για την ανάλυση του χώρου καταστάσεων του μοντέλου και την εύρεση αποδοτικών και προβληματικών μονοπατιών μεταξύ των εισόδων και των εξόδων. Μία προσομοίωση αντιστοιχεί σε ένα μονοπάτι του χώρου καταστάσεων και δεν μπορεί να θεωρηθεί ικανή και αντιπροσωπευτική όλων των πιθανών μονοπατιών. Οι επίσημες τεχνικές ανάλυσης επιτρέπουν την ανάλυση ενός μοντέλου έγχρωμου δικτύου Πέτρι χωρίς την ανάγκη διεξαγωγής προσομοιώσεων. Ωστόσο, εάν κάποιος ενδιαφέρεται για την στατιστική αξία κάποιων συγκεκριμένων μεταβλητών σε ένα μοντέλο, τότε οι προσομοιώσεις θεωρούνται αναγκαίες.

Το Θεμελιώδες Επίπεδο Του Δικτύου

Στο επίπεδο περιλαμβάνονται οι βασικές δομές και τα υποσυστήματα του υλοποιημένου δικτύου. Όλα τα υπόλοιπα υποσυστήματα που θα παρουσιαστούν στην συνέχεια περιέχονται εσωτερικά στις θεμελιώδεις αυτές δομές. Το επίπεδο αυτό φαίνεται στην εικόνα παρακάτω.



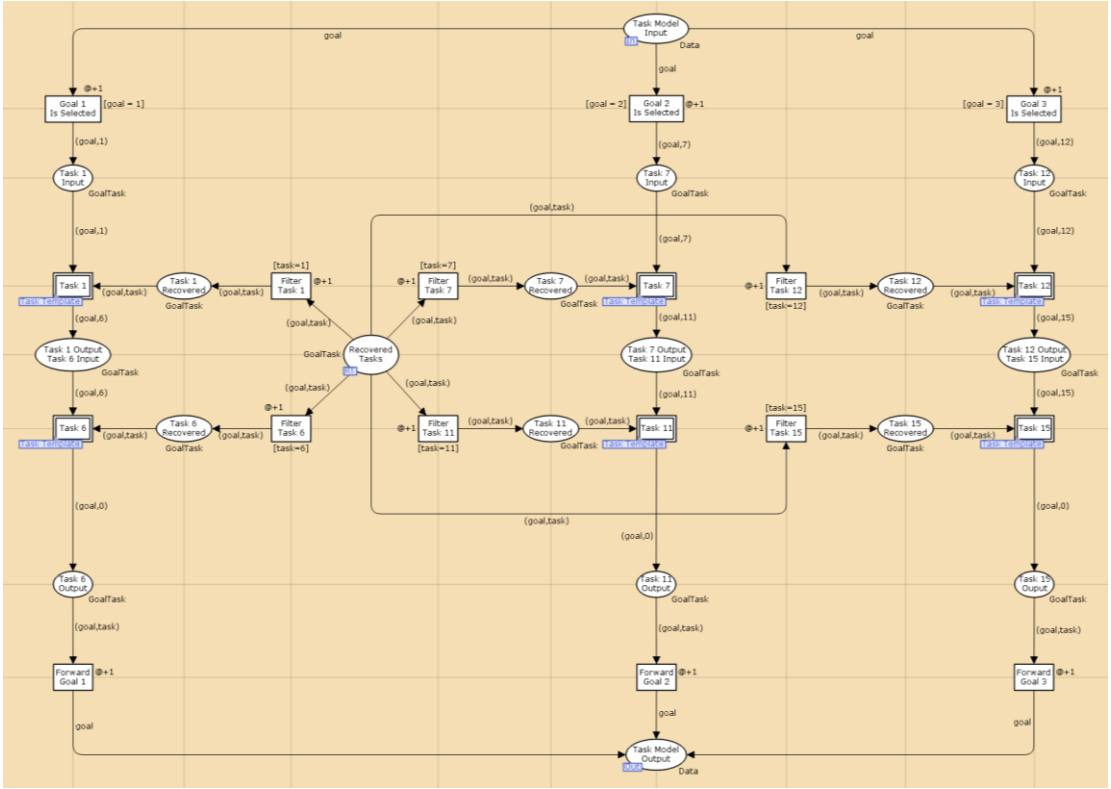
Εικόνα 5: Το Θεμελιώδες Επίπεδο του Δικτύου

Όπως φαίνεται και στην εικόνα, τα δύο βασικά υποσυστήματα του δικτύου είναι ο εκτελεστικό μηχανισμός και το μοντέλο εργασιών. Τα υποσυστήματα αυτά συνδέονται μέσω δύο θέσεων: Η θέση που περιλαμβάνει τους στόχους που έχουν ολοκληρωθεί ή διακοπεί αποτελεί την έξοδο του μοντέλου εργασιών και ταυτόχρονα την είσοδο του εκτελεστικού μηχανισμού, ενώ η θέση που περιλαμβάνει τους στόχους που έχουν εκκινηθεί ή επανεκκινηθεί αποτελεί την έξοδο του εκτελεστικού μηχανισμού και ταυτόχρονα την είσοδο του μοντέλου εργασιών. Και τα δύο υποσυστήματα θα περιγραφούν λεπτομερώς στην συνέχεια.

Το Μοντέλο Εργασιών

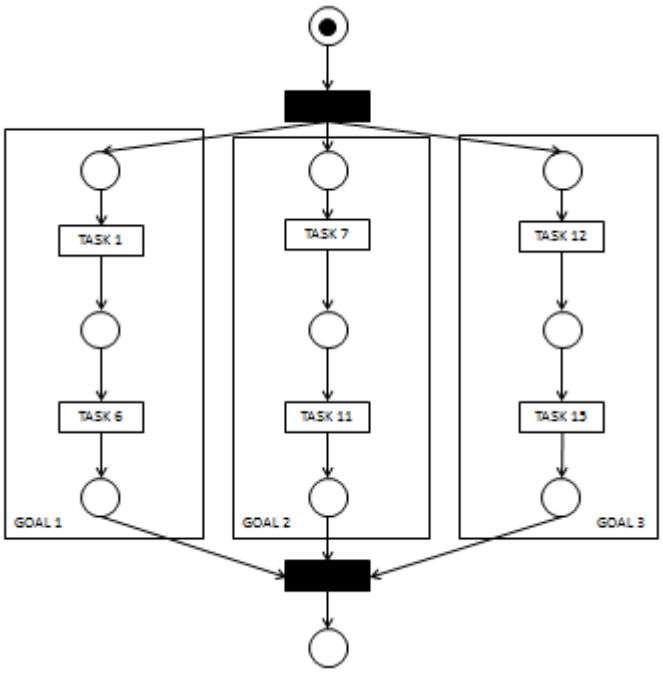
Το μοντέλο εργασιών περιγράφει την ακολουθία των εργασιών η οποία μπορεί να προσαρμοστεί στις μεταβαλλόμενες εργασιακές συνθήκες με δύο τρόπους: (1) *online προσαρμογή,* μέσω του μοντέλου των νοητικών διαδικασιών των εργαζομένων και (2) *offline προσαρμογή,* αλλάζοντας την δομή του μοντέλου εργασιών. Για παράδειγμα, η προσαρμογή της έναρξης ή διάρκειας των

εργασιών, η αλλαγή της σειράς των εργασιών, και η εναλλαγή μεταξύ της σειριακής και παράλληλης επεξεργασίας μπορεί να πραγματοποιηθεί στην online κατάσταση. Από την άλλη πλευρά, η πραγματοποίηση νέων ακολουθιών εργασιών μπορεί να συμβεί μόνο σε offline κατάσταση επειδή αυτό προϋποθέτει ορισμένες αλλαγές στην οργάνωση του δικτύου εργασιών. Η αιτία της offline προσαρμογής είναι πως το μοντέλο εργασιών υλοποιήθηκε ως ένας γράφος δέντρου (Εικόνα 6) με σκοπό να διατηρηθεί συμβατό με τις υπάρχουσες μεθόδους ανάλυσης εργασιών.



Εικόνα 6: Το Μοντέλο Εργασιών (CPN Tools)

Για την καλύτερη κατανόηση του συστήματος, αφαιρώντας τον μηχανισμό ανάκλησης των υπό αναστολή εργασιών, καταλήγουμε στην απλοποιημένη μορφή του παραπάνω συστήματος όπως φαίνεται στην Εικόνα 7:



Εικόνα 7: Το Μοντέλο Εργασιών (αρχιτεκτονική)

Όπως φαίνεται στο διάγραμμα, έχουν προκαθοριστεί 3 στόχοι. Για την εκπλήρωση του πρώτου στόχου πρέπει να ολοκληρωθούν οι εργασίες 1 και 6 σειριακά. Όμοια, για την εκπλήρωση του δεύτερου στόχου πρέπει να ολοκληρωθούν οι εργασίες 7 και 11 σειριακά, ενώ για την εκπλήρωση του τρίτου στόχου πρέπει να ολοκληρωθούν οι εργασίες 12 και 15 σειριακά. Οι τρεις στόχοι εκτελούνται παράλληλα. Στους δύο παρακάτω πίνακες παρουσιάζονται οι θέσεις και οι μεταβάσεις του μοντέλου εργασιών με την περιγραφή τους.

ΘΕΣΗ	ΤΥΠΟΣ ΔΕΔΟΜΕΝΩΝ	ΠΕΡΙΓΡΑΦΗ
Recovered Tasks	GoalTask (INT, INT)	Η θέση η οποία περιέχει όλες τις ανακτημένες εργασίες που προέρχονται από αναστολή τους.
Task 1 Input	GoalTask (INT, INT)	Η θέση εισόδου του προτύπου εργασιών της εργασίας 1 (στόχος 1).
Task 1 Output Task 6 Input	GoalTask (INT, INT)	Η θέση εξόδου του προτύπου εργασιών της εργασίας 1 και παράλληλα η θέση εισόδου του προτύπου εργασιών της εργασίας 6.
Task 1 Recovered	GoalTask (INT, INT)	Η θέση εισόδου του προτύπου εργασιών της εργασίας 1 στην περίπτωση που αυτή ανακτήθηκε ύστερα από αναστολή.
Task 11 Output	GoalTask (INT, INT)	Η θέση εξόδου του προτύπου εργασιών της εργασίας 11.
Task 11 Recovered	GoalTask (INT, INT)	Η θέση εισόδου του προτύπου εργασιών της εργασίας 11 στην περίπτωση που αυτή ανακτήθηκε ύστερα από αναστολή.
Task 12 Input	GoalTask (INT, INT)	Η θέση εισόδου του προτύπου εργασιών της εργασίας 12 (στόχος 3).
Task 12 Output Task 15 Input	GoalTask (INT, INT)	Η θέση εξόδου του προτύπου εργασιών της εργασίας 12 και παράλληλα η θέση εισόδου του προτύπου εργασιών της εργασίας 15.
Task 12 Recovered	GoalTask (INT, INT)	Η θέση εισόδου του προτύπου εργασιών της εργασίας 12 στην περίπτωση που αυτή ανακτήθηκε ύστερα από αναστολή.
Task 15 Output	GoalTask (INT, INT)	Η θέση εξόδου του προτύπου εργασιών της εργασίας 15.
Task 15 Recovered	GoalTask (INT, INT)	Η θέση εισόδου του προτύπου εργασιών της εργασίας 15 στην περίπτωση που αυτή ανακτήθηκε ύστερα από αναστολή.
Task 6 Output	GoalTask (INT, INT)	Η θέση εξόδου του προτύπου εργασιών της εργασίας 6.
Task 6 Recovered	GoalTask (INT, INT)	Η θέση εισόδου του προτύπου εργασιών της εργασίας 6 στην περίπτωση που αυτή ανακτήθηκε ύστερα από αναστολή.
Task 7 Input	GoalTask (INT, INT)	Η θέση εισόδου του προτύπου εργασιών της εργασίας 7.
Task 7 Output Task 11 Input	GoalTask (INT, INT)	Η θέση εξόδου του προτύπου εργασιών της εργασίας 7 και παράλληλα η θέση εισόδου του προτύπου εργασιών της εργασίας 11.
Task 7 Recovered	GoalTask (INT, INT)	Η θέση εισόδου του προτύπου εργασιών της εργασίας 7 στην περίπτωση που αυτή ανακτήθηκε ύστερα από αναστολή.
Task Model Input	Goal (INT)	Η θέση εισόδου του μοντέλου εργασιών.
Task Model	Goal (INT)	Η θέση εξόδου του μοντέλου εργασιών.

Output		

Πίνακας 1: Πίνακας Θέσεων Μοντέλου Εργασιών

ΜΕΤΑΒΑΣΗ	ΦΥΛΑΚΑΣ	ΧΡΟΝΟΣ	ΠΕΡΙΓΡΑΦΗ
Filter Task 1	task = 1	1	Βοηθητική μετάβαση η οποία ενεργοποιείται μόνο στην περίπτωση που η εργασία που εισέρχεται είναι η 1.
Filter Task 11	task = 11	1	Βοηθητική μετάβαση η οποία ενεργοποιείται μόνο στην περίπτωση που η εργασία που εισέρχεται είναι η 11.
Filter Task 12	task = 12	1	Βοηθητική μετάβαση η οποία ενεργοποιείται μόνο στην περίπτωση που η εργασία που εισέρχεται είναι η 12.
Filter Task 15	task = 15	1	Βοηθητική μετάβαση η οποία ενεργοποιείται μόνο στην περίπτωση που η εργασία που εισέρχεται είναι η 15.
Filter Task 6	task = 6	1	Βοηθητική μετάβαση η οποία ενεργοποιείται μόνο στην περίπτωση που η εργασία που εισέρχεται είναι η 6.
Filter Task 7	task = 7	1	Βοηθητική μετάβαση η οποία ενεργοποιείται μόνο στην περίπτωση που η εργασία που εισέρχεται είναι η 7.
Forward Goal 1	N/A	1	Η μετάβαση που είναι υπεύθυνη να προωθήσει τον επεξεργασμένο στόχο 1 στην έξοδο του μοντέλου εργασιών.
Forward Goal 2	N/A	1	Η μετάβαση που είναι υπεύθυνη να προωθήσει τον επεξεργασμένο στόχο 2 στην έξοδο του μοντέλου εργασιών.
Forward Goal 3	N/A	1	Η μετάβαση που είναι υπεύθυνη να προωθήσει τον επεξεργασμένο στόχο 3 στην έξοδο του μοντέλου εργασιών.
Goal 1 Is Selected	goal = 1	1	Η μετάβαση εισόδου για τον στόχο 1.
Goal 2 Is Selected	goal = 2	1	Η μετάβαση εισόδου για τον στόχο 2.
Goal 3 Is Selected	goal = 3	1	Η μετάβαση εισόδου για τον στόχο 3.
Task 1	N/A	N/A	Το πρότυπο εργασιών της εργασίας 1.
Task 11	N/A	N/A	Το πρότυπο εργασιών της εργασίας 11.
Task 12	N/A	N/A	Το πρότυπο εργασιών της εργασίας 12.
Task 15	N/A	N/A	Το πρότυπο εργασιών της εργασίας 15.
Task 6	N/A	N/A	Το πρότυπο εργασιών της εργασίας 6.
Task 7	N/A	N/A	Το πρότυπο εργασιών της εργασίας 7.

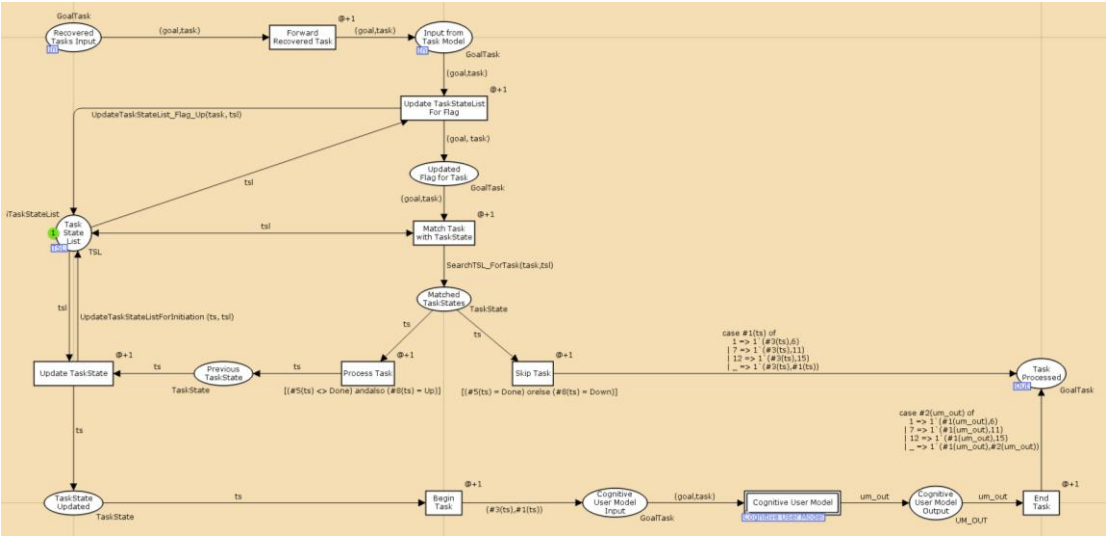
Πίνακας 2: Πίνακας Μεταβάσεων Μοντέλου Εργασιών

Όπως παρουσιάζεται στο [38], αναπτύχθηκε μία σημειολογία δικτύων Πέτρι η οποία περιελάμβανε 15 πλάνα ή ακολουθίες εργασιών οι οποίες μπορούν να ταιριάξουν με τα πλάνα τα οποία συναντώνται στην ανάλυση ιεραρχικών εργασιών (για παράδειγμα, σειριακή εκτέλεση (πρώτα το Α και μετά το Β), παράλληλη εκτέλεση (το Α μαζί με το Β), επιλεκτική εκτέλεση (ή το Α ή το Β), εκτέλεση με προτεραιότητες (και το Α και το Β, δίνοντας προτεραιότητα στο Α)). Πράγματι, η μοντελοποίηση εργασιών επιτρέπει έναν πιο ακριβή προσδιορισμό των πλάνων σε σύγκριση με την ανάλυση των εργασιών, με σημειολογία η οποία είναι επιπλέον εκτελέσιμη. Για μία πλήρη παρουσίαση μοτίβων ροών εργασίας (workflow patterns) και των μετασχηματισμών τους σε δίκτυα Πέτρι μπορεί κάποιος να ανατρέξει στο [39].

Οι εργασίες αναπαριστώνται ως μεταβάσεις (transitions – άσπρα ορθογώνια κουτιά στην Εικόνα 7) οι οποίες διαθέτουν θέσεις εισόδου και εξόδου ενώ άλλες μεταβάσεις μπορούν να χρησιμοποιηθούν για την δημιουργία διαφορετικών δρομολογήσεων εργασιών οι οποίες να αντιστοιχούν σε συγκεκριμένα πλάνα ενεργειών (μαύρα ορθογώνια κουτιά στην Εικόνα 7). Οι θέσεις (places) διατηρούν πληροφορίες σχετικά με τις ιδιότητες των εργασιών και των triggers των εργασιών (η θέση εκκίνησης στην Εικόνα 7), ενώ οι μάρκες (tokens) κυκλοφορούν αυτή την πληροφορία σε όλο το δίκτυο (μικροί συμπαγείς μαύροι κύκλοι στην Εικόνα 7). Οι ακολουθίες εργασιών είναι οργανωμένες με τέτοιο τρόπο ώστε να σχηματίζουν διαφορετικά σύνολα τα οποία να αντιστοιχούν στους στόχους των εργαζομένων. Με τον τρόπο αυτό, παράγεται μία ιεραρχική οργάνωση από στόχους (Εικόνα 7). Ορισμένα μοντέλα έγχρωμων δικτύων Πέτρι χρησιμοποιούν τις μάρκες ως αναπαράσταση εργασιών επειδή αυτό επιτρέπει την δυναμική εισαγωγή και αφαίρεση των εργασιών [40][41]. Ωστόσο, αυτή η αρχιτεκτονική χάνει στην απεικόνιση των ακολουθιών των εργασιών η οποία επιτυγχάνεται με την αναπαράσταση των μεταβάσεων ως εργασίες όπως απεικονίζεται στην Εικόνα 7. Με σκοπό την αύξηση του βαθμού της online προσαρμογής, το προτεινόμενο πλαίσιο μετασχηματίζει τις

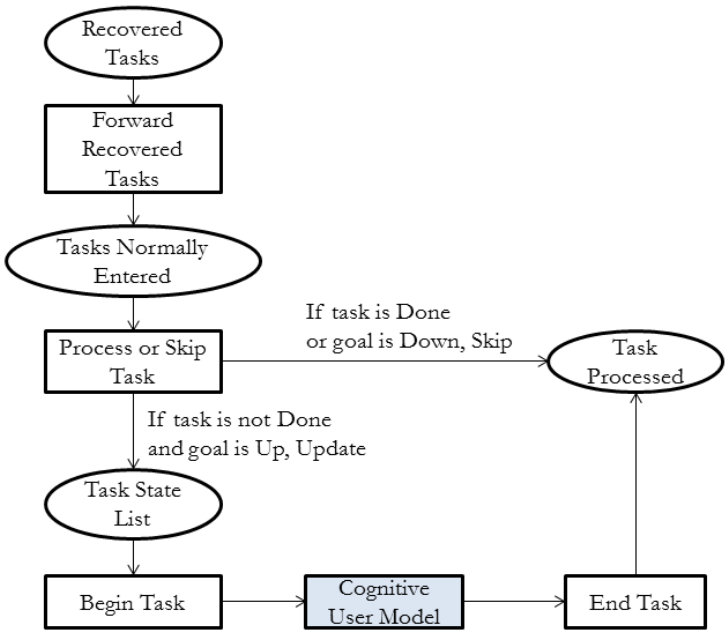
εργασίες σε μάρκες σε επόμενο στάδιο, όταν εισέρχονται στο μοντέλο των νοητικών διαδικασιών των εργαζομένων.

Αυτό επιτυγχάνεται με την χρήση ενός προτύπου εργασιών (όπως φαίνεται στην Εικόνα 8 και στην Εικόνα 9) το οποίο αποσυνθέτει κάθε «μετάβαση εργασίας» σε ένα σύνολο από θέσεις και μεταβάσεις. Η δρομολόγηση της μάρκας από το μοντέλο εργασιών προς το μοντέλο των νοητικών διαδικασιών των εργαζομένων προσδιορίζεται από τον κώδικα στις μεταβάσεις «Process Task» και «Skip Task». Σε αυτό το στάδιο η μάρκα που αναπαριστά την εργασία περιέχει μόνο πληροφορίες σχετικά με την ταυτότητα της εργασίας και του στόχου, ενώ επιπλέον ιδιότητες λαμβάνονται υπόψη στο μοντέλο των νοητικών διαδικασιών των εργαζομένων. Μία εργασία μπορεί να βρίσκεται σε οκτώ καταστάσεις (default, started, processing, done, deferred, upset, failed, interrupted) οι οποίες επηρεάζουν την περαιτέρω επεξεργασία της εργασίας και κατ' επέκταση του στόχου. Οι πληροφορίες σχετικά με την κατάσταση των εργασιών και άλλων ιδιοτήτων των εργασιών διατηρούνται στη θέση με την ονομασία «Task State List» η οποία ενημερώνεται όταν αυτό κρίνεται αναγκαίο και είναι προσβάσιμη σε ολόκληρο το δίκτυο.



Εικόνα 8: Το Πρότυπο Εργασιών (CPN Tools)

Για την καλύτερη κατανόηση του συστήματος, καταλήγουμε στην απλοποιημένη μορφή του παραπάνω συστήματος όπως φαίνεται στην Εικόνα 9:



Εικόνα 9: Το Πρότυπο Εργασιών (αρχιτεκτονική)

Στους δύο παρακάτω πίνακες παρουσιάζονται οι θέσεις και οι μεταβάσεις του προτύπου εργασιών με τις περιγραφές τους.

ΘΕΣΗ	ΤΥΠΟΣ ΔΕΔΟΜΕΝΩΝ	ΠΕΡΙΓΡΑΦΗ
Cognitive User Model Input	GoalTask (INT, INT)	Η θέση εισόδου του μοντέλου των νοητικών διαδικασιών των εργαζομένων.
Cognitive User Model Output	UM_OUT (INT, INT, ENUM)	Η θέση εξόδου του μοντέλου των νοητικών διαδικασιών των εργαζομένων.
Input from Task Model	GoalTask (INT, INT)	Η θέση από την οποία εισέρχονται υπό κανονικές συνθήκες οι εργασίες.
Matched TaskStates	TaskState (INT, INT, INT, ENUM, ENUM, STRING, STRING, ENUM, INT)	Η θέση περιέχει μία δομή που περιλαμβάνει σημαντικές πληροφορίες για την υπό επεξεργασία εργασία.
Previous TaskState	TaskState (INT, INT, INT, ENUM, ENUM, STRING, STRING, ENUM, INT)	Η θέση της εργασίας πριν ενημερωθεί η κατάλληλη "βάση δεδομένων" για την έναρξη της επεξεργασίας της.
Recovered Tasks Input	GoalTask (INT, INT)	Η θέση από την οποία εισέρχονται οι ανακτημένες εργασίες που προέρχονται από αναστολή.
Task Processed	GoalTask (INT, INT)	Η θέση εξόδου των εργασιών από το πρότυπο εργασιών.
Task State List	TSL (list of (INT, INT, INT, ENUM, ENUM, STRING, STRING, ENUM, INT))	Η θέση η οποία περιέχει δεδομένα σχετικά με τις ιδιότητες των εργασιών του δικτύου.
TaskState Updated	TaskState (INT, INT, INT, ENUM, ENUM, STRING, STRING, ENUM, INT)	Η θέση της εργασίας μετά την ολοκλήρωση της ενημέρωσης της κατάλληλης "βάσης δεδομένων" για την έναρξη της επεξεργασίας της.
Updated Flag for Task	GoalTask (INT, INT)	Η θέση από την οποία περνάει μία εργασία όταν έχει ενημερωθεί η κατάστασή της σχετικά με την ετοιμασία της για επεξεργασία.

Πίνακας 3: Πίνακας Θέσεων Προτύπου Εργασιών

ΜΕΤΑΒΑΣΗ	ΦΥΛΑΚΑΣ	ΧΡΟΝΟΣ	ΠΕΡΙΓΡΑΦΗ
Begin Task	N/A	1	Η μετάβαση που σηματοδοτεί την είσοδο της εργασίας στο μοντέλο των νοητικών διαδικασιών των εργαζομένων.
Cognitive User Model	N/A	N/A	Το μοντέλο των νοητικών διαδικασιών των εργαζομένων.
End Task	N/A	1	Η μετάβαση που σηματοδοτεί την έξοδο της εργασίας από το μοντέλο των νοητικών διαδικασιών των εργαζομένων.
Forward Recovered Task	N/A	1	Η μετάβαση που προωθεί τις ανακτημένες εργασίες στο πρότυπο εργασιών.
Match Task with TaskState	N/A	1	Η μετάβαση από την οποία περνώντας μία εργασία αντιστοιχίζεται στην κατάλληλη δομή που περιέχει πληροφορίες σχετικά με τις ιδιότητες της εργασίας.
Process Task	[(#5(ts) <> Done) andalso (#8(ts) = Up)]	1	Η μετάβαση που σηματοδοτεί την έναρξη επεξεργασίας της εργασίας.
Skip Task	[(#5(ts) = Done) orelse (#8(ts) = Down)]	1	Η μετάβαση που σηματοδοτεί την παράλειψη της εργασίας.
Update TaskState	N/A	1	Η μετάβαση από την οποία περνώντας μία εργασία ενημερώνεται η κατάλληλη "βάση δεδομένων" για την έναρξη της επεξεργασίας της.
Update TaskStateList For Flag	N/A	1	Η μετάβαση από την οποία περνώντας μία εργασία ενημερώνεται η κατάλληλη "βάση δεδομένων" για την κατάστασή της (πως ετοιμάζεται για επεξεργασία).

Πίνακας 4: Πίνακας Μεταβάσεων Προτύπου Εργασιών

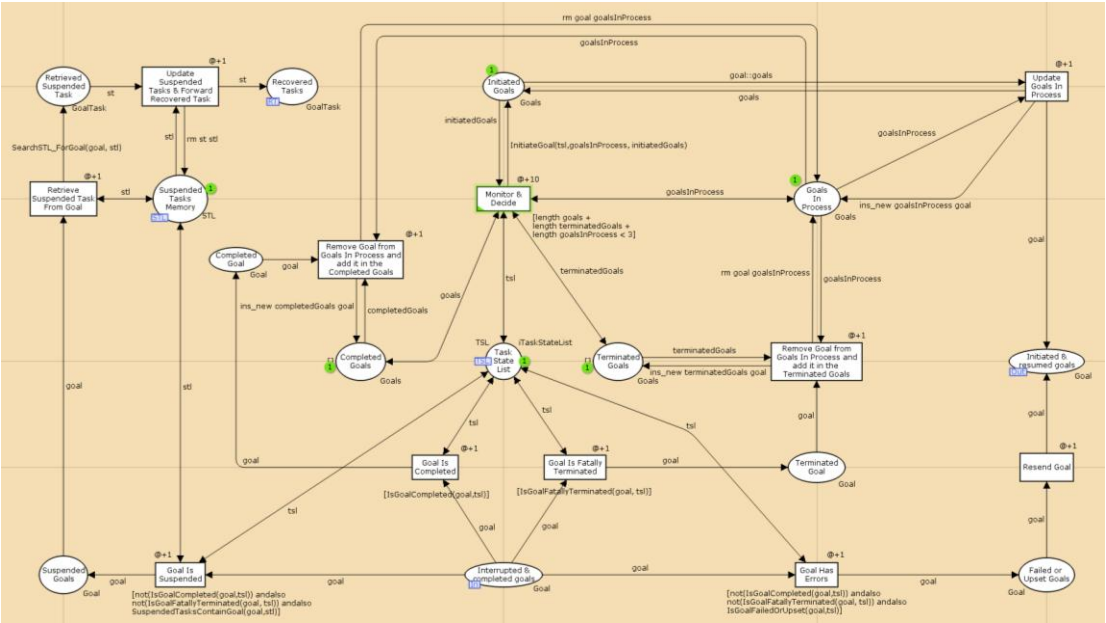
Η λειτουργία του προτύπου εργασιών είναι η εξής: στην αρχή του υποσυστήματος εισάγονται οι ανακτημένες από την μνήμη εργασίες των οποίων η λειτουργία σε κάποιο στάδιο της λειτουργίας του δικτύου είχε ανασταλεί. Επιπλέον εισέρχονται και οι εργασίες οι οποίες ξεκινούν για πρώτη φορά και προέρχονται από το μοντέλο εργασιών. Έπειτα ενημερώνεται η κατάλληλη "βάση δεδομένων" για την κατάσταση της εργασίας, δηλαδή πως ετοιμάζεται για επεξεργασία. Στην συνέχεια, ελέγχεται αν η εργασία έχει ήδη ολοκληρωθεί με επιτυχία ή εάν ανήκει σε κάποιο στόχο ο οποίος βρίσκεται υπό αναστολή στην οποία περίπτωση προωθείται απ' ευθείας στην έξοδο του

προτύπου εργασιών, ενώ, στην αντίθετη περίπτωση, η εργασία προωθείται στο μοντέλο των νοητικών διαδικασιών των εργαζομένων αφού πρώτα ενημερωθεί η μνήμη που διατηρεί την κατάσταση των εργασιών για την έναρξη της εργασίας. Μετά την ολοκλήρωση της εκτέλεσης της εργασίας από το μοντέλο των νοητικών διαδικασιών των εργαζομένων αυτή προωθείται στην έξοδο του προτύπου εργασιών.

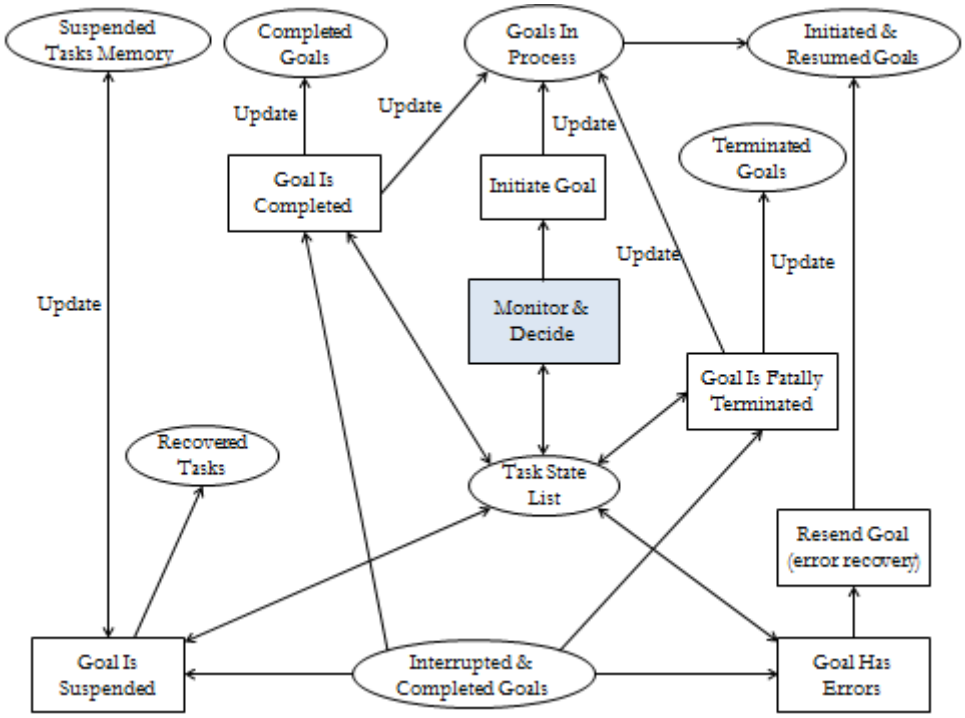
Οι στόχοι επανεκκινούνται από τον εκτελεστικό μηχανισμό ο οποίος παρακολουθεί την κατάσταση του συστήματος και λαμβάνει αποφάσεις σχετικά με τον προγραμματισμό των στόχων. Αξίζει να σημειωθεί πως η ανάκτηση των στόχων που περιέχουν μία ή περισσότερες εργασίες των οποίων η λειτουργία έχει ανασταλεί (interrupted, deferred) διαφέρει από την επανεκκίνηση των στόχων που περιέχουν μία ή περισσότερες εργασίες οι οποίες διακόπηκαν λόγω λάθους (failed, upset). Όταν ένας στόχος περιέχει μία εργασία υπό αναστολή τότε επανεκκινείται μόνο η συγκεκριμένη εργασία χωρίς να υπάρχει ανάγκη επανεκκίνησης όλων των εργασιών από τις οποίες αποτελείται ο στόχος, ενώ όταν ένας στόχος περιέχει μία εργασία η οποία έχει διακοπεί λόγω λάθους τότε επανεκκινείται όλος ο στόχος μέσω της εισαγωγής μίας μάρκας στο μοντέλο εργασιών το οποίο περνάει από όλα τα πρότυπα εργασιών αλλά παρακάμπτει εκείνα τα οποία έχουν ολοκληρωθεί με επιτυχία στο παρελθόν.

Εκτελεστικός Μηχανισμός

Ο εκτελεστικός μηχανισμός ουσιαστικά αποτελείται από ένα σύστημα παρακολούθησης της λειτουργίας του δικτύου και λήψης αποφάσεων καθώς και από ένα σύστημα ανάκλησης των υπό αναστολή εργασιών (Εικόνα 10 και Εικόνα 11). Ο εκτελεστικός μηχανισμός λειτουργεί παράλληλα με τα υπόλοιπα υποσυστήματα του δικτύου. Οι διαδικασίες της παρακολούθησης και λήψης αποφάσεων πραγματοποιούνται ανά τακτά χρονικά διαστήματα (10 μονάδες χρόνου). Οι αποφάσεις αφορούν την έναρξη νέων στόχων, τις αλλαγές στην επεξεργασία των στόχων όταν αυτοί περιέχουν εργασίες υπό αναστολή ή εργασίες των οποίων η λειτουργία έχει διακοπεί λόγω λαθών, και την ανάκτηση στόχων που εμφάνισαν κάποιο πρόβλημα στην διεκπεραίωση τους. Επιπλέον, όπως φαίνεται στην Εικόνα 10 και στην Εικόνα 11, η ανάκτηση των υπό αναστολή εργασιών γίνεται με διαφορετικό τρόπο από την ανάκτηση εργασιών που διακόπηκαν λόγω λαθών.



Εικόνα 10: Ο Εκτελεστικός Μηχανισμός (CPN Tools)



Εικόνα 11: Ο Εκτελεστικός Μηχανισμός (αρχιτεκτονική)

Στους δύο παρακάτω πίνακες παρουσιάζονται οι θέσεις και οι μεταβάσεις του εκτελεστικού μηχανισμού με τις περιγραφές τους.

ΘΕΣΗ	ΤΥΠΟΣ ΔΕΔΟΜΕΝΩΝ	ΠΕΡΙΓΡΑΦΗ
Completed Goal	Goal (INT)	Ο στόχος ο οποίος έχει ολοκληρωθεί, πριν την ενημέρωση των υπό εκτέλεση στόχων σχετικά με την ολοκλήρωση της εκτέλεσής του.
Completed Goals	Goals (list of INT)	Η λίστα με τους ολοκληρωμένους στόχους.
Failed or Upset Goals	Goal (INT)	Ο στόχος ο οποίος έχει αποτύχει λόγω λαθών και ετοιμάζεται για την εκ νέου προώθησή του στο δίκτυο.
Goals In Process	Goals (list of INT)	Η λίστα με τους υπό επεξεργασία στόχους.
Initiated & resumed goals	Goal (INT)	Οι στόχοι που έχουν εκκινηθεί ή επανεκκινηθεί ύστερα από λάθη κατά την διάρκεια εκτέλεσής τους.
Initiated Goals	Goals (list of INT)	Περιέχει μία λίστα με τις νέες εργασίες που έχουν εκκινηθεί.
Interrupted & completed goals	Goal (INT)	Οι στόχοι που έχουν ολοκληρωθεί με επιτυχία ή χωρίς επιτυχία.
Recovered Tasks	GoalTask (INT, INT)	Η θέση η οποία περιέχει όλες τις ανακτημένες εργασίες που προέρχονται από αναστολή τους.
Retrieved Suspended Task	GoalTask (INT, INT)	Η θέση η οποία περιέχει τις εργασίες που βρίσκονται υπό αναστολή ύστερα από την ανάκτησή της από την κατάλληλη "βάση δεδομένων".
Suspended Goals	Goal (INT)	Ο στόχος ο οποίος βρίσκεται υπό αναστολή πριν την ανάκτηση των κατάλληλων πληροφοριών από την λίστα με τις υπό αναστολή εργασίες.
Suspended Tasks Memory	STL(list of (INT, INT))	Η λίστα με τις υπό αναστολή εργασίες.
Task State List	TSL (list of (INT, INT, INT, ENUM, ENUM, STRING, STRING, ENUM, INT))	Η θέση η οποία περιέχει πληροφορίες σχετικά με τις ιδιότητες των εργασιών του δικτύου.
Terminated Goal	Goal (INT)	Ο στόχος ο οποίος έχει τερματιστεί λόγω πολλών λαθών, πριν την ενημέρωση των υπό εκτέλεση στόχων σχετικά με τον τερματισμό της εκτέλεσής του.
Terminated Goals	Goals (list of INT)	Η λίστα με τους τερματισμένους (λόγω λαθών) στόχους.

Πίνακας 5: Πίνακας Θέσεων Εκτελεστικού Μηχανισμού

ΜΕΤΑΒΑΣΗ	ΦΥΛΑΚΑΣ	ΧΡΟΝΟΣ	ΠΕΡΙΓΡΑΦΗ
Goal Has Errors	[not(IsGoalCompleted(goal,tsl)) andalso not(IsGoalFatallyTerminated(goal, tsl)) andalso IsGoalFailedOrUpset(goal,tsl)]	1	Ενεργοποιείται μόνο για τους στόχους οι οποίοι έχουν διακοπεί λόγω λαθών.
Goal Is Completed	IsGoalCompleted(goal, tsl)	1	Ενεργοποιείται μόνο για τους ολοκληρωμένους στόχους.
Goal Is Fatally Terminated	IsGoalFatallyTerminated(goal, tsl)	1	Ενεργοποιείται μόνο για τους μόνιμα αποτυχημένους στόχους.
Goal Is Suspended	not(IsGoalCompleted(goal,tsl)) andalso not(IsGoalFatallyTerminated(goal, tsl)) andalso SuspendedTasksContainGoal(goal,stl)	1	Ενεργοποιείται μόνο για τους στόχους οι οποίοι βρίσκονται υπό αναστολή.
Monitor & Decide	length goals + length terminatedGoals + length goalsInProcess < 3	10	Η μετάβαση παρακολούθησης και λήψης αποφάσεων.
Remove Goal from Goals In Process and add it in the Completed Goals	N/A	1	Ενημερώνει τη λίστα με τους υπό επεξεργασία στόχους για την ολοκλήρωση του στόχου και προσθέτει τον ολοκληρωμένο στόχο στην κατάλληλη λίστα με τους υπόλοιπους ολοκληρωμένους στόχους.
Remove Goal from Goals In Process and add it in the Terminated Goals	N/A	1	Ενημερώνει τη λίστα με τους υπό επεξεργασία στόχους για τον τερματισμό του στόχου και προσθέτει τον αποτυχημένο

			στόχο στην κατάλληλη λίστα με τους υπόλοιπους στόχους που έχουν μόνιμα αποτύχει.
Resend Goal	N/A	1	Εκ νέου προώθηση ενός στόχου που είχε διακοπεί λόγω λαθών στο δίκτυο.
Retrieve Suspended Task From Goal	N/A	1	Ανακτά την υπό αναστολή εργασία από την λίστα με τις υπό αναστολή εργασίες με βάση τον στόχο στον οποίο ανήκει η εργασία.
Update Goals In Process	N/A	1	Ενημερώνει την λίστα με τους υπό επεξεργασία στόχους για την έναρξη επεξεργασίας του στόχου στον οποίο ανήκει η εργασία και την προωθεί προς επεξεργασία.
Update Suspended Tasks & Forward Recovered Task	N/A	1	Ενημερώνει την λίστα με τις υπό αναστολή εργασίες για την επανεκκίνηση της εργασίας και προωθεί την εργασία στις ανακτημένες εργασίες.

Πίνακας 6: Πίνακας Μεταβάσεων Εκτελεστικού Μηχανισμού

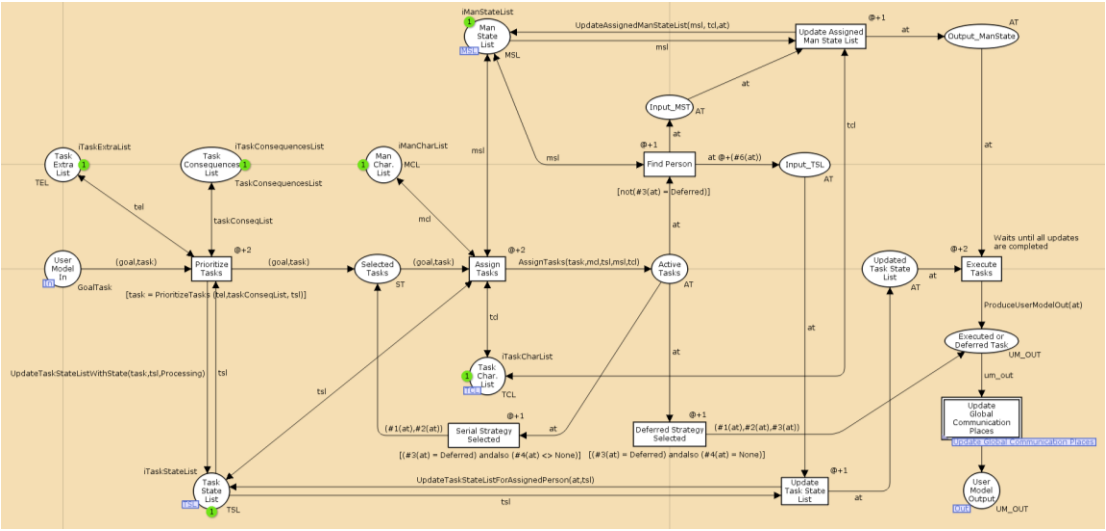
Ο εκτελεστικός μηχανισμός αναλαμβάνει αρχικά να εκκινήσει τους στόχους που έχουν τεθεί στο δίκτυο χρησιμοποιώντας τις πληροφορίες που περιέχονται στην βάση δεδομένων «Task State List» η οποία περιέχει όλους τους στόχους

και τις εργασίες τους καθώς και την κατάσταση των εργασιών. Η διαδικασία φαίνεται στην Εικόνα 11 από την διαδρομή «Monitor & Decide» → «Initiate Goal» → «Goals In Process» → «Initiated & Resumed Goals». Κατά την διάρκεια διεκπεραίωσης ενός στόχου υπάρχει πιθανότητα μία ή περισσότερες εργασίες από τις οποίες αποτελείται ο στόχος να τεθούν υπό αναστολή ή να διακοπούν λόγω κάποιου λάθους. Όταν ολοκληρωθούν (με ή χωρίς επιτυχία) όλες οι εργασίες του στόχου, αυτός επιστρέφει στον εκτελεστικό μηχανισμό μέσω του «Interrupted & Completed Goals». Στην περίπτωση που όλες οι εργασίες του στόχου ολοκληρωθούν με επιτυχία τότε, αφού ενημερωθούν οι κατάλληλες βάσεις δεδομένων («Completed Goals», «Goals In Process») για την επιτυχημένη ολοκλήρωση των εργασιών, ο στόχος ολοκληρώνεται. Στην περίπτωση που ο στόχος περιέχει εργασίες των οποίων η λειτουργία τέθηκε σε αναστολή τότε επανεκκινούνται μέσω του «Recovered Tasks» μόνο οι εργασίες που έχουν τεθεί υπό αναστολή (περιέχονται στην δομή «Suspended Tasks Memory») χωρίς να χρειάζεται να επανεκκινηθεί ολόκληρος ο στόχος. Στην περίπτωση που ο στόχος περιέχει εργασίες των οποίων η λειτουργία διακόπηκε λόγω λαθών τότε επανεκκινείται ολόκληρος ο στόχος και απλά όταν επιχειρείται η επανεκκίνηση εργασιών οι οποίες έχουν ολοκληρωθεί με επιτυχία αυτές παραβλέπονται και δεν εκτελούνται ξανά. Η διαδικασία φαίνεται στην Εικόνα 11 από την διαδρομή «Interrupted & Completed Goals» → «Goal Has Errors» → «Resend Goal (error recovery)» → «Initiated & Resumed Goals». Στην περίπτωση που ο στόχος περιέχει τόσο εργασίες οι οποίες βρίσκονται υπό αναστολή όσο και εργασίες οι οποίες έχουν διακοπεί λόγω λαθών τότε πραγματοποιείται ο συνδυασμός όσων αναφέρθηκαν παραπάνω. Στην περίπτωση που οι εργασίες ενός στόχου αποτύχουν τέσσερις φορές αυτός διακόπτεται μόνιμα καθώς θεωρείται πως παρουσιάζει κάποιο σοβαρό πρόβλημα. Στην περίπτωση αυτή, αφού ενημερωθούν οι κατάλληλες βάσεις δεδομένων («Terminated Goals», «Goals In Process») για την αποτυχημένη ολοκλήρωση των εργασιών του στόχου, ο στόχος τερματίζεται.

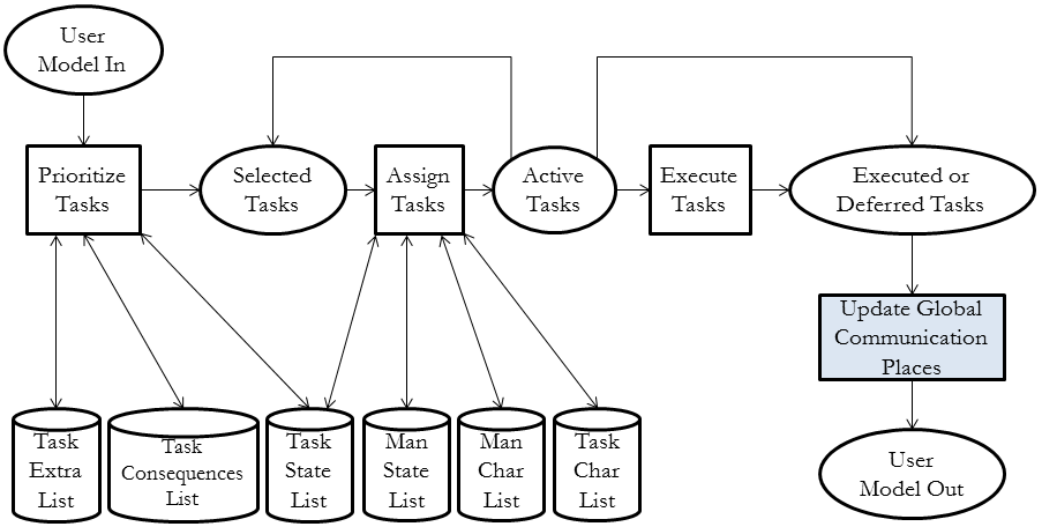
Η παραπάνω διαδικασία συνεχίζεται και επαναλαμβάνεται μέχρι να ολοκληρωθούν όλοι οι στόχοι. Ολοκληρωμένος θεωρείται είτε ο στόχος του οποίου όλες οι εργασίες έχουν ολοκληρωθεί με επιτυχία είτε ο στόχος ο οποίος έχει μόνιμα τερματιστεί λόγω μεγάλου αριθμού λαθών.

Μοντέλο Νοητικών Διαδικασιών Εργαζομένων

Το μοντέλο των νοητικών διαδικασιών των εργαζομένων προσαρμόζει το δίκτυο εργασιών στις μεταβαλλόμενες εργασιακές συνθήκες και περιλαμβάνει ένα σύνολο από υποσυστήματα τα οποία διαχειρίζονται την εργασιακή προσοχή, την διαχείριση του φόρτου εργασίας, την πνευματική παρακολούθηση των εργασιών και την ανθρώπινη αξιοπιστία. Το μοντέλο των νοητικών διαδικασιών των εργαζομένων αποτελείται ουσιαστικά από τέσσερις μεταβάσεις οι οποίες αναλαμβάνουν να διαβάζουν και να γράφουν σε τέσσερις θέσεις οι οποίες διατηρούν τις απαραίτητες για την λειτουργία του δικτύου πληροφορίες και είναι καθολικά προσβάσιμες σε όλο το δίκτυο (global communication places).



Εικόνα 12: Το Μοντέλο των Νοητικών Διαδικασιών των Εργαζομένων (CPN Tools)



Εικόνα 13: Το Μοντέλο των Νοητικών Διαδικασιών των Εργαζομένων (αρχιτεκτονική)

Στους δύο παρακάτω πίνακες παρουσιάζονται οι θέσεις και οι μεταβάσεις του μοντέλου των νοητικών διαδικασιών των εργαζομένων με τις περιγραφές τους.

ΘΕΣΗ	ΤΥΠΟΣ ΔΕΔΟΜΕΝΩΝ	ΠΕΡΙΓΡΑΦΗ
Active Tasks	AT (INT, INT, ENUM, ENUM, INT, INT)	Η ενεργή εργασία η οποία προωθείται προς εκτέλεση.
Executed or Deferred Task	UM_OUT (INT, INT, ENUM)	Η εργασία ύστερα από την εκτέλεση ή αναστολή της.
Input_MST	AT (INT, INT, ENUM, ENUM, INT, INT)	Η ενεργή εργασία (η οποία έχει ανατεθεί σε κάποιον εργαζόμενο) πριν από την ενημέρωση της δομής που διατηρεί την κατάσταση κάθε εργαζόμενου.
Input_TSL	AT (INT, INT, ENUM, ENUM, INT, INT)	Η ενεργή εργασία (η οποία έχει ανατεθεί σε κάποιον εργαζόμενο) πριν από την ενημέρωση της δομής που διατηρεί τα βασικά χαρακτηριστικά κάθε εργασίας.
Man Char. List	MCL (list of (INT, PADs))	Η λίστα που περιέχει πληροφορίες σχετικά με τις ικανότητες και αποδοτικότητα κάθε υπαλλήλου για κάθε τύπο εργασίας.
Man State List	MSL (ENUM, INT, INT)	Η λίστα με τα χαρακτηριστικά του κάθε εργαζόμενου.
Output_ManState	AT (INT, INT, ENUM, ENUM, INT, INT)	Η ενεργή εργασία (η οποία έχει ανατεθεί σε κάποιον εργαζόμενο) ύστερα από την ενημέρωση της δομής που διατηρεί την κατάσταση κάθε εργαζόμενου.
Selected Tasks	ST (INT, INT)	Η επιλεγμένη εργασία ύστερα από την προτεραιοποίηση των υποψήφιων εργασιών.
Task Char. List	TCL (list of (INT, INT, INT, INT, INT))	Η λίστα που περιέχει ορισμένα προκαθορισμένα χαρακτηριστικά για κάθε εργασία (π.χ. φόρτος για κάποιον εργαζόμενο στον οποίο θα ανατεθεί σε νοητικό επίπεδο).
Task Consequences List	TaskConsequencesList (list of (INT, INT, INT))	Η λίστα με ορισμένες προκαθορισμένες συνέπειες για κάθε εργασία από την μη εκτέλεση της εκάστοτε εργασίας και οι αντίστοιχες πιθανότητες εμφάνισης των συνεπειών αυτών.
Task Extra List	TEL (list of (INT, INT, INT))	Η λίστα με ορισμένα επιπλέον χαρακτηριστικά κάθε εργασίας (π.χ. κόστος διακοπής).
Task State List	TSL (list of (INT, INT, INT, ENUM, ENUM, STRING, STRING, ENUM, INT))	Η θέση η οποία περιέχει δεδομένα σχετικά με τις ιδιότητες των εργασιών του δικτύου.

Updated Task State List	AT (INT, INT, ENUM, ENUM, INT, INT)	Η ενεργή εργασία (η οποία έχει ανατεθεί σε κάποιον εργαζόμενο) ύστερα από την ενημέρωση της δομής που διατηρεί τα βασικά χαρακτηριστικά κάθε εργασίας.
User Model In	GoalTask (INT, INT)	Η είσοδος του μοντέλου των νοητικών διαδικασιών των εργαζομένων.
User Model Output	UM_OUT (INT, INT, ENUM)	Η έξοδος του μοντέλου των νοητικών διαδικασιών των εργαζομένων.

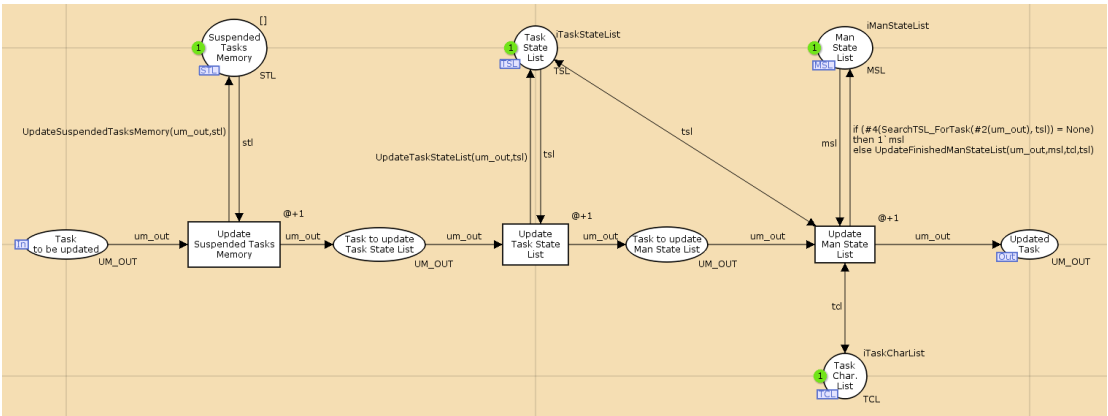
Πίνακας 7: Πίνακας Θέσεων Μοντέλου Νοητικών Διαδικασιών Εργαζομένων

ΜΕΤΑΒΑΣΗ	ΦΥΛΑΚΑΣ	ΧΡΟΝΟΣ	ΠΕΡΙΓΡΑΦΗ
Update Assigned Man State List	N/A	1	Ενημερώνει την κατάλληλη δομή (Man State List) σχετικά με την ανάθεση της εργασίας σε κάποιον συγκεκριμένο εργαζόμενο ενημερώνοντας την εγγραφή που αντιστοιχεί στον εργαζόμενο.
Assign Tasks	N/A	2	Αναθέτει την εργασία σε κάποιο εργαζόμενο με την χρήση κατάλληλων αλγορίθμων όπως αυτοί περιγράφονται στην διατριβή.
Deferred Strategy Selected	[(#3(at) = Deferred) andalso (#4(at) = None)]	1	Υλοποιεί την στρατηγική της αναστολής της εκτέλεσης μίας εργασίας.
Execute Tasks	N/A	2	Εκτελεί την εργασία.
Find Person	[not(#3(at) = Deferred)]	1	Ενεργοποιείται όταν ανατεθεί μία εργασία σε κάποιον εργαζόμενο ώστε να ενημερωθούν οι κατάλληλες εγγραφές.
Prioritize Tasks	[task = PrioritizeTasks (tel,taskConseqList, tsl)]	2	Ταξινομεί τις εργασίες ανάλογα με την σπουδαιότητά τους και επιλέγει την πιο σημαντική.
Serial Strategy Selected	[(#3(at) = Deferred) andalso (#4(at) <> None)]	1	Υλοποιεί την στρατηγική της σειριακής εκτέλεσης δύο εργασιών αντί για την παράλληλη εκτέλεσή τους.
Update Global Communication Places	N/A	N/A	Η μονάδα ενημέρωσης των διαθέσιμων «βάσεων δεδομένων» μετά την ολοκλήρωση της εργασίας.
Update Task State List	N/A	1	Ενημέρωση της λίστας που διατηρεί τα βασικά χαρακτηριστικά των εργασιών σχετικά με την ανάθεση της εργασίας σε κάποιον εργαζόμενο.

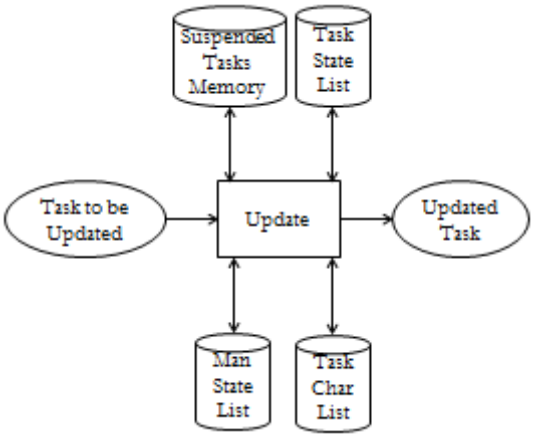
Πίνακας 8: Πίνακας Μεταβάσεων Μοντέλου Νοητικών Διαδικασιών Εργαζομένων

Κάθε μάρκα, η οποία αντιστοιχεί σε μία εργασία, εισέρχεται στο μοντέλο των νοητικών διαδικασιών των εργαζομένων από τη θέση «User Model In». Η μετάβαση «Prioritize Tasks», χρησιμοποιώντας την εξίσωση (1) του κεφαλαίου 5 , ταξινομεί τις εργασίες ανάλογα με την σπουδαιότητά τους και επιλέγει την

πιο σημαντική δίνοντάς της προτεραιότητα συγκριτικά με τις υπόλοιπες υποψήφιες εργασίες. Από την ουρά των εργασιών που έχουν προτεραιότητα στη θέση «Selected Tasks», η μετάβαση «Assign Tasks» επιλέγει εργασίες με σκοπό να τις αναθέσει στο κατάλληλο προσωπικό το οποίο θα επωμιστεί την υποχρέωση της εκτέλεσής τους. Όταν ολοκληρωθεί η ανάθεση του κατάλληλου προσωπικού στις εργασίες, η μετάβαση «Assign Tasks» παράγει μία ακόμα ουρά εργασιών στη θέση «Active Tasks» οι οποίες στην συνέχεια επεξεργάζονται στην μετάβαση «Execute Tasks» από τον εργαζόμενο στον οποίο κάθε εργασία έχει ανατεθεί. Η μετάβαση «Assign Tasks» υπολογίζει τον φόρτο εργασίας των εργαζομένων και υλοποιεί μία στρατηγική διαχείρισης του φόρτου εργασίας σε περίπτωση που ο επιλεγμένος εργαζόμενος αδυνατεί να αναλάβει την εκτέλεση της εργασίας. Για παράδειγμα, όπως φαίνεται στην Εικόνα 12 και στην Εικόνα 13, μπορεί να αποφασιστεί δύο εργασίες να εκτελεστούν σειριακά και όχι παράλληλα, στην οποία περίπτωση μία από τις δύο εργασίες θα επιστρέψει στη θέση «Selected Tasks», ή μπορεί να αποφασιστεί μία εργασία να ανασταλεί οπότε θα προωθηθεί απευθείας στην θέση «Executed or Deferred Task», ή ακόμα μπορεί να αποφασιστεί μία εργασία να ανατεθεί σε κάποιον λιγότερο απασχολημένο εργαζόμενο από τον εργαζόμενο στον οποίο αρχικά είχε ανατεθεί. Επιπλέον, ανθρώπινα λάθη μπορεί να συμβούν κατά την διάρκεια της εκτέλεσης των εργασιών στην μετάβαση «Execute Tasks». Τέλος, μετά την ολοκλήρωση της εκτέλεσης κάθε εργασίας, χρησιμοποιώντας το υποσύστημα «Update Global Communication Places», ενημερώνονται όλες οι κατάλληλες βάσεις δεδομένων οι οποίες περιέχουν πληροφορίες αναφορικά με την εργασία της οποίας η εκτέλεση ολοκληρώθηκε. Στην Εικόνα 14 φαίνεται το υποσύστημα «Update Global Communication Places» και στην Εικόνα 15 φαίνεται η απλοποιημένη έκδοση του ίδιου υποσυστήματος.



Εικόνα 14: Update Global Communication Places (CPN Tools)



Εικόνα 15: Update Global Communication Places (αρχιτεκτονική)

Στους δύο παρακάτω πίνακες παρουσιάζονται οι θέσεις και οι μεταβάσεις του υποσυστήματος «Update Global Communication Places» με τις περιγραφές τους.

ΘΕΣΗ	ΤΥΠΟΣ ΔΕΔΟΜΕΝΩΝ	ΠΕΡΙΓΡΑΦΗ
Man State List	MSL (ENUM, INT, INT)	Η λίστα με τα χαρακτηριστικά του κάθε εργαζόμενου.
Suspended Tasks Memory	STL(list of (INT, INT))	Η λίστα με τις υπό αναστολή εργασίες.
Task Char. List	TCL (list of (INT, INT, INT, INT, INT))	Η λίστα που περιέχει ορισμένα προκαθορισμένα χαρακτηριστικά για κάθε εργασία (π.χ. φόρτος για κάποιον εργαζόμενο στον οποίο θα ανατεθεί σε νοητικό επίπεδο).
Task State List	TSL (list of (INT, INT, INT, ENUM, ENUM, STRING, STRING, ENUM, INT))	Η θέση η οποία περιέχει δεδομένα σχετικά με τις ιδιότητες των εργασιών του δικτύου.
Task to be Updated	UM_OUT (INT, INT, ENUM)	Η είσοδος του υποσυστήματος.
Task to update Man State List	UM_OUT (INT, INT, ENUM)	Η εργασία μετά την ενημέρωση της δομής «Task State List» και πριν την ενημέρωση της δομής «Man State List».
Task to update Task State List	UM_OUT (INT, INT, ENUM)	Η εργασία μετά την ενημέρωση της λίστας των υπό αναστολή εργασιών και πριν την ενημέρωση της δομής «Task State List».
Updated Task	UM_OUT (INT, INT, ENUM)	Η έξοδος του υποσυστήματος.

Πίνακας 9: Πίνακας Θέσεων Υποσυστήματος "Updated Global Communication Places"

ΜΕΤΑΒΑΣΗ	ΦΥΛΑΚΑΣ	ΧΡΟΝΟΣ	ΠΕΡΙΓΡΑΦΗ
Update Man State List	N/A	1	Ενημέρωση της λίστας που περιέχει τα βασικά χαρακτηριστικά των εργαζομένων.
Update Suspended Tasks Memory	N/A	1	Ενημέρωση της λίστας με τις υπό αναστολή εργασίες.
Update Task State List	N/A	1	Ενημέρωση της λίστας που περιέχει τα βασικά χαρακτηριστικά των εργασιών.

Πίνακας 10: Πίνακας Μεταβάσεων Υποσυστήματος "Updated Global Communication Places"

Παρακολούθηση εργασιών στο Μοντέλο Νοητικών Διαδικασιών Εργαζομένων

Η παρακολούθηση των εργασιών περιλαμβάνει την είσοδο των υποψήφιων προς εκτέλεση εργασιών στο μοντέλο των νοητικών διαδικασιών των εργαζομένων από τη θέση «User Model In» (Εικόνα 12 και Εικόνα 13). Η μετάβαση «Prioritize Tasks» δίνει προτεραιότητα σε εργασίες σύμφωνα με τους περιορισμούς κάθε εργασίας, τους χρονικούς περιορισμούς και τις προτεραιότητες οι οποίες υπολογίζονται για κάθε εργασία ξεχωριστά (παράγραφος «Διαχείριση της Προτεραιότητας των Εργασιών» κεφάλαιο 5). Τα δεδομένα τα οποία παραμένουν αμετάβλητα κατά την διάρκεια της παρακολούθησης βρίσκονται αποθηκευμένα σε ξεχωριστά εξωτερικά αρχεία δεδομένων τα οποία περιέχουν, μεταξύ άλλων, πληροφορίες σχετικά με τους περιορισμούς των εργασιών και τα κόστη διακοπής των εργασιών. Τα δυναμικά δεδομένα λαμβάνονται από τη θέση «Task State List» με σκοπό τον υπολογισμό ενός δείκτη προτεραιότητας για κάθε υποψήφια εργασία στο συγκεκριμένο χρονικό σημείο. Οι εργασίες οι οποίες δεν επιλέγονται στο συγκεκριμένο χρονικό σημείο παραμένουν στην θέση τους ενώ η επιλεγμένη εργασία προωθείται στην επόμενη θέση η οποία είναι η «Selected Tasks».

Εκτέλεση εργασιών στο Μοντέλο Νοητικών Διαδικασιών Εργαζομένων

Η εκτέλεση των εργασιών περιλαμβάνει την δημιουργία μίας λίστας ενεργών εργασιών οι οποίες εκτελούνται σύμφωνα με τον χρόνο στον οποίο έχουν καθοριστεί να εκτελεστούν. Η μετάβαση «Assign Tasks» χρησιμοποιεί τον αλγόριθμο ο οποίος περιγράφεται στην ενότητα «Οργάνωση Εργασιών» του κεφαλαίου 5 με σκοπό την ανάθεση των εργασιών στους εργαζομένους. Τα δεδομένα των εργασιών που χρησιμοποιούνται στον αλγόριθμο που προαναφέρθηκε περιλαμβάνονται στις θέσεις «Task State List», «Man State List», «Man Char List» και «Task Char List» οι οποίες περιέχουν, μεταξύ άλλων, εγγραφές οι οποίες ορίζουν τον καταλληλότερο εργαζόμενο για κάθε εργασία, και τις δυνατότητες των εργαζομένων με όρους ακρίβειας και ταχύτητας εκτέλεσης (ή διάρκειας εκτέλεσης) κάθε εργασίας. Για τον λόγο αυτό, παράγεται μία λίστα εργαζομένων και το πρόσωπο με το μεγαλύτερο δείκτη ανάθεσης επιλέγεται να φέρει εις πέρας την τρέχουσα εργασία. Με σκοπό να ληφθεί υπόψη το επίπεδο του φόρτου εργασίας των εργαζομένων, η μετάβαση «Assign Tasks» υπολογίζει ένα δείκτη εργασιακού φόρτου. Όταν ο φόρτος εργασίας των επιλεγμένων εργαζομένων υπερβαίνει ένα προκαθορισμένο όριο, υπάρχει η δυνατότητα της επιλογής μίας από τις διαθέσιμες στρατηγικές διαχείρισης του φόρτου εργασίας όπως περιγράφεται

στην ενότητα «Διαχείριση της Προτεραιότητας των Εργασιών» στο κεφάλαιο 5. Για παράδειγμα, εάν επιλεγεί η σειριακή επεξεργασία δύο εργασιών, τότε η μάρκα που αντιστοιχεί σε μία από αυτές θα επέστρεφε στη θέση «Selected Tasks» όπως φαίνεται στην Εικόνα 12 και στην Εικόνα 13. Για τον υπολογισμό του δείκτη φόρτου εργασίας, τα δεδομένα των εργασιών λαμβάνονται από τη θέση «Task Char List» η οποία περιέχει εγγραφές αναφορικά με τον βαθμό που επιβαρύνεται ένας εργαζόμενος αναλαμβάνοντας την εκτέλεση κάθε εργασίας σε επίπεδο αντίληψης, νόησης και μηχανικής επιβάρυνσης. Το προσδοκώμενο εργασιακό φορτίο υπολογίζεται για κάθε εργαζόμενο ο οποίος μπορεί να εκτελεί επιπλέον εργασίες παράλληλα με την υποψήφια προς ανάθεση εργασία. Οι πληροφορίες αναφορικά με τις υπό εκτέλεση εργασίες οι οποίες εκτελούνται παράλληλα παρέχεται από τη θέση «Task State List». Τέλος, τα επίπεδα εργασιακού φόρτου και της διαθεσιμότητας κάθε εργαζομένου ενημερώνονται στη θέση «Man State List».

Οι εργασίες οι οποίες ανατέθηκαν σε κάποιον εργαζόμενο προστίθενται σε μία ουρά στη θέση «Active Tasks». Μετά το πέρας του χρόνου διάρκειας εκτέλεσης της εκάστοτε εργασίας, οι αντίστοιχες μάρκες των εργασιών αυτών εκτελούνται από την μετάβαση «Execute Tasks». Ανθρώπινα σφάλματα μπορούν να συμβούν κατά την διάρκεια της εκτέλεσης τα οποία καταγράφονται στη θέση «Task State List». Μετά το πέρας της ενημέρωσης όλων των βάσεων δεδομένων σχετικά με τις αλλαγές που πραγματοποιήθηκαν κατά την διάρκεια εκτέλεσης της εργασίας μέσω της μετάβασης «Update Global Communication Places» και ανεξάρτητα του βαθμού της επιτυχίας της εργασίας, παράγεται μία μάρκα που αντιστοιχεί στην υπό επεξεργασία εργασία στη θέση «User Model Output» ώστε να μεταβιβαστεί η πληροφορία στα πρότυπα των εργασιών καθώς και στο δίκτυο εργασιών.

Τα ανθρώπινα λάθη εισάγονται στην εξομοίωση μέσω δύο μεθόδων ή με τον συνδυασμό αυτών των δύο μεθόδων: (1) χειροκίνητα, σύμφωνα με την οποία ο σχεδιαστής του δικτύου μπορεί να εισάγει χειροκίνητα διαφορετικούς τύπους λαθών με σκοπό να αξιολογήσει την επίδραση στην δρομολόγηση και οργάνωση των εργασιών και (2) μέσω της χρήσης μίας συνάρτησης εισαγωγής λαθών (error inducing function – EIF), με την χρήση της οποίας υπολογίζεται η επίδραση του εργασιακού φόρτου, της κούρασης και της πνευματικής κόπωσης στην ανθρώπινη αξιοπιστία. Σύμφωνα με τη δεύτερη μέθοδο, στα πλαίσια της παρούσας διατριβής, η έκβαση μίας εργασίας κατά την διάρκεια εκτέλεσής της ορίζεται τυχαία με βάση κάποιες προκαθορισμένες πιθανότητες

και πιο συγκεκριμένα μία εργασία έχει 10% πιθανότητα να διακοπεί (interrupted), 10% πιθανότητα να αναβληθεί (deferred), 10% πιθανότητα να αποτύχει (failed), 10% πιθανότητα να ανατραπεί (upset) και 60% πιθανότητα να ολοκληρωθεί με επιτυχία. Επιπλέον ενέργειες οι οποίες οδηγούν σε λάθη, όπως ορίζεται στο [3], μπορούν να μοντελοποιηθούν ξεχωριστά στο μοντέλο εργασιών καθώς συσχετίζονται με αλλαγές στην ακολουθία των εργασιών.

Οι συνέπειες των λαθών μπορούν να πάρουν τις ακόλουθες μορφές: (1) αντίκτυπο στην τρέχουσα εργασία αναφορικά με την ακρίβεια εκτέλεσης ή διάρκεια ολοκλήρωσης της εργασίας, (2) αντίκτυπο στον εργαζόμενο στον οποίο η εργασία ανατέθηκε μέσω της αύξησης του φόρτου εργασίας του εργαζομένου ή την ανάθεση της εργασίας σε άλλον εργαζόμενο, (3) παρενέργειες στις ολοκληρωμένες εργασίες λόγω της καθυστέρησης ολοκλήρωσης του στόχου μέρους του οποίου αποτελεί η εργασία που παρουσίασε πρόβλημα και (4) καθυστέρηση της επεξεργασίας των εργασιών που ακολουθούν σειριακά την εργασία που παρουσίασε το πρόβλημα μέχρι να ξεπεραστεί το πρόβλημα. Οι συνέπειες των λαθών μπορούν να συμβούν είτε άμεσα με την αποτυχία της εργασίας είτε σε επόμενο στάδιο όταν θα πραγματοποιηθεί μία ακόμα προσπάθεια επανάληψης της εργασίας. Όταν μία εργασία διακοπεί ή αναβληθεί τότε ενημερώνεται η θέση «Suspended Tasks Memory». Όταν συμβεί κάποιο λάθος κατά την διάρκεια εκτέλεσης μίας εργασίας, οι πληροφορίες σχετικά με την αποτυχία της εργασίας καταγράφονται στη θέση «Task State List». Τόσο τα δεδομένα της θέσης «Suspended Tasks Memory» όσο και τα δεδομένα που περιέχονται στη θέση «Task State List» χρησιμοποιούνται από τον εκτελεστικό μηχανισμό του δικτύου για την ορθή διαχείριση και επανεκκίνηση των εργασιών που παρουσίασαν κάποιας μορφής πρόβλημα κατά την διάρκεια εκτέλεσής τους.

Οι Τύποι Δεδομένων που Χρησιμοποιούνται στο Δίκτυο

Στον παρακάτω πίνακα παρουσιάζονται οι τύποι δεδομένων που ορίστηκαν στην υλοποίηση της παρούσας διατριβής.

COLSET	ΤΥΠΟΣ ΔΕΔΟΜΕΝΩΝ	ΠΕΡΙΓΡΑΦΗ
INT	int	Ο τύπος δεδομένων integer.
STRING	string	Ο τύπος δεδομένων string.
Goal	INT	Ένας στόχος του δικτύου ο οποίος έχει την μορφή ακεραίου.

Goals	list Goal	Λίστα με τους στόχους του δικτύου.
Task	INT	Μία εργασία του δικτύου η οποία έχει την μορφή ακεραίου.
Accuracy	INT	Η ακρίβεια εκτέλεσης μίας εργασίας από έναν εργαζόμενο. Παίρνει τιμές από 0-100.
Duration	INT	Η διάρκεια εκτέλεσης, υπό κανονικές συνθήκες, μίας εργασίας από έναν εργαζόμενο.
Person	with None | Supervisor | Assistant_A | Assistant_B | Worker_A | Worker_B | Worker_C	Οι εργαζόμενοι του δικτύου. Υπάρχουν διαθέσιμοι ένας επόπτης, δύο βοηθοί και τρεις εργάτες.
PAD	Person * Accuracy * Duration	Βοηθητική δομή που περιέχει την ακρίβεια και την διάρκεια εκτέλεσης μίας εργασίας από έναν εργαζόμενο.
PADs	list PAD	Μία λίστα που αποτελείται από την παραπάνω δομή.
State	with Default | Started | Processing | Done | Deferred | Upset | Failed | Interrupted	Περιγράφει τις πιθανές καταστάσεις που μπορεί να βρεθεί μία εργασία: Default (δεν έχει εκκινηθεί), Started (έχει εκκινηθεί), Processing (υπό επεξεργασία), Done (ολοκληρωμένη), Deferred (υπό αναστολή), Upset (έχει ανατραπεί), Failed (έχει αποτύχει), Interrupted (έχει διακοπεί).
UM_OUT	Goal * Task * State	Η έξοδος του μοντέλου των νοητικών διαδικασιών των εργαζομένων.
ST	Goal * Task	Μία υπό αναστολή εργασία και ο στόχος στον οποίο ανήκει.
AT	Goal * Task * State * Person * Accuracy * Duration	Μία ενεργή εργασία (εργασία που έχει ανατεθεί σε κάποιον εργαζόμενο) που συνοδεύεται από ορισμένα χαρακτηριστικά που χρειάζονται στην συνέχεια για την επεξεργασία της.
ManChar	Task * PADs	Περιλαμβάνει την ακρίβεια και διάρκεια εκτέλεσης μίας εργασίας για όλους τους εργαζομένους του δικτύου.
Perception	INT	Η επιβάρυνση σε επίπεδο αντίληψης για έναν εργαζόμενο όταν αναλάβει μία συγκεκριμένη εργασία. Παίρνει τιμές από 0-100.

Cognitive	INT	Η επιβάρυνση σε νοητικό επίπεδο για έναν εργαζόμενο όταν αναλάβει μία συγκεκριμένη εργασία. Παίρνει τιμές από 0-100.
Motor	INT	Η επιβάρυνση σε μηχανικό επίπεδο για έναν εργαζόμενο όταν αναλάβει μία συγκεκριμένη εργασία. Παίρνει τιμές από 0-100.
TaskChar	Goal * Task * Perception * Cognitive * Motor	Περιλαμβάνει την επιβάρυνση σε νοητικό, μηχανικό και επίπεδο αντίληψης για τον εργαζόμενο που θα αναλάβει την συγκεκριμένη εργασία.
AvailableAt	INT	Το χρονικό σημείο στο οποίο ο εργαζόμενος θα είναι πάλι διαθέσιμος να αναλάβει μία εργασία.
Current Workload	INT	Ο τρέχων φόρτος εργασίας ενός εργαζομένου. Παίρνει τιμές από 0-150.
ManState	Person * AvailableAt * CurrentWorkload	Καταγράφει το χρονικό σημείο που θα είναι διαθέσιμος και τον φόρτο εργασίας ενός εργαζομένου.
Priority Level	INT	Ένα ενδεικτικό επίπεδο προτεραιότητας μίας εργασίας που χρησιμοποιείται σε αλγορίθμους μέσα στο δίκτυο.
StartTime	STRING	Το χρονικό σημείο που ξεκίνησε μία εργασία.
EndTime	STRING	Το χρονικό σημείο που ολοκληρώθηκε μία εργασία.
Flag	with Up | Down	Το Up αντιστοιχεί σε στόχο ο οποίος βρίσκεται σε επεξεργασία, ενώ το Down σε στόχο ο οποίος έχει διακοπεί.
Errors Counter	INT	Είναι ένας μετρητής ο οποίος αυξάνεται κατά μία μονάδα κάθε φορά που η εργασία διακόπτεται. Όταν μία εργασία διακοπεί τέσσερις φορές τότε ο στόχος αποτυγχάνει μόνιμα.
TaskState	Task * PriorityLevel * Goal * Person * State * StartTime * EndTime * Flag * ErrorsCounter	Περιλαμβάνει τα βασικά χαρακτηριστικά που συνοδεύουν μία εργασία και τα οποία έχουν ήδη περιγραφεί παραπάνω.
STL	list ST	Η λίστα των εργασιών που βρίσκονται υπό αναστολή.
TSL	list TaskState	Η λίστα όλων των εργασιών και των βασικών τους χαρακτηριστικών.

MSL	list ManState	Η λίστα που καταγράφει το χρονικό σημείο που θα είναι διαθέσιμος και τον φόρτο εργασίας για καθέναν από τους εργαζόμενους.
MCL	list ManChar	Η λίστα η οποία περιλαμβάνει την ακρίβεια και διάρκεια εκτέλεσης για όλες τις εργασίες από όλους τους εργαζομένους του δικτύου.
TCL	list TaskChar	Η λίστα η οποία περιλαμβάνει την επιβάρυνση σε νοητικό, μηχανικό και επίπεδο αντίληψης για τον εργαζόμενο που θα αναλάβει την εκάστοτε εργασία.
GoalTask	Goal * Task	Δομή που περιλαμβάνει την εργασία και τον στόχο στον οποίο ανήκει.
Interruption Cost	INT	Το κόστος του να διακοπεί ή να αναβληθεί μία εργασία. Χρησιμοποιείται σε αλγόριθμους στο δίκτυο.
PriorityIndex	INT	Ένας δείκτης προτεραιότητας μίας εργασίας που χρησιμοποιείται σε αλγορίθμους μέσα στο δίκτυο.
TaskExtra	Task * InterruptionCost * PriorityIndex	Επιπλέον πληροφορίες για μία εργασία οι οποίες περιλαμβάνουν το κόστος διακοπής ή αναβολής της εργασίας και τον δείκτη προτεραιότητας της εργασίας.
TEL	list TaskExtra	Μία λίστα που περιλαμβάνει τις επιπλέον πληροφορίες για όλες τις εργασίες.
Probability Of Consequence	INT	Η πιθανότητα να συμβεί κάποια συνέπεια για μία εργασία. Παίρνει τιμές από 0-100.
CostOf Consequence	INT	Το κόστος από μία συνέπεια για μία εργασία.
Task Consequence	Task * ProbabilityOf Consequence * CostOf Consequence	Η δομή που περιλαμβάνει για μία εργασία την πιθανότητα να συμβεί κάποια συνέπεια για την συγκεκριμένη εργασία και το κόστος της συνέπειας αυτής.
Task Consequences List	list Task Consequence	Η λίστα που περιλαμβάνει για κάθε εργασία τις πιθανότητες να συμβούν κάποιες συνέπειες για την εκάστοτε εργασία και το κόστος των συνεπειών αυτών.

Πίνακας 11: Οι Τύποι Δεδομένων που Χρησιμοποιούνται στο Δίκτυο

Οι Συναρτήσεις (Functions) που Χρησιμοποιούνται στο Δίκτυο

Στον παρακάτω πίνακα παρουσιάζονται οι συναρτήσεις (functions) που ορίστηκαν και χρησιμοποιούνται στην υλοποίηση της παρούσας διατριβής. Ο κώδικας των συναρτήσεων είναι διαθέσιμος στο Παράρτημα.

ΣΥΝΑΡΤΗΣΗ	**ΠΑΡΑΜΕΤΡΟΙ**	**ΒΟΗΘΗΤΙΚΕΣ ΣΥΝΑΡΤΗΣΕΙΣ (*)**
AssignTasks	task:Task mcl:MCL tsl:TSL msl:MSL tcl:TCL	-ChooseStrategy -CheckSelectedOperator -PADS_LT_Function
ΠΕΡΙΓΡΑΦΗ		
Σκοπός της συνάρτησης είναι η ανάθεση μίας εργασίας στον καλύτερο διαθέσιμο υποψήφιο εργαζόμενο. Αρχικά, η συνάρτηση ανακτά τα απαραίτητα στοιχεία αναφορικά με την συγκεκριμένη εργασία και ταξινομεί άλλους τους διαθέσιμους εργαζόμενους σύμφωνα με τον δείκτη ανάθεσης εργασίας όπως αυτός περιγράφεται στο κεφαλαίο 5 στην ενότητα "Οργάνωση Εργασιών". Στην συνέχεια ελέγχει εάν ο καλύτερος υποψήφιος μπορεί να αναλάβει την εργασία, σύμφωνα με τον φόρτο εργασίας του, διαφορετικά επιλέγεται μία από τις εναλλακτικές στρατηγικές όπως περιγράφονται στο κεφάλαιο 5 στην ενότητα "Διαχείριση της Προτεραιότητας των Εργασιών".		
ΣΥΝΑΡΤΗΣΗ	**ΠΑΡΑΜΕΤΡΟΙ**	**ΒΟΗΘΗΤΙΚΕΣ ΣΥΝΑΡΤΗΣΕΙΣ (*)**
PrioritizeTasks	tel:TEL tcl:TaskConsequencesList tsl:TSL	-TraverseAllTSL -FindTaskConsequences - TaskExtra_LT_Function
ΠΕΡΙΓΡΑΦΗ		
Σκοπός της συνάρτησης είναι η προτεραιοποίηση των εργασιών με βάση την σπουδαιότητά τους. Αρχικά, η συνάρτηση ανακτά όλες τις εργασίες οι οποίες έχουν status «Started» και στην συνέχεια ανακτώνται οι απαραίτητες πληροφορίες για αυτές τις εργασίες. Έπειτα, οι εργασίες ταξινομούνται σύμφωνα με τον δείκτη προτεραιότητας της κάθε εργασίας όπως αυτός περιγράφεται στο κεφάλαιο 5 στην ενότητα "Διαχείριση της Προτεραιότητας των Εργασιών". Τέλος, επιλέγεται η εργασία με τον υψηλότερο δείκτη προτεραιότητας.		
ΣΥΝΑΡΤΗΣΗ	**ΠΑΡΑΜΕΤΡΟΙ**	**ΒΟΗΘΗΤΙΚΕΣ ΣΥΝΑΡΤΗΣΕΙΣ (*)**
SearchMCL_ForTask	task:Task mcl:MCL	N/A
ΠΕΡΙΓΡΑΦΗ		
Αναζήτηση σε μία λίστα τύπου ManCharList της εγγραφής που αντιστοιχεί σε μία συγκεκριμένη εργασία.		
ΣΥΝΑΡΤΗΣΗ	**ΠΑΡΑΜΕΤΡΟΙ**	**ΒΟΗΘΗΤΙΚΕΣ ΣΥΝΑΡΤΗΣΕΙΣ (*)**
SearchMSL_ForPerson	person:Person msl:MSL	N/A

ΠΕΡΙΓΡΑΦΗ		
Αναζήτηση σε μία λίστα τύπου ManStateList της εγγραφής που αντιστοιχεί σε έναν συγκεκριμένο εργαζόμενο.		
ΣΥΝΑΡΤΗΣΗ	**ΠΑΡΑΜΕΤΡΟΙ**	**ΒΟΗΘΗΤΙΚΕΣ ΣΥΝΑΡΤΗΣΕΙΣ (*)**
SearchTCL_ForTask	task:Task tcl:TCL	N/A
ΠΕΡΙΓΡΑΦΗ		
Αναζήτηση σε μία λίστα τύπου TaskCharList της εγγραφής που αντιστοιχεί σε μία συγκεκριμένη εργασία.		
ΣΥΝΑΡΤΗΣΗ	**ΠΑΡΑΜΕΤΡΟΙ**	**ΒΟΗΘΗΤΙΚΕΣ ΣΥΝΑΡΤΗΣΕΙΣ (*)**
SearchTSL_ForTask	task:Task tsl:TSL	N/A
ΠΕΡΙΓΡΑΦΗ		
Αναζήτηση σε μία λίστα τύπου TaskStateList της εγγραφής που αντιστοιχεί σε μία συγκεκριμένη εργασία.		
ΣΥΝΑΡΤΗΣΗ	**ΠΑΡΑΜΕΤΡΟΙ**	**ΒΟΗΘΗΤΙΚΕΣ ΣΥΝΑΡΤΗΣΕΙΣ (*)**
SearchTSL_ForStatus	status:State tsl:TSL returnedTSL:TSL	N/A
ΠΕΡΙΓΡΑΦΗ		
Αναζήτηση σε μία λίστα τύπου TaskStateList των εγγραφών που αντιστοιχούν σε ένα συγκεκριμένο status και επιστροφή μίας λίστας που περιλαμβάνει τις εγγραφές αυτές.		
ΣΥΝΑΡΤΗΣΗ	**ΠΑΡΑΜΕΤΡΟΙ**	**ΒΟΗΘΗΤΙΚΕΣ ΣΥΝΑΡΤΗΣΕΙΣ (*)**
SearchTEL_ForTask	task:Task tel:TEL	N/A
ΠΕΡΙΓΡΑΦΗ		
Αναζήτηση σε μία λίστα τύπου TaskExtraList της εγγραφής που αντιστοιχεί σε μία συγκεκριμένη εργασία.		
ΣΥΝΑΡΤΗΣΗ	**ΠΑΡΑΜΕΤΡΟΙ**	**ΒΟΗΘΗΤΙΚΕΣ ΣΥΝΑΡΤΗΣΕΙΣ (*)**
SearchSTL_ForGoal	goal:Goal stl:STL	N/A
ΠΕΡΙΓΡΑΦΗ		
Αναζήτηση σε μία λίστα τύπου SuspendedTasksList της εγγραφής που αντιστοιχεί σε ένα συγκεκριμένο στόχο.		
ΣΥΝΑΡΤΗΣΗ	**ΠΑΡΑΜΕΤΡΟΙ**	**ΒΟΗΘΗΤΙΚΕΣ ΣΥΝΑΡΤΗΣΕΙΣ (*)**
UpdateFinishedManStateList	um_out:UM_OUT msl:MSL	N/A

	tcl:TCL tsl:TSL	
ΠΕΡΙΓΡΑΦΗ		
Ενημέρωση μίας λίστας τύπου ManStateList της εγγραφής που αντιστοιχεί σε ένα συγκεκριμένο εργαζόμενο ο οποίος ολοκλήρωσε την εκτέλεση μίας εργασίας. Ουσιαστικά ενημερώνει τον φόρτο εργασίας του εργαζόμενου.		
ΣΥΝΑΡΤΗΣΗ	**ΠΑΡΑΜΕΤΡΟΙ**	**ΒΟΗΘΗΤΙΚΕΣ ΣΥΝΑΡΤΗΣΕΙΣ (*)**
UpdateAssignedManStateList	msl:MSL tcl:TCL at:AT	N/A
ΠΕΡΙΓΡΑΦΗ		
Ενημέρωση μίας λίστας τύπου ManStateList της εγγραφής που αντιστοιχεί σε ένα συγκεκριμένο εργαζόμενο ο οποίος ολοκλήρωσε την εκτέλεση μίας εργασίας. Ουσιαστικά ενημερώνει τον χρόνο στον οποίο θα είναι ο εργαζόμενος διαθέσιμος και τον φόρτο εργασίας του εργαζόμενου.		
ΣΥΝΑΡΤΗΣΗ	**ΠΑΡΑΜΕΤΡΟΙ**	**ΒΟΗΘΗΤΙΚΕΣ ΣΥΝΑΡΤΗΣΕΙΣ (*)**
UpdateTaskStateListFor AssignedPerson	at:AT tsl:TSL	N/A
ΠΕΡΙΓΡΑΦΗ		
Ενημέρωση μίας λίστας τύπου TaskStateList της εγγραφής που αντιστοιχεί σε μία συγκεκριμένη εργασία, η οποία ανατέθηκε σε κάποιον εργαζόμενο, σχετικά με την ανάθεση αυτή.		
ΣΥΝΑΡΤΗΣΗ	**ΠΑΡΑΜΕΤΡΟΙ**	**ΒΟΗΘΗΤΙΚΕΣ ΣΥΝΑΡΤΗΣΕΙΣ (*)**
UpdateTaskStateListWithState	task:Task tsl:TSL state:State	N/A
ΠΕΡΙΓΡΑΦΗ		
Ενημέρωση μίας λίστας τύπου TaskStateList της εγγραφής που αντιστοιχεί σε μία συγκεκριμένη εργασία, της οποίας η κατάσταση μεταβλήθηκε, σχετικά με την μεταβολή αυτή.		
ΣΥΝΑΡΤΗΣΗ	**ΠΑΡΑΜΕΤΡΟΙ**	**ΒΟΗΘΗΤΙΚΕΣ ΣΥΝΑΡΤΗΣΕΙΣ (*)**
UpdateTaskStateList	um_out:UM_OUT tsl:TSL	-SetGoalInactiveIn -TaskStateList -SetGoalInactiveIn -TaskStateList_Task
ΠΕΡΙΓΡΑΦΗ		
Ενημέρωση μίας λίστας τύπου TaskStateList της εγγραφής που αντιστοιχεί σε μία συγκεκριμένη εργασία ή των εγγραφών που αντιστοιχούν στον στόχο της εργασίας, της οποίας η κατάσταση μεταβλήθηκε, σχετικά με την μεταβολή αυτή. Πιο συγκεκριμένα, σε περίπτωση που μία εργασία αποτύχει για οποιοδήποτε λόγο, οι εγγραφές που αντιστοιχούν στον στόχο της εργασίας τίθενται ίσες με "Down", ενώ σε περίπτωση που η εργασία έχει ολοκληρωθεί με επιτυχία τότε ενημερώνεται η εγγραφή που αντιστοιχεί στην εργασία αυτή		

για την ολοκλήρωσή της.		
ΣΥΝΑΡΤΗΣΗ	**ΠΑΡΑΜΕΤΡΟΙ**	**ΒΟΗΘΗΤΙΚΕΣ ΣΥΝΑΡΤΗΣΕΙΣ (*)**
UpdateSuspendedTasksMemory	um_out:UM_OUT stl:STL	N/A
ΠΕΡΙΓΡΑΦΗ		
Προσθήκη σε μία λίστα τύπου SuspendedTasksList μίας εγγραφής που αντιστοιχεί σε μία εργασία της οποίας η λειτουργία έχει διακοπεί ή αναβληθεί.		
ΣΥΝΑΡΤΗΣΗ	**ΠΑΡΑΜΕΤΡΟΙ**	**ΒΟΗΘΗΤΙΚΕΣ ΣΥΝΑΡΤΗΣΕΙΣ (*)**
UpdateTaskStateList_Flag_Up	task:Task tsl:TSL	-SetGoalActiveIn TaskStateList -SetGoalActiveIn TaskStateList_Task -CheckIfTaskDown
ΠΕΡΙΓΡΑΦΗ		
Ενημέρωση μίας λίστας τύπου TaskStateList των εγγραφών που αντιστοιχούν στον στόχο μίας συγκεκριμένης εργασίας, της οποίας η λειτουργία ξεκίνησε, σχετικά με την μεταβολή αυτή (θέτοντας όλες τις εγγραφές του στόχου ως «Up»).		
ΣΥΝΑΡΤΗΣΗ	**ΠΑΡΑΜΕΤΡΟΙ**	**ΒΟΗΘΗΤΙΚΕΣ ΣΥΝΑΡΤΗΣΕΙΣ (*)**
UpdateTaskStateListForInitiation	ts:TaskState tsl:TSL	N/A
ΠΕΡΙΓΡΑΦΗ		
Ενημέρωση μίας λίστας τύπου TaskStateList του χρόνου έναρξης και λήξης της εγγραφής που αντιστοιχεί σε μία συγκεκριμένη εργασία.		
ΣΥΝΑΡΤΗΣΗ	**ΠΑΡΑΜΕΤΡΟΙ**	**ΒΟΗΘΗΤΙΚΕΣ ΣΥΝΑΡΤΗΣΕΙΣ (*)**
SuspendedTasksContainGoal	goal:Goal stl:STL	N/A
ΠΕΡΙΓΡΑΦΗ		
Η συνάρτηση επιστρέφει εάν μία λίστα τύπου SuspendedTasksList περιέχει εργασίες από έναν συγκεκριμένο στόχο.		
ΣΥΝΑΡΤΗΣΗ	**ΠΑΡΑΜΕΤΡΟΙ**	**ΒΟΗΘΗΤΙΚΕΣ ΣΥΝΑΡΤΗΣΕΙΣ (*)**
ProduceUserModelOut	at:AT	N/A
ΠΕΡΙΓΡΑΦΗ		
Η συνάρτηση παράγει το αποτέλεσμα της εκτέλεσης μίας εργασίας. Μία εργασία έχει 10% πιθανότητα να διακοπεί, 10% πιθανότητα να αναβληθεί, 10% πιθανότητα να αποτύχει, 10% πιθανότητα να ανατραπεί και 60% πιθανότητα να εκτελεστεί με επιτυχία.		
ΣΥΝΑΡΤΗΣΗ	**ΠΑΡΑΜΕΤΡΟΙ**	**ΒΟΗΘΗΤΙΚΕΣ**

		ΣΥΝΑΡΤΗΣΕΙΣ (*)
IsGoalFailedOrUpset	goal:Goal tsl:TSL	-FilterTSL_ForGoal
ΠΕΡΙΓΡΑΦΗ		
Η συνάρτηση επιστρέφει εάν μία λίστα τύπου TaskStateList περιέχει εργασίες οι οποίες είτε έχουν αποτύχει είτε έχουν ανατραπεί για έναν συγκεκριμένο στόχο.		
ΣΥΝΑΡΤΗΣΗ	**ΠΑΡΑΜΕΤΡΟΙ**	**ΒΟΗΘΗΤΙΚΕΣ ΣΥΝΑΡΤΗΣΕΙΣ (*)**
IsGoalFatallyTerminated	goal:Goal tsl:TSL	FilterTSL_ForGoal
ΠΕΡΙΓΡΑΦΗ		
Η συνάρτηση επιστρέφει εάν μία λίστα τύπου TaskStateList περιέχει εργασίες οι οποίες να έχουν αποτύχει για οποιοδήποτε λόγο πάνω από τρεις φορές για έναν συγκεκριμένο στόχο.		
ΣΥΝΑΡΤΗΣΗ	**ΠΑΡΑΜΕΤΡΟΙ**	**ΒΟΗΘΗΤΙΚΕΣ ΣΥΝΑΡΤΗΣΕΙΣ (*)**
IsGoalCompleted	goal:Goal tsl:TSL	FilterTSL_ForGoal
ΠΕΡΙΓΡΑΦΗ		
Η συνάρτηση ελέγχει εάν όλες οι εργασίες ενός συγκεκριμένου στόχου, που περιέχονται σε μία λίστα τύπου TaskStateList, έχουν ολοκληρωθεί με επιτυχία.		
ΣΥΝΑΡΤΗΣΗ	**ΠΑΡΑΜΕΤΡΟΙ**	**ΒΟΗΘΗΤΙΚΕΣ ΣΥΝΑΡΤΗΣΕΙΣ (*)**
InitiateGoal	tsl:TSL goalsInProcess:Goals initiatedGoals:Goals	N/A
ΠΕΡΙΓΡΑΦΗ		
Η συνάρτηση αναλαμβάνει να παράξει τις νέες εργασίες που θα διοχετευτούν στο δίκτυο.		
(*) = συναρτήσεις οι οποίες χρησιμοποιούνται αποκλειστικά από τις βασικές συναρτήσεις.		

Πίνακας 12: Οι Συναρτήσεις που Χρησιμοποιούνται στο Δίκτυο

Ανακεφαλαίωση

Στο κεφάλαιο πραγματοποιήθηκε αναφορά στις βασικές έννοιες των έγχρωμων δικτύων Πέτρι και στην συνέχεια περιγραφή της υλοποίησης του προτεινόμενου πλαισίου χρησιμοποιώντας τα έγχρωμα δίκτυα Πέτρι και το λογισμικό πακέτο CPN Tools. Τα υποσυστήματα περιγράφηκαν λεπτομερώς τόσο με εικόνες όσο και με πίνακες και κείμενο. Τα υποσυστήματα που παρουσιάστηκαν είναι τα ακόλουθα: (1) το θεμελιώδες επίπεδο του δικτύου, (2) το μοντέλο εργασιών, (3) το πρότυπο εργασιών, (4) ο εκτελεστικός

μηχανισμός, (5) το μοντέλο των νοητικών διαδικασιών των εργαζομένων, και (6) το υποσύστημα «Update Global Communication Places». Επιπλέον παρουσιάστηκαν λεπτομερώς οι τύποι δεδομένων και οι συναρτήσεις που χρησιμοποιούνται στο δίκτυο.

Κεφάλαιο 7: Ανάλυση Ιδιοτήτων Δικτύων Πέτρι

Εισαγωγή

Το κεφάλαιο παρουσιάζει την ανάλυση ορισμένων από τις βασικές ιδιότητες [48] των δικτύων Πέτρι και πιο συγκεκριμένα των ιδιοτήτων της περιοριστικότητας (boundedness), της ζωτικότητας (liveness), της αντιστρεψιμότητας (reversibility) καθώς και της ύπαρξης home σημάνσεων ή μεταβάσεων που να μπορούν δυνητικά να εκτελεστούν άπειρες φορές για το σύστημα που υλοποιήθηκε. Ο σκοπός των αναλύσεων αυτών είναι η επαλήθευση της δομής και της συμπεριφοράς του δικτύου εργασιών. Αξίζει να σημειωθεί πως τα συμπεράσματα σχετικά με τις ιδιότητες αυτές δεν επαληθεύονται πλήρως από την χρήση του εργαλείου State Space Analysis της πλατφόρμας CPN Tools. Για τον λόγο αυτό διερευνήθηκε η αιτία ή οι αιτίες στις οποίες οφείλεται αυτή η ασυνέπεια και καταλήξαμε στο συμπέρασμα πως η πιθανότερη αιτία της απόκλισης των αποτελεσμάτων μπορεί να θεωρηθεί η αδυναμία υπολογισμού του συνόλου των δυνατών καταστάσεων στις οποίες μπορεί να περιέλθει το δίκτυο καθώς φαίνεται πως το δίκτυο είναι μεγαλύτερο σε όγκο από τον όγκο που μπορεί να διαχειριστεί η πλατφόρμα CPN Tools.

Ανάλυση Χώρου Καταστάσεων

Η μαθηματική θεμελίωση των δικτύων Πέτρι παρέχει την βάση για τη χρήση ποικίλων τεχνικών ανάλυσης οι οποίες είναι διαθέσιμες στην πλατφόρμα CPN Tools με σκοπό την επαλήθευση των δυναμικών ιδιοτήτων του συστήματος. Δίνεται η δυνατότητα να εξεταστεί, για παράδειγμα, αν το δίκτυο εργασιών

περιέχει αδιέξοδα, εργασίες που δεν τελειώνουν ποτέ, εργασίες που δεν συνεισφέρουν πουθενά, και εργασίες που ενεργοποιούνται ακούσια με αποτέλεσμα την έλλειψη συγχρονισμού. Αυτές οι ιδιότητες μπορούν να διερευνηθούν με επίσημες τεχνικές ανάλυσης, όπως είναι, για παράδειγμα, οι γράφοι προσβασιμότητας.

Κάθε κατάσταση ενός μοντέλου έγχρωμου δικτύου Πέτρι αντιστοιχεί σε μια λίστα με όλες τις θέσεις του δικτύου και μία κατανομή των μαρκών στις θέσεις αυτές. Ένας χώρος καταστάσεων αποτελείται από ένα μεγάλο αριθμό καταστάσεων ο οποίος μπορεί να αναζητηθεί και να αναλυθεί με τυπικές τεχνικές ανάλυσης. Ο χώρος των καταστάσεων του δικτύου, ο οποίος υπολογίζεται αυτόματα από την πλατφόρμα CPN Tools και λειτούργησε βοηθητικά στην ανάλυση των ιδιοτήτων του υλοποιημένου δικτύου, είναι επιπλέον ένας κατευθυνόμενος γράφος ο οποίος αποτελείται από κόμβους οι οποίοι αντιστοιχούν σε συγκεκριμένες καταστάσεις (markings) του δικτύου και οι οποίοι συνδέονται με τόξα τα οποία φανερώνουν την μετάβαση από την μία κατάσταση στην επόμενη.

Ο χώρος των καταστάσεων του δικτύου ο οποίος υπολογίστηκε αποτελούνταν από 240.558 κόμβους οι οποίοι αντιστοιχούν σε στιγμιότυπα του δικτύου και 290.742 τόξα (arcs) τα οποία αντιστοιχούν στις μεταβάσεις από την μία κατάσταση στην άλλη. Ο χώρος των καταστάσεων ήταν ημιτελής και ο υπολογισμός του διήρκησε 3 ώρες. Επιπλέον, το γεγονός ότι ο SCC (Strongly Connected Component) γράφος που υπολογίζεται από την πλατφόρμα έχει τον ίδιο αριθμό κόμβων και ακμών φανερώνει πως το υλοποιημένο δίκτυο δεν έχει κυκλικές διαδρομές (loops). Ο SCC γράφος είναι ο μέγιστος υπο-γράφος του χώρου των καταστάσεων του δικτύου στον οποίο όλοι οι κόμβοι είναι προσβάσιμοι μεταξύ τους. Οι κόμβοι ενός SCC γράφου είναι αμοιβαία ξένοι μεταξύ τους.

Στην πρόσφατη βιβλιογραφία, υπάρχει ένας αυξανόμενος αριθμός από έρευνες οι οποίες διερευνούν την εφαρμογή της ανάλυσης του χώρου καταστάσεων

στον σχεδιασμό λειτουργιών χρησιμοποιώντας τα έγχρωμα δίκτυα Πέτρι [41][51]. Στην συνέχεια παρουσιάζονται τα συμπεράσματα σχετικά με τις ιδιότητες αυτές ύστερα από την διεξοδική ανάλυση του υλοποιημένου συστήματος.

Περιοριστικότητα (boundedness) Δικτύου

Ένα δίκτυο δεν είναι περιορισμένο όταν ο αριθμών των μαρκών σε τουλάχιστον μία από τις θέσεις που το αποτελούν αυξάνεται διαρκώς και ανεξέλεγκτα με την πάροδο του χρόνου. Το υλοποιημένο δίκτυο είναι περιορισμένο, καθώς ο αριθμός των μαρκών σε όλες τις θέσεις του συστήματος δεν υπερβαίνουν έναν πεπερασμένο μέγιστο και ελάχιστο αριθμό. Η ιδιότητα αυτή επιβεβαιώνεται και από την χρήση του εργαλείου State Space Analysis της πλατφόρμας CPN Tools.

Home Σήμανση

Μία home σήμανση είναι μία σήμανση η οποία είναι προσβάσιμη από οποιαδήποτε άλλα σήμανση του δικτύου. Το δίκτυο δεν περιλαμβάνει home σήμανση καθώς δεν υπάρχει τρόπος επιστροφής στην αρχική σήμανση M_0 επειδή το δίκτυο περιλαμβάνει πεπερασμένο πλήθος εκτελέσεων μέχρι να ολοκληρωθούν οι εργασίες οι οποίες έχουν διοχετευτεί στο δίκτυο και επιπλέον υπάρχουν θέσεις οι οποίες καταγράφουν λεπτομέρειες για την λειτουργία του δικτύου και κατά συνέπεια στο τέλος της λειτουργίας του δικτύου αυτές οι θέσεις θα έχουν διαφορετικές τιμές από αυτές που είχαν στην έναρξη λειτουργίας του δικτύου.

Ιδιότητες Ζωτικότητας (liveness)

Μία σήμανση (marking) M ενός δικτύου είναι νεκρή (dead) στην περίπτωση κατά την οποία δεν περιλαμβάνει καμία ενεργοποιημένη μετάβαση και κατά

συνέπεια το δίκτυο αδυνατεί να λειτουργήσει περαιτέρω. Το υλοποιημένο δίκτυο περιλαμβάνει νεκρές σημάνσεις οι οποίες αντιστοιχούν σε όλες τις πιθανές καταληκτικές καταστάσεις του δικτύου με δεδομένο πως η λειτουργία του δικτύου λήγει μετά την εξυπηρέτηση όλων των εργασιών που διοχετεύτηκαν στο δίκτυο. Κατά συνέπεια, το πλήθος των νεκρών σημάνσεων του υλοποιημένου δικτύου είναι γνωστό και πεπερασμένο.

Μία μετάβαση t ενός δικτύου είναι νεκρή όταν δεν μπορεί να ενεργοποιηθεί ποτέ από τις διαθέσιμες σημάνσεις του δικτύου. Το υλοποιημένο δίκτυο δεν περιλαμβάνει νεκρές μεταβάσεις επειδή όλες οι μεταβάσεις του ενεργοποιούνται τουλάχιστον μία φορά από το σύνολο των διαθέσιμων σημάνσεων του.

Σύμφωνα με τον ορισμό των ζωτικών μεταβάσεων «μία μετάβαση t ενός δικτύου Πέτρι λέγεται ζωτική εάν μπορεί να ενεργοποιηθεί από οποιαδήποτε σήμανση M, η οποία είναι προσεγγίσιμη από την αρχική σήμανση M_0». Καταλήξαμε στο συμπέρασμα πως το υλοποιημένο δίκτυο δεν είναι ζωτικό επειδή αφενός έχει πεπερασμένο πλήθος βημάτων μέχρι να ολοκληρωθούν οι εργασίες οι οποίες έχουν διοχετευτεί στο δίκτυο, και επιπλέον δεν μπορεί να επιστρέψει στην αρχική του σήμανση M_0, επιτρέποντας την πρόσβαση όλων των μεταβάσεων από όλες τις δυνατές σημάνσεις, αφού στο τέλος λειτουργίας του δικτύου υπάρχουν θέσεις οι οποίες καταγράφουν λεπτομέρειες για την λειτουργία του δικτύου και κατά συνέπεια στο τέλος της λειτουργίας του δικτύου αυτές οι θέσεις θα έχουν διαφορετικές τιμές από αυτές που είχαν στην έναρξη λειτουργίας του δικτύου. Η ιδιότητα αυτή επαληθεύεται και από την χρήση της λειτουργίας State Space Analysis της πλατφόρμας CPN Tools.

Αντιστρεψιμότητα (reversibility)

Το υλοποιημένο δίκτυο δεν είναι αντιστρέψιμο επειδή περιλαμβάνει θέσεις καταγραφής λειτουργίας του συστήματος και κατά συνέπεια δεν μπορεί να

επιστρέψει στην αρχική του σήμανση M_0 αφού οι εν λόγω θέσεις μεταβάλλονται με το πέρασμα του χρόνου.

Άπειρες μεταβάσεις (fairness)

Στο δίκτυο δεν υπάρχουν μεταβάσεις οι οποίες δυνητικά να μπορούν να εκτελεστούν άπειρες φορές καθώς το δίκτυο περιλαμβάνει πεπερασμένο πλήθος εκτελέσεων μέχρι να ολοκληρωθούν οι εργασίες οι οποίες έχουν διοχετευτεί στο δίκτυο αδυνατώντας να επιστρέψει με άπειρο τρόπο σε προηγούμενες μεταβάσεις. Η ιδιότητα αυτή επαληθεύεται και από την χρήση της λειτουργίας State Space Analysis της πλατφόρμας CPN Tools.

Ανακεφαλαίωση

Στο κεφάλαιο παρουσιάστηκαν ορισμένες από τις βασικές ιδιότητες των δικτύων Πέτρι και πιο συγκεκριμένα των ιδιοτήτων της περιοριστικότητας (boundedness), της ζωτικότητας (liveness), της αντιστρεψιμότητας (reversibility) καθώς και της ύπαρξης home σημάνσεων ή μεταβάσεων που να μπορούν να εκτελεστούν άπειρες φορές για το σύστημα που υλοποιήθηκε. Επιπλέον έγινε λεπτομερής περιγραφή της τεχνικής της ανάλυσης του χώρου των καταστάσεων του δικτύου. Αξίζει να σημειωθεί πως τα συμπεράσματα σχετικά με τις ιδιότητες αυτές δεν επαληθεύονται πλήρως από την χρήση του εργαλείου State Space Analysis της πλατφόρμας CPN Tools. Οι αιτίες της διαφοροποίησης αυτής εξετάστηκαν και παρουσιάστηκαν στο κεφάλαιο.

Κεφάλαιο 8: Προσομοίωση

Εισαγωγή

Το κεφάλαιο περιλαμβάνει την περιγραφή μίας σειράς από δοκιμαστικές προσομοιώσεις του υλοποιημένου δικτύου πάνω στην πλατφόρμα CPN Tools ώστε να επαληθευτούν και να παρουσιαστούν οι δυνατότητες ανάλυσης του εργαλείου στα πλαίσια της μοντελοποίησης των νοητικών διαδικασιών και των ανθρωπίνων λαθών για τον σχεδιασμό και την ασφαλή διαχείριση ροών εργασίας.

Το εργαλείο προσομοίωσης μπορεί να μοντελοποιήσει την απόδοση των εργαζομένων και να εξετάσει την αλληλεπίδραση μεταξύ των εργασιακών απαιτήσεων, των στρατηγικών διαχείρισης φόρτου εργασίας και των σχημάτων οργάνωσης των εργασιών. Η απόδοση του συστήματος και η ανθρώπινη αξιοπιστία μπορεί να βελτιωθεί με την τροποποίηση αυτών των τριών παραγόντων. Για παράδειγμα, οι εργασιακές απαιτήσεις μπορούν να μειωθούν προσαρμόζοντας την οργάνωση των εργασιών στο μοντέλο εργασιών, με τον επανασχεδιασμό των εργασιών (δηλαδή, της διάρκειας των εργασιών, και των προτεραιοτήτων των εργασιών) και με την αφαίρεση ορισμένων συνεπειών από ενδεχόμενες αποτυχίες των εργασιών. Από την άλλη πλευρά, οι εργαζόμενοι μπορούν να καταφύγουν σε διαφορετικές στρατηγικές για τη μείωση του φόρτου εργασίας τους και την ενίσχυση της αξιοπιστίας τους. Τέλος, τα σχήματα οργάνωσης των εργασιών μπορούν να βελτιστοποιηθούν με την αλλαγή των κανόνων κατανομής των εργασιών, του εύρους των εργασιών και

του βαθμού εμπειρίας των εργαζομένων. Αξίζει επίσης να σημειωθεί η αλληλεπίδραση μεταξύ αυτών των παραγόντων. Για παράδειγμα, το εύρος των βιώσιμων στρατηγικών φόρτου εργασίας μπορεί να εξαρτάται από τις εργασιακές απαιτήσεις (π.χ., μπορεί να μην είναι δυνατόν να αναβληθούν εργασίες ή να εκτελεστούν εργασίες με τρόπο σειριακό) και από την εμπειρία των εργαζομένων (π.χ., τα υψηλότερα επίπεδα εμπειρίας μπορεί να μειώσουν τον φόρτο εργασίας ή να επηρεάσουν την επιλογή των στρατηγικών φόρτου εργασίας).

Το εργαλείο προσομοίωσης παράγει αρκετές συστεμικές και ψυχολογικές μετρικές για την αξιολόγηση των επιπτώσεων από τις αλλαγές στις εργασιακές απαιτήσεις, στις στρατηγικές του φόρτου εργασίας και της οργάνωσης των εργασιών. Οι συστεμικές μετρικές μπορεί να περιλαμβάνουν την ακρίβεια και τον χρόνο που απαιτείται για να εκτελεστεί μία ακολουθία εργασιών, την κατάσταση των εργασιών σε κάθε χρονικό σημείο (π.χ., εργασίες που έχουν διακοπεί ή αναβληθεί), την ανάθεση των εργασιών σε εργαζόμενους και τις χρονικές περιόδους κατά τις οποίες οι προτιμώμενοι εργαζόμενοι δεν ήταν διαθέσιμοι. Οι ψυχολογικές μετρικές μπορεί να περιλαμβάνουν δείκτες φόρτου εργασίας, κόπωσης και νοητικής φόρτισης.

Στα πλαίσια της παρούσας διατριβής διεξήχθησαν δύο σενάρια. Στο πρώτο σενάριο οι απαιτήσεις των εργασιών δεν ήταν μεγάλες με αποτέλεσμα ένας εργαζόμενος να είναι σε θέση να ασχοληθεί με παραπάνω από μία εργασίες ταυτόχρονα, ενώ στο δεύτερο σενάριο οι εργασίες κρίνονται ιδιαίτερα απαιτητικές με αποτέλεσμα να είναι δύσκολη η ενασχόληση ενός εργαζομένου με παραπάνω από μία εργασίες. Κάθε σενάριο εκτελέστηκε τρεις φορές και τα αποτελέσματα των προσομοιώσεων παρουσιάζονται στη συνέχεια.

Αρχική Κατάσταση Δικτύου

Στο υποθετικό δίκτυο υπάρχουν έξι εργαζόμενοι: ένας προϊστάμενος, δύο βοηθοί, και τρεις εργάτες. Στον παρακάτω πίνακα παρουσιάζεται για κάθε

εργαζόμενο η αρχική του κατάσταση και πιο συγκεκριμένα το χρονικό διάστημα στο οποίο θα είναι διαθέσιμος και ο φόρτος εργασίας του κάθε εργαζομένου.

Person	Available At	Current Workload
Supervisor	0	0
Assistant_A	0	0
Assistant_B	0	0
Worker_A	0	0
Worker_B	0	0
Worker_C	0	0

Πίνακας 13: Αρχική Κατάσταση Εργαζομένων

Κάθε εργαζόμενος έχει ορισμένες συγκεκριμένες δυνατότητες αναφορικά με την ακρίβεια εκτέλεσης κάθε εργασίας και τον χρόνο που δαπανά για την εκτέλεσή της. Στους παρακάτω δύο πίνακες παρουσιάζονται για κάθε εργαζόμενο τα χαρακτηριστικά αυτά τα οποία είναι σταθερά (δεν μεταβάλλονται με τον χρόνο), δηλαδή για κάθε εργασία η ακρίβεια και η διάρκεια εκτέλεσης της από τον κάθε εργαζόμενο.

Person	Task 1 Accuracy	Task 1 Duration	Task 6 Accuracy	Task 6 Duration	Task 7 Accuracy	Task 7 Duration
Supervisor	45	50	80	20	70	10
Assistant_A	70	30	70	30	60	20
Assistant_B	65	35	65	35	55	25
Worker_A	80	20	50	40	50	40
Worker_B	50	40	45	50	45	50
Worker_C	40	60	40	60	40	60

Πίνακας 14: Χαρακτηριστικά Εργαζομένων Α'

Person	Task 11 Accuracy	Task 11 Duration	Task 12 Accuracy	Task 12 Duration	Task 15 Accuracy	Task 15 Duration
Supervisor	70	10	90	30	90	30
Assistant_A	60	20	80	40	80	40
Assistant_B	55	25	75	45	75	45
Worker_A	50	40	50	40	50	40
Worker_B	45	50	45	50	45	50
Worker_C	40	60	40	60	40	60

Πίνακας 15: Χαρακτηριστικά Εργαζομένων Β'

Κάθε εργασία κατά την διάρκεια λειτουργίας της διατηρεί ορισμένες χαρακτηριστικές ιδιότητες της κατάστασής της οι οποίες είτε παραμένουν αμετάβλητες είτε μεταβάλλονται δυναμικά με τον χρόνο. Οι ιδιότητες αυτές επιγραμματικά είναι ο στόχος στον οποίο ανήκει η εργασία, ο εργαζόμενος ο οποίος έχει αναλάβει την εκτέλεσή της, η κατάσταση λειτουργίας της, ο χρόνος έναρξής της, ο χρόνος λήξης της, μία σημαία που εκφράζει αν η εργασία είναι ενεργή ή όχι, και ένας μετρητής ο οποίος αυξάνεται κατά μία μονάδα κάθε φορά που η εργασία αποτυγχάνει. Η αρχική κατάσταση των εργασιών του δικτύου φαίνεται στον παρακάτω πίνακα:

Task	Goal	Person	Status	Start Time	End Time	Flag	Errors Counter
1	1	None	Default	0	0	Down	0
6	1	None	Default	0	0	Down	0
7	2	None	Default	0	0	Down	0
11	2	None	Default	0	0	Down	0
12	3	None	Default	0	0	Down	0
15	3	None	Default	0	0	Down	0

Πίνακας 16: Χαρακτηριστικά Γνωρίσματα της Κατάστασης κάθε Εργασίας

Επιπλέον κάθε εργασία χαρακτηρίζεται από το πόσο επιβαρύνει τον φόρτο εργασίας κάθε εργαζομένου σε επίπεδο αντίληψης, σε νοητικό επίπεδο και σε μηχανικό επίπεδο. Για το πρώτο σενάριο, οι εργασίες χαρακτηρίζονται ως μη

απαιτητικές το οποίο αλλάζει στο δεύτερο σενάριο στο οποίο οι εργασίες είναι ιδιαίτερα απαιτητικές. Η διαφοροποίηση πραγματοποιήθηκε για να μελετηθεί η προσαρμογή του υλοποιημένου δικτύου στις νέες συνθήκες. Πρέπει φυσικά να σημειωθεί πως θα μπορούσαν να τροποποιηθούν και άλλες μεταβλητές της αρχικής κατάστασης του δικτύου ωστόσο επιλέχθηκε να τροποποιηθούν μόνο τα συγκεκριμένα χαρακτηριστικά. Τα χαρακτηριστικά του πρώτου και του δεύτερου σεναρίου περιέχονται στους δύο παρακάτω πίνακες:

Task	Perception	Cognitive	Motor
1	10	10	10
6	5	15	10
7	15	5	10
11	4	6	10
12	8	7	10
15	10	5	15

Πίνακας 17: Σενάριο 1: Φόρτος κάθε εργαζομένου από τις εργασίες

Task	Perception	Cognitive	Motor
1	50	50	40
6	45	15	50
7	35	35	30
11	40	60	10
12	80	20	20
15	90	15	15

Πίνακας 18: Σενάριο 2: Φόρτος κάθε εργαζομένου από τις εργασίες

Κάθε εργασία διακρίνεται από ορισμένα επιπλέον χαρακτηριστικά τα οποία χρησιμοποιούνται στους αλγορίθμους κατά την διάρκεια λειτουργίας του δικτύου. Πιο συγκεκριμένα, είναι η σταθερά «επίπεδο προτεραιότητας» του αλγορίθμου (2) όπως περιγράφεται στο κεφάλαιο 5, η σταθερά «κόστος διακοπής» του αλγορίθμου (1) όπως περιγράφεται στο κεφάλαιο 5 και η

μεταβλητή «δείκτης προτεραιότητας» του αλγορίθμου (1) όπως περιγράφεται στο κεφάλαιο 5. Οι τιμές που λαμβάνουν τα χαρακτηριστικά αυτά φαίνονται στον παρακάτω πίνακα:

Task	Priority Level	Interruption Cost	Priority Index
1	50	5	0
6	60	10	0
7	40	15	0
11	20	20	0
12	70	25	0
15	110	30	0

Πίνακας 19: Σενάριο 1: Επιπλέον Χαρακτηριστικά Εργασιών

Τέλος στον αλγόριθμο (1) του κεφαλαίου 5 αναφέρεται πως για κάθε εργασία υπάρχουν ορισμένες συνέπειες οι οποίες συνοδεύουν την διακοπή της οι οποίες θα πρέπει να συνυπολογιστούν στον καθορισμό του δείκτη προτεραιότητας της εκάστοτε εργασίας. Αυτές οι συνέπειες και οι πιθανότητες εμφάνισής τους ορίστηκαν με τυχαίο τρόπο και παρουσιάζονται στον παρακάτω πίνακα:

Task	Probability Of Consequence (i)	Cost Of Consequence (i)
1	10	5
1	5	10
1	5	15
6	15	5
6	5	20
7	10	10
7	10	15
11	5	10
11	10	5
12	15	10
15	20	10

Πίνακας 20: Οι συνέπειες διακοπής των εργασιών και οι πιθανότητες εμφάνισής τους

Για παράδειγμα, η εργασία 1 έχει τρεις πιθανές συνέπειες από μία ενδεχόμενη διακοπή της, η καθεμία με την δική της πιθανότητα εμφάνισης και το αντίστοιχο κόστος, η εργασία 6 έχει δύο πιθανές συνέπειες από μία ενδεχόμενη διακοπή της κτλ.

Σενάριο 1: Εκτέλεση Προσομοίωσης

Για την ορθή αξιολόγηση του υλοποιημένου δικτύου πραγματοποιήθηκαν μία σειρά από δοκιμαστικές προσομοιώσεις με σκοπό τον έλεγχο της λειτουργίας και την αξιολόγηση της διαχείρισης των εργασιών κάτω από κανονικές και μη συνθήκες με ιδιαίτερη έμφαση στις μη κανονικές συνθήκες δηλαδή στις συνθήκες κατά τις οποίες οι εργασίες για οποιοδήποτε λόγο διακόπηκαν ή αναβλήθηκαν. Τελικά επιλέχθηκαν τρεις από αυτές τις προσομοιώσεις οι οποίες παρουσιάζουν ενδιαφέρον αναφορικά με τις δυνατότητες του δικτύου. Οι δοκιμές αυτές περιλήφθησαν υπό μορφή πινάκων για την καλύτερη κατανόηση της πορείας και της διαχείρισης των εργασιών. Κάθε προσομοίωση συνοδεύεται από κείμενο στο οποίο περιγράφονται τα γεγονότα που έλαβαν χώρα. Η λειτουργία του δικτύου επιλέχθηκε να ελέγχεται κάθε 40 μονάδες χρόνου. Ανάμεσα σε διαδοχικούς ελέγχους οι καταστάσεις που μεταβάλλονται από το ένα χρονικό διάστημα στο άλλο τονίζονται με «bold» γραμματοσειρά.

Πρέπει να υπενθυμιστεί πως ο στόχος 1 αποτελείται από τις εργασίες 1 και 6 οι οποίες εκτελούνται σειριακά, ο στόχος 2 αποτελείται από τις εργασίες 7 και 11 οι οποίες εκτελούνται και αυτές σειριακά, και ο στόχος 3 αποτελείται από τις εργασίες 12 και 15 οι οποίες και αυτές με την σειρά τους εκτελούνται σειριακά. Η σειριακή εκτέλεση όπως γίνεται κατανοητό προϋποθέτει την ολοκλήρωση της πρώτης εργασίας για την εκτέλεση της δεύτερης. Στην συνέχεια παρουσιάζονται οι δοκιμές του πρώτου σεναρίου:

Προσομοίωση 1

Step (Time)	0 (0)	40 (72)
Task|Goal|Person|State|Flag|Start|End|Errors	1|1|None|Default|Down|0|0|0	1|1|**Worker_A**|**Done**|**Up**|**24**|**71**|0
	6|1|None|Default|Down|0|0|0	6|1|None|Default|**Up**|0|0|0
	7|2|None|Default|Down|0|0|0	7|2|None|**Processing**|**Up**|**40**|0|0
	11|2|None|Default|Down|0|0|0	11|2|None|Default|**Up**|0|0|0
	12|3|None|Default|Down|0|0|0	12|3|None|**Processing**|**Up**|**41**|0|0
	15|3|None|Default|Down|0|0|0	15|3|None|Default|**Up**|0|0|0

Πίνακας 21: Προσομοίωση 1 (χρόνοι 0-72)

Στο πρώτο στιγμιότυπο της προσομοίωσης 1 (χρόνοι 0-72), παρατηρούμε πως η εργασία 1 έχει ολοκληρωθεί από τον εργαζόμενο «Worker_A». Η εργασία 1 ξεκίνησε σε χρόνο 24 και ολοκληρώθηκε σε χρόνο 71 χωρίς λάθη κατά την διάρκεια εκτέλεσής της. Επιπλέον έχει ξεκινήσει η επεξεργασία της εργασίας 7 σε χρόνο 40 χωρίς ωστόσο να έχει ανατεθεί σε κάποιον εργαζόμενο. Το ίδιο ισχύει και για την εργασία 12 της οποίας η επεξεργασία ξεκίνησε σε χρόνο 41 χωρίς ωστόσο να έχει ανατεθεί η εκτέλεσή της σε κάποιον εργαζόμενο. Οι εργασίες 6, 11 και 15 δεν έχουν στον χρόνο αυτό ξεκινήσει.

Step (Time)	80 (104)	120 (146)
Task|Goal|Person|State|Flag|Start|End|Errors	1|1|Worker_A|Done|Up|24|71|0	1|1|Worker_A|Done|**Down**|24|71|0
	6|1|None|**Processing**|Up|**77**|0|0	6|1|**Supervisor**|**Done**|**Down**|77|**111**|0
	7|2|**Assistant_A**|Processing|Up|40|**81**|**1**	7|2|Assistant_A|**Done**|Up|40|**131**|1
	11|2|None|Default|Up|0|0|**1**	11|2|None|**Processing**|Up|**137**|0|1
	12|3|**Assistant_A**|**Interrupted**|**Down**|41|**97**|**1**	12|3|Assistant_A|**Processing**|**Up**|41|97|1
	15|3|None|Default|**Down**|0|0|**1**	15|3|None|Default|**Up**|0|0|1

Πίνακας 22: Προσομοίωση 1 (χρόνοι 72-146)

Στο επόμενο στιγμιότυπο (χρόνοι 72-104) παρατηρούμε πως έχει ξεκινήσει η επεξεργασία της εργασίας 6 σε χρόνο 77 χωρίς ωστόσο να έχει ανατεθεί σε κάποιο εργαζόμενο. Η εργασία 7 ανατέθηκε στον εργαζόμενο «Assistant_A» ωστόσο η επεξεργασία της απέτυχε σε χρόνο 81 και στην συνέχεια επανεκκινήθηκε. Η εργασία 12 ανατέθηκε και αυτή στον εργαζόμενο «Assistant_A» ωστόσο διακόπηκε σε χρόνο 97 και μέχρι το χρονικό σημείο 104 δεν είχε επανεκκινηθεί. Οι εργασίες 11 και 15 παρέμειναν αμετάβλητες.

Στην συνέχεια (χρόνοι 104-146) η εργασία 6 ανατέθηκε στον εργαζόμενο «Supervisor» ο οποίος την ολοκλήρωσε σε χρόνο 111 ολοκληρώνοντας

παράλληλα και τον πρώτο στόχο. Η εργασία 7 ολοκληρώθηκε από τον εργαζόμενο «Assistant_A» σε χρόνο 131 και στην συνέχεια ξεκίνησε η επεξεργασία της εργασίας 11 σε χρόνο 137. Η εργασία 12 η οποία είχε διακοπεί επανεκκινήθηκε και ανατέθηκε εκ νέου στον εργαζόμενο «Assistant_A». Η εργασία 15 παρέμεινε αμετάβλητη.

Step (Time)	160 (230)	166 (231)
Task|Goal|Person|State|Flag|Start|End|Errors	1|1|Worker_A|Done|Down|24|71|0	1|1|Worker_A|Done|Down|24|71|0
	6|1|Supervisor|Done|Down|77|111|0	6|1|Supervisor|Done|Down|77|111|0
	7|2|Assistant_A|Done|**Down**|40|131|1	7|2|Assistant_A|Done|Down|40|131|1
	11|2|**Assistant_A**|**Done**|**Down**|137|**169**|1	11|2|Assistant_A|Done|Down|137|169|1
	12|3|Assistant_A|**Done**|**Down**|41|**167**|1	12|3|Assistant_A|Done|Down|41|167|1
	15|3|**Assistant_A**|**Done**|**Down**|**173**|**225**|1	15|3|Assistant_A|Done|Down|173|225|1

Πίνακας 23: Προσομοίωση 1 (χρόνοι 146-231)

Στο επόμενο χρονικό διάστημα (χρόνοι 146-230) η εργασία 11 ολοκληρώθηκε από τον εργαζόμενο «Assistant_A» σε χρόνο 169 ολοκληρώνοντας τον δεύτερο στόχο , ενώ τόσο η εργασία 12 όσο και η εργασία 15 ολοκληρώθηκαν από τον ίδιο εργαζόμενο σε χρόνους 167 και 225 αντίστοιχα. Να σημειωθεί πως η εργασία 15 ξεκίνησε σε χρόνο 173 μετά την ολοκλήρωση της εργασίας 12. Με τον τρόπο αυτό ολοκληρώθηκε και ο τρίτος στόχος φτάνοντας στην λήξη λειτουργίας του δικτύου σε χρόνο 231.

Προσομοίωση 2

Step (Time)	0 (0)	40 (102)
Task|Goal|Person|State|Flag|Start|End|Errors	1|1|None|Default|Down|0|0|0	1|1|**Worker_A**|**Processing**|**Up**|**69**|0|0
	6|1|None|Default|Down|0|0|0	6|1|None|Default|**Up**|0|0|0
	7|2|None|Default|Down|0|0|0	7|2|None|**Processing**|**Up**|**70**|0|0
	11|2|None|Default|Down|0|0|0	11|2|None|Default|**Up**|0|0|0
	12|3|None|Default|Down|0|0|0	12|3|None|**Processing**|**Up**|**68**|0|0
	15|3|None|Default|Down|0|0|0	15|3|None|Default|**Up**|0|0|0

Πίνακας 24: Προσομοίωση 2 (χρόνοι 0-102)

Στο πρώτο στιγμιότυπο της λειτουργίας του δικτύου (χρόνοι 0-102) παρατηρούμε πως ξεκίνησε η επεξεργασία της εργασίας 1 σε χρόνο 69 η οποία ανατέθηκε στον εργαζόμενο «Worker_A». Επιπλέον ξεκίνησε η επεξεργασία

των εργασιών 7 και 12 σε χρόνους 70 και 68 αντίστοιχα χωρίς ωστόσο να έχουν ανατεθεί σε κάποιον εργαζόμενο. Οι εργασίες 6, 11 και 15 παρέμειναν αμετάβλητες.

Step (Time)	80 (133)	120 (171)
Task|Goal|Person|State|Flag|Start|End|Errors	1|1|Worker_A|**Done**|Up|69|**105**|0	1|1|Worker_A|Done|**Down**|69|105|0
	6|1|None|**Processing**|Up|**114**|0|0	6|1|**Supervisor**|**Done**|**Down**|114|**148**|0
	7|2|**Assistant_A**|Processing|Up|70|**106**|**1**	7|2|Assistant_A|**Started**|Up|70|**153**|**2**
	11|2|None|Default|Up|0|0|**1**	11|2|None|Default|Up|0|0|**2**
	12|3|**Assistant_A**|**Done**|Up|68|**132**|0	12|3|Assistant_A|Done|Up|68|132|0
	15|3|None|Default|Up|0|0|0	15|3|None|**Processing**|Up|**138**|0|0

Πίνακας 25: Προσομοίωση 2 (χρόνοι 102-171)

Στο επόμενο στιγμιότυπο (χρόνοι 102-133) η εργασία 1 ολοκληρώθηκε σε χρόνο 105 από τον εργαζόμενο «Worker_A», και στην συνέχεια ξεκίνησε η επεξεργασία της εργασίας 6 σε χρόνο 114. Η εκτέλεση της εργασίας 7 ανατέθηκε στον εργαζόμενο «Assistant_A» ωστόσο παρουσίασε κάποιο σφάλμα σε χρόνο 106 ενώ σε χρόνο 133 ήταν πάλι υπό επεξεργασία. Η εργασία 12 ανατέθηκε στον εργαζόμενο «Assistant_A» ο οποίος την ολοκληρώθηκε με επιτυχία σε χρόνο 132. Οι εργασίες 11 και 15 παρέμειναν αμετάβλητες.

Στο επόμενο χρονικό σημείο (χρόνοι 133-171) η εργασία 6 ανατέθηκε στον εργαζόμενο «Supervisor» ο οποίος την ολοκλήρωσε με επιτυχία σε χρόνο 148 ολοκληρώνοντας έτσι τον πρώτο στόχο. Η εργασία 7 απέτυχε εκ νέου από τον εργαζόμενο «Assistant_A» σε χρόνο 153 ωστόσο σε χρόνο 171 είχε επανεκκινηθεί. Επιπλέον ξεκίνησε η επεξεργασία της εργασίας 15 σε χρόνο 138 ενώ η εργασία 11 παρέμεινε αμετάβλητη.

Step (Time)	160 (246)	169 (255)
Task|Goal|Person|State|Flag|Start|End|Errors	1|1|Worker_A|Done|Down|69|105|0	1|1|Worker_A|Done|Down|69|105|0
	6|1|Supervisor|Done|Down|114|148|0	6|1|Supervisor|Done|Down|114|148|0
	7|2|Assistant_A|**Processing**|Up|70|**200**|**3**	7|2|Assistant_A|**Deferred**|**Down**|70|**246**|**4**
	11|2|None|Default|Up|0|0|**3**	11|2|None|Default|**Down**|0|0|**4**
	12|3|Assistant_A|Done|**Down**|68|132|0	12|3|Assistant_A|Done|Down|68|132|0
	15|3|**Assistant_A**|**Done**|**Down**|138|**189**|0	15|3|Assistant_A|Done|Down|138|189|0

Πίνακας 26: Προσομοίωση 2 (χρόνοι 171-255)

Στο επόμενο χρονικό διάστημα (χρόνοι 171-246) η εργασία 7 απέτυχε για τρίτη φορά από τον εργαζόμενο «Assistant_A» σε χρόνο 200 και μέχρι το χρονικό σημείο 246 είχε ξεκινήσει εκ νέου η τέταρτη και τελευταία απόπειρα εκτέλεσής της. Εδώ πρέπει να σημειωθεί πως στην περίπτωση που ένας στόχος αποτύχει τέσσερις φορές τότε θεωρείται αποτυχημένος και τερματίζεται. Η εργασία 15 ολοκληρώθηκε από τον εργαζόμενο «Assistant_A» σε χρόνο 189 ολοκληρώνοντας τον τρίτο στόχο, ενώ η εργασία 11 παρέμεινε αμετάβλητη.

Στο επόμενο στιγμιότυπο (χρόνοι 246-255) παρατηρούμε πως η εργασία 7 απέτυχε για τέταρτη και τελευταία φορά σε χρόνο 246 οπότε ο δεύτερος στόχος διακόπηκε μόνιμα και το δίκτυο τερμάτισε την λειτουργία του σε χρόνο 255.

Προσομοίωση 3

Step (Time)	0 (0)	40 (105)
Task|Goal|Person|State|Flag|Start|End|Errors	1|1|None|Default|Down|0|0|0	1|1|None|**Processing**|**Up**|**84**|0|0
	6|1|None|Default|Down|0|0|0	6|1|None|Default|**Up**|0|0|0
	7|2|None|Default|Down|0|0|0	7|2|None|**Processing**|**Up**|**92**|0|0
	11|2|None|Default|Down|0|0|0	11|2|None|Default|**Up**|0|0|0
	12|3|None|Default|Down|0|0|0	12|3|None|**Processing**|**Up**|**93**|0|0
	15|3|None|Default|Down|0|0|0	15|3|None|Default|**Up**|0|0|0

Πίνακας 27: Προσομοίωση 3 (χρόνοι 0-105)

Στο πρώτο στιγμιότυπο (χρόνοι 0-105) ξεκίνησε η επεξεργασία των εργασιών 1, 7 και 12 σε χρόνους 84, 92 και 93 αντίστοιχα ενώ οι εργασίες 6, 11 και 15 παρέμειναν αμετάβλητες κάτι που είναι φυσιολογικό αφού για να εκτελεστούν θα πρέπει πρώτα να ολοκληρωθούν οι εργασίες που προηγούνται.

Step (Time)	80 (149)	120 (182)
Task|Goal|Person|State|Flag|Start|End|Errors	1|1|**Worker_A**|**Done**|Up|84|**117**|0	1|1|Worker_A|Done|**Down**|84|117|0
	6|1|None|**Processing**|Up|**123**|0|0	6|1|**Supervisor**|**Done**|**Down**|123|**155**|0
	7|2|**Assistant_A**|Processing|Up|92|**129**|**1**	7|2|Assistant_A|**Done**|Up|92|**174**|1
	11|2|None|Default|Up|0|0|**1**	11|2|None|**Started**|Up|**180**|0|1
	12|3|**Assistant_A**|Processing|Up|93|0|0	12|3|Assistant_A|Processing|Up|93|**149**|**1**
	15|3|None|Default|Up|0|0|0	15|3|None|Default|Up|0|0|**1**

Πίνακας 28: Προσομοίωση 3 (χρόνοι 105-182)

Στο επόμενο χρονικό διάστημα (χρόνοι 105-149) η εργασία 1 ανατέθηκε στον εργαζόμενο «Worker_A» ο οποίος την ολοκλήρωσε σε χρόνο 117, και στην συνέχεια ξεκίνησε η επεξεργασία της εργασίας 6 σε χρόνο 123. Η εργασία 7 ανατέθηκε στον εργαζόμενο «Assistant_A» ωστόσο απέτυχε σε χρόνο 129 ενώ στην συνέχεια επανεκκινήθηκε. Η εργασία 12 ανατέθηκε στον εργαζόμενο «Assistant_A», ενώ οι εργασίες 11 και 15 παρέμειναν αμετάβλητες.

Στο επόμενο χρονικό διάστημα (χρόνοι 149-182), η εργασία 6 ανατέθηκε στον εργαζόμενο «Supervisor» ο οποίος την ολοκλήρωσε σε χρόνο 155 ολοκληρώνοντας με τον τρόπο αυτό τον πρώτο στόχο. Η εργασία 7 ολοκληρώθηκε από τον εργαζόμενο «Assistant_A» σε χρόνο 174 και στην συνέχεια εκκινήθηκε η εργασία 11 σε χρόνο 180. Η εργασία 12 απέτυχε σε χρόνο 149 από τον εργαζόμενο «Assistant_A» και ξεκίνησε εκ νέου η επεξεργασία της σε αυτό το χρονικό διάστημα, ενώ η εργασία 15 παρέμεινε αμετάβλητη.

Step (Time)	**160 (281)**	**179 (342)**
Task | Goal | Person | State | Flag | Start | End | Errors	1 | 1 | Worker_A | Done | Down | 84 | 117 | 0	1 | 1 | Worker_A | Done | Down | 84 | 117 | 0
	6 | 1 | Supervisor | Done | Down | 123 | 155 | 0	6 | 1 | Supervisor | Done | Down | 123 | 155 | 0
	7 | 2 | Assistant_A | Done | **Down** | 92 | 174 | 1	7 | 2 | Assistant_A | Done | Down | 92 | 174 | 1
	11 | 2 | **Assistant_A** | **Done** | **Down** | 180 | **211** | 1	11 | 2 | Assistant_A | Done | Down | 180 | 211 | 1
	12 | 3 | Assistant_A | **Done** | Up | 93 | **214** | **2**	12 | 3 | Assistant_A | Done | **Down** | 93 | 214 | 2
	15 | 3 | Assistant_A | **Failed** | Up | **221** | **272** | **2**	15 | 3 | Assistant_A | **Done** | **Down** | 221 | **336** | 2

Πίνακας 29: Προσομοίωση 3 (χρόνοι 182-342)

Στην συνέχεια (χρόνοι 182-281), η εργασία 11 ανατέθηκε στον εργαζόμενο «Assistant_A» ο οποίος την ολοκλήρωσε σε χρόνο 211 ολοκληρώνοντας με τον τρόπο αυτό τον δεύτερο στόχο. Η εργασία 12 ολοκληρώθηκε από τον εργαζόμενο «Assistant_A» σε χρόνο 214. Στην συνέχεια ξεκίνησε η επεξεργασία της εργασίας 15 σε χρόνο 221 η οποία ανατέθηκε στον εργαζόμενο «Assistant_A» ωστόσο απέτυχε σε χρόνο 272.

Στο επόμενο χρονικό διάστημα (χρόνοι 281-342), η εργασία 15 ολοκληρώθηκε από τον εργαζόμενο «Assistant_A» σε χρόνο 336 ολοκληρώνοντας τον τρίτο στόχο και στην συνέχεια τερματίστηκε η λειτουργία του δικτύου σε χρόνο 342.

Σενάριο 2: Εκτέλεση Προσομοίωσης

Στα πλαίσια του δεύτερου σεναρίου, όπως προαναφέρθηκε, η μοναδική διαφορά με το πρώτο σενάριο είναι πως οι εργασίες είναι ιδιαίτερα απαιτητικές συγκριτικά με το πρώτο σενάριο και κατά συνέπεια κάθε εργαζόμενος θα ήταν ιδιαίτερα δύσκολο να ασχοληθεί με παραπάνω από μία εργασίες ταυτόχρονα. Στην συνέχεια παρουσιάζονται οι προσομοιώσεις του δεύτερου σεναρίου:

Προσομοίωση 4

Step (Time)	0 (0)	40 (91)
Task|Goal|Person|State|Flag|Start|End|Errors	1|1|None|Default|Down|0|0|0	1|1|**Worker_A**|**Processing**|**Up**|**59**|0|0
	6|1|None|Default|Down|0|0|0	6|1|None|Default|**Up**|0|0|0
	7|2|None|Default|Down|0|0|0	7|2|None|**Processing**|**Up**|**57**|0|0
	11|2|None|Default|Down|0|0|0	11|2|None|Default|**Up**|0|0|0
	12|3|None|Default|Down|0|0|0	12|3|None|**Processing**|**Up**|**58**|0|0
	15|3|None|Default|Down|0|0|0	15|3|None|Default|**Up**|0|0|0

Πίνακας 30: Προσομοίωση 4 (χρόνοι 0-91)

Στο πρώτο στιγμιότυπο της προσομοίωσης 4 (χρόνοι 0-91), παρατηρούμε πως ξεκίνησε η επεξεργασία της εργασίας 1 σε χρόνο 59 και ανατέθηκε στον εργαζόμενο «Worker_A». Επιπλέον έχει ξεκινήσει η επεξεργασία των εργασιών 7 και 12 σε χρόνους 57 και 58 αντίστοιχα. Οι εργασίες 6, 11 και 15 δεν έχουν στον χρόνο αυτό ξεκινήσει.

Step (Time)	80 (114)	120 (155)
Task|Goal|Person|State|Flag|Start|End|Errors	1|1|Worker_A|**Started**|Up|59|**93**|**1**	1|1|Worker_A|**Deferred**|**Down**|59|**145**|**2**
	6|1|None|Default|Up|0|0|**1**	6|1|None|Default|**Down**|0|0|**2**
	7|2|**Assistant_A**|Processing|Up|57|**95**|**1**	7|2|Assistant_A|**Done**|Up|57|**148**|1
	11|2|None|Default|Up|0|0|**1**	11|2|None|Default|Up|0|0|1
	12|3|**Supervisor**|Processing|Up|58|0|0	12|3|Supervisor|**Done**|Up|58|**114**|0
	15|3|None|Default|Up|0|0|0	15|3|None|**Processing**|Up|**125**|0|0

Πίνακας 31: Προσομοίωση 4 (χρόνοι 91-155)

Στο επόμενο στιγμιότυπο (χρόνοι 91-114), παρατηρούμε πως η εργασία 1 απέτυχε σε χρόνο 93, ωστόσο είχε πάλι ξεκινήσει μέσα στο συγκεκριμένο χρονικό διάστημα. Η εργασία 7 ανατέθηκε στον εργαζόμενο «Assistant_A» ωστόσο η επεξεργασία της απέτυχε σε χρόνο 95 και στην συνέχεια επανεκκινήθηκε. Η εργασία 12 ανατέθηκε στον εργαζόμενο «Supervisor» ενώ οι εργασίες 6, 11 και 15 παρέμειναν αμετάβλητες.

Στην συνέχεια (χρόνοι 114-155), η εργασία 1 απέτυχε για δεύτερη φορά (αναβλήθηκε) σε χρόνο 145. Η εργασία 7 ολοκληρώθηκε από τον εργαζόμενο «Assistant_A» σε χρόνο 148. Η εργασία 12 ολοκληρώθηκε από τον εργαζόμενο «Supervisor» σε χρόνο 114 και στην συνέχεια ξεκίνησε η επεξεργασία της εργασίας 15 σε χρόνο 125. Η εργασίες 6 και 11 παρέμειναν αμετάβλητες.

Step (Time)	160 (195)	200 (237)
Task | Goal | Person | State | Flag | Start | End | Errors	1 | 1 | Worker_A | **Processing** | **Up** | 59 | 145 | 2	1 | 1 | Worker_A | Processing | Up | 59 | **195** | **3**
	6 | 1 | None | Default | **Up** | 0 | 0 | 2	6 | 1 | None | Default | Up | 0 | 0 | **3**
	7 | 2 | Assistant_A | Done | **Down** | 57 | 148 | **2**	7 | 2 | Assistant_A | Done | Down | 57 | 148 | 2
	11 | 2 | **Assistant_A** | **Interrupted** | **Down** | **156** | **189** | **2**	11 | 2 | Assistant_A | **Done** | Down | 156 | **234** | 2
	12 | 3 | Supervisor | Done | **Down** | 58 | 114 | 0	12 | 3 | Supervisor | Done | Down | 58 | 114 | 0
	15 | 3 | **Supervisor** | **Done** | **Down** | 125 | **170** | 0	15 | 3 | Supervisor | Done | Down | 125 | 170 | 0

Πίνακας 32: Προσομοίωση 4 (χρόνοι 155-237)

Στο επόμενο χρονικό διάστημα (χρόνοι 155-195), ξεκίνησε για τρίτη φορά η επεξεργασία της εργασίας 1. Η εργασία 11 ξεκίνησε σε χρόνο 156, ανατέθηκε στον εργαζόμενο «Assistant_A», ωστόσο διακόπηκε σε χρόνο 189. Η εργασία 15 ανατέθηκε στον εργαζόμενο «Supervisor» ο οποίος την ολοκλήρωσε σε χρόνο 170 ολοκληρώνοντας με τον τρόπο αυτό τον τρίτο στόχο. Οι εργασία 6 παρέμεινε αμετάβλητη.

Στη συνέχεια (χρόνοι 195-237), ο εργαζόμενος «Worker_A» απέτυχε για τρίτη φορά να ολοκληρώσει την εργασία 1 σε χρόνο 195 ωστόσο η εργασία επανεκκινήθηκε μέσα σε αυτό το χρονικό διάστημα. Η εργασία 11

ολοκληρώθηκε από τον εργαζόμενο «Assistant_A» σε χρόνο 234 ολοκληρώνοντας με τον τρόπο αυτό τον δεύτερο στόχο, ενώ η εργασία 6 παρέμεινε αμετάβλητη.

Step (Time)	**227 (286)**
Task | Goal | Person | State | Flag | Start | End | Errors	1 | 1 | Worker_A | **Done** | **Down** | 59 | **243** | 3
	6 | 1 | **Supervisor** | **Done** | **Down** | **249** | **280** | 3
	7 | 2 | Assistant_A | Done | Down | 57 | 148 | 2
	11 | 2 | Assistant_A | Done | Down | 156 | 234 | 2
	12 | 3 | Supervisor | Done | Down | 58 | 114 | 0
	15 | 3 | Supervisor | Done | Down | 125 | 170 | 0

Πίνακας 33: Προσομοίωση 4 (χρόνοι 237-286)

Στο επόμενο και τελευταίο χρονικό διάστημα (χρόνοι 237-286), η εργασία 1 ολοκληρώθηκε από τον εργαζόμενο «Worker_A» σε χρόνο 243, και στη συνέχεια ξεκίνησε η επεξεργασία της εργασίας 6 σε χρόνο 249, η οποία ανατέθηκε στον εργαζόμενο «Supervisor» και ολοκληρώθηκε σε χρόνο 280 ολοκληρώνοντας με τον τρόπο αυτό τον πρώτο στόχο. Στην συνέχεια τερματίστηκε η λειτουργία του δικτύου σε χρόνο 286.

Προσομοίωση 5

Step (Time)	**0 (0)**	**40 (67)**
Task | Goal | Person | State | Flag | Start | End | Errors	1 | 1 | None | Default | Down | 0 | 0 | 0	1 | 1 | **Worker_A** | **Deferred** | Down | **34** | **66** | **1**
	6 | 1 | None | Default | Down | 0 | 0 | 0	6 | 1 | None | Default | Down | 0 | 0 | **1**
	7 | 2 | None | Default | Down | 0 | 0 | 0	7 | 2 | None | **Processing** | **Up** | **38** | 0 | 0
	11 | 2 | None | Default | Down | 0 | 0 | 0	11 | 2 | None | Default | **Up** | 0 | 0 | 0
	12 | 3 | None | Default | Down | 0 | 0 | 0	12 | 3 | None | **Processing** | **Up** | **45** | 0 | 0
	15 | 3 | None | Default | Down | 0 | 0 | 0	15 | 3 | None | Default | **Up** | 0 | 0 | 0

Πίνακας 34: Προσομοίωση 5 (χρόνοι 0-67)

Στο πρώτο στιγμιότυπο της λειτουργίας του δικτύου (χρόνοι 0-67) παρατηρούμε πως ξεκίνησε η επεξεργασία της εργασίας 1 σε χρόνο 34 η οποία ανατέθηκε στον εργαζόμενο «Worker_A», ωστόσο αναβλήθηκε σε χρόνο 66. Επιπλέον ξεκίνησε η επεξεργασία των εργασιών 7 και 12 σε χρόνους 38 και 45

αντίστοιχα χωρίς ωστόσο να έχουν ανατεθεί σε κάποιον εργαζόμενο. Οι εργασίες 6, 11 και 15 παρέμειναν αμετάβλητες.

Step (Time)	80 (93)	120 (129)
Task|Goal|Person|State|Flag|Start|End|Errors	1|1|Worker_A|**Processing**|**Up**|34|66|1	1|1|Worker_A|**Deferred**|Up|34|**115**|**2**
	6|1|None|Default|**Up**|0|0|1	6|1|None|Default|Up|0|0|**2**
	7|2|**Assistant_A**|**Failed**|Up|38|**75**|**1**	7|2|Assistant_A|**Processing**|Up|38|75|1
	11|2|None|Default|Up|0|0|**1**	11|2|None|Default|Up|0|0|1
	12|3|**Supervisor**|**Done**|Up|45|**90**|0	12|3|Supervisor|Done|Up|45|90|0
	15|3|None|Default|Up|0|0|0	15|3|None|**Processing**|Up|**97**|0|0

Πίνακας 35: Προσομοίωση 5 (χρόνοι 67-129)

Στο επόμενο στιγμιότυπο (χρόνοι 67-93) ξεκίνησε εκ νέου η επεξεργασία της εργασίας 1, ενώ η εργασία 7 ανατέθηκε στον εργαζόμενο «Assistant_A» ωστόσο απέτυχε σε χρόνο 75. Η εργασία 12 ανατέθηκε στον εργαζόμενο «Supervisor» ο οποίος την ολοκλήρωσε σε χρόνο 90. Οι εργασίες 6, 11 και 15 παρέμειναν αμετάβλητες.

Στο επόμενο χρονικό σημείο (χρόνοι 93-129), η εργασία 1 απέτυχε για δεύτερη φορά (αναβλήθηκε) σε χρόνο 115, ενώ ξεκίνησε εκ νέου η επεξεργασία της εργασίας 7. Επιπλέον ξεκίνησε η επεξεργασία της εργασίας 15 σε χρόνο 97 ενώ η εργασίες 6 και 11 παρέμειναν αμετάβλητες.

Step (Time)	160 (169)	200 (219)
Task|Goal|Person|State|Flag|Start|End|Errors	1|1|Worker_A|**Interrupted**|**Down**|34|**163**|**3**	1|1|Worker_A|**Done**|**Up**|34|**217**|3
	6|1|None|Default|**Down**|0|0|**3**	6|1|None|Default|**Up**|0|0|3
	7|2|Assistant_A|**Done**|Up|38|**129**|1	7|2|Assistant_A|Done|Up|38|129|**2**
	11|2|**Assistant_A**|**Processing**|Up|**138**|0|1	11|2|Assistant_A|Processing|Up|138|**169**|**2**
	12|3|Supervisor|Done|**Down**|45|90|0	12|3|Supervisor|Done|Down|45|90|0
	15|3|**Supervisor**|**Done**|**Down**|97|**148**|0	15|3|Supervisor|Done|Down|97|148|0

Πίνακας 36: Προσομοίωση 5 (χρόνοι 129-219)

Στο επόμενο χρονικό διάστημα (χρόνοι 129-169) η εργασία 1 απέτυχε για τρίτη φορά (διακόπηκε) από τον εργαζόμενο «Worker_A» σε χρόνο 163. Η εργασία 7 ολοκληρώθηκε από τον εργαζόμενο «Assistant_A» σε χρόνο 129 και στην συνέχεια ξεκίνησε η επεξεργασία της εργασίας 11 σε χρόνο 138 η οποία ανατέθηκε και αυτή στον εργαζόμενο «Assistant_A». Επιπλέον, η

εργασία 15 ανατέθηκε στον εργαζόμενο «Supervisor» ο οποίος την ολοκλήρωσε σε χρόνο 148 ολοκληρώνοντας με τον τρόπο αυτό τον τρίτο στόχο. Η εργασία 6 παρέμεινε αμετάβλητη.

Στο επόμενο στιγμιότυπο (χρόνοι 169-219) παρατηρούμε πως η εργασία 1 ολοκληρώθηκε από τον εργαζόμενο «Worker_A» σε χρόνο 217, ενώ η εργασία 11 απέτυχε για δεύτερη φορά σε χρόνο 169, ωστόσο επανεκκινήθηκε. Η εργασία 6 παρέμεινε αμετάβλητη.

Step (Time)	**225 (261)**
Task | Goal | Person | State | Flag | Start | End | Errors	1 | 1 | Worker_A | Done | **Down** | 34 | 217 | 3
	6 | 1 | **Supervisor** | **Done** | **Down** | **223** | **255** | 3
	7 | 2 | Assistant_A | Done | **Down** | 38 | 129 | 2
	11 | 2 | Assistant_A | **Done** | **Down** | 138 | **219** | 2
	12 | 3 | Supervisor | Done | Down | 45 | 90 | 0
	15 | 3 | Supervisor | Done | Down | 97 | 148 | 0

Πίνακας 37: Προσομοίωση 5 (χρόνοι 219-261)

Στο επόμενο και τελευταίο χρονικό διάστημα (χρόνοι 219-261), ξεκίνησε η επεξεργασία της εργασίας 6 σε χρόνο 223 και ανατέθηκε στον εργαζόμενο «Supervisor» ο οποίος την ολοκλήρωσε σε χρόνο 255 ολοκληρώνοντας τον πρώτο στόχο. Επιπλέον ολοκληρώθηκε η εκτέλεση της εργασίας 11 από τον εργαζόμενο «Assistant_A» σε χρόνο 219 ολοκληρώνοντας με τον τρόπο αυτό τον δεύτερο στόχο. Στην συνέχεια τερματίστηκε η λειτουργία του δικτύου σε χρόνο 261.

Προσομοίωση 6

Step (Time)	**0 (0)**	**40 (61)**
Task | Goal | Person | State | Flag | Start | End | Errors	1 | 1 | None | Default | Down | 0 | 0 | 0	1 | 1 | None | **Processing** | **Up** | **34** | 0 | 0
	6 | 1 | None | Default | Down | 0 | 0 | 0	6 | 1 | None | Default | **Up** | 0 | 0 | 0
	7 | 2 | None | Default | Down | 0 | 0 | 0	7 | 2 | None | **Processing** | **Up** | **41** | 0 | 0
	11 | 2 | None | Default | Down | 0 | 0 | 0	11 | 2 | None | Default | **Up** | 0 | 0 | 0
	12 | 3 | None | Default | Down | 0 | 0 | 0	12 | 3 | None | **Processing** | **Up** | **45** | 0 | 0
	15 | 3 | None | Default | Down | 0 | 0 | 0	15 | 3 | None | Default | **Up** | 0 | 0 | 0

Πίνακας 38: Προσομοίωση 6 (χρόνοι 0-61)

Στο πρώτο στιγμιότυπο (χρόνοι 0-61) ξεκίνησε η επεξεργασία των εργασιών 1, 7 και 12 σε χρόνους 34, 41 και 45 αντίστοιχα ενώ οι εργασίες 6, 11 και 15 παρέμειναν αμετάβλητες κάτι που είναι φυσιολογικό αφού για να εκτελεστούν θα πρέπει πρώτα να ολοκληρωθούν οι εργασίες που προηγούνται.

Step (Time)	80 (83)	120 (132)
Task|Goal|Person|State|Flag|Start|End|Errors	1|1|**Worker_A**|**Done**|Up|34|**65**|0	1|1|Worker_A|Done|Up|34|65|**1**
	6|1|None|**Processing**|Up|**75**|0|0	6|1|**Supervisor**|Processing|Up|75|**113**|**1**
	7|2|**Assistant_A**|**Done**|Up|41|**73**|0	7|2|Assistant_A|Done|**Down**|41|73|0
	11|2|None|**Processing**|Up|**81**|0|0	11|2|**Assistant_A**|**Done**|**Down**|81|**123**|0
	12|3|None|**Deferred**|Up|45|**63**|**1**	12|3|None|**Processing**|Up|45|63|1
	15|3|None|Default|Up|0|0|**1**	15|3|None|Default|Up|0|0|1

Πίνακας 39: Προσομοίωση 6 (χρόνοι 61-132)

Στο επόμενο χρονικό διάστημα (χρόνοι 61-83) η εργασία 1 ανατέθηκε στον εργαζόμενο «Worker_A» ο οποίος την ολοκλήρωσε σε χρόνο 65, και στην συνέχεια ξεκίνησε η επεξεργασία της εργασίας 6 σε χρόνο 75. Η εργασία 7 ανατέθηκε στον εργαζόμενο «Assistant_A» ο οποίος την ολοκλήρωσε σε χρόνο 73 και στην συνέχεια ξεκίνησε η επεξεργασία της εργασίας 11 σε χρόνο 81. Η εργασία 12 αναβλήθηκε σε χρόνο 63 καθώς ο εργαζόμενος που επιλέχθηκε να την εκτελέσει δεν ήταν διαθέσιμος οπότε επιλέχθηκε η στρατηγική της αναβολής της εργασίας. Οι εργασία 15 παρέμεινε αμετάβλητη.
Στο επόμενο χρονικό διάστημα (χρόνοι 83-132), η εργασία 6 ανατέθηκε στον εργαζόμενο «Supervisor» ο οποίος απέτυχε να την ολοκληρώσει σε χρόνο 113 ωστόσο επανεκκινήθηκε. Η εργασία 11 ανατέθηκε στην εργαζόμενο «Assistant_A» ο οποίος την ολοκλήρωσε σε χρόνο 123 ολοκληρώνοντας με τον τρόπο αυτό τον δεύτερο στόχο. Επιπλέον ξεκίνησε η επεξεργασία της εργασίας 12 ενώ η εργασία 15 παρέμεινε αμετάβλητη.

Step (Time)	160 (171)	200 (221)
Task | Goal | Person | State | Flag | Start | End | Errors	1 | 1 | Worker_A | Done | Up | 34 | 65 | **2**	1 | 1 | Worker_A | Done | Up | 34 | 65 | **3**
	6 | 1 | Supervisor | **Deferred** | Up | 75 | **160** | **2**	6 | 1 | Supervisor | **Started** | Up | 75 | **204** | **3**
	7 | 2 | Assistant_A | Done | Down | 41 | 73 | 0	7 | 2 | Assistant_A | Done | Down | 41 | 73 | 0
	11 | 2 | Assistant_A | Done | Down | 81 | 123 | 0	11 | 2 | Assistant_A | Done | Down | 81 | 123 | 0
	12 | 3 | **Assistant_A** | Processing | Up | 45 | **139** | **2**	12 | 3 | Assistant_A | **Done** | Up | 45 | **206** | 2
	15 | 3 | None | Default | Up | 0 | 0 | **2**	15 | 3 | None | **Processing** | Up | **213** | 0 | 2

Πίνακας 40: Προσομοίωση 6 (χρόνοι 132-221)

Στην συνέχεια (χρόνοι 132-171), η εργασία 6 απέτυχε για δεύτερη φορά (αναβλήθηκε) σε χρόνο 160. Η εργασία 12 ανατέθηκε στον εργαζόμενο «Assistant_A» ο οποίος απέτυχε να την ολοκληρώσει σε χρόνο 139, ωστόσο η εργασία στην συνέχεια επανεκκινήθηκε. Η εργασία 15 παρέμεινε αμετάβλητη.

Στο επόμενο χρονικό διάστημα (χρόνοι 171-221), η εργασία 6 απέτυχε για τρίτη φορά σε χρόνο 204, ωστόσο επανεκκινήθηκε. Η εργασία 12 ολοκληρώθηκε από τον εργαζόμενο «Assistant_A» σε χρόνο 206 και στην συνέχεια ξεκίνησε η επεξεργασία της εργασίας 15 σε χρόνο 213.

Step (Time)	223 (271)
Task | Goal | Person | State | Flag | Start | End | Errors	1 | 1 | Worker_A | Done | **Down** | 34 | 65 | 3
	6 | 1 | Supervisor | **Done** | **Down** | 75 | **251** | 3
	7 | 2 | Assistant_A | Done | Down | 41 | 73 | 0
	11 | 2 | Assistant_A | Done | Down | 81 | 123 | 0
	12 | 3 | Assistant_A | Done | **Down** | 45 | 206 | 2
	15 | 3 | **Assistant_A** | **Done** | **Down** | 213 | **265** | 2

Πίνακας 41: Προσομοίωση 6 (χρόνοι 221-271)

Στο επόμενο και τελευταίο χρονικό διάστημα (χρόνοι 221-271), ολοκληρώθηκε η επεξεργασία της εργασίας 6 σε χρόνο 251 από τον εργαζόμενο «Supervisor» ολοκληρώνοντας τον πρώτο στόχο. Επιπλέον ολοκληρώθηκε η εκτέλεση της εργασίας 15 από τον εργαζόμενο «Assistant_A» σε χρόνο 265 ολοκληρώνοντας με τον τρόπο αυτό τον τρίτο στόχο και στην συνέχεια τερματίστηκε η λειτουργία του δικτύου σε χρόνο 271.

Ανακεφαλαίωση

Συγκεντρωτικά, τα αποτελέσματα των δύο σεναρίων φαίνονται στους παρακάτω πίνακες.

Simulation #	Task	Person	Duration	# of Errors
1	1	Worker_A	47	0
1	6	Supervisor	34	0
1	7	Assistant_A	91	1
1	11	Assistant_A	32	0
1	12	Assistant_A	126	1
1	15	Assistant_A	52	0
2	1	Worker_A	36	0
2	6	Supervisor	34	0
2	7	Assistant_A	176	4
2	11	None	0	0
2	12	Assistant_A	64	0
2	15	Assistant_A	51	0
3	1	Worker_A	33	0
3	6	Supervisor	32	0
3	7	Assistant_A	82	1
3	11	Assistant_A	31	0
3	12	Assistant_A	121	1
3	15	Assistant_A	115	1

Πίνακας 42: Συγκεντρωτικά αποτελέσματα 1ου σεναρίου Α'

Κατά την διάρκεια της πρώτης προσομοίωσης, η εργασία 1 ολοκληρώθηκε χωρίς σφάλματα από τον εργαζόμενο «Worker_A» και διήρκησε 47 μονάδες χρόνου, η εργασία 6 ολοκληρώθηκε χωρίς σφάλματα από τον εργαζόμενο «Supervisor» και διήρκησε 34 μονάδες χρόνου ολοκληρώνοντας τον πρώτο στόχο, η εργασία 7 ολοκληρώθηκε από τον εργαζόμενο «Assistant_A» με ένα σφάλμα και διήρκησε 91 μονάδες χρόνου, η εργασία 11 ολοκληρώθηκε από τον εργαζόμενο «Assistant_A» χωρίς σφάλματα και διήρκησε 32 μονάδες χρόνου ολοκληρώνοντας τον δεύτερο στόχο, η εργασία 12 ολοκληρώθηκε από τον εργαζόμενο «Assistant_A» με ένα σφάλμα και διήρκησε 126 μονάδες χρόνου, και η εργασία 15 ολοκληρώθηκε από τον εργαζόμενο «Assistant_A» χωρίς σφάλματα και διήρκησε 52 μονάδες χρόνου ολοκληρώνοντας τον τρίτο στόχο.

Κατά την διάρκεια της δεύτερης προσομοίωσης, η εργασία 1 ολοκληρώθηκε χωρίς σφάλματα από τον εργαζόμενο «Worker_A» και διήρκησε 36 μονάδες χρόνου, η εργασία 6 ολοκληρώθηκε χωρίς σφάλματα από τον εργαζόμενο «Supervisor» και διήρκησε 34 μονάδες χρόνου ολοκληρώνοντας τον πρώτο

στόχο, ενώ η εργασία 7 δεν ολοκληρώθηκε ποτέ αφού απέτυχε τέσσερις φορές και κατά συνέπεια δεν προχώρησε και η εκτέλεση της εργασίας 11 ενώ ο δεύτερος στόχος τερματίστηκε ανεπιτυχώς. Η εργασία 12 ολοκληρώθηκε από τον εργαζόμενο «Assistant_A» χωρίς σφάλματα και διήρκησε 64 μονάδες χρόνου, και η εργασία 15 ολοκληρώθηκε από τον εργαζόμενο «Assistant_A» χωρίς σφάλματα και διήρκησε 51 μονάδες χρόνου ολοκληρώνοντας τον τρίτο στόχο.

Κατά την διάρκεια της τρίτης προσομοίωσης, η εργασία 1 ολοκληρώθηκε χωρίς σφάλματα από τον εργαζόμενο «Worker_A» και διήρκησε 33 μονάδες χρόνου, η εργασία 6 ολοκληρώθηκε χωρίς σφάλματα από τον εργαζόμενο «Supervisor» και διήρκησε 32 μονάδες χρόνου ολοκληρώνοντας τον πρώτο στόχο, η εργασία 7 ολοκληρώθηκε από τον εργαζόμενο «Assistant_A» με ένα σφάλμα και διήρκησε 82 μονάδες χρόνου, η εργασία 11 ολοκληρώθηκε από τον εργαζόμενο «Assistant_A» χωρίς σφάλματα και διήρκησε 31 μονάδες χρόνου ολοκληρώνοντας τον δεύτερο στόχο, η εργασία 12 ολοκληρώθηκε από τον εργαζόμενο «Assistant_A» με ένα σφάλμα και διήρκησε 121 μονάδες χρόνου, και η εργασία 15 ολοκληρώθηκε από τον εργαζόμενο «Assistant_A» με ένα σφάλμα και διήρκησε 115 μονάδες χρόνου ολοκληρώνοντας τον τρίτο στόχο.

Task	Avg. Duration	Total # of Errors
1	38,66666667	0
6	33,33333333	0
7	116,3333333	6
11	31,5	0
12	103,6666667	2
15	72,66666667	1

Πίνακας 43: Συγκεντρωτικά αποτελέσματα 1ου σεναρίου Β'

Τα συμπεράσματα που μπορούν να ληφθούν ύστερα από την εκτέλεση των προσομοιώσεων του πρώτου σεναρίου, αναφορικά με τις εργασίες, όπως φαίνεται και στον παραπάνω πίνακα, είναι πως η εργασία 7 φαίνεται πως είναι υποψήφια να παρουσιάσει κάποιο πρόβλημα (λόγω των περισσότερων λαθών κατά την διάρκεια επεξεργασίας της) περισσότερο από όλες τις υπόλοιπες εργασίες, ενώ η αμέσως επόμενη είναι η εργασία 12 χωρίς ωστόσο να θεωρείται ιδιαίτερα μεγάλος ο αριθμός των λαθών που παρουσίασε η συγκεκριμένη εργασία. Οι υπόλοιπες εργασίες θεωρούνται μάλλον ασφαλείς.

Όσον αφορά την διάρκεια εκτέλεσης κάθε εργασίας παρατηρούμε πως η εργασία 7 κρίνεται ως η πιο χρονοβόρα με την εργασία 12 να ακολουθεί, ενώ και η εργασία 15 φαίνεται να διαρκεί αρκετά περισσότερο από τις υπόλοιπες εργασίες. Οι εργασίες 1, 6 και 11 φαίνονται να βρίσκονται σε φυσιολογικά πλαίσια. Σύμφωνα με τα παραπάνω, η εργασία 7 φαίνεται πως χρειάζεται περισσότερο προσοχή από όλες τις εργασίες, ενώ οι εργασίες 12 και 15 θα έπρεπε και αυτές να βρίσκονται υπό μερική παρακολούθηση, ενώ οι εργασίες 1, 6 και 11 κρίνονται μάλλον ασφαλείς.

Simulation #	Task	Person	Duration	# of Errors
4	1	Worker_A	184	3
4	6	Supervisor	31	0
4	7	Assistant_A	91	1
4	11	Assistant_A	78	1
4	12	Supervisor	56	0
4	15	Supervisor	45	0
5	1	Worker_A	183	3
5	6	Supervisor	32	0
5	7	Assistant_A	91	1
5	11	Assistant_A	81	1
5	12	Supervisor	45	0
5	15	Supervisor	51	0
6	1	Worker_A	31	0
6	6	Supervisor	176	3
6	7	Assistant_A	32	0
6	11	Assistant_A	42	0
6	12	Assistant_A	161	2
6	15	Assistant_A	52	0

Πίνακας 44: Συγκεντρωτικά αποτελέσματα 2ου σεναρίου Α'

Κατά την διάρκεια της τέταρτης προσομοίωσης, η εργασία 1 ολοκληρώθηκε με τρία σφάλματα από τον εργαζόμενο «Worker_A» και διήρκησε 184 μονάδες χρόνου, η εργασία 6 ολοκληρώθηκε χωρίς σφάλματα από τον εργαζόμενο «Supervisor» και διήρκησε 31 μονάδες χρόνου ολοκληρώνοντας τον πρώτο στόχο, η εργασία 7 ολοκληρώθηκε από τον εργαζόμενο «Assistant_A» με ένα σφάλμα και διήρκησε 91 μονάδες χρόνου, η εργασία 11 ολοκληρώθηκε από τον εργαζόμενο «Assistant_A» με ένα σφάλμα και διήρκησε 78 μονάδες χρόνου ολοκληρώνοντας τον δεύτερο στόχο, η εργασία

12 ολοκληρώθηκε από τον εργαζόμενο «Supervisor» χωρίς σφάλματα και διήρκησε 56 μονάδες χρόνου, και η εργασία 15 ολοκληρώθηκε από τον εργαζόμενο «Supervisor» χωρίς σφάλματα και διήρκησε 45 μονάδες χρόνου ολοκληρώνοντας τον τρίτο στόχο.

Κατά την διάρκεια της πέμπτης προσομοίωσης, η εργασία 1 ολοκληρώθηκε με τρία σφάλματα από τον εργαζόμενο «Worker_A» και διήρκησε 183 μονάδες χρόνου, η εργασία 6 ολοκληρώθηκε χωρίς σφάλματα από τον εργαζόμενο «Supervisor» και διήρκησε 32 μονάδες χρόνου ολοκληρώνοντας τον πρώτο στόχο, η εργασία 7 ολοκληρώθηκε από τον εργαζόμενο «Assistant_A» με ένα σφάλμα και διήρκησε 91 μονάδες χρόνου, η εργασία 11 ολοκληρώθηκε από τον εργαζόμενο «Assistant_A» με ένα σφάλμα και διήρκησε 81 μονάδες χρόνου ολοκληρώνοντας τον δεύτερο στόχο, η εργασία 12 ολοκληρώθηκε από τον εργαζόμενο «Supervisor» χωρίς σφάλματα και διήρκησε 45 μονάδες χρόνου, και η εργασία 15 ολοκληρώθηκε από τον εργαζόμενο «Supervisor» χωρίς σφάλματα και διήρκησε 51 μονάδες χρόνου ολοκληρώνοντας τον τρίτο στόχο.

Κατά την διάρκεια της έκτης προσομοίωσης, η εργασία 1 ολοκληρώθηκε χωρίς σφάλματα από τον εργαζόμενο «Worker_A» και διήρκησε 31 μονάδες χρόνου, η εργασία 6 ολοκληρώθηκε με τρία σφάλματα από τον εργαζόμενο «Supervisor» και διήρκησε 176 μονάδες χρόνου ολοκληρώνοντας τον πρώτο στόχο, η εργασία 7 ολοκληρώθηκε από τον εργαζόμενο «Assistant_A» χωρίς σφάλματα και διήρκησε 32 μονάδες χρόνου, η εργασία 11 ολοκληρώθηκε από τον εργαζόμενο «Assistant_A» χωρίς σφάλματα και διήρκησε 42 μονάδες χρόνου ολοκληρώνοντας τον δεύτερο στόχο, η εργασία 12 ολοκληρώθηκε από τον εργαζόμενο «Assistant_A» με δύο σφάλματα και διήρκησε 161 μονάδες χρόνου, και η εργασία 15 ολοκληρώθηκε από τον εργαζόμενο «Assistant_A» χωρίς σφάλματα και διήρκησε 52 μονάδες χρόνου ολοκληρώνοντας τον τρίτο στόχο.

Task	Avg. Duration	Total # of Errors
1	132,6666667	6
6	79,66666667	3
7	71,33333333	2
11	67	2
12	87,33333333	2
15	49,33333333	0

Πίνακας 45: Συγκεντρωτικά αποτελέσματα 2ου σεναρίου Β'

Τα συμπεράσματα που μπορούν να ληφθούν ύστερα από την εκτέλεση των προσομοιώσεων του δεύτερου σεναρίου, αναφορικά με τις εργασίες, όπως φαίνεται και στον παραπάνω πίνακα, είναι πως η εργασία 1 φαίνεται πως είναι υποψήφια να παρουσιάσει κάποιο πρόβλημα (λόγω των περισσότερων λαθών κατά την διάρκεια επεξεργασίας της) περισσότερο από όλες τις υπόλοιπες εργασίες, ενώ οι αμέσως επόμενες είναι οι εργασίες 6, 7, 11 και 12 χωρίς ωστόσο να θεωρείται ιδιαίτερα μεγάλος ο αριθμός των λαθών που παρουσίασαν οι συγκεκριμένες εργασίες. Η εργασία 15 θεωρείται μάλλον ασφαλής. Όσον αφορά την διάρκεια εκτέλεσης κάθε εργασίας, παρατηρούμε πως η εργασία 1 κρίνεται ως η πιο χρονοβόρα με την εργασία 12 να ακολουθεί, ενώ και οι εργασίες 6, 7 και 11 φαίνεται να διαρκούν για αρκετό χρονικό διάστημα. Η εργασία 15 φαίνεται να βρίσκεται σε φυσιολογικά πλαίσια. Σύμφωνα με τα παραπάνω, η εργασία 1 φαίνεται πως χρειάζεται περισσότερο προσοχή από όλες τις εργασίες, ενώ οι εργασίες 6, 7, 11 και 12 θα έπρεπε και αυτές να βρίσκονται υπό μερική παρακολούθηση, ενώ η εργασία 15 κρίνεται μάλλον ασφαλής. Τα αποτελέσματα κρίνονται απόλυτα φυσιολογικά αφού βασική παραδοχή του δεύτερου σεναρίου ήταν το γεγονός πως οι εργασίες ήταν πολύ πιο απαιτητικές από το πρώτο σενάριο.

Person	Total # of Tasks
Supervisor	10
Assistant_A	19
Assistant_B	0
Worker_A	6
Worker_B	0
Worker_C	0

Πίνακας 46: Συγκεντρωτικά Εργαζομένων

Κατά την διάρκεια των έξι προσομοιώσεων, τρεις από τους έξι εργαζόμενους χρησιμοποιήθηκαν για την εκτέλεση των εργασιών. Αυτό σημαίνει πως οι υπόλοιποι εργαζόμενοι πιθανότατα δεν χρειάζονταν για την εκτέλεση των προκαθορισμένων εργασιών των δύο σεναρίων και κατά συνέπεια θα

μπορούσαν να είχαν αποδεσμευτεί ώστε να αξιοποιηθούν σε άλλες εργασίες και καθήκοντα. Επιπλέον παρατηρούμε πως ο εργαζόμενος με την μεγαλύτερη κινητικότητα και αποδοτικότητα ήταν ο εργαζόμενος «Assistant_A» με 19 ολοκληρωμένες εργασίες στο ενεργητικό του ενώ ο εργαζόμενος «Supervisor» είχε 10 ολοκληρωμένες εργασίες στο ενεργητικό του και τελευταίος ο εργαζόμενος «Worker_A» με 6 ολοκληρωμένες εργασίες. Τα στοιχεία αυτά δίνουν μία σαφή εικόνα αναφορικά με την απόδοση των εργαζομένων στα πλαίσια των πραγματοποιημένων προσομοιώσεων.

Στο τέλος των προσομοιώσεων κρίθηκε πως η υλοποίηση είναι επαρκής, καλύπτοντας μεγάλο φάσμα περιπτώσεων και καταστάσεων οι οποίες λήφθησαν υπόψη καθώς το σύστημα ελέγχθηκε με μία σειρά από προσομοιώσεις οι οποίες επαλήθευσαν την λειτουργία του. Στο κεφάλαιο περιγράφηκαν λεπτομερώς έξι από αυτές τις δοκιμαστικές εκτελέσεις (προσομοιώσεις) του υλοποιημένου δικτύου πάνω στην πλατφόρμα CPN Tools ώστε να επαληθευτούν και να παρουσιαστούν οι δυνατότητες ανάλυσης του εργαλείου στα πλαίσια του ελέγχου και της διαχείρισης ροών εργασιών. Οι δοκιμές και τα αποτελέσματά τους παρουσιάστηκαν τόσο σε μορφή πινάκων όσο και σε μεγαλύτερη λεπτομέρεια με κείμενο. Αρχικά είχε προηγηθεί η λεπτομερής παρουσίαση της αρχικής κατάστασης των οντοτήτων του δικτύου.

Αξίζει φυσικά να σημειωθεί πως δεν αποκλείονται εξαιρέσεις (exceptions), καταστάσεις δηλαδή οι οποίες να μην λήφθηκαν υπόψη και κατά συνέπεια να μην καλύπτονται από την παρούσα υλοποίηση. Είναι λογικό εξάλλου πως για να ληφθούν ασφαλή συμπεράσματα απαιτούνται μερικές χιλιάδες προσομοιώσεις πάνω σε βιομηχανικά δεδομένα. Συνολικά, η υλοποίηση και τα αποτελέσματα από τις προσομοιώσεις κρίνονται ιδιαίτερα ικανοποιητικά.

Κεφάλαιο 9: Συμπεράσματα

Το κεφάλαιο περιλαμβάνει την περιγραφή των δυνατοτήτων και των περιορισμών του προτεινόμενου πλαισίου και της αντίστοιχης υλοποίησης. Το εργαλείο προσομοίωσης αναπτύχθηκε για να εξυπηρετήσει τις απαιτήσεις της ενσωμάτωσης των δικτύων εργασιών με μοντέλα των νοητικών διαδικασιών των εργαζομένων. Τα δίκτυα εργασιών μπορούν να δημιουργήσουν αρκετές μετρικές απόδοσης αλλά δεν μπορούν να αντιμετωπίσουν μεταβολές στον σχεδιασμό των εργασιών, όπως τροποποιήσεις στις εργασιακές διαδικασίες, στις προτεραιότητες των στόχων και στην κατανομή των εργασιών. Η ενσωμάτωση των δικτύων εργασιών με μοντέλα των νοητικών διαδικασιών των εργαζομένων επιτρέπει στους σχεδιαστές να χρησιμοποιούν αρκετές συστεμικές και ψυχολογικές μετρικές (δηλαδή, βαθμολογίες σχετικά με την διάρκεια και την ακρίβεια εκτέλεσης των εργασιών, δείκτες φόρτου εργασίας και κούρασης) καθώς και να μοντελοποιούν τον τρόπο με τον οποίο η απόδοση ενός συνόλου εργαζομένων μπορεί να προσαρμοστεί στις μεταβαλλόμενες συνθήκες (δηλαδή, πνευματική παρακολούθηση των υπό αναστολή εργασιών, αντιμετώπιση της διακοπής και ανακατανομής των εργασιών, και διαχείριση του φόρτου εργασίας). Με τον τρόπο αυτό, η προσαρμογή στις εργασιακές μεταβολές και η ανάκαμψη από ανεπιτυχείς προσπάθειες γίνεται η βάση του σχεδιασμού των εργασιών όσον αφορά την ασφάλεια και την παραγωγικότητα.

Ένα σημαντικό χαρακτηριστικό του εργαλείου προσομοίωσης είναι το γεγονός ότι η ανθρώπινη απόδοση εξετάζεται μέσα στο ευρύτερο πλαίσιο εργασίας. Για

παράδειγμα, η απόδοση μίας εργασίας επηρεάζεται μεταξύ άλλων από την κατάσταση άλλων εργασιών που βρίσκονται υπό λειτουργία (δηλαδή, υποψήφιες, υπό αναστολή και αποτυχημένες εργασίες) και την κατάσταση των νοητικών δραστηριοτήτων των εργαζομένων (δηλαδή, παρακολούθηση γεγονότων, πνευματική παρακολούθηση εργασιών, διαχείριση φόρτου εργασίας). Κατά συνέπεια, μπορούν να εξαχθούν συγκεκριμένες συστάσεις αναφορικά με την βελτίωση της παραγωγικότητας και της αξιοπιστίας μέσω των αλλαγών των απαιτήσεων των εργασιών, των στρατηγικών του φόρτου εργασίας, και της οργάνωσης των εργασιών. Το εργαλείο προσομοίωσης μπορεί να επεκταθεί περαιτέρω με στόχο την ικανοποίηση τριών ευρύτερων ζητημάτων: την εγκυρότητα των μοντέλων των νοητικών διαδικασιών, τους μηχανισμούς των μοντέλων των εργαζομένων που επηρεάζουν την ανθρώπινη αξιοπιστία, και την εγκυρότητα των πληροφοριών που χρησιμοποιούνται στην μοντελοποίηση των εργασιών.

Για να αποκτηθούν έγκυρα αποτελέσματα, τα μοντέλα των εργαζομένων θα πρέπει να είναι έγκυρα μοντέλα της αλληλεπίδρασης μεταξύ της νόησης και της εργασίας. Υπό αυτή την έννοια, τα μοντέλα της επιλογής και της οργάνωσης των εργασιών είναι προκαταρκτικά μοντέλα τα οποία δεν έχουν επικυρωθεί στο πλαίσιο πραγματικών εργασιακών σεναρίων. Κατά συνέπεια, κρίνεται αναγκαία η διεξαγωγή εμπειρικών μελετών για την επαλήθευση των χρησιμοποιημένων μοντέλων εργαζομένων στα πλαίσια του ελέγχου των διαδικασιών. Ωστόσο, αναμένεται ότι η προσομοίωση από μόνη της είναι ένα χρήσιμο εργαλείο για την σύγκριση των προσομοιωμένων μοντέλων εργαζομένων με τα εμπειρικά δεδομένα. Οι μελλοντικές επεκτάσεις των μοντέλων των νοητικών διαδικασιών των εργαζομένων του εργαλείου προσομοίωσης θα πρέπει να βρίσκονται προς αυτή την κατεύθυνση.

Ένα ακόμα ζήτημα αφορά τους μηχανισμούς του μοντέλου των νοητικών διαδικασιών των εργαζομένων που επηρεάζουν την ανθρώπινη αξιοπιστία. Στην παρούσα υλοποίηση, το εργαλείο προσομοίωσης υποθέτει πως τα ανθρώπινα

λάθη προκύπτουν ως αποτέλεσμα ενός συνόλου από παράγοντες (Performance Shaping Factors - PSF) όπως είναι ο φόρτος εργασίας, η κούραση και η παρακολούθηση των εργασιών οι οποίοι σε κάποιο σημείο υπερβαίνουν ένα συγκεκριμένο όριο. Ωστόσο η ανθρώπινη απόδοση είναι ευμετάβλητη και οι εργαζόμενοι μπορούν δυνητικά να αποφύγουν ένα σφάλμα μειώνοντας τις απαιτήσεις της απόδοσης (π.χ. εκτελώντας μία εργασία με μικρότερη ακρίβεια). Για τον λόγο αυτό, είναι δύσκολο να καθοριστεί αν υψηλά επίπεδα των παραγόντων που επηρεάζουν την απόδοση θα οδηγήσουν σε αύξηση των ανθρωπίνων λαθών ή μείωση της απόδοσης. Επιπλέον, δεν είναι δυνατόν να προσδιοριστεί το είδος των λαθών που μπορούν να προκύψουν από τις υψηλές εργασιακές απαιτήσεις. Το εργαλείο προσομοίωσης θα πρέπει να χρησιμοποιείται κυρίως για την σύγκριση διαφορετικών εργασιακών απαιτήσεων και να εντοπίζει ακραία σενάρια τα οποία να επηρεάζουν την απόδοση του συστήματος και όχι ως μία αναπαράσταση των νοητικών διαδικασιών ακολουθώντας ως πρότυπο την εργασία που περιέχεται στο [23].

Ένα ακόμα σημαντικό ζήτημα είναι τα δεδομένα και οι πληροφορίες των μοντέλων των εργαζομένων ή των αλγορίθμων που χρησιμοποιούνται. Στην παρούσα υλοποίηση, οι προσομοιώσεις χρησιμοποιούν περισσότερο αυθαίρετες τιμές για πολλές από τις μεταβλητές που αφορούν, για παράδειγμα, τις ιδιότητες των εργασιών (π.χ. ιδιότητες ακολουθιών ή χρονικές ιδιότητες), τις συνέπειες των σφαλμάτων ή τις προθεσμίες που συνδέονται με ανταγωνιστικές εργασίες. Αυτά τα δεδομένα είναι άμεσα εξαρτημένα από την φύση του τομέα των εργασιών και θα πρέπει να λαμβάνονται μέσα από παρατήρηση και πειραματισμό στα πλαίσια του εκάστοτε τομέα. Μία σειρά από εμπειρικές μελέτες απαιτούνται σε σκοπό την συλλογή δεδομένων τα οποία μπορούν να χρησιμοποιηθούν για την βαθμονόμηση των μεταβλητών και των ορίων των αλγορίθμων του μοντέλου των νοητικών διαδικασιών των εργαζομένων. Αυτό είναι απαραίτητο προκειμένου να χρησιμοποιηθούν οι

προσομοιώσεις για να γίνουν προβλέψεις σχετικά με την ανθρώπινη απόδοση σε νέες εργασίες.

Όταν επιλυθούν αυτά τα ζητήματα εγκυρότητας, το εργαλείο προσομοίωσης μπορεί να χρησιμοποιηθεί για τον επανασχεδιασμό εργασιών και ομάδων, με σκοπό την αύξηση της παραγωγικότητας και της αξιοπιστίας. Οι σχεδιαστές μπορούν να είναι σε θέση να επαληθεύσουν λειτουργικές διαδικασίες για ένα ευρύ φάσμα εργασιακών μεταβολών και να αναγνωρίσουν κρίσιμες εργασιακές απαιτήσεις οι οποίες θα μπορούσαν να οδηγήσουν σε αύξηση των ανθρωπίνων σφαλμάτων. Ο σχεδιασμός των εργασιών μπορεί επιπλέον να επωφεληθεί από το εργαλείο προσομοίωσης μέσω του επανασχεδιασμού των εργασιακών χαρακτηριστικών με σκοπό την μείωση του φόρτου εργασίας αλλάζοντας τις δυσμενείς συνέπειες που προκύπτουν από τις υπό αναστολή ή αποτυχημένες εργασίες. Τέλος, οι στρατηγικές για την διαχείριση του φόρτου εργασίας, η αλλαγή των προτεραιοτήτων των στόχων και ο επαναπρογραμματισμός των εργασιών μπορούν να διαμορφώσουν την βάση νέων σχημάτων εκπαίδευσης.

Αξίζει να σημειωθεί πως τα έγχρωμα δίκτυα Πέτρι, η γλώσσα προγραμματισμού ML καθώς και η πλατφόρμα ανάπτυξης CPN Tools υπήρξαν μία πολύ χρήσιμη πλατφόρμα μοντελοποίησης των νοητικών διαδικασιών και των ανθρωπίνων λαθών με σκοπό τον σχεδιασμό και την ασφαλή διαχείριση ροών εργασίας. Το σύνολο των προσομοιώσεων κρίνεται ιδιαίτερα πετυχημένο ωστόσο κρίνεται χρήσιμη η περαιτέρω επέκταση του υλοποιημένου δικτύου με επιπλέον αλγορίθμους για την πρόβλεψη και διαχείριση επιπλέον καταστάσεων καθώς και η χρήση δεδομένων προσομοίωσης από την βιομηχανία.

Δεν υπάρχει αμφιβολία πως οι προσπάθειες μοντελοποίησης με την χρήση υπολογιστών μπορούν να απαιτήσουν σημαντικούς πόρους όσον αφορά τα δεδομένα, την τεχνογνωσία και τον χρόνο που απαιτείται. Ωστόσο, τα σύγχρονα συστήματα απαιτούν έναν αυξανόμενο βαθμό ευελιξίας στον προγραμματισμό και προσαρμογή των εργασιακών διαδικασιών στις

μεταβαλλόμενες εργασιακές συνθήκες. Το προτεινόμενο εργαλείο προσομοίωσης προσφέρει μία πλατφόρμα για την μοντελοποίηση της απόδοσης ενός συστήματος και της ανθρώπινης αξιοπιστίας με μία παρόμοια γλώσσα μοντελοποίησης που μπορεί να εφαρμοστεί τόσο σε επίπεδο προσομοίωσης όσο και σε επίπεδο επίσημης ανάλυσης.

ΠΑΡΑΡΤΗΜΑ

Οι αρχικές τιμές των δεδομένων του δικτύου

```
val iTaskStateList =
[
	(1,50,1,None,Default,"0","0", Down, 0),
	(6,60,1,None,Default,"0","0", Down, 0),
	(7,40,2,None,Default,"0","0", Down, 0),
	(11,20,2,None,Default,"0","0", Down, 0),
	(12,70,3,None,Default,"0","0", Down, 0),
	(15,110,3,None,Default,"0","0", Down, 0)
];

val iManStateList =
[
	 (Supervisor,0,0),
	 (Assistant_A,0,0),
	 (Assistant_B,0,0),
	 (Worker_A,0,0),
	 (Worker_B,0,0),
	 (Worker_C,0,0)
];

val iManCharList =
[
		(1,[(Supervisor,45,50),(Assistant_A,70,30),(Assistant_B,65,35),(Worker_A,80,20),(Worker_B,50,40
),(Worker_C,40,60)]),
		(6,[(Supervisor,80,20),(Assistant_A,70,30),(Assistant_B,65,35),(Worker_A,50,40),(Worker_B,45,50
),(Worker_C,40,60)]),
		(7,[(Supervisor,70,10),(Assistant_A,60,20),(Assistant_B,55,25),(Worker_A,50,40),(Worker_B,45,50
),(Worker_C,40,60)]),
		(11,[(Supervisor,70,10),(Assistant_A,60,20),(Assistant_B,55,25),(Worker_A,50,40),(Worker_B,45,5
0),(Worker_C,40,60)]),
		(12,[(Supervisor,90,30),(Assistant_A,80,40),(Assistant_B,75,45),(Worker_A,50,40),(Worker_B,45,5
0),(Worker_C,40,60)]),
		(15,[(Supervisor,90,30),(Assistant_A,80,40),(Assistant_B,75,45),(Worker_A,50,40),(Worker_B,45,5
0),(Worker_C,40,60)])
];

val iTaskCharList =
[
	(1,1,10,10,10),
	(1,6,5,15,10),
	(2,7,15,5,10),
	(2,11,4,6,10),
	(3,12,8,7,10),
	(3,15,10,5,15)
];

val iTaskExtraList =
[
	(1,5,0),
	(6,10,0),
	(7,15,0),
	(11,20,0),
	(12,25,0),
	(15,30,0)
```

```
];

val iTaskConsequencesList =
[
        (1,10,5),
        (1,5,10),
        (1,5,15),
        (6,15,5),
        (6,5,20),
        (7,10,10),
        (7,10,15),
        (11,5,10),
        (11,10,5),
        (12,15,10),
        (15,20,10)
];
```

Συνάρτηση «AssignTasks» και οι βοηθητικές της συναρτήσεις

```
fun PADS_LT_Function (pad_A : PAD, pad_B : PAD) =
   if ((#2(pad_A)) div (#3(pad_A))) + ((#2(pad_A)) mod (#3(pad_A))) < ((#2(pad_B)) div (#3(pad_B)))
+ ((#2(pad_B)) mod (#3(pad_B)))
   then true
   else false

fun CheckSelectedOperator (person:Person,task:Task,msl:MSL,tcl:TCL) =
   let
      val ms = SearchMSL_ForPerson(person,msl)
      val tc = SearchTCL_ForTask(task,tcl)
   in
      if ( (#3(ms) + #3(tc) + #4(tc) + #5(tc)) > 150 )
      then false
      else true
   end;

fun ChooseStrategy (pads:PADs,ts:TaskState) =
   let
      val strategy = discrete(1,4)
      val first_pad = List.hd pads
      val pads_tail = List.tl pads
      val second_pad = List.hd pads_tail
   in
      case strategy of
            1 => 1`(#3(ts),#1(ts),Started,#1(first_pad),(#2(first_pad)) div 2,(#3(first_pad)) * 2)
         | 2 => 1`(#3(ts),#1(ts),Deferred,None,0,0)
         | 3 => 1`(#3(ts),#1(ts),Deferred,#1(first_pad),#2(first_pad),#3(first_pad))
         | 4 => 1`(#3(ts),#1(ts),Started,#1(second_pad),#2(second_pad),#3(second_pad))
   end;

fun AssignTasks (task:Task,mcl:MCL,tsl:TSL,msl:MSL,tcl:TCL) =
   let
      val mc = SearchMCL_ForTask(task,mcl)
      val ts = SearchTSL_ForTask(task,tsl)
      val pads = sort PADS_LT_Function (#2(mc))
      val pad = List.hd pads
   in
      if (CheckSelectedOperator (#1(pad),#1(ts),msl:MSL,tcl:TCL))
      then 1`(#3(ts),#1(ts),Started,#1(pad),#2(pad),#3(pad))
      else ChooseStrategy(pads,ts)
```

```
end;
```

Συνάρτηση «PrioritizeTasks» και οι βοηθητικές της συναρτήσεις

```
fun TaskExtra_LT_Function (te_A:TaskExtra,te_B:TaskExtra) =
   if #3(te_A) > #3(te_B)
   then true
   else false

fun FindTaskConsequences (task:Task,tcl:TaskConsequencesList,value:INT) =
   let
      val tcl_head = List.hd tcl
      val tcl_tail = List.tl tcl
   in
      if ((#1(tcl_head) = task) andalso (tcl_tail = []))
      then (value + (#2(tcl_head) * #3(tcl_head)))
      else if  ((#1(tcl_head) = task) andalso (tcl_tail <> []))
            then FindTaskConsequences(task,tcl_tail,(value + (#2(tcl_head) * #3(tcl_head))))
            else if  ((#1(tcl_head) <> task) andalso (tcl_tail = []))
                  then value
                  else FindTaskConsequences(task,tcl_tail,value)
   end;

fun TraverseAllTSL (tel:TEL,tcl:TaskConsequencesList, tsl_filtered:TSL,tel_temp:TEL) =
   let
      val tsl_head = List.hd tsl_filtered
      val tsl_tail = List.tl tsl_filtered
      val te = SearchTEL_ForTask(#1(tsl_head),tel)
      val tsl_head_consequences_cost = FindTaskConsequences(#1(tsl_head),tcl,0)
      val priority_index = #2(te) + tsl_head_consequences_cost
      val te_temp = (#1(te),#2(te),priority_index)
   in
      if (tsl_tail = [])
      then tel_temp ^^ [te_temp]
      else TraverseAllTSL(tel,tcl,tsl_tail,tel_temp ^^ [te_temp])
   end;

fun PrioritizeTasks (tel:TEL,tcl:TaskConsequencesList, tsl:TSL) =
   let
      val tsl_filtered = SearchTSL_ForStatus (Started,tsl,[])
      val tel_filtered = TraverseAllTSL(tel,tcl,tsl_filtered,[])
      val tel_sorted =  sort TaskExtra_LT_Function (tel_filtered)
      val tel_sorted_head = List.hd tel_sorted
   in
      #1(tel_sorted_head)
   end;
```

Συναρτήσεις αναζήτησης

```
fun SearchMCL_ForTask (task:Task,mcl:MCL) =
   let
      val mcl_head = List.hd mcl
      val mcl_tail = List.tl mcl
   in
      if (#1(mcl_head) = task)
      then mcl_head
      else SearchMCL_ForTask(task,mcl_tail)
   end;

fun SearchMSL_ForPerson (person:Person,msl:MSL) =
   let
      val msl_head = List.hd msl
      val msl_tail = List.tl msl
   in
      if (#1(msl_head) = person)
      then msl_head
      else SearchMSL_ForPerson(person,msl_tail)
   end;

fun SearchTCL_ForTask (task:Task,tcl:TCL) =
   let
      val tcl_head = List.hd tcl
      val tcl_tail = List.tl tcl
   in
      if (#2(tcl_head) = task)
      then tcl_head
      else SearchTCL_ForTask(task,tcl_tail)
   end;

fun SearchTSL_ForTask (task:Task,tsl:TSL) =
   let
      val tsl_head = List.hd tsl
      val tsl_tail = List.tl tsl
   in
      if (#1(tsl_head) = task)
      then tsl_head
      else SearchTSL_ForTask(task,tsl_tail)
   end;

fun SearchTSL_ForStatus (status:State,tsl:TSL,returnedTSL:TSL) =
   let
      val tsl_head = List.hd tsl
      val tsl_tail = List.tl tsl
   in
      if (#5(tsl_head) = status) andalso (tsl_tail = [])
      then (returnedTSL^^[tsl_head])
      else if (#5(tsl_head) = status) andalso (tsl_tail <> [])
           then SearchTSL_ForStatus(status,tsl_tail,(returnedTSL^^[tsl_head]))
           else if (#5(tsl_head) <> status) andalso (tsl_tail = [])
                then returnedTSL
                else SearchTSL_ForStatus(status,tsl_tail,returnedTSL)
   end;

fun SearchTEL_ForTask (task:Task,tel:TEL) =
   let
      val tel_head = List.hd tel
      val tel_tail = List.tl tel
   in
```

```
      if (#1(tel_head) = task)
      then tel_head
      else SearchTEL_ForTask(task,tel_tail)
   end;

fun SearchSTL_ForGoal (goal:Goal,stl:STL) =
   let
      val stl_head = List.hd stl
      val stl_tail = List.tl stl
   in
      if (#1(stl_head) = goal)
      then stl_head
      else SearchSTL_ForGoal(goal,stl_tail)
   end;
```

Συναρτήσεις ενημέρωσης και οι βοηθητικές τους συναρτήσεις

```
fun  UpdateTaskStateListForInitiation (ts:TaskState, tsl:TSL) =
   let
      val temp_tsl = rm ts tsl;
   in
      if (#6(ts) = "0")
       then   1`   (temp_tsl^^[(#1(ts),#2(ts),#3(ts),#4(ts),Started,ModelTime.toString(time()),#7(ts),   #8(ts),
#9(ts))])
      else 1` (temp_tsl^^[(#1(ts),#2(ts),#3(ts),#4(ts),Started,#6(ts),#7(ts), #8(ts), #9(ts))])
   end;

fun  SetGoalActiveInTaskStateList_Task (ts:TaskState, tsl:TSL) =
   let
      val ret_tsl = ins_new tsl (#1(ts),#2(ts),#3(ts),#4(ts),#5(ts),#6(ts),#7(ts), Up, #9(ts));
   in
      ret_tsl
   end;

fun  SetGoalActiveInTaskStateList (goal:Goal, tsl:TSL, returnedTSL:TSL) =
   let
      val tsl_head = List.hd tsl
      val tsl_tail = List.tl tsl
   in
      if (tsl_tail = [])
      then if ((#3(tsl_head) = goal) andalso (#8(tsl_head) = Down))
             then SetGoalActiveInTaskStateList_Task(tsl_head, returnedTSL)
             else returnedTSL^^[tsl_head]
      else if ((#3(tsl_head) = goal) andalso (#8(tsl_head) = Down))
             then  SetGoalActiveInTaskStateList(goal,  tsl_tail,  SetGoalActiveInTaskStateList_Task(tsl_head,
returnedTSL))
            else SetGoalActiveInTaskStateList(goal, tsl_tail, returnedTSL^^[tsl_head])
   end;

fun CheckIfTaskDown (task:Task, tsl:TSL) =
   let
      val ts = SearchTSL_ForTask(task, tsl);
   in
      if (#5(ts) <> Done)
      then tsl
      else SetGoalActiveInTaskStateList (#3(ts), tsl, [])
   end;
```

```
fun  UpdateTaskStateList_Flag_Up (task:Task, tsl:TSL) =
   let
      val ts = SearchTSL_ForTask(task, tsl);
   in
      if ((#8(ts) = Down) andalso (#1(ts) = 1 orelse #1(ts) = 7 orelse #1(ts) = 12) )
      then SetGoalActiveInTaskStateList(#3(ts), tsl, [])
      else if ((#8(ts) = Down) andalso (#1(ts) = 6 orelse #1(ts) = 11 orelse #1(ts) = 15) )
            then if (#1(ts) = 6)
                  then CheckIfTaskDown (1, tsl)
                  else if (#1(ts) = 11)
                        then CheckIfTaskDown (7, tsl)
                        else if (#1(ts) = 15)
                              then CheckIfTaskDown (12, tsl)
                              else empty
            else tsl
   end;

fun  SetGoalInactiveInTaskStateList_Task (original_ts:TaskState, ts:TaskState, tsl:TSL, state:State) =
   let
   in
      if ((state = Interrupted) orelse (state = Deferred) orelse (state = Failed) orelse (state = Upset))
      then if (#1(original_ts) = #1(ts))
             then  ins_new  tsl  (#1(ts),#2(ts),#3(ts),#4(ts),state,#6(ts),ModelTime.toString(time()),  Down,
#9(ts) + 1)
             else ins_new tsl (#1(ts),#2(ts),#3(ts),#4(ts),#5(ts),#6(ts),#7(ts), Down, #9(ts) + 1)
      else if (#1(original_ts) = #1(ts))
            then  ins_new  tsl  (#1(ts),#2(ts),#3(ts),#4(ts),state,#6(ts),ModelTime.toString(time()),  Down,
#9(ts))
            else ins_new tsl (#1(ts),#2(ts),#3(ts),#4(ts),#5(ts),#6(ts),#7(ts), Down, #9(ts))
   end;

fun  SetGoalInactiveInTaskStateList (ts:TaskState, tsl:TSL, state:State, returnedTSL:TSL) =
   let
      val tsl_head = List.hd tsl
      val tsl_tail = List.tl tsl
      val goal = #3(ts)
   in
      if (tsl_tail = [])
      then if (#3(tsl_head) = goal)
             then SetGoalInactiveInTaskStateList_Task(ts, tsl_head, returnedTSL, state)
             else returnedTSL^^[tsl_head]
      else if (#3(tsl_head) = goal)
             then  SetGoalInactiveInTaskStateList(ts,  tsl_tail,  state,  SetGoalInactiveInTaskStateList_Task(ts,
tsl_head, returnedTSL, state))
            else SetGoalInactiveInTaskStateList(ts, tsl_tail, state, returnedTSL^^[tsl_head])
   end;

fun  UpdateSuspendedTasksMemory (um_out:UM_OUT, stl:STL) =
   let
      val updatedList = ins_new stl (#1(um_out),#2(um_out));
   in
      if #3(um_out) = Interrupted orelse #3(um_out) = Deferred
      then 1`updatedList
      else 1`stl
   end;

fun  UpdateTaskStateList (um_out:UM_OUT, tsl:TSL) =
   let
      val ts = SearchTSL_ForTask(#2(um_out), tsl);
      val temp_tsl = rm ts tsl;
```

```
   in
      case #3(um_out) of
         Interrupted => SetGoalInactiveInTaskStateList(ts, tsl, Interrupted, [])
      | Deferred => SetGoalInactiveInTaskStateList(ts, tsl, Deferred, [])
      | Failed => SetGoalInactiveInTaskStateList(ts, tsl, Failed, [])
      | Upset => SetGoalInactiveInTaskStateList(ts, tsl, Upset, [])
      | _ => if (#1(ts) = 1 orelse #1(ts) = 7 orelse #1(ts) = 12)
                then ins_new temp_tsl (#1(ts),#2(ts),#3(ts),#4(ts),Done,#6(ts),ModelTime.toString(time()),
#8(ts), #9(ts))
             else SetGoalInactiveInTaskStateList(ts, tsl, Done, [])
   end;

fun  UpdateTaskStateListWithState (task:Task, tsl:TSL, state:State) =
   let
      val ts = SearchTSL_ForTask(task, tsl);
      val temp_tsl = rm ts tsl;
   in
      ins_new temp_tsl (#1(ts),#2(ts),#3(ts),#4(ts),state,#6(ts),#7(ts),#8(ts), #9(ts))
   end;

fun  UpdateTaskStateListForAssignedPerson (at:AT, tsl:TSL) =
   let
      val ts = SearchTSL_ForTask(#2(at), tsl);
      val temp_tsl = rm ts tsl;
   in
      1`ins_new temp_tsl (#1(ts),#2(ts),#3(ts),#4(at),#5(ts),#6(ts),#7(ts), #8(ts), #9(ts))
   end;

fun UpdateAssignedManStateList (msl:MSL, tcl:TCL, at:AT) =
   let
      val manState = SearchMSL_ForPerson(#4(at), msl);
      val tc = SearchTCL_ForTask(#2(at), tcl);
      val temp = rm manState msl;
       val updatedList = ins_new temp (#1(manState),#2(manState) + #6(at),#3(manState) + #3(tc) +
#4(tc) + #5(tc));
   in
      1` updatedList
   end;

fun UpdateFinishedManStateList (um_out:UM_OUT, msl:MSL, tcl:TCL, tsl:TSL) =
   let
      val ts = SearchTSL_ForTask(#2(um_out), tsl)
      val tc = SearchTCL_ForTask(#2(um_out), tcl);
      val manState = SearchMSL_ForPerson(#4(ts),msl)
      val temp = rm manState msl;
       val updatedList = ins_new temp (#1(manState),#2(manState),#3(manState) - #3(tc) - #4(tc) -
#5(tc));
   in
      1`updatedList
   end;
```

Διάφορες συναρτήσεις και οι βοηθητικές τους συναρτήσεις

```
fun InitiateGoal (tsl:TSL, goalsInProcess:Goals, initiatedGoals:Goals) =
   if List.null tsl
   then empty
   else
      let
         val tsl_head = List.hd tsl
         val tsl_tail = List.tl tsl
      in
         if List.null tsl_tail
          then if (#5(tsl_head) = Default andalso not(mem goalsInProcess (#3(tsl_head))) andalso not(mem
initiatedGoals (#3(tsl_head))) )
                then ins_new initiatedGoals (#3(tsl_head))
                else initiatedGoals
          else if (#5(tsl_head) = Default andalso not(mem goalsInProcess (#3(tsl_head))) andalso not(mem
initiatedGoals (#3(tsl_head))) )
               then ins_new initiatedGoals (#3(tsl_head))
               else InitiateGoal(tsl_tail,goalsInProcess, initiatedGoals)
   end;

fun FilterTSL_ForGoal(goal:Goal,tsl:TSL,returnedTSL:TSL) =
   let
      val tsl_head = List.hd tsl
      val tsl_tail = List.tl tsl
   in
      if (#3(tsl_head) = goal) andalso (tsl_tail = [])
      then (returnedTSL^^[tsl_head])
      else if (#3(tsl_head) = goal) andalso (tsl_tail <> [])
            then FilterTSL_ForGoal(goal,tsl_tail,(returnedTSL^^[tsl_head]))
            else if (#3(tsl_head) <> goal) andalso (tsl_tail = [])
                  then returnedTSL
                  else FilterTSL_ForGoal(goal,tsl_tail,returnedTSL)
   end;

fun IsGoalCompleted (goal:Goal,tsl:TSL) =
   let
      val tsl_filtered = FilterTSL_ForGoal(goal,tsl,[])
      val tsl_head = List.hd tsl_filtered
      val tsl_tail = List.tl tsl_filtered
   in
      if (#5(tsl_head) <> Done)
      then false
      else if (tsl_tail = [])
            then true
            else IsGoalCompleted(goal,tsl_tail)
   end;

fun IsGoalFatallyTerminated (goal:Goal,tsl:TSL) =
   let
      val tsl_filtered = FilterTSL_ForGoal(goal,tsl,[])
      val tsl_head = List.hd tsl_filtered
      val tsl_tail = List.tl tsl_filtered
   in
      if (#9(tsl_head) > 3)
      then true
      else if (tsl_tail = [])
            then false
            else IsGoalFatallyTerminated(goal,tsl_tail)
   end;
```

```
fun IsGoalFailedOrUpset (goal:Goal,tsl:TSL) =
   let
      val tsl_filtered = FilterTSL_ForGoal(goal,tsl,[])
      val tsl_head = List.hd tsl_filtered
      val tsl_tail = List.tl tsl_filtered
   in
      if ((#5(tsl_head) = Failed) orelse (#5(tsl_head) = Upset))
      then true
      else if (tsl_tail = [])
            then false
            else IsGoalFailedOrUpset(goal,tsl_tail)
   end;

fun ProduceUserModelOut (at:AT) =
   let
      val strategy = discrete(1,10)
   in
      case strategy of
         1 => 1`(#1(at),#2(at),Interrupted)
       | 2 => 1`(#1(at),#2(at),Deferred)
       | 3 => 1`(#1(at),#2(at),Failed)
       | 4 => 1`(#1(at),#2(at),Upset)
       | _ => 1`(#1(at),#2(at),Done)
   end;

fun SuspendedTasksContainGoal (goal:Goal,stl:STL) =
   if List.null stl
   then false
   else
      let
         val stl_head = List.hd stl
         val stl_tail = List.tl stl
      in
          if(stl_tail = [] andalso #1(stl_head) <> goal)
          then false
          else if (#1(stl_head) = goal)
                then true
                else SuspendedTasksContainGoal(goal, stl_tail)
   end;
```

ΒΙΒΛΙΟΓΡΑΦΙΑ

[1] Kontogiannis Tom (2005), Integration of task networks and cognitive user models using coloured Petri nets and its application to job design for safety and productivity, Cogn Tech Work, Springer-Verlag London Limited.

[2] Kletz TA (1991), An engineer's view of human error, 2nd edition, Institution of Chemical Engineers, Rugby.

[3] Hollnagel E (1993), Human reliability: context and control, Academic, London.

[4] Embrey DE, Kontogiannis T, Green M (1994), Guidelines for preventing human error in process safety, Center for Chemical process safety, American Institute of Chemical Engineers, New York.

[5] Kjellen U (2000), Prevention of accidents through experience feedback, Taylor & Francis, London.

[6] Kirwan B, Ainsworth LK (1992), A guide to task analysis, Taylor & Francis, London.

[7] Stanton N, Young MS (1999), A guide to methodology in ergonomics, Taylor & Francis, London.

[8] Laughery KR, Corker KM (1997), Computer modeling and simulation of human-system performance. In: Salvendy G (ed) Handbook of human factors and ergonomics, 2nd edition, Wiley, New York, pp 1375-1408.

[9] John BE, Kieras DE (1996), The GOMS family of user interfaces analysis techniques: comparison and constructs. ACM Trans Comput Hum Interact 3(4):20-351.

[10] Mori G, Paterno F, Santoro C (2002), CTTE: support for developing and analyzing task models for interactive system design, IEEE Trans Software Eng 28(9):1-17.

[11] Uhr H (2003) TOMBOLA: simulation and user-specific presentation of executable task models. In: Jacko J, Staphanidis C (eds), Human computer

interaction: theory and practice (Part I), Lawrence Erlbaum Associates, Mahwah, pp 263-267.

[12] Abed M, Tabary D, Kolski C (2003), Using formal specification techniques for modeling tasks and generation of HCI specifications. In: Diaper D, Stanton N (eds), The handbook of task analysis for human computer interaction, Chapter 30, Lawrence Erlbaum Associates, Mahwah, pp 503-529.

[13] Tarby JC, Barthet MF (1996) The DIANE+ method. In: Vanderdonckt J (ed), Proceedings of the 2nd international workshop on computer aided design of user interfaces (CADUI'96). Presses Universitaires de Namur, Namur, vol 15, pp 95-120.

[14] Lebiere C, Biefeld F, Archer S, Allender L, Kelley TD (2002) IMPRINT/ACT-R: integration of a task network architecture with a cognitive architecture and its application to human error modeling. In: Chini MJ (ed) Military, government and aerospace simulation, Society for Modelling and Simulation International, San Diego 34(3):13-19.

[15] Cacciabue PC, Mauri C, Owen D (2003) The development of a model and simulation of an aviation maintenance technician performance. Cogn Technol Work 5:229-247.

[16] Lindsay P, Connelly S (2002) Modelling erroneous operator behaviors for an air traffic control tasks. In: Grundy J, Calder P (eds) Third Australian user interface conference, Australian Computer Society Inc, vol 7, pp 43-54.

[17] Hendy KC, Farrell PS (1997) Implementing a model of information processing in a task network simulation environment (DCIEM 97-R-71). Defense and Civil Institute of Environmental Medicine, Ontario, Canada.

[18] Lacaze X, Palanque P, Bastide R (2002) Performance evaluations as a tool for quantitative assessment of complexity of interactive systems. Lecture Notes in Computer Science, Springer Verlag, No 2545, pp 208-222.

[19] Palanque P, Bastide R (1997) Synergistic modelling of tasks, users and systems using formal specification techniques. Interact Comput 9(2):129-153.

[20] Card SK, Moran TP, Newell A (1983) The psychology of human computer interaction. Lawrence Erlbaum Associates, Mahwah.

[21] Rauterberg M (1993) AMME: an automatic mental model evaluation to analyze user behavior traced in a finite, discrete state space. Ergonomics 36:1369-1380.

[22] Moher T, Dirda V (1995) Revising mental models to accommodate expectation failures in human-computer dialogues. In: 2nd DSVIS workshop on design, specification and verification of interactive systems, Speringer, Berlin Heidelberg New York.

[23] Yoshikawa H, Nakagawa T, Nakatani Y, Furutra T, Hasegawa A (1997) Development of an analysis support system for man-machine system design information. Reliability Eng Syst Saf 5(3):417-425.

[24] Stutz P, Onken R (2001) Adaptive pilot modelling for cockpit crew assistance: concept, realization and results. In: Proceedings of the 8th conference of cognitive science approaches to process control. The cognitive work process: automation and interaction. Munich, 24-26 September 2001.

[25] Blom HAP, Daams J, Nijhuis HB (2000) Human cognition modelling in ATM safety assessment. In: Proceedings of the 3rd USA/Europe AirTraffic Management R & D Seminar, Napoli, 13-16 June 2000.

[26] Handley HAH, Levis AH (2003) Modelling for future command and control architectures. In: Proceedings of the 8th international command and control research and technology symposium, National Defense University, Washington DC, 17-19 June 2003.

[27] Levenson N, Stolzy (1987) Safety analysis using Petri nets. IEE Trans Software Eng 13:386-397.

[28] Jensen K (1997a) Coloured Petri nets: basic concepts, analysis methods and practical use. Vol 1: basic concepts. Monographs is theoretical computer science, Springer, Berlin Heidelberg New York.

[29] Jensen K (1997b) Coloured Petri nets: basic concepts, analysis methods and practical use. Vol 2: analysis methods. Monographs is theoretical computer science, Springer, Berlin Heidelberg New York.

[30] Cacciabue PC (1998) Modelling and simulation of human behavior in system analysis. Springer, Berlin Heidelberg New York.

[31] Pew RW, Mavor AS (1998) Modelling human and organizational behavior: application to military simulations. National Academy Press, Washington, DC.

[32] Zachary W, Campbell G, Laughery R, Glenn F (1998) Application of human modelling technology to the design, operation and evaluation of complex systems, Technical Memo No 980727.9705. CHI Systems, Inc., Lower Gwynedd, PA.

[33] Schlick C, Reuth R, Luczak H (2002) A comparative simulation study of work processes in autonomous production cells. Hum Factors Ergon Manuf 12(1):31-54.

[34] Freed M (1998) Managing multiple tasks in complex, dynamic environments. In: Proceedings of the 15th national conference on artificial intelligence, AAAI Press/MIT Press, Madison/Wisconsin, pp. 921-927.

[35] Freed M (2000) Reactive prioritization. In: Proceedings of the 2nd NASA international workshop on planning and scheduling in space, March 2000, San Francisco.

[36] Lockett JF (1997) Task network modelling of human workload coping strategies. In: Smith MJ, Salvendy G, Koubek RJ (eds) Design of computing systems: social and ergonomic considerations, proceedings of the seventh international conference on human-computer interaction, (HCI International '97). Elsevier, San Francisco vol 2, pp 71-74.

[37] Shepherd A (2001) Hierarchical task analysis. Taylor & Francis, London.

[38] Kontogiannis T (2003) A Petri-net based approach for ergonomic task analysis and modelling with emphasis on adaptation to system changes. Saf Sci 41:803-835.

[39] van der Aalst WMP, Hofstede AHM (2002) Workflow patterns: on the expressive power of (Petri nets based) workflow languages. In: Jensen K (eds) Proceedings of the fourth workshop on the practical use of coloured Petri nets and CPN tools, Department of Computer Science, University of Aarhus, Denmark, vol 560, ppl-20.

[40] Handley HAH, Zaidi ZR, Levis A (1999) The use of simulation models in model-driven experiments. Syst Eng 2(2): 108-128.

[41] Petrucci L, Kristensen LM, Billington J, Qureshi Z (2002) Towards formal specification and analysis of avionics mission systems: In: Proceedings of conference in research and practice in information technology, Workshop on formal methods applied to defense systems, Australian Computer Society, vol 2, pp 95-104.

[42] Definition of Petri Nets on Wikipedia: http://en.wikipedia.org/wiki/Petri_net.

[43] Carl Adam Petri and Wolfgang Reisig (2008) Petri net. Scholarpedia, 3(4):6477.

[44] Definition of Standard ML on Wikipedia: http://en.wikipedia.org/wiki/Standard_ML.

[45] Milner, R.; M. Tofte, R. Harper and D. MacQueen. (1997). The Definition of Standard ML (Revised). MIT Press. ISBN 0-262-63181-4.

[46] Colored Petri Nets, October 10 2007, by Mark C. Chu-Carroll: http://scienceblogs.com/goodmath/2007/10/colored_petri_nets.php.

[47] CPN Tools Homepage: http://cpntools.org/.

[48] Coloured Petri Nets, Modeling and Validation of Concurrent Systems, Chapter 7: State Spaces and Behavioral Properties, Kurt Jensen & Lars Michael Kristensen, Aarhus University.

[49] Altman EM, Trafton JG (2002) Memory for goals: an activation-based model. Cogn Sci 26:39–83.

[50] Lovett MC, Daily LZ, Reder LM (2000) A source activation theory of working memory: cross task prediction of performance in ACT-R. J Cogn Syst Res 1:99–118.

[51] Kristensen LM, Mitchell B, Zhang L, Billington J (2002) Modeling and initial analysis of operational planning processes using colored Petri nets. In: Proceedings of workshop on formal methods applied to defense systems, conference in research and practice in information technology, Australian Computer Society, vol 12, pp105–114.