

Μοντελοποίηση και Ανάλυση
Θεμελιωδών Στοιχείων Συστημάτων
Παραγωγής
με χρήση Έγχρωμων Δικτύων Petri

Πέτρος Γαβαλάκης
Τμήμα Μηχανικών Παραγωγής και Διοίκησης
Πολυτεχνείο Κρήτης
Χανιά

Αύγουστος 2006

Περιεχόμενα

1	Εισαγωγή	7
1.1	Εισαγωγή	8
1.1.1	Χρήσεις των δικτύων Petri στο χώρο της παραγωγής	9
1.1.2	Δομή της παρούσας εργασίας	11
2	Απλά και έγχρωμα δίκτυα Petri	13
2.1	Απλά δίκτυα Petri	14
2.1.1	Μαθηματική περιγραφή των απλών δικτύων Petri	15
2.1.2	Κανόνες εκτέλεσης των απλών δικτύων Petri	16
2.1.3	Επεκτάσεις των απλών δικτύων Petri	17
2.2	Τα έγχρωμα δίκτυα Petri	18
2.2.1	Βασικά στοιχεία των έγχρωμων δικτύων Petri	19
2.3	Ιδιότητες και ανάλυση των έγχρωμων δικτύων Petri	22
2.4	Αποτίμηση συμπεριφοράς, ορθότητας και επίδοσης	25
2.4.1	Ανάλυση επίδοσης των δικτύων CP	25
2.4.2	Ανάλυση συμπεριφοράς και ορθότητας των δικτύων CP	26
2.4.3	Το εργαλείο προσομοίωσης των δικτύων CP	26
2.4.4	Γραφική επισκόπηση του χώρου καταστάσεων	27
2.4.5	Αναφορά γράφου καλυψιμότητας	29
3	Μοντελοποίηση συστημάτων παραγωγής	33
3.1	Στοιχεία ενός συστήματος παραγωγής	34
3.1.1	Μοντελοποίηση συστημάτων παραγωγής με δίκτυα Petri	36
3.1.2	Η φυσική σημασία των ιδιοτήτων των δικτύων Petri στα συστήματα παραγωγής	39

4	Μοντελοποίηση θεμελιωδών στοιχείων	43
4.1	Μοντελοποίηση διαδικασίας παραγωγής	44
4.2	Μοντελοποίηση μηχανών	47
4.2.1	Αποκλειστικές μηχανές	47
4.2.2	Μηχανές με πολλαπλές λειτουργίες	50
4.2.3	Μηχανές με παράλληλες λειτουργίες	57
4.2.4	Μηχανές συναρμολόγησης	67
4.2.5	Μηχανές αποσυναρμολόγησης	69
4.2.6	Μηχανές που είναι δυνατόν να χαλάσουν	69
4.3	Μοντελοποίηση συστημάτων μεταφοράς	69
4.3.1	Οχήματα μεταφοράς	69
4.3.2	Ιμάντες (ή ταινίες) μεταφοράς	72
4.3.3	Ρομποτικοί βραχίονες	75
4.4	Μοντελοποίηση συστημάτων αποθήκευσης	77
4.5	Μοντελοποίηση εργαλειοθηκών	78
5	Μοντελοποίηση σύνθετων συστημάτων	81
5.1	Ιεραρχική αποσύνθεση	82
5.1.1	Ιεραρχικά έγχρωμα δίκτυα Petri	82
5.1.2	Παράδειγμα ιεραρχικού δικτύου	83
6	Συμπεράσματα και μελλοντικές επεκτάσεις	89
A'	Χρήση του εργαλείου υλοποίησης δικτύων	93
A'.1	Εγκατάσταση και ρυθμίσεις του εργαλείου Design/CPN	93
A'.1.1	Δημιουργία νέου δικτύου (page)	94
A'.1.2	Καθορισμός κύριας σελίδας	95
A'.1.3	Το μενού CPN	96
A'.1.4	Δημιουργία του κεντρικού σημείου δηλώσεων	97
A'.1.5	Προσθήκη θέσεων και μεταβάσεων	97
A'.1.6	Εισαγωγή τόξων	99
A'.1.7	Αρχική σήμανση	101
A'.1.8	Έλεγχος σύνταξης	102
A'.1.9	Προσομοίωση του δικτύου	102

Α'.1.10 Επαναφορά στην αρχική κατάσταση	105
Α'.1.11 Κατάσταση εξαγωγής γράφου καλυψιμότητας	105
Α'.1.12 Αποθήκευση του μοντέλου CPN	107

Κεφάλαιο 1

Εισαγωγή

1.1 Εισαγωγή

Τα δίκτυα Petri (Petri Nets, PN) γεννήθηκαν από τη διδακτορική διατριβή [17] του Carl Adam Petri με αντικείμενο την επικοινωνία μεταξύ αυτόματων μηχανών, το 1962. Χρησιμοποιώντας ένα δίκτυο, περιέγραψε την τυπική σχέση μεταξύ των γεγονότων σε ένα σύστημα υπολογιστών. Στην πορεία ανακαλύφθηκε ότι τα δίκτυα Petri είναι ένα ισχυρό εργαλείο στην περιγραφή συστημάτων που εξαρτώνται από γεγονότα (event-driven systems), μιας και σε αντίθεση με τις περισσότερες γλώσσες αναπαράστασης, τα δίκτυα Petri την ίδια στιγμή δίνουν σημασία τόσο στην κατάσταση του συστήματος όσο και στις διάφορες ενέργειες. Τα συστήματα που βασίζονται σε γεγονότα μπορεί να είναι ασύγχρονα, να περιέχουν σειριακές ή παράλληλες εργασίες, να περιλαμβάνουν συγκρούσεις, αμοιβαίο αποκλεισμό και εν γένει να μην είναι ντετερμινιστικά. Τέτοια συστήματα χαρακτηρίζονται ως Δυναμικά Συστήματα Διακριτών Γεγονότων (Discrete Event Dynamic Systems, DEDES).

Η δυνατότητα γραφικής απεικόνισης των δικτύων Petri και το ισχυρό μαθηματικό υπόβαθρό τους τα βοήθησε να αποτελέσουν βασικό εργαλείο για πολλές εργασίες σε προβλήματα στο πεδίο των μηχανικών είτε αυτά ήταν η μοντελοποίηση και η σύνθεση μεθόδων για τον έλεγχο συστημάτων κατεργασιών μέσω διακριτών γεγονότων είτε η γενικότερη χρήση τους στην ανάλυση, την αποτίμηση των επιδόσεων και τον έλεγχο συστημάτων.

Η ανάπτυξη των *έγχρωμων* δικτύων Petri προέκυψε από την επιθυμία να αναπτυχθεί μια γλώσσα μοντελοποίησης τόσο θεωρητικά θεμελιωμένης όσο και πρακτικά ευέλικτης ώστε να χρησιμοποιηθεί για τα συστήματα του μεγέθους και της πολυπλοκότητας που συναντώνται στη βιομηχανία. Για να επιτευχθεί κάτι τέτοιο, έγινε προσπάθεια να συνδυαστεί η δύναμη αναπαράστασης των απλών δικτύων Petri με τις δυνατότητες των γλωσσών προγραμματισμού. Τα πρώτα δίνουν τη δυνατότητα της περιγραφή συγχρονισμού των ταυτόχρονων διεργασιών που παρατηρούνται στην πράξη ενώ οι γλώσσες προγραμματισμού παρέχουν τα μέσα για να καθορίζονται οι τύποι των στοιχείων του συστήματος και να χειρίζονται οι τιμές των στοιχείων αυτών.

Μέσω των εμπλουτισμένων μορφών των δικτύων Petri (π.χ. χρονικά, στοχαστικά, *έγχρωμα* δίκτυα), οι εφαρμογές τους επεκτάθηκαν σε πεδία όπως η σχεδίαση και ο έλεγχος καλής λειτουργίας δικτυακών πρωτοκόλλων, η μοντελοποίηση της ε-

πικοινωνίας ανθρώπου - υπολογιστή και άλλων διαδραστικών συστημάτων, η αναπαράσταση επιχειρησιακών διαδικασιών (π.χ. ροής εργασιών), η εργονομία, η επιστήμη υπολογιστών (π.χ. στη μοντελοποίηση ταυτόχρονων διεργασιών και αποκλεισμού, στην επαναχρησιμοποίηση τμημάτων κώδικα, στην επικοινωνία συστημάτων πολλαπλών πρακτόρων, στα κατανεμημένα συστήματα) κ.α. ¹

1.1.1 Χρήσεις των δικτύων Petri στο χώρο της παραγωγής

Στον ευρύτερο τομέα της αυτοματοποίησης της παραγωγής, η χρήση των δικτύων Petri (είτε με την αρχική τους μορφή είτε ως έγχρωμα δίκτυα) μπορεί να συνοψιστεί ως εξής:

1. Το αρχικό ενδιαφέρον ξεκίνησε από την ανάγκη του προσδιορισμού και της μοντελοποίησης συστημάτων παραγωγής. Σε πρώτη φάση, τα απλά δίκτυα Petri χρησιμοποιήθηκαν για την αναπαράσταση απλών γραμμών παραγωγής με ουρές, μηχανουργείων και αυτοματοποιημένων συστημάτων παραγωγής ενώ στη συνέχεια χρησιμοποιήθηκαν για τη μοντελοποίηση ευέλικτων συστημάτων παραγωγής, αυτοματοποιημένων γραμμών συναρμολόγησης και συστημάτων με κοινή χρήση πόρων με στόχο την καλύτερη διαχείρισή τους. Η εργασία προχώρησε στην προσπάθεια προσδιορισμού και μοντελοποίησης συστημάτων παραγωγής μεγάλης κλίμακας κάτω από διαφορετικές επιχειρησιακές συνθήκες.
2. Αργότερα, η έρευνα προσανατολίστηκε στην εξέταση των ποιοτικών χαρακτηριστικών των μοντέλων με δίκτυα Petri για συστήματα παραγωγής. Για παράδειγμα, η ανάλυση της προσβασιμότητας (reachability analysis) εξετάζει αν το σύστημα μπορεί να φτάσει σε μια συγκεκριμένη κατάσταση. Επιθυμητές ακολουθίες γεγονότων ελέγχονται και επικυρώνονται (verification analysis) σύμφωνα με τις ανάγκες του συστήματος. Άλλες ιδιότητες των δικτύων Petri χρησιμοποιούνται για να εξεταστεί η σταθερότητα των συστημάτων DEDS, η

¹Ένας πιο πλήρης κατάλογος εφαρμογών μεγάλης κλίμακας των εμπλουτισμένων μορφών των δικτύων Petri μπορεί να βρεθεί στο [3].

επαναληπτική συμπεριφορά αλλά και η εξάλειψη των αδιεξόδων στη λειτουργία τους. Ειδικά οι τεχνικές αποφυγής των αδιεξόδων και η εκτίμηση τους αποτελεί ακόμη και σήμερα ένα ιδιαίτερο πλαίσιο έρευνας.

3. Όταν οι χρονικές ή ποσοτικές «επιδόσεις» του συστήματος προς μοντελοποίηση ήταν σημαντικές, τότε τα χρονικά (*timed*) δίκτυα *Petri* χρησιμοποιήθηκαν για να προκύψει ο χρόνος του κύκλου λειτουργίας για συστήματα που επαναλαμβάνονται και με γεγονότα που συμβαίνουν ταυτόχρονα. Από την άλλη, για να αντιμετωπιστεί η στοχαστική φύση πολλών ενεργειών στην παραγωγή, τα στοχαστικά *PN* χρησιμοποιήθηκαν ώστε να προκύψουν οι ρυθμοί παραγωγής και οι υπόλοιπες παράμετροι του συστήματος. Η αντιμετώπιση των ζητημάτων αυτών συνήθως στηρίζεται σε κατασκευή (εκπαίδευση - εύρεση κατάλληλων παραμέτρων) αλυσίδων Markov κάτι που μπορεί να αποφευχθεί χάρη στους αλγόριθμους μετατροπής των υπάρχοντων μοντέλων Markov σε στοχαστικά δίκτυα *Petri*.
4. Σε προβλήματα με πάρα πολλές καταστάσεις ή εκεί όπου το φυσικό σύστημα δεν υπάγεται σε μαθηματική ανάλυση, τότε χρειάζεται προσομοίωση για να είναι δυνατή η εξαγωγή τόσο των ποιοτικών όσο και των ποσοτικών χαρακτηριστικών. Ευτυχώς, τα μοντέλα *PN* μπορούν εύκολα να αξιοποιηθούν για προσομοίωση πολύπλοκων διακριτών γεγονότων. Πολλά πακέτα λογισμικού προσομοίωσης στηρίζονται σε δίκτυα *Petri*. Η χρήση τεχνικών προσομοίωσης για την εκτίμηση ευέλικτων συστημάτων παραγωγής και ρομποτικών συστημάτων συναρμολόγησης επιρρεπών σε λάθη είναι μόνο μερικοί τομείς όπου χρησιμοποιήθηκαν οι δυνατότητες προσομοίωσης των μοντέλων με δίκτυα *Petri*.
5. Οι ελεγκτές προγραμματιζόμενης λογικής (*Programmable Logic Controllers, PLCs*) χρησιμοποιούνται ευρέως για τον έλεγχο βιομηχανικών αυτοματοποιημένων συστημάτων. Η σχεδίασή τους όμως στηρίζεται στα λογικά διαγράμματα τύπου σκάλας (*ladder logic diagrams*), τα οποία είναι γενικά δύσκολο να αποσφαλματωθούν. Παρατηρήθηκε ότι ένας ελεγκτής *PLC* μπορεί να μετατραπεί σε ένα δίκτυο *Petri* και αντίστροφα. Στις περισσότερες περιπτώσεις, σε τέτοια μοντέλα *Petri* πρέπει να ενσωματωθούν επιπρόσθετες πληροφορίες που

αφορούν σε σήματα πραγματικού χρόνου (real-time signals). Τα πλεονεκτήματα των δικτύων Petri αφορούν στην σχετική ευκολία τους να παραστήσουν και να δώσουν την δυνατότητα για μετατροπή της λογικής του ελεγκτή, αλλά και στην εν γένει δυναμικότητά τους για περαιτέρω μαθηματική ανάλυση και γραφική προσομοίωση κατά τη διαδικασία ελέγχου της αξιοπιστίας του μοντέλου. Είναι εύκολο να παρατηρηθεί ότι η πολυπλοκότητα στα γραφήματα των δικτύων Petri μεγαλώνει λιγότερο από ότι αυτή των διαγραμμάτων τύπου σκάλας αναλογικά με την αύξηση της πολυπλοκότητας του συστήματος. Συμπερασματικά, τα δίκτυα Petri προτιμήθηκαν έναντι των διαγραμμάτων τύπου σκάλας για την δημιουργία ελεγκτών προγραμματιζόμενης λογικής.

6. Η δυνατότητα μαθηματικής αναπαράστασης επιτρέπει στους σχεδιαστές να χρησιμοποιήσουν τα δίκτυα Petri για την δημιουργία ενός πρωτότυπου ενός συστήματος ελέγχου κατεργασιών ή ελέγχου διακριτών γεγονότων σε σύντομο χρονικό διάστημα (rapid prototyping). Ολόκληρα εικονικά εργοστάσια μπορούν να υλοποιηθούν μέσω της γραφικής αναπαράστασης των δικτύων Petri. Ο σταδιακή δοκιμή και υλοποίηση μπορεί να επιτευχθεί συνδέοντας τον πραγματικό εξοπλισμό σε ένα σύστημα στηριζόμενο σε δίκτυα Petri, επιτυγχάνοντας μείωση του χρόνου σχεδίασης και ανάπτυξης.
7. Τέλος, τα δίκτυα Petri έχουν συνδυασθεί και με άλλες προσεγγίσεις ώστε να είναι δυνατή η επίτευξη διαφορετικών σκοπών σε θέματα που αφορούν στον σχεδιασμό διεργασιών (process planning) και στο χρονικό τους προγραμματισμό (scheduling), στον ευφυή έλεγχο (intelligent control) αλλά και στη δημιουργία εμπειρων συστημάτων (expert systems).

1.1.2 Δομή της παρούσας εργασίας

Στην εργασία αυτή επικεντρώνουμε το ενδιαφέρον μας στη μοντελοποίηση των θεμελιωδών στοιχείων ενός ευέλικτου συστήματος παραγωγής με τη βοήθεια των *έγχρωμων* χρονικών δικτύων Petri ενώ γίνεται και σύγκριση των μοντέλων αυτών με τα αντίστοιχα μοντέλα απλών δικτύων Petri μέσω της ενδεικτικής παρουσίασης μοντέλων και των δύο τύπων για κάποια από τα θεμελιώδη στοιχεία.

Στο κεφάλαιο 2 γίνεται μια εισαγωγή στα απλά και έγχρωμα δίκτυα Petri και, μαζί με το μαθηματικό ορισμό τους, παρουσιάζονται τα βασικά χαρακτηριστικά και οι ιδιότητές τους. Στο κεφάλαιο 3 αναλύεται η έννοια των ευέλικτων συστημάτων παραγωγής και οι κυρίαρχες προσεγγίσεις αναπαράστασής τους ενώ διακρίνονται τα βασικά τους υποσυστήματα. Στη συνέχεια, στο κεφάλαιο 4 παρουσιάζονται και αναλύονται τα μοντέλα των θεμελιωδών στοιχείων ενός ευέλικτου συστήματος παραγωγής, για να ακολουθήσει στο κεφάλαιο 5 ένα ενδεικτικό παράδειγμα σύνθεσης ενός πιο πολύπλοκου μοντέλου βασιζόμενο στα θεμελιώδη στοιχεία. Τέλος, στο κεφάλαιο 6 παρουσιάζονται συνοπτικά τα συμπεράσματα και προτείνονται διάφορες προοπτικές επέκτασης της παρούσας προσέγγισης.

Κεφάλαιο 2

Απλά και έγχρωμα δίκτυα Petri

2.1 Απλά δίκτυα Petri

Εάν φανταστούμε ένα οποιοδήποτε φυσικό δίκτυο, βρίσκουμε δύο θεμελιώδη στοιχεία: τους κόμβους και τις ζεύξεις μεταξύ τους. Τόσο οι κόμβοι όσο και οι ζεύξεις παίζουν διαφορετικό ρόλο στο δίκτυο. Οποιαδήποτε αλλαγή σε οποιοδήποτε στοιχείο έχει ως αποτέλεσμα τελείως διαφορετικά δίκτυα ως προς τις ιδιότητές τους.

Ένα απλό (ή σύννηθες) δίκτυο Petri περιέχει κόμβους δύο ειδών: τις θέσεις (places) και τις μεταβάσεις (transitions) γι' αυτό και αναφέρονται και ως δίκτυα καταστάσεων/ μεταβάσεων (place / transition nets - P/T). Οι θέσεις αναπαριστώνται με έναν κύκλο ενώ οι μεταβάσεις συνήθως με κατακόρυφες γραμμές (μπάρες). Οι θέσεις μπορεί να αναπαραστούν συνθήκες, κατάσταση ενός συστατικού ή μια εργασία του συστήματος ενώ οι μεταβάσεις συνήθως αναπαραριστούν τα γεγονότα και/ή τις εργασίες.

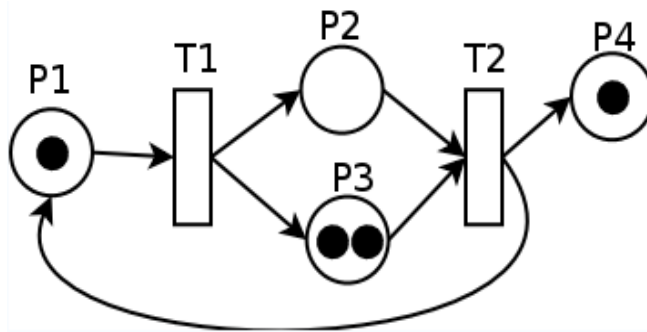
Τα δίκτυα Petri είναι *κατευθυνόμενα*. Έτσι, αντίθετα με τις ζεύξεις διπλής κατεύθυνσης των φυσικών δικτύων, σε ένα δίκτυο Petri υπάρχουν τόξα που συνδέουν θέσεις, οι οποίες αποτελούν τις θέσεις εισόδου, με μεταβάσεις ή τόξα που συνδέουν μεταβάσεις με θέσεις, οι οποίες αποκαλούνται θέσεις εξόδου.

Με άλλα λόγια, η μεταφορά των πληροφοριών από μία θέση σε μία μετάβαση ή από μια μετάβαση σε μία θέση είναι μονής κατεύθυνσης (one-way). Μια μεταφορά διπλής κατεύθυνσης (two-way) επιτυγχάνεται με ένα τόξο από μια θέση με μια μετάβαση, και εν συνεχεία με ένα άλλο τόξο από την μετάβαση στην αρχική θέση. Αυτό αποτελεί και έναν *κλειστό βρόχο* (self-loop).

Η δυναμική του δικτύου καθορίζεται από την δυνατότητα των θέσεων να μην έχουν καμία ή να έχουν κάποιο θετικό αριθμό μαρκών (ή κουπονιών - tokens), οι οποίες αναπαριστώνται με μικρούς γεμάτους κύκλους. Η κατανομή των μαρκών στις θέσεις ονομάζεται *σήμανση* (marking) του δικτύου. Οι μάρκες μπορούν, ανάμεσα σε πολλά άλλα, να σημαίνουν τον αριθμό των πόρων, να υποδηλώνουν αν μια συνθήκη είναι αληθής, ή να δηλώνουν αν μια εργασία είναι σε λειτουργία. Αυτό εξαρτάται από το τι υποδηλώνει η θέση στην οποία βρίσκονται.

Στο σχήμα 2.1 διακρίνονται 4 θέσεις P_1, P_2, P_3, P_4 και 2 μεταβάσεις T_1, T_2 που ενώνονται μεταξύ τους με τόξα. Η τρέχουσα σήμανση του δικτύου αποτελείται από 1 μάρκα στη θέση P_1 , 2 μάρκες στη θέση P_3 και 1 μάρκα στη θέση P_4 .

Όταν όλες οι θέσεις εισόδου μιας μετάβασης διαθέτουν ικανό αριθμό μαρκών, το



Σχήμα 2.1: Παράδειγμα απλού δικτύου Petri

γεγονός που υποδηλώνει η μετάβαση μπορεί να πραγματοποιηθεί. Αυτό αποκαλείται *εκपुरσοκρότηση μετάβασης* (transition firing). Η εκपुरσοκρότηση αλλάζει την κατανομή των μαρκών στις θέσεις, δηλώνοντας την αλλαγή στην κατάσταση του συστήματος.

Τόσο οι δομικές ιδιότητες όσο και οι ιδιότητες της συμπεριφοράς του δικτύου Petri μπορούν να οριστούν και να μελετηθούν. Οι πρώτες αποφασίζονται αποκλειστικά από την δομή ή την τοπολογία του δικτύου, ενώ οι τελευταίες εξαρτώνται τόσο από την αρχική σήμανση που μοντελοποιεί την αρχική κατάσταση όσο και από τις προϋποθέσεις αλλά και την τελική σειρά εκपुरσοκρότησης των διαφόρων μεταβάσεων.

Πιο σημαντικό χαρακτηριστικό των δικτύων Petri είναι η δυνατότητα που προσφέρουν για έναν τυπικό έλεγχο των ιδιοτήτων του συστήματος και της συμπεριφοράς του, όπως π.χ. οι σχέσεις προτεραιότητας των γεγονότων, οι ταυτόχρονες ενέργειες, ο απαιτούμενος συγχρονισμός, η αποφυγή αδιεξόδων, η επαναληψιμότητα ενεργειών, ή η αμοιβαία απαγόρευση χρήσης μοιραζόμενων πόρων. Ένας τέτοιος έλεγχος πριν την υλοποίηση του συστήματος στο φυσικό επίπεδο είναι ιδιαίτερα σημαντικός στην δημιουργία και την ανάπτυξη ευέλικτων αυτοματοποιημένων συστημάτων διεργασιών.

2.1.1 Μαθηματική περιγραφή των απλών δικτύων Petri

Ένα σημασμένο δίκτυο Petri ορίζεται ως ένα σύνολο πέντε στοιχείων

$$Z = \langle P, T, I, O, M \rangle, \quad (2.1)$$

όπου:

- $P = p_1, p_2, \dots, p_n, n > 0$, είναι ένας πεπερασμένος πίνακας των θέσεων που αναπαριστούνται γραφικά με κύκλους
- $T = t_1, t_2, \dots, t_n, n > 0$, είναι ένας πεπερασμένος πίνακας των μεταβάσεων που παρουσιάζονται με κατακόρυφες γραμμές (μπάρες), όπου $P \cup T \neq \emptyset$ και $P \cap T = \emptyset$
- $I : P \times T \rightarrow \mathbb{N}$, είναι η συνάρτηση εισόδου που ορίζει το σύνολο των προσανατολισμένων τόξων από τις θέσεις P στις μεταβάσεις T , όπου $\mathbb{N} = 0, 1, 2, \dots$
- $O : P \times T \rightarrow \mathbb{N}$, είναι η συνάρτηση εξόδου που ορίζει το σύνολο των προσανατολισμένων τόξων από τις μεταβάσεις T στις θέσεις P
- $M : P \rightarrow \mathbb{N}$, είναι η σήμανση της οποίας ο i -οστός όρος παριστάνει τον αριθμό των μαρκών στην i -οστή θέση. Η αρχική σήμανση δηλώνεται ως m_0 .

Τα τέσσερα στοιχεία (P, T, I, O) αποτελούν την δομή του δικτύου Petri που ορίζεται από ένα προσανατολισμένο γράφο. Όπως τονίστηκε και παραπάνω, η εισαγωγή κουπονιών σε θέσεις και η ροή τους μέσα από μεταβάσεις δίνει την δυνατότητα περιγραφής και ανάλυσης της δυναμικής συμπεριφοράς των διακριτών γεγονότων ενός δικτύου Petri και κατ' επέκταση του συστήματος υπό μοντελοποίηση.

Όταν $I(p, t) > 0$, τότε ο αριθμός $I(p, t)$ ονομάζεται και βάρος του κλάδου ή των κλάδων από τη θέση p στη μετάβαση t . Το ίδιο ισχύει και για την τιμή $O(p, t)$ το οποίο ονομάζεται βάρος του κλάδου ή των κλάδων από το t στο p . Το I και το O παριστάνουν δύο πίνακες $n \times s$ θετικών ακέραιων στοιχείων. Η αφαίρεση του I από το O έχει ως αποτέλεσμα έναν πίνακα συμβάντων, ο οποίος δηλώνεται ως $C = O - I$.

2.1.2 Κανόνες εκτέλεσης των απλών δικτύων Petri

Οι κανόνες εκτέλεσης ενός δικτύου Petri αναφέρονται στους κανόνες εκκίνησης, ενεργοποίησης και εκπυροσρότησης.

- Μια μετάβαση $t \in T$ ενεργοποιείται όταν και μόνο όταν $m(p) \geq I(p, t), \forall p \in P$

- Η μετάβαση t που ενεργοποιείται σε μια σήμανση m εκπυρσοκροτεί και ως αποτέλεσμα προκύπτει μια νέα σήμανση \acute{m} , όπου:

$$\acute{m}(p) = m(p) - I(p, t) + O(p, t), \forall p \in P \quad (2.2)$$

Η σήμανση \acute{m} λέγεται ότι είναι (άμεσα) προσβάσιμη από την σήμανση m . Ο κανόνας ενεργοποίησης λέει ότι αν όλες οι θέσεις εισόδου μιας μετάβασης t έχουν ικανό αριθμό κουπονιών, τότε η t ενεργοποιείται. Αυτό σημαίνει ότι αν οι συνθήκες σχετικά με την πραγματοποίηση ενός γεγονότος ικανοποιούνται πλήρως, τότε το γεγονός λαμβάνει χώρα. Σημειώνεται ότι, από πλευράς λογικής, οι συνθήκες αυτές δημιουργούν μια σχέση AND. Πολύ συχνά περισσότερες από μία μεταβάσεις μπορεί να ικανοποιούν τις προϋποθέσεις για ενεργοποίηση. Στην περίπτωση αυτή υπάρχει σύγκρουση (collision) και, αν δεν υπάρχουν προτεραιότητες, μια από όλες τις μεταβάσεις αυτές επιλέγεται τυχαία προς εκπυρσοκρότηση.

Ο κανόνας εκπυρσοκρότησης λέει ότι μια ενεργοποιημένη μετάβαση t εκπυρσοκροτεί (δηλαδή ένα γεγονός συμβαίνει). Η εκπυρσοκρότηση αυτή μπορεί να εξεταστεί σε δύο διαφορετικά στάδια. Στο πρώτο στάδιο, αφαιρείται ο αριθμός των κουπονιών που χρειάζονται από την κάθε μία θέση εισόδου, και ο αριθμός αυτός ισούται με το άθροισμα των τόξων που συνδέουν την θέση εισόδου με την μετάβαση t . Στην εξίσωση 2.2, η αφαίρεση μαρκών δηλώνεται από τον όρο $I(p, t)$. Σε δεύτερο στάδιο, τοποθετούνται τα κουπόνια στην κάθε μία θέση εξόδου από την μετάβαση t , και ο αριθμός των κουπονιών ισούται με το άθροισμα των τόξων από την t στις συνδεδεμένες θέσεις εξόδου. Στην παραπάνω εξίσωση, αυτό δηλώνεται από τον όρο $O(p, t)$.

2.1.3 Επεκτάσεις των απλών δικτύων Petri

Πέρα από τις παραπάνω δυνατότητες, σε άλλες εργασίες ([11]) έχουν προταθεί αρκετές τεχνικές βελτίωσης και εμπλουτισμού των δυνατοτήτων μοντελοποίησης (αύξησης της ισχύος τους), όπως π.χ. η προσθήκη της δυνατότητας μηδενικού ελέγχου (zero testing ability), δηλαδή της δυνατότητας ελέγχου αν μια θέση έχει μηδενικό αριθμό κουπονιών και η επακόλουθη ενεργοποίηση ενός δεσμευτικού τόξου (inhibitory arc), η ανάθεση προτεραιοτήτων στις μεταβάσεις ¹ ([9]) ή η χρονική

¹Η προσθήκη προτεραιοτήτων μπορεί να έχει το ίδιο αποτέλεσμα με το δεσμευτικό τόξο.

καθυστέρηση σε θέσεις, μεταβάσεις ή/και σε τόξα (χρονικά δίκτυα, [22],[8]).

2.2 Τα έγχρωμα δίκτυα Petri

Όπως τονίστηκε και στην παράγραφο 1.1, η ανάπτυξη των έγχρωμων δικτύων Petri (Colored Petri Nets - δίκτυα CP) προέκυψε από την επιθυμία να αναπτυχθεί μια γλώσσα μοντελοποίησης τόσο θεωρητικά θεμελιωμένης όσο και πρακτικά ευέλικτης ώστε να χρησιμοποιηθεί για τα συστήματα του μεγέθους και της πολυπλοκότητας που συναντώνται στη βιομηχανία.

Για να επιτευχθεί κάτι τέτοιο, έγινε προσπάθεια να συνδυαστεί η δύναμη αναπαράστασης των απλών δικτύων Petri με τις δυνατότητες των γλωσσών προγραμματισμού. Τα πρώτα δίνουν τη δυνατότητα της περιγραφή συγχρονισμού των ταυτόχρονου διεργασιών που παρατηρούνται στην πράξη, ενώ οι γλώσσες προγραμματισμού παρέχουν τα μέσα για να καθορίζονται οι τύποι των στοιχείων του συστήματος και να χειρίζονται οι τιμές των στοιχείων αυτών ([12]).

Στο σημείο αυτό πρέπει να τονιστεί ότι μπορεί να αποδειχτεί ότι κάθε έγχρωμο δίκτυο Petri μπορεί να μεταφραστεί σε ένα ισοδύναμο (ως προς τη συμπεριφορά) απλό δίκτυο Petri και αντίστροφα. Αυτό σημαίνει ότι τα δίκτυα CP δεν έχουν κάποιο θεωρητικό πλεονέκτημα ως προς τη δύναμη έκφρασης και αναπαράστασης σε σχέση με τα απλά δίκτυα Petri.

Στην πράξη όμως τα δίκτυα CP αποτελούν μια πιο συμπαγή και βολική γλώσσα αναπαράστασης (κατ' αντιστοιχία με τη σχέση των γλωσσών προγραμματισμού υψηλού επιπέδου και των γλωσσών μηχανής - επιπέδου assembly). Η δυνατότητα ανάθεσης τύπων και τιμών στις μάρκες (σε σχέση με τα απλά δίκτυα Petri όπου ο μόνος τρόπος διάκρισης μεταξύ των μαρκών είναι να τοποθετούνται σε διαφορετικές θέσεις) επιτρέπει να χρησιμοποιηθεί ένας σημαντικά μικρότερος αριθμός θέσεων για την αναπαράσταση ενός συγκεκριμένου στοιχείου ή συστήματος.

Στα ακόλουθα παρουσιάζονται τα βασικά χαρακτηριστικά των έγχρωμων δικτύων Petri καθώς επίσης και οι τρόποι αναπαράστασης και μελέτης τους με τη βοήθεια των εργαλείων που προσφέρει το πακέτο Design/CPN ².

²Πιο αναλυτικές οδηγίες χρήσης του πακέτου Design/CPN μπορούν να βρεθούν στο παράρτημα Α'

2.2.1 Βασικά στοιχεία των έγχρωμων δικτύων Petri

Και στην περίπτωση των έγχρωμων δικτύων Petri βασικό ρόλο παίζει η έννοια των θέσεων (places) που συμβολίζονται με κύκλους (ή ελλείψεις). Κάθε θέση συσχετίζεται με έναν τύπο που καθορίζει το είδος των δεδομένων που μπορεί να περιέχει η συγκεκριμένη θέση. Ο τύπος από σύμβαση γράφεται με πλάγια γράμματα δίπλα σε κάθε θέση.

Κατά τη διάρκεια της εκτέλεσης, κάθε θέση θα περιέχει έναν μεταβλητό αριθμό από μάρκες (tokens). Κάθε μάρκα έχει μια τιμή που ανήκει στο συγκεκριμένο τύπο δεδομένων με τον οποίο είναι συσχετισμένη κάθε θέση.

Η κατάσταση ενός δικτύου CP ονομάζεται *σήμανση* (marking). Αποτελείται από έναν αριθμό από μάρκες τοποθετημένες σε διάφορες θέσεις του δικτύου. Οι μάρκες που είναι τοποθετημένες σε μια θέση αποτελούν τη σήμανση της θέσης. Για λόγους σύμβασης, αναγράφεται η αρχική σήμανση κάθε θέσης με υπογραμμισμένο κείμενο κοντά στην αντίστοιχη θέση.

Στη γενική περίπτωση, πολλές μάρκες που είναι σε μια θέση μπορεί να έχουν την ίδια τιμή. Με τον τρόπο αυτό προκύπτει ένα πολυσύνολο (multiset), σε αντίθεση με το απλό σύνολο όπου μόνο μια μάρκα μπορεί να έχει μια συγκεκριμένη τιμή. Για τα πολυσύνολα ορίζονται τελεστές και πράξεις όπως πρόσθεση, πολλαπλασιασμός, σύγκριση, μέγεθος, αφαίρεση. Ο συντελεστής μπροστά από το σύμβολο (') καθορίζει τον αριθμό των μαρκών μιας συγκεκριμένης τιμής στο πολυσύνολο μιας θέσης.

Για ιστορικούς λόγους, οι τιμές των διαφόρων μαρκών αναφέρονται ως χρώματα (colours) ενώ οι αντίστοιχοι τύποι ως σύνολα από χρώματα (colour sets) για να τονιστεί η σχέση τους με τα απλά δίκτυα Petri όπου κάθε μάρκα θεωρείται μονόχρωμη («μαύρη») και είναι αυτού του «χρώματος» ανεξάρτητα από το τι αναπαριστά.

Οι ενέργειες σε ένα δίκτυο CP μοντελοποιούνται με την έννοια των μεταβάσεων (transitions), οι οποίες γραφικά αναπαριστώνται με παραλληλόγραμμα. Συνήθως σε κάθε θέση ή μετάβαση ανατίθεται και ένα όνομα που απλά και μόνο βοηθά στην αναγνωσιμότητα του μοντέλου ³.

Ένα εισερχόμενο τόξο (arc) δείχνει ότι μια μετάβαση μπορεί να αφαιρέσει μάρκες από μια θέση ενώ ένα εξερχόμενο τόξο δείχνει ότι η ίδια μετάβαση μπορεί να

³Τον ίδιο «διακοσμητικό» ρόλο παίζουν και τα υπόλοιπα γραφιστικά στοιχεία (π.χ. το πάχος ή το χρώμα γραμμής κλπ).

προσθέσει μάρκες (τις ίδιες ή «επεξεργασμένες») σε άλλες θέσεις. Ο αριθμός και οι τιμές των μαρκών που μετακινούνται εξαρτώνται από τις εκφράσεις⁴ (arc expressions) που αναγράφονται δίπλα σε κάθε τόξο. Υπάρχουν και τα διπλά βέλη που ουσιαστικά είναι σύντομος τρόπος αναπαράστασης δύο τόξων με αντίθετη «κατεύθυνση» και ίδια «έκφραση».

Αλγόριθμος εκτέλεσης

Σε αντιστοιχία με τους κανόνες που περιγράφονται στην παράγραφο 2.1.2 σχετικά με την διαδικασία εξέλιξης των απλών δικτύων Petri, ορίζεται ο ακόλουθος αλγόριθμος εκτέλεσης των έγχρωμων δικτύων Petri.

Για να μιλήσουμε για την εκτέλεση (occurrence) ή εκπυρσοκρότηση μιας μετάβασης θα πρέπει να αρχικά γίνει μια συσχέτιση/πρόσδεση (bind) των τιμών των μεταβλητών στις εκφράσεις των τόξων εισόδου με τιμές μαρκών που πρέπει ήδη να υπάρχουν στις θέσεις από όπου ξεκινούν τα εισερχόμενα (στη μετάβαση) τόξα. Τότε λέμε ότι η μετάβαση είναι ενεργοποιημένη (enabled) με τη συγκεκριμένη συσχέτιση τιμών. Μία από όλες τις ενεργοποιημένες μεταβάσεις επιλέγεται τυχαία προς εκτέλεση όπως και στην περίπτωση των απλών δικτύων.

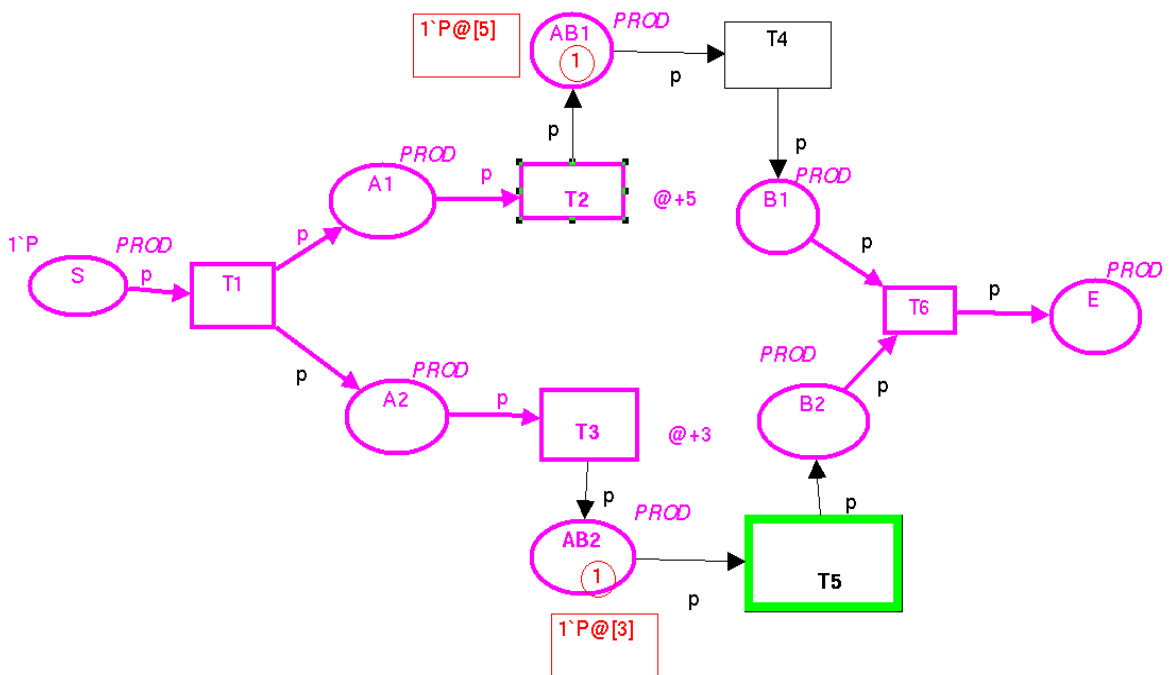
Όταν η μετάβαση εκτελεστεί, θα απορροφηθούν οι μάρκες που έχουν συσχετιστεί με τις μεταβλητές των εκφράσεων τόξων ενώ θα εμφανιστούν (είτε ταυτόχρονα είτε με κάποια χρονική καθυστέρηση που μπορεί να καθοριστεί όπως θα δούμε παρακάτω) κάποιες μάρκες μέσω των εκφράσεων των εξερχόμενων τόξων στις θέσεις όπου αυτά καταλήγουν. Κατά την εκτέλεση του δικτύου, οι ενεργοποιημένες μεταβάσεις συνήθως αναπαριστώνται γραφικά με μια πιο παχιά γραμμή γύρω από τη μετάβαση.

Η εκτέλεση ενός CPN μοντέλου περιγράφεται μέσω της ακολουθίας εκτέλεσης των διαφόρων μεταβάσεων (occurrence sequence) και ουσιαστικά περιλαμβάνει τις διάφορες σημάνσεις και τα βήματα μέσω των οποίων αυτά επιτυγχάνονται. Γενικά μπορεί να υπάρχουν πολλές δυνατές ακολουθίες μεταβάσεων κάτι που σημαίνει ότι ένα δίκτυο CPN έχει μια μη-ντετερμινιστική συμπεριφορά.

Η χρήση μεταβλητών στις εκφράσεις των τόξων σημαίνει ότι κάθε μετάβαση μπορεί να εκτελεστεί με πολλές διαφορετικές συσχετίσεις/αναθέσεις τιμών (κατά

⁴Οι εκφράσεις των τόξων είναι τμήματα από κώδικα (code segments) τα οποία συσχετίζονται με συγκεκριμένες μεταβάσεις. Όταν μια μετάβαση εκτελεστεί, το αντίστοιχο τμήμα κώδικα εκτελείται.

αντιστοιχία με την κλήση μιας διαδικασίας με πολλές διαφορετικές τιμές παραμέτρων σε μια συμβατική γλώσσα προγραμματισμού) οπότε μπορεί να χρησιμοποιηθεί μια μόνο μετάβαση για την αναπαράσταση μιας δραστηριότητας, ενώ στα απλά δίκτυα Petri θα χρειαζόταν μια μετάβαση για κάθε στιγμιότυπο/επανεμφάνιση της δραστηριότητας.



Σχήμα 2.2: Παράδειγμα έγχρωμου δικτύου Petri

```

use "/CPN_EXAMPLES/tod.sml";

color PROD = with P timed;
color RES = with Tom | Mark;
var r:RES;
var p:PROD;
    
```

Σχήμα 2.3: Ορισμοί για το παράδειγμα CPN δικτύου

Στο σχήμα 2.2 απεικονίζεται ένα παράδειγμα CPN. Διακρίνονται οι θέσεις με

τους τύπους που είναι συσχετισμένοι με αυτές δίπλα τους με πλάγια γράμματα. Οι τύποι αυτοί ορίζονται στο τμήμα δηλώσεων του σχήματος 2.2.1. Υπάρχουν μεταβάσεις που εκτελούνται ακαριαία (όπως η T_1) αλλά και άλλες που είναι συσχετισμένες με χρόνο μέσω του τελεστή @ (όπως η T_2). Στο στιγμιότυπο του συγκεκριμένου σχήματος (τη χρονική στιγμή $t = 3$) είναι ενεργοποιημένη η μετάβαση T_5 λόγω της ύπαρξης μιας μάρκας στη θέση AB_2 η οποία θα συσχετιστεί με τη μεταβλητή p .

Η παραπάνω σύντομη περιγραφή για τον τρόπο ενεργοποίησης και εκτέλεσης των κανόνων εξηγεί διαισθητικά τον τρόπο λειτουργίας των δικτύων CP. Ο μαθηματικός ορισμός των παραπάνω εννοιών μπορεί να βρεθεί στο [12].

Το εργαλείο Design/CPN (βλ. παράρτημα Α') παρέχει έναν CPN editor καθώς και ένα εργαλείο ελέγχου της σύνταξης (syntax checker) ενώ μέσα από τα εργαλεία της προσομοίωσης και της ανάλυσης του χώρου καταστάσεων μπορεί να ανιχνευτεί η σωστή συμπεριφορά και λειτουργία του μοντέλου.

Το Design/CPN χρησιμοποιεί τη συναρτησιακή γλώσσα Standard ML ([10],[15]) για τον ορισμό των τύπων και των εκφράσεων του δικτύου. Εδώ πρέπει να τονιστεί ότι αν και τόσο το Design/CPN όσο και το εργαλείο CPNTools χρησιμοποιούν την παραπάνω γλώσσα, αντ' αυτής μπορούν χρησιμοποιηθούν και άλλες γλώσσες για την αναπαράσταση των έγχρωμων δικτύων Petri και υπάρχουν άλλα εργαλεία που τις χρησιμοποιούν ([16], Matlab Stateflow κλπ.).

Η γλώσσα SML επιτρέπει, ανάμεσα σε άλλα, τον ορισμό σύνθετων τύποι δεδομένων (χρώματα) μέσω της δεσμευμένης λέξης **color** καθώς και τιμές και τύπους για κάθε μεταβλητή (**var**) ή σταθερά (**val**) που μπορεί να εμφανιστεί ενώ μπορούν να οριστούν και συναρτήσεις (**fun**). Στο σχήμα 2.2.1 διακρίνονται τύποι όπως ακέραιοι, πραγματικοί αριθμοί, ζεύγη (products), λίστες αλλά και πιο σύνθετοι τύποι που μπορούν να προκύψουν με τη βοήθειά τους.

2.3 Ιδιότητες και ανάλυση των έγχρωμων δικτύων Petri

Τα δίκτυα Petri ως μαθηματικά εργαλεία κατέχουν έναν αριθμό ιδιοτήτων. Οι ιδιότητες αυτές διακρίνονται σε δύο είδη, στις ιδιότητες που αφορούν στην δομή του συστήματος και σε αυτές που αφορούν στην συμπεριφορά του συστήματος. Οι ιδιό-

```

color INT = int;
color DATA = string;
color AMOUNT = real;
color INTxDATA = product INT * DATA;
color BUF = list INT;
var n, k: INT;
var p, str: DATA;
val stop = "###";

color Ten0 = int with 0. .10;
color Ten1 = int with 1. .10;

var s: Ten0;
var r: Ten1;
var lista: BUF;

fun Ok(s:Ten0, r:Ten1) = (r<=s);

```

Σχήμα 2.4: Παράδειγμα ορισμών σύνθετων τύπων και συναρτήσεων στη γλώσσα SML

τητες συμπεριφοράς είναι οι ιδιότητες που εξαρτώνται από την αρχική σήμανση του δικτύου. Στον αντίποδα, οι ιδιότητες της δομής του δικτύου δεν εξαρτώνται από την αρχική σήμανση. Εξαρτώνται μόνο από την δομή ή την τοπολογία του δικτύου.

Όπως τονίστηκε και παραπάνω, μπορεί να αποδειχτεί η μαθηματική ισοδυναμία των έγχρωμων και των απλών δικτύων Petri. Στα παρακάτω θα παρουσιαστούν εν συντομία οι βασικές ιδιότητες των απλών δικτύων Petri, πολλές από τις οποίες συναντούνται και στο πεδίο των δικτύων CP. Γενικά, όμως, όταν κάποιο δίκτυο CP χρησιμοποιεί σύνθετους τύπους, τα ισοδύναμα ως προς τη συμπεριφορά απλά δίκτυα Petri έχουν πολύ μεγάλο και συχνά αστρονομικού μεγέθους αριθμό από θέσεις, οπότε και η εξέταση πολλών από τις ιδιότητες αυτές γίνεται δύσκολη διαδικασία (τόσο από την άποψη της απαιτούμενης υπολογιστικής ισχύος όσο και της πρακτικής χρησιμότητας).

Οι πιο σημαντικές, από πλευράς πρακτικότητας, ιδιότητες είναι η προσβασιμότητα (reachability), η φραγή (boundness), η ασφάλεια (safeness), η συντηρητικότητα (conservativeness), η ζωτικότητα (liveness), η αντιστρεπτότητα (reversibility) και

η αρχική κατάσταση (home state). Επιπρόσθετες δομικές ιδιότητες περιλαμβάνουν την επαναληψιμότητα (repetitiveness) και τη συνέπεια (consistency). Επίσης, υπάρχουν και άλλες ιδιότητες όπως η κάλυψη (coverability), η απόσταση συγχρονισμού (synchronic distance), η αμεροληψία (fairness) κ.α.

Η σημασία αυτών των ιδιοτήτων στο πεδίο των ευέλικτων συστημάτων παραγωγής είναι εν γένει πολύ σημαντική, για αυτό και θα δούμε αναλυτικά τη φυσική τους σημασία στην εφαρμογή των δικτύων Petri στο πεδίο αυτό στο κεφάλαιο 3.

Η *προσβασιμότητα* είναι μια ιδιότητα συμπεριφοράς εφόσον εξαρτάται από την αρχική σήμανση. Μια πολύ σημαντική παράμετρος κατά την σχεδίαση ενός συστήματος είναι το ερώτημα κατά πόσον το σύστημα μπορεί να φτάσει σε μια συγκεκριμένη κατάσταση, ή να επιδείξει συγκεκριμένη λειτουργική συμπεριφορά. Γενικά, θα μπορούσαμε να πούμε ότι το ερώτημα είναι αν το σύστημα που έχει μοντελοποιηθεί με ένα δίκτυο Petri διαθέτει όλα τα χαρακτηριστικά, όπως αυτά ορίζονται από τις προδιαγραφές, και καμία ανεπιθύμητη ιδιότητα.

Η *φραγή* εγγυάται τη μη υπερχειλίση της χωρητικότητας κάποιας θέσης.

Η *ασφάλεια*⁵ είναι μια ειδική περίπτωση της προσβασιμότητας. Η ασφάλεια μιας θέσης που υποδηλώνει μια εργασία εγγυάται ότι δεν θα γίνει καμία προσπάθεια εκτέλεσης μιας υπό-εκτέλεση εργασίας. Η ασφάλεια μιας θέσης που υποδηλώνει έναν πόρο εγγυάται την διαθεσιμότητα ενός μόνο πόρου.

Η *ζωτικότητα* ενός δικτύου Petri δηλώνει ότι το δίκτυο δεν θα καταλήξει σε αδιέξοδο (deadlock), είτε από ακινησία ενός τμήματος είτε του συνόλου του συστήματος. Το αδιέξοδο είναι αποτέλεσμα λανθασμένης κατανομής των πόρων ή εξάντλησης συγκεκριμένων τύπων πόρων.

Η *αντιστρεπτότητα* σε ένα μοντέλο δικτύου Petri εγγυάται την επαναληπτική⁶ και κυκλική συμπεριφορά του μοντελοποιούμενου συστήματος.

⁵Μαθηματικά, *ασφαλές* ορίζεται ένα δίκτυο Petri που είναι 1-φραγμένο

⁶Ένα δίκτυο Petri ονομάζεται *επαναληπτικό* αν υπάρχει μια πεπερασμένη σήμανση m_0 και μια ακολουθία εκπυροσροτήσεων S από την m_0 , έτσι ώστε κάθε μετάβαση να συμβαίνει απείρως συχνά μέσα στην S

2.4 Αποτίμηση συμπεριφοράς, ορθότητας και επίδοσης

Τόσο το πακέτο Design/CPN όσο και το CPNTools παρέχουν εργαλεία με στόχο την εκτίμηση της ορθότητας και της επίδοσης ενός μοντέλου.

2.4.1 Ανάλυση επίδοσης των δικτύων CP

Για την εξέταση της επίδοσης των μοντέλων (πέρα από την ορθότητα της λειτουργίας τους) δηλαδή της ταχύτητας με την οποία λειτουργούν τα αντίστοιχα συστήματα, επεκτείνεται η λειτουργικότητά τους με την έννοια του χρόνου. Εισάγεται έτσι ένα κεντρικό ρολόι, οι τιμές του οποίου αναπαριστούν το *χρόνο εκτέλεσης* του μοντέλου (*model time*) και μπορεί να είναι είτε διακριτές είτε ακόμη και συνεχείς στη γενική περίπτωση. Πλέον κάθε μάρκα συσχετίζεται με μια *τιμή χρόνου* (*time stamp*). Ουσιαστικά η χρονική αυτή στιγμή αντιστοιχεί στην ενωρίτερη τιμή του χρόνου του μοντέλου στην οποία η μάρκα αυτή μπορεί να χρησιμοποιηθεί (δηλαδή να αφαιρεθεί μέσω της *ανάθεσης* σε κάποιο στοιχείο πρόσδεσης (π.χ. σε κάποια μεταβλητή).

Στα χρονικά δίκτυα Petri, για την εκτέλεση μιας μετάβασης, εκτός από τις υπόλοιπες προϋποθέσεις της ενεργοποίησης αλλά και της ικανοποίησης των συνθηκών προφύλαξης (*guard*) της μετάβασης, θα πρέπει να είναι διαθέσιμες και οι μάρκες υπό πρόσδεση, δηλαδή οι τιμές χρόνου τους να είναι μικρότερες από την τιμή του κεντρικού ρολογιού.

Για να μοντελοποιηθεί το γεγονός ότι μια διαδικασία ή ενέργεια μπορεί να χρειαστεί r χρονικές μονάδες για να ολοκληρωθεί, η αντίστοιχη μετάβαση αναθέτει *timestamps* για τις μάρκες εξόδου της που έχουν τιμές χρόνου r χρονικές μονάδες μεγαλύτερες από τη χρονική στιγμή που εκτελέστηκε η συγκεκριμένη μετάβαση. Αυτό ουσιαστικά σημαίνει ότι οι μάρκες αυτές (εξόδου) γίνονται διαθέσιμες r χρονικές μονάδες αργότερα από τη χρονική στιγμή εκτέλεσης της μετάβασης.

Πρακτικά, η «εκτέλεση» ενός *χρονικού* δικτύου CP λειτουργεί με παρόμοιο τρόπο με αυτόν των ουρών γεγονότων που συναντάται σε πολλές άλλες γλώσσες μοντελοποίησης διακριτών γεγονότων. Το σύστημα παραμένει με σταθερή την τιμή του κεντρικού ρολογιού όσο υπάρχουν στοιχεία που είναι διαθέσιμα και μεταβάσεις που ενεργοποιούνται από τέτοια στοιχεία. Όταν δεν υπάρχουν τα παραπάνω, τότε η

τιμή του κεντρικού ρολογιού αυξάνεται στην επόμενη τιμή κατά την οποία γίνονται διαθέσιμες μάρκες και μπορούν να ενεργοποιηθούν μεταβάσεις.

Στο εργαλείο Design/CPN, για να μπορέσει μια μάρκα να συσχετιστεί με μια τιμή χρόνου θα πρέπει να υπάρχει μια δήλωση του αντίστοιχου τύπου που να χαρακτηρίζεται ως **timed**. Μέσω μιας δήλωσης χρονικής διάρκειας (time inscription) που ξεκινά με το σύμβολο «@+» μπορεί σε κάθε μετάβαση να αντιστοιχηθεί η διάρκεια ολοκλήρωσής της.

2.4.2 Ανάλυση συμπεριφοράς και ορθότητας των δικτύων CP

Βασικές τεχνικές ελέγχου της συμπεριφοράς των δικτύων CP είναι η προσομοίωση και η ανάλυση του χώρου καταστάσεων. Για τα απλά δίκτυα Petri, η συνήθως μειωμένης πολυπλοκότητας μαθηματική αναπαράστασή τους επιτρέπει την ύπαρξη και άλλων εναλλακτικών ελέγχων όπως η ανάλυση μέσω αναλλοίωτων (invariant analysis) και η μέθοδος ανάλυσης μέσω απλοποίησης (reduction method), περισσότερα για τις οποίες αναφέρονται στο [18].

Μέσω της εξέτασης της συμπεριφοράς μπορούμε να έχουμε μια εικόνα για την ορθότητα της λειτουργίας ενός δικτύου (σε σχέση πάντα με τη συμπεριφορά του φυσικού συστήματος που προσπαθούμε να αναπαραστήσουμε). Ειδικά στην περίπτωση των δικτύων CP για τον έλεγχο της ορθότητας μπορούν να χρησιμοποιηθούν και τεχνικές ημιαυτόματης επαλήθευσης χρήσης (use verification) μέσω εισαγωγής εκφράσεων χρονικής λογικής (temporal logic) [6].

2.4.3 Το εργαλείο προσομοίωσης των δικτύων CP

Η εξέταση της λειτουργίας και ο έλεγχος ενός μοντέλου γίνονται με τη βοήθεια του εργαλείου προσομοίωσης⁷. Σε κάθε βήμα εκτέλεσης μπορεί ο χρήστης να δει το σύνολο των ενεργοποιημένων μεταβάσεων και των δυνατών προσδέσεων (bindings) σε κάθε περίπτωση. Έχει ακόμη τη δυνατότητα να επιλέξει ποιά από αυτές τις συσχετίσεις θα εκτελεστεί (occur) σε κάθε βήμα. Η τελευταία κατάσταση λειτουργίας αναφέρεται ως *διαδραστική* και είναι από τη φύση της αρκετά αργή μιας και ο

⁷Πιο αναλυτικές οδηγίες για τη χρήση του εργαλείου μπορούν να βρεθούν στο παράρτημα Α'

σχεδιαστής του συστήματος θα πρέπει σε κάθε βήμα να εμπλέκεται και να παίρνει αποφάσεις για την εξέλιξη του συστήματος.

Σε αντίθεση με την παραπάνω κατάσταση, κατά την *αυτόματη προσομοίωση* μπορούν να εκτελείται ένας μεγάλος αριθμός μεταβάσεων το δευτερόλεπτο ενώ πλέον το ίδιο το εργαλείο παίρνει αποφάσεις για το ποια από τις ενεργοποιημένες μεταβάσεις τελικά θα συμβεί με τη βοήθεια κάποιας γεννήτριας τυχαίων αριθμών ώστε να αποφεύγονται φαινόμενα πόλωσης (biasing) των επιλογών.

Με τους παραπάνω τρόπους είναι δυνατόν να παρατηρηθεί η εξέλιξη της κατάστασης ενός μοντέλου μέσω της καταγραφής της σήμανσης (marking) του δικτύου. Μέσω της αναφοράς προσομοίωσης, μπορούν να επισημανθούν όλες οι συσχετίσεις που έχουν τελικά εκτελεστεί κατά της διάρκεια μιας (και μόνο) προσομοίωσης. Υπάρχουν και εναλλακτικοί τρόποι καταγραφής της πορείας εξέλιξης της κατάστασης ενός μοντέλου όπως, για παράδειγμα, η προσθήκη μιας σειράς θέσεων ειδικού ρόλου (αναφοράς) που δεν επηρεάζουν τη συμπεριφορά του δικτύου.

2.4.4 Γραφική επισκόπηση του χώρου καταστάσεων

Με την προσομοίωση δεν είναι δυνατόν να αποφανθούμε για την ορθότητα ενός μοντέλου μιας και ο περιορισμένος αριθμός προσομοιώσεων δεν επιτρέπει να εξεταστούν όλες οι δυνατές εξελίξεις της κατάστασης ενός δικτύου. Αυτό το επιδιώκει η ανάλυση του χώρου καταστάσεων (state space analysis) (ή αλλιώς ανάλυση μέσω ελέγχου προσβασιμότητας), όπου βασικός στόχος είναι η κατασκευή ενός γράφου του οποίου κάθε κόμβος αντιστοιχεί σε μια σήμανση από την οποία μπορεί να «περάσει» η κατάσταση ενώ κάθε τόξο αντιστοιχεί σε εκείνη τη συσχέτιση στοιχείων που «οδηγεί» στη συγκεκριμένη κατάσταση. Η τελική αναπαράσταση αυτών ονομάζεται δένδρο ή γράφημα προσβασιμότητας (reachability tree/graph) ή καλυψιμότητας (occurrence graph) και εκεί περιλαμβάνονται όλα τα δυνατά μονοπάτια εξέλιξης των σημάνσεων του μοντέλου. Μιας και εξετάζονται όλες οι δυνατές καταστάσεις, συχνά χρειάζεται η απλούστευση του μοντέλου ή ο καθορισμός συνθηκών τερματισμού της εξέτασης για να αποφευχθεί ο μεγάλος χρόνος αλλά και να μπορεί να παρουσιαστεί γραφικά το δέντρο των επισκέψιμων καταστάσεων. Εν συνεχεία, όλες οι ιδιότητες συμπεριφοράς που σχολιάστηκαν παραπάνω μπορούν να προκύψουν εφόσον ο αριθμός των καταστάσεων είναι πεπερασμένος.

Ως προς την υλοποίηση της τεχνικής αυτής, υπάρχουν δύο στρατηγικές οι οποίες δημιουργούν και ονοματίζουν όλες τις σημάνσεις σε ένα σημασμένο δίκτυο Petri: κατά βάθος και κατά πλάτος αναζήτηση.

Στην κατά βάθος στρατηγική, ξεκινώντας από την αρχική σήμανση, αναγνωρίζουμε όλες τις μεταβάσεις που μπορούν να εκपुरσοκροτήσουν και εκपुरσοκροτούμε μία στην τύχη προκαλώντας μία νέα σήμανση. Εάν η σήμανση αυτή είναι παλιά ή νεκρή, τότε δεν την μελετάμε αλλά επιστρέφουμε στην σήμανση από την οποία προέκυψε η τρέχουσα και συνεχίζουμε με κάποια άλλη μετάβαση. Σε διαφορετική περίπτωση, αναγνωρίζουμε ξανά τις μεταβάσεις που μπορούν να εκपुरσοκροτήσουν, και εκपुरσοκροτούμε μία στην τύχη, προκαλώντας ξανά μια νέα σήμανση. Αυτή η διαδικασία συνεχίζεται έως ότου όλες οι ενεργοποιημένες μεταβάσεις έχουν τελικά εκपुरσοκροτήσει και όλες οι σημάνσεις έχουν δημιουργηθεί, εφόσον ο αριθμός των σημάνσεων είναι πεπερασμένος.

Στην κατά πλάτος στρατηγική, αναγνωρίζουμε όλες τις μεταβάσεις που μπορούν να εκपुरσοκροτήσουν, και τις εκपुरσοκροτούμε όλες, προκαλώντας πολλαπλές νέες σημάνσεις. Για κάθε σήμανση, εάν έχει εξεταστεί προηγουμένως ή είναι νεκρή σήμανση, προχωράμε στην επόμενη. Σε διαφορετική περίπτωση, αναγνωρίζουμε όλες τις μεταβάσεις που μπορούν να εκपुरσοκροτήσουν και τις εκपुरσοκροτούμε, προκαλώντας νέες σημάνσεις, και προχωράμε στην επόμενη σήμανση, μέχρις ότου οι σημάνσεις του ίδιου επιπέδου να εξαντληθούν. Εν συνεχεία, ξεκινάμε στο επόμενο επίπεδο σημάνσεων.

Η γραφική απεικόνιση του αποτελέσματος αποτελείται από κόμβους, που αντιστοιχούν σε σημάνσεις, και τόξα. Ένα τόξο συνδέει έναν κόμβο με έναν άλλο και έχει σημειωμένη την μετάβαση η οποία εκपुरσοκρότησε και δημιούργησε την αλλαγή στη σήμανση. Η διαγραφή διπλών σημάνσεων από το δένδρο οδηγεί στην δημιουργία ενός γραφήματος προσβασιμότητας ή κάλυψης. Και οι δύο στρατηγικές οδηγούν στο ίδιο γράφημα προσβασιμότητας. Στην πραγματικότητα, κάθε δοσμένο δίκτυο με μια αρχική σήμανση έχει μοναδικό γράφημα προσβασιμότητας.

Όταν ένα δίκτυο Petri είναι μη-φραγμένο ή ο αριθμός των καταστάσεων του μοντελοποιημένου συστήματος είναι μη πεπερασμένος, τότε η παραπάνω διαδικασία θα συνεχίζεται για πάντα. Υπάρχουν τεχνικές που υιοθετούνται ώστε ακόμη και σ' αυτή την περίπτωση να διατηρηθεί το δέντρο πεπερασμένο.

Γενικά η ανάλυση του χώρου των καταστάσεων θεωρείται ένας πολύ αποδοτικός

τρόπος για να εξετάσει κανείς τη δυναμική συμπεριφορά ενός συστήματος. Τόσο η κατασκευή όσο και η ανάλυση του χώρου καταστάσεων είναι αυτόματες διαδικασίες. Επιπλέον των προ-κατασκευασμένων αναφορών, στα περισσότερα εργαλεία υπάρχει και η δυνατότητα να εισαχθούν νέες ερωτήσεις (queries) σχετικά με τη συμπεριφορά του συστήματος και την εξέλιξή της με το πέρασμα του χρόνου.

Το βασικό πρόβλημα της προσέγγισης αυτής είναι το γεγονός ότι για πολύπλοκα συστήματα ο χώρος καταστάσεων γίνεται πολύ μεγάλος και η κατασκευή του μπορεί να πάρει αρκετές ώρες (κάτι που εξαρτάται από τη διαθέσιμη υπολογιστική ισχύ). Παρόλα αυτά εξακολουθεί να θεωρείται πιο γρήγορος και αξιόπιστος τρόπος σε σχέση με μια -έστω και εκ πρώτης όψεως εξονυχιστική - ανάλυση μέσω προσομοίωσης. Το πρόβλημα αυτό συνήθως αναφέρεται στη βιβλιογραφία ως «πρόβλημα έκρηξης του αριθμού των καταστάσεων» (state explosion problem). Για να αντιμετωπιστεί το θέμα αυτό υπάρχουν διάφορες μέθοδοι που είτε εκμεταλλεύονται κρυμμένες συμμετρίες του συστήματος ⁸ είτε καταφεύγουν σε αρθρωτές (modular) προσεγγίσεις.

Άλλο βασικό θέμα της μεθόδου του χώρου καταστάσεων είναι ότι ο τελικός γράφος κατασκευάζεται για μια συγκεκριμένη αρχική σήμανση που συχνά αντιστοιχεί σε μια από τις πολλές δυνατές αρχικές διαμορφώσεις του συστήματος. Δυστυχώς δεν υπάρχει δυνατότητα εξαγωγής συμπερασμάτων για διαφορετικές αρχικές συνθήκες ή για τη γενική συμπεριφορά του συστήματος παρά μόνο ενδείξεις.

Άλλες μέθοδοι ανάλυσης είναι ακόμη πιο γενικές και παρέχουν μια πλήρη απόδειξη ότι ένα σύστημα δουλεύει όπως αναμένεται να λειτουργεί. Μια τέτοια μέθοδος είναι και η ανάλυση αναλοίωτων (place invariant) που όμως είναι λιγότερο αυτόματη μέθοδος. Πολύ συχνά εμπλέκονται αρκετές μαθηματικές αποδείξεις και άρα είναι περισσότερο χρονοβόρα αλλά και ευάλωτη σε ανθρώπινα λάθη.

2.4.5 Αναφορά γράφου καλυψιμότητας

Εναλλακτικά ως προς τη γραφική απεικόνιση, το εργαλείο Design/CPN μπορεί να χρησιμοποιήσει τον γράφο καλυψιμότητας για να εξάγει μια αναφορά που δίνει αρκετά σημαντικές πληροφορίες σχετικά με τη συμπεριφορά του δικτύου. Ακολουθεί μια

⁸Περισσότερα για τη συμπίεσμένη μορφή του γράφου καλυψιμότητας (Strongly Connected Components - SCC graph) μπορεί να βρει κανείς στα [12],[13]

παρουσίαση των στοιχείων της αναφοράς που συνήθως ενδιαφέρουν περισσότερο.

Ιδιότητα φραγής (Boundedness)

Η ιδιότητα αυτή αφορά το άνω και κάτω όριο σε κάθε θέση του μοντέλου καθόλη τη λειτουργία του. Τα όρια αυτά αφορούν στον αριθμό των μαρκών κάθε τύπου που μπορεί να βρίσκονται σε μια συγκεκριμένη θέση. Από τα εργαλεία ανάλυσης παρέχονται στοιχεία τόσο ως προς τον αριθμό (integer bounds) αλλά και τον ακριβή τύπο και τιμή των μαρκών (multiset bounds) για τις τιμές του μέγιστου και ελάχιστου ορίου για κάθε θέση.

Ιδιότητα επαναληπτικότητας (Home property)

Μια σήμανση ορίζεται ως *αρχική* (home) αν είναι πάντα δυνατόν να ξαναπαρατηρηθεί όσο εξελίσσεται το μοντέλο.

Ιδιότητα ζωτικότητας (Liveness property)

Η *ζωτικότητα* ενός μοντέλου εξετάζει κατά πόσον κάθε μετάβαση είναι δυνατόν να ενεργοποιηθεί μετά από την ενεργοποίηση και εκπυρσοκρότηση ενός αριθμού άλλων μεταβάσεων. Σαν *νεκρή* σήμανση (dead marking) ορίζεται μια κατάσταση του μοντέλου όπου καμιά μετάβαση δεν ενεργοποιείται. Είναι προφανές ότι στη γενική περίπτωση, αν αυτές οι καταστάσεις δεν αφορούν τελικές καταστάσεις του συστήματος (π.χ. μάρκες που αναπαριστούν το τελικό προϊόν σε μια διαδικασία συναρμολόγησης), δεν είναι επιθυμητές. Με τη χρήση του σχετικού εργαλείου «μετάφρασης» μιας σήμανσης στην αντίστοιχη κατάσταση του μοντέλου μπορεί να γίνει έλεγχος όλων των αναφερόμενων ως νεκρών σημάνσεων και να ελεγχθεί αν όντως αντιστοιχούν σε έγκυρες τελικές καταστάσεις.

Άλλες ενδιαφέρουσες ιδιότητες είναι η έννοια της ύπαρξης νεκρών μεταβάσεων (dead transitions) δηλαδή μεταβάσεων που δεν ενεργοποιούνται ποτέ ή της ύπαρξης ζωντανών μεταβάσεων (live transitions), δηλαδή μεταβάσεων που διαρκώς ενεργοποιούνται ακόμη και αν μόλις έχουν εκπυρσοκροτήσει.

Ιδιότητα τιμιότητας (Fairness property)

Η ιδιότητα της τιμιότητας εξετάζει τη συχνότητα ενεργοποίησης των μεταβάσεων. Μια μετάβαση χαρακτηρίζεται ως αμερόληπτη (impartial) αν είναι δυνατόν να παρατηρηθεί απείρως συχνά σε μια άπειρη σειρά καλυψιμότητας. Υπάρχουν και ασθενέστερες έννοιες δικαιοσύνης όπως η *fair* και *just*, ενώ αν δεν τηρείται καμιά προδιαγραφή δικαιοσύνης (και άρα υπάρχουν διαδοχές καταστάσεων όπου δεν ενεργοποιούνται οι αντίστοιχες καταστάσεις), τότε χρησιμοποιείται ο χαρακτηρισμός *no fairness*.

Κεφάλαιο 3

Μοντελοποίηση συστημάτων παραγωγής

3.1 Στοιχεία ενός συστήματος παραγωγής

Η λειτουργία ενός συστήματος παραγωγής (manufacturing system) στηρίζεται:

- στους διαθέσιμους πόρους (resources). Το σύνολο από τους πόρους ονομάζεται *φυσικό σύστημα* (physical system).
- το *σχέδιο παραγωγής* (manufacturing layout). Το σχέδιο καθορίζει τον τρόπο με τον οποίο κατανέμονται χωρικά οι διάφοροι πόροι και πώς συνδέονται μεταξύ τους, και
- τη *διαχείριση* (management) του συστήματος.

Στόχος ενός συστήματος παραγωγής είναι η παραγωγή ενός συνόλου από προϊόντα διαφορετικού τύπου. Για να ολοκληρωθεί η παραγωγή ενός προϊόντος, το τελευταίο υπόκειται μια σειρά από κατεργασίες (operations). Κάθε τέτοια κατεργασία πραγματοποιείται από μία ή περισσότερες μηχανές (machines). Η πραγματοποίηση κάθε κατεργασίας πιθανά απαιτεί ένα ή περισσότερα εργαλεία (tools). Τέτοια μπορεί να είναι, για παράδειγμα, εργαλεία κοπής, μέτρησης, προσαρμογής κ.α. Η μεταφορά του - ημιτελούς - προϊόντος από μια μηχανή σε μια άλλη γίνεται με τη χρήση μηχανισμών μεταφοράς (transportation resources). Τέλος, συστήματα αποθήκευσης (storage systems), όπως προσωρινές αποθήκες (buffers) ή αποθήκες για τα τελικά προϊόντα χρησιμοποιούνται για την απορρόφηση τυχαίων γεγονότων, που μπορούν να προκύψουν και να διαταράξουν τη συνηθισμένη λειτουργία του συστήματος, ώστε να το διατηρήσουν σε έναν σταθερό ρυθμό λειτουργίας.

Το σχέδιο παραγωγής εξαρτάται από τους διαθέσιμους πόρους (π.χ. πρώτες ύλες), τα προϊόντα που πρέπει να κατασκευαστούν όπως επίσης και από τις απαιτήσεις των πελατών. Οι πιο συχνά χρησιμοποιούμενοι τύποι σχεδίων παραγωγής είναι:

- Γραμμικό σχέδιο (linear manufacturing layout). Συνήθως ακολουθείται από τις γραμμές μαζικής παραγωγής.
- Σχέδιο που έχει σαν στόχο την ομαδοποίηση των πόρων βάσει της λειτουργικότητάς τους. Χρησιμοποιείται όταν η γκάμα των προϊόντων είναι μεγάλη ενώ η ζήτηση για κάθε τύπο προϊόντος είναι μικρή.

- Κυκλικό/επαναληπτικό σχέδιο (circular manufacturing layout), που χρησιμοποιείται όταν τόσο η γκάμα όσο και η ζήτηση για κάθε τύπο προϊόντος είναι μέση.

Η διαχείριση του συστήματος παραγωγής στοχεύει στον συντονισμό των διαφόρων ενεργειών με στόχο την ικανοποίηση των απαιτήσεων που έχουν τεθεί από τους πελάτες αλλά και την όσο το δυνατό βέλτιστη χρησιμοποίηση/εκμετάλλευση των διατιθέμενων πόρων. Η διαχείριση, εκτός των άλλων, περιλαμβάνει διαδικασίες σχεδιασμού (planning), χρονοπρογραμματισμού (scheduling) και παρακολούθησης/επίβλεψης (monitoring).

Ως *ευέλικτο σύστημα παραγωγής* (Flexible Manufacturing Systems, FMS) θεωρείται ένα σύστημα παραγωγής που μπορεί να προσαρμοστεί τόσο εύκολα όσο και γρήγορα στις διαρκώς μεταβαλλόμενες συνθήκες της αγοράς ([21]).

Ένα FMS μπορεί να ειπωθεί σαν ένα σύνολο από πόρους παραγωγής, μεταφοράς και αποθήκευσης ([1]). Οι πόροι αποθήκευσης χρησιμοποιούνται για την προσωρινή ή μόνιμη αποθήκευση των πρώτων υλών, των ενδιάμεσων αλλά και των τελικών προϊόντων. Οι πόροι παραγωγής είναι τα τμήματα των μηχανών που επεξεργάζονται τις πρώτες ύλες ή τα ενδιάμεσα προϊόντα ενώ οι πόροι μεταφοράς αναφέρονται στις περισσότερες φορές είτε σε ρομποτικές διατάξεις είτε σε οχήματα μεταφοράς (automatic guided vehicles - AGV) που χρησιμοποιούνται για τη μεταφορά των πρώτων υλών και των προϊόντων μεταξύ των αποθηκευτικών χώρων.

Σε ένα ευέλικτο σύστημα παραγωγής οι μηχανές ομαδοποιούνται χωρικά σε *κυψέλες* ή *κύτταρα* παραγωγής (manufacturing cells). Κάθε τέτοια κυψέλη χρησιμοποιείται για να επεξεργαστεί είτε πρώτες ύλες είτε ενδιάμεσα προϊόντα. Έτσι, σε κάθε τελικό προϊόν αντιστοιχεί μια ροή στις κυψέλες παραγωγής που ονομάζεται *διαδρομή παραγωγής* (production route).

Μαθηματικά, ένα ευέλικτο σύστημα παραγωγής μπορεί να οριστεί ως σύνολο έξι στοιχείων

$$\langle P, B, Y, C, M, H \rangle \quad (3.1)$$

όπου:

- Το P είναι ένα περατό και μη κενό σύνολο προϊόντων
- Το B είναι ένα περατό και μη κενό σύνολο αποθηκευτικών χώρων

- Το Y είναι ένα περατό και μη κενό σύνολο από πόρους
- Το C είναι ένα περατό και μη κενό σύνολο από κυψέλες
- Το M είναι ένα περατό και μη κενό σύνολο μηχανών
- Το H είναι ένα περατό και μη κενό σύνολο διαδρομών

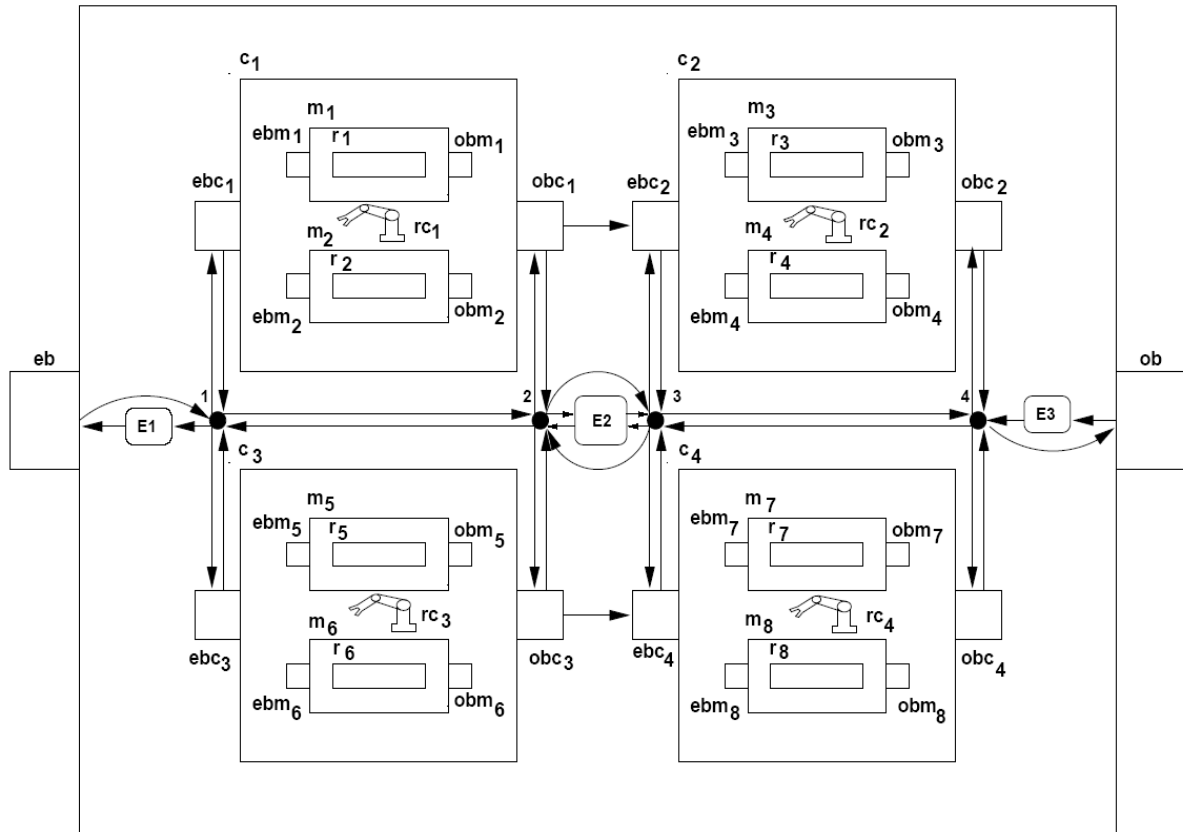
Σύμφωνα με τον παραπάνω ορισμό, μπορούν να εντοπιστούν τα στοιχεία ενός ευέλικτου συστήματος παραγωγής στο σχήμα 3.1. Πιο συγκεκριμένα, διακρίνονται τέσσερις κυψέλες παραγωγής (c_1, c_2, c_3, c_4) και οκτώ μηχανές (m_1, m_2, \dots, m_8) με τους αντίστοιχους πόρους παραγωγής (r_1, \dots, r_8). Σε κάθε κυψέλη και για κάθε μηχανή υπάρχει ένα ζευγάρι από αποθηκευτικούς χώρους (εισόδου (eb) και εξόδου (ob)). Τέλος, διακρίνονται και συστήματα μεταφοράς, δηλαδή ρομποτικές διατάξεις (r_{c_1}, \dots, r_{c_4}) και οχήματα μεταφοράς (AGVs). Τα πρώτα χρησιμοποιούνται για μεταφορές εντός μιας κυψέλης ενώ τα τελευταία αναλαμβάνουν τις μεταφορές από κυψέλη σε κυψέλη.

3.1.1 Μοντελοποίηση συστημάτων παραγωγής με δίκτυα Petri

Όπως αναφέρθηκε και στην παράγραφο 1.1, τα δίκτυα Petri έχουν χρησιμοποιηθεί στη μοντελοποίηση ευέλικτων συστημάτων παραγωγής ([18]). Για να αποφευχθεί όμως το πρόβλημα της «έκρηξης» του χώρου των καταστάσεων συχνά ενισχύεται η λεγόμενη αρθρωτή (modular) προσέγγιση και η σύνθεση του πολύπλοκου μοντέλου από μικρότερα μοντέλα που αναπαριστούν τα διάφορα υποσυστήματα. Με τον τρόπο αυτό, για τα θεμελιώδη υποσυστήματα (που συναντούνται συχνά με μικρό αριθμό αρχικών διαμορφώσεων) μπορεί να επαληθευτεί η σωστή τους λειτουργία με τη μέθοδο του χώρου των καταστάσεων.

Στο [7] προτείνονται οι ακόλουθες φάσεις για την ανάλυση πολύπλοκων συστημάτων παραγωγής με την αρθρωτή μέθοδο:

1. κατάτμηση του συστήματος σε μικρότερα στοιχειώδη τμήματα - αρθρώματα (modules) τα οποία μπορούν εύκολα να μοντελοποιηθούν και να αναλυθούν,
2. μοντελοποίηση των στοιχειωδών τμημάτων με χρήση δικτύων Petri



Σχήμα 3.1: Σχηματική αναπαράσταση ενός ευέλικτου συστήματος παραγωγής

3. έλεγχος των ποιοτικών ιδιοτήτων των μοντέλων που απαιτείται για τη χρήση τους σε συστήματα παραγωγής (π.χ. φραγή, ζωτικότητα κλπ.)
4. ανάλυση των ποσοτικών ιδιοτήτων κάθε τμήματος (όπως η παραγωγικότητα, ο ρυθμός χρησιμοποίησης των πόρων κλπ.)
5. ενσωμάτωση (integration) των διαφόρων υποσυστημάτων με τέτοιο τρόπο ώστε να διατηρούνται οι ποιοτικές ιδιότητες των στοιχειωδών μοντέλων
6. εξαγωγή των συμπερασμάτων για την συμπεριφορά του συστήματος βάσει της συμπεριφοράς των υποσυστημάτων

Ένα τυχαίας τοπολογίας σύστημα παραγωγής μπορεί να αναλυθεί σε τέσσερα σύνολα από στοιχειώδη τμήματα (modules) που αντιστοιχούν στα συστήματα γραμμής παραγωγής (production line) ή αλυσίδας μεταφοράς (transfer chain), συναρμο-

λόγησης (assembly), αποσυναρμολόγησης (disassembly) και παράλληλων μηχανών (parallel machines). Έτσι προτείνονται μοντέλα απλών δικτύων Petri τόσο για τις απλουστευμένες (generic) περιπτώσεις όπου έχουμε μικρό αριθμό εισόδων/εξόδων στους ενδιάμεσους χώρους αποθήκευσης (buffers) και μικρό αριθμό μηχανών (ένα ή δύο) όσο και για τις πιο πολύπλοκες περιπτώσεις (generalized) με n μηχανές και buffers με n εισόδους ή/και εξόδους. Τα συστήματα παραγωγής που εξετάζονται στην εργασία [19] ανήκουν στην κατηγορία αυτών με μηχανές αποκλειστικής χρήσης ή, αλλιώς, αφροσιωμένες μηχανές (dedicated machine production systems). Στα συστήματα αυτά, σε κάθε μηχανή έχει ανατεθεί μια συγκεκριμένη επεξεργασία ενός μόνο τύπου προϊόντος. Όλες οι πρώτες ύλες και τα ενδιάμεσα προϊόντα (parts) ακολουθούν την ίδια διαδρομή (route) περνώντας από τις ίδιες μηχανές με την ίδια σειρά με χρόνους επεξεργασίας που αναμένεται να είναι ίδιοι και τελικά ανακτώνται από τον ίδιο τελικό ενδιάμεσο αποθηκευτικό χώρο.

Το βασικό στοιχειώδες τμήμα που αναπαριστά τη γραμμή παραγωγής περιλαμβάνει μια μηχανή M_i , η οποία παίρνει μισοεπεξεργασμένα κομμάτια (items) από έναν προσωρινό χώρο αποθήκευσης (buffer) B_{ji} και μετά από επεξεργασία τα στέλνει σε έναν άλλο χώρο αποθήκευσης B_{il} . Κατά τη διαδικασία συναρμολόγησης μια μηχανή M_i παίρνει κομμάτια (parts) από δύο ή περισσότερους χώρους αποθήκευσης (B_{ji} ως B_{ki}), τα συναρμολογεί έτσι ώστε να αποτελέσουν μία μονάδα και τα στέλνει σε έναν αποθηκευτικό χώρο εξερχόμενης ροής (downstream buffer) B_{il} . Κατά την αποσυναρμολόγηση, μια μηχανή M_i παίρνει κάποια κομμάτια από έναν upstream buffer B_{ji} , τα διαχωρίζει σε δύο ή περισσότερα τμήματα και τα στέλνει στους αντίστοιχους αποθηκευτικούς χώρους (B_{ik} ως B_{il}) μετά την έξοδό τους από τη μηχανή. Τέλος, το άρθρωμα (module) των παράλληλων μηχανών περιλαμβάνει μηχανές με πιθανά διαφορετικές ρυθμίσεις και λειτουργίες να δουλεύουν για την παραγωγή του ίδιου προϊόντος ανεξάρτητα η μια από την άλλη, επεξεργαζόμενες διαφορετικά κομμάτια του προϊόντος αλλά πάντα του ίδιου τύπου.

Τα παραπάνω τέσσερα θεμελιώδη στοιχεία συνδεόμενα κατάλληλα μεταξύ τους μπορούν να αναπαραστήσουν δίκτυα παραγωγής πολλών διαφορετικών τύπων και μορφών ([19]).

3.1.2 Η φυσική σημασία των ιδιοτήτων των δικτύων Petri στα συστήματα παραγωγής

Απλά δίκτυα Petri

Κατά τη μοντελοποίηση των συστημάτων παραγωγής, οι μαθηματικές ιδιότητες που παρουσιάστηκαν στο κεφάλαιο 2 αποκτούν μια πιο συγκεκριμένη φυσική σημασία, κάτι που καθιστά την παρουσία (ή όχι) μιας ιδιότητας σημαντική ένδειξη για τη σωστή ή μη λειτουργία του αντίστοιχου συστήματος.

Έτσι, στην περίπτωση των απλών δικτύων Petri, οι θέσεις συνήθως χρησιμοποιούνται για να μοντελοποιηθούν χώροι αποθήκευσης τεμαχίων, εργαλείων, παλετών, και ΟΑΠ (Οχημάτων Αυτόνομης Πλοήγησης) σε ένα σύστημα διεργασιών. Μπορούν επίσης να αναπαραστήσουν την διαθεσιμότητα πόρων. Είναι, λοιπόν, πολύ σημαντικό να μπορεί να διαπιστωθεί ότι η προτεινόμενη μοντελοποίηση εμποδίζει την υπερχειλίση των αποθηκευτικών χώρων ή την υπερκατανάλωση των πόρων. Η φραγή ενός δικτύου Petri χρησιμοποιείται ακριβώς για να προσδιοριστεί αν υπάρχουν υπερχειλίσεις στο μοντελοποιημένο σύστημα. Η ιδιότητα της φραγής ερμηνεύεται επίσης ως σταθερότητα ενός διακριτού συστήματος κατεργασιών, ιδίως αν η μοντελοποίηση του περιλαμβάνει συστήματα ουρών.

Από την άλλη, όταν μια θέση ορίζει μια εργασία, η ιδιότητα της ασφάλειας στο δίκτυο εγγυάται ότι το σύστημα δεν θα προχωρήσει σε καμία προσπάθεια να εκκινήσει μια ήδη υπό εκτέλεση εργασία.

Στα πραγματικά συστήματα, ο αριθμός των πόρων που χρησιμοποιούνται συνήθως περιορίζεται από οικονομικούς ή άλλους παράγοντες. Αν για την αναπαράσταση των πόρων χρησιμοποιούνται μάρκες, ο αριθμός των οποίων είναι σταθερός, τότε ο αριθμός των μαρκών σε ένα δίκτυο Petri θα έπρεπε να είναι αμετάβλητος ανεξάρτητα από την σήμανση του δικτύου. Αυτό συνεπάγεται *συντηρητικότητα* καθώς οι πόροι δεν μπορούν ούτε να καταστραφούν ούτε να δημιουργηθούν, εκτός αν υπάρχει μια διάταξη τροφοδότησης που να επιτρέπει κάτι τέτοιο. Για παράδειγμα, ένα σπασμένο εργαλείο μπορεί να αφαιρεθεί από μια κυψέλη κατεργασιών με αποτέλεσμα την μείωση των διαθέσιμων εργαλείων κατά ένα. Η *αυστηρή συντηρητικότητα* σημαίνει ότι ο αριθμός των κουπονιών διατηρείται. Από δομικής πλευράς του δικτύου, η αυστηρή συντηρητικότητα σημαίνει ότι ο αριθμός των τόξων εισόδου σε μία μετάβαση πρέπει να ισούται με τον αριθμό των τόξων εξόδου. Η συντηρητικότητα σημαίνει ότι το

ζυγισμένο άθροισμα των τόξων παραμένει σταθερό για κάθε σήμανση.

Στα πραγματικά συστήματα, οι πόροι συχνά συνδυάζονται ώστε να επιτευχθεί κάποια συγκεκριμένη εργασία. Μετά το πέρας της εργασίας, χωρίζονται ξανά. Για παράδειγμα, ένα ρομπότ σηκώνει ένα τεμάχιο, το φορτώνει σε μια μηχανή για κατεργασία, και μετά το ξεφορτώνει. Μια μάρκα μπορεί να παριστάνει πολλούς πόρους που συνεργάζονται. Αυτή η μάρκα μπορεί να χρησιμοποιηθεί για να δημιουργηθούν πολλαπλές μάρκες (μία ανά πόρο) με την εκपुरσοκρότηση μιας μετάβασης με περισσότερα τόξα εξόδου από ότι εισόδου.

Καταστάσεις με *αδιέξοδα* είναι συνήθως αποτέλεσμα λανθασμένης κατανομής των πόρων ή εξάντλησης συγκεκριμένων τύπων πόρων. Σε ένα αυτοματοποιημένο σύστημα κατεργασιών πολλοί πόροι χρησιμοποιούνται από κοινού. Τέτοιοι πόροι μπορούν να είναι μηχανές, συστήματα χειρισμού υλικού συμπεριλαμβανομένων ρομπότ και ΟΑΠ ή/και χώροι αποθήκευσης. Για παράδειγμα, τα ρομπότ χρησιμοποιούνται από κοινού από διάφορες μηχανές, οι μηχανές από διαφορετικές ή ίδιες εργασίες, και οι αποθήκες από πρώτη ύλη, ενδιάμεσο, και τελικό προϊόν. Συνθήκες που οδηγούν σε *αδιέξοδα* μπορεί να είναι π.χ. η αμοιβαία απαγόρευση, όπου ένας πόρος είτε είναι διαθέσιμος είτε διατίθεται σε μία διεργασία που έχει αποκλειστική πρόσβαση σε αυτόν τον πόρο ή κυκλική αναμονή, όπου δύο ή περισσότερες διεργασίες είναι κανονισμένες σε αλυσίδα στην οποία η κάθε διεργασία περιμένει τους πόρους που είναι κρατημένοι από την επόμενη διεργασία της αλυσίδας.

Εκτός αν υπάρχει μια διάταξη στο λογισμικό του ελέγχου για εντοπισμό και διόρθωση του *αδιεξόδου*, μια τέτοια κατάσταση, που συνήθως συμβαίνει σε ένα μικρό υποσύστημα του δικτύου, μπορεί να διογκωθεί και να επηρεάσει ένα μεγάλο κομμάτι του συστήματος. Αυτό συνήθως συνεπάγεται ολοκληρωτική στασιμότητα του συστήματος.

Η *ζωτικότητα* ενός δικτύου Petri σημαίνει ότι από μια οποιαδήποτε σήμανση m προσβάσιμη από την αρχική m_0 , είναι απολύτως πιθανό να εκपुरσοκροτήσει οποιαδήποτε μετάβαση μέσα στο δίκτυο προχωρώντας μέσα από ακολουθίες ενεργοποιήσεων. Γι' αυτό τον λόγο, αν ένα δίκτυο Petri είναι *ζωτικό*, δεν είναι δυνατόν να εμφανισθεί *αδιέξοδο*.

Η *αντιστρεπτότητα* ενός δικτύου Petri σημαίνει ότι για μια οποιαδήποτε σήμανση m προσβάσιμη από την m_0 , η m_0 είναι επίσης προσβάσιμη από την m . Πολλά συστήματα απαιτείται να επιστρέφουν από μια κατάσταση αποτυχίας στις επερχό-

μενες σωστές καταστάσεις. Γι αυτό τον λόγο, η συγκεκριμένη ιδιότητα είναι πολύ σημαντική για την ανάκαμψη από λάθος ενός συστήματος κατεργασιών. Επιπροσθέτως, η ιδιότητα εγγυάται την κυκλική συμπεριφορά του συστήματος, κάτι το οποίο απαιτείται για όλα τα επαναληπτικά συστήματα κατεργασιών. Αυτή η απαίτηση είναι στενά συνδεδεμένη με τις ιδιότητες της αντιστρεπτότητας και της αρχικής κατάστασης ενός δικτύου Petri.

Έγχρωμα δίκτυα Petri

Στο πεδίο των έγχρωμων δικτύων Petri μερικές από τις παραπάνω ιδιότητες αποκτούν νέο νόημα ή χάνουν τη «δύναμή» τους ως προς την ένδειξη για επίδειξη συγκεκριμένης συμπεριφοράς από τα μοντελοποιούμενα συστήματα.

Αν για κάθε θέση υπάρχει τόσο ελάχιστο όσο και μέγιστο όριο αυτό σημαίνει ότι το μοντέλο είναι φραγμένο, κάτι που στην περίπτωση των μοντέλων συστημάτων παραγωγής είναι βέβαια το επιθυμητό μιας και αποφεύγονται οι υπερσυγκεντρώσεις μαρκών/προϊόντων ή μαρκών/ενεργειών σε συγκεκριμένα σημεία στο σύστημα.

Ως προς τις αρχικές (home) σημάνσεις, στα συστήματα που εξετάζουμε δεν είναι πάντα θετικό να υπάρχουν τέτοιες σημάνσεις στο δίκτυο μιας και πολλές φορές αυτές αφορούν αδικαιολόγητους βρόχους (loops). Μια τέτοια περίπτωση είναι και αυτή π.χ. της αναπαράστασης του μοντέλου προϊόντων (product model), όπου δεν έχει νόημα μια τέτοια ιδιότητα μιας και η συναρμολόγηση ενός προϊόντος από πρώτες ύλες ή άλλα ενδιάμεσα προϊόντα δεν εμπλέκει κάποια διαδικασία αποσύνθεσης και επαναφοράς στην αρχική κατάσταση.

Η παραπάνω ανάλυση της φυσικής σημασίας των διαφόρων ιδιοτήτων των δικτύων Petri σε συνδυασμό με τις αναφορές που προκύπτουν από τα εργαλεία ανάλυσης όπως παρουσιάστηκαν στην παράγραφο 2.4 θα βοηθήσουν στην αξιολόγηση των θεμελιωδών μοντέλων που θα παρουσιαστούν στο κεφάλαιο 4.

Κεφάλαιο 4

Μοντελοποίηση θεμελιωδών στοιχείων

Στις επόμενες παραγράφους θα παρουσιαστούν τα μοντέλα των θεμελιωδών στοιχείων ενός συστήματος παραγωγής χρησιμοποιώντας τα έγχρωμα δίκτυα Petri όπως αυτά παρουσιάστηκαν στο προηγούμενο κεφάλαιο. Η προτεινόμενη μοντελοποίηση δεν είναι απαραίτητα η βέλτιστη, όμως παρουσιάζονται τα αποτελέσματα των διαφόρων αναλύσεων και ελέγχεται η λειτουργία των μοντέλων σε σχέση με την αναμενόμενη.

Κύριος στόχος της ανάπτυξης των μοντέλων των βασικών στοιχείων ενός συστήματος παραγωγής, όπως παρουσιάστηκε και προηγουμένως, είναι η επαναχρησιμοποίησή τους ως δομικά συστατικά ενός πιο σύνθετου μοντέλου. Γι' αυτό, βασική τους ιδιότητα θα πρέπει να είναι η επικοινωνία τους με άλλα στοιχεία κάτι που σημαίνει ότι κάποιες θέσεις και μεταβάσεις θα παίζουν το ρόλο της διεπαφής με άλλα υποσυστήματα. Έτσι προβλέπονται ενδιάμεσοι χώροι αποθήκευσης (buffers) εισόδου και εξόδου.

Για τη αναπαράσταση των θεμελιωδών αυτών μοντέλων γίνεται εξαρχής χρήση χρονικών (timed) δικτύων. Οι περισσότερες ενέργειες (μεταβάσεις) είναι συνδεδεμένες με μια χρονική διάρκεια t (και άρα δεν υπάρχει εξ' ορισμού παραδοχή ως προς την ακαριαία εκτέλεσή τους) με αποτέλεσμα η μοντελοποίηση να είναι πιο ρεαλιστική αλλά και ο αριθμός των συγκρούσεων που πρέπει να επιλυθούν σε κάθε βήμα εκτέλεσης να είναι σημαντικά μικρότερος.

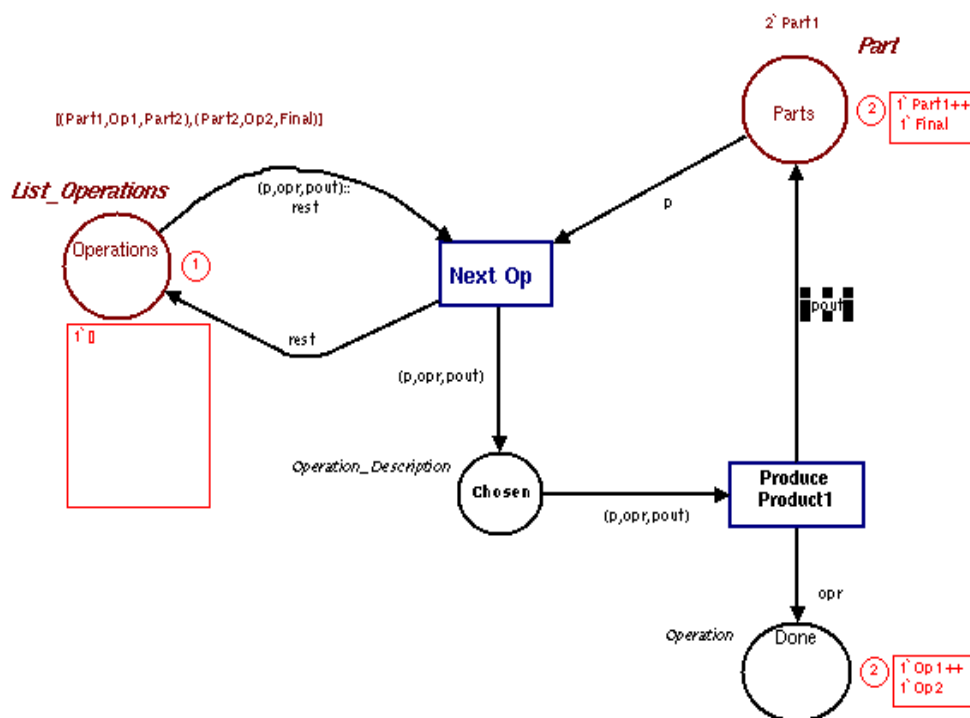
4.1 Μοντελοποίηση διαδικασίας παραγωγής

Κατά την κατασκευή (manufacturing) ενός προϊόντος η πρώτη ύλη χρειάζεται μια σειρά από επεξεργασίες ή κατεργασίες (operations), καθεμιά από τις οποίες απαιτεί κάποιο χρόνο για την εκτέλεσή της. Το σύνολο των επεξεργασιών ονομάζεται δρομολόγηση (routing) του προϊόντος ενώ σε συνδυασμό με τους χρόνους εκτέλεσης της κάθε επεξεργασίας συνιστούν τη διαδικασία παραγωγής (manufacturing process). Στις γραμμικές (linear) διαδικασίες παραγωγής οι παραπάνω διεργασίες πραγματοποιούνται διαδοχικά ή μια μετά την άλλη. Μια τέτοια περίπτωση αναπαριστάται με το μοντέλο του σχήματος 4.1, όπου κάθε ενδιάμεσο προϊόν υπόκειται μια κατεργασία και το νέο προϊόν τοποθετείται στον αποθηκευτικό χώρο ώστε να υποστεί την επόμενη επεξεργασία.

Η θέση Operations περιέχει μάρκες με τις περιγραφές όλων των ενεργειών που

πρέπει να γίνουν. Κάθε ενέργεια έχει μια ονομασία, αναφέρεται σε έναν συγκεκριμένο τύπο προϊόντος και έχει σαν αποτέλεσμα της ένα άλλο ενδιάμεσο προϊόν. Για να επιλεγεί μια κατεργασία (και να προωθηθεί στη θέση **Chosen**) θα πρέπει το κατάλληλο προϊόν/πρώτη ύλη να βρίσκεται στον κατάλληλο προσωρινό αποθηκευτικό χώρο (στη θέση **Parts**).

Διαφορετικά, ή όλη διαδικασία σταματά μέχρι να γίνει διαθέσιμο το κατάλληλο προϊόν που θα χρησιμοποιηθεί ως πρώτη ύλη. Αφού επιλεγεί κάποια κατεργασία, το τελικό προϊόν επανεισάγεται στον προσωρινό αποθηκευτικό χώρο για να χρησιμοποιηθεί ως πρώτη ύλη σε κάποια επόμενη κατεργασία. Η κατεργασία που ολοκληρώνεται επιτυχώς «καταγράφεται» (μέσω μιας μάρκας που σαν τιμή έχει τον τύπο της κατεργασίας) στη θέση **Done**.



Σχήμα 4.1: Μοντέλο γραμμικής διαδικασίας παραγωγής

Στη γενικότερη περίπτωση, όμως, κατά την παραγωγή εμπλέκονται διαδικασίες

```

use "/CPN_EXAMPLES/tod.sml";
color Int = int;

color Part = with Part1 | Part2 | Final;
color Product = with Product1 | Product2;
color Operation = with Op1|Op2|Op3;
color Operation_Description = product Part*Operation*Part;
color List_Operations = list Operation_Description;

var rest>List_Operations;
var p:Part;
var pout:Part;
var opr:Operation;

```

Σχήμα 4.2: Ορισμοί για το μοντέλο παραγωγής προϊόντων

συναρμολόγησης και αποσυναρμολόγησης οπότε η όλη διαδικασία παύει να είναι γραμμική. Μια τέτοια διαδικασία αναπαρίσταται συνήθως με τη χρήση της δενδρικής δομής Bills of Material(BOM). Για να αποτυπωθεί μια τέτοια δομή με τη βοήθεια των απλών δικτύων Petri απαιτείται μια αρκετά πολύπλοκη και σχετικά στατική αποτύπωση των σχέσεων μέσω των μεταβάσεων των μαρκών (που αναπαριστούν τη διαθεσιμότητα ή όχι μιας πρώτης ύλης ή ενός ενδιάμεσου προϊόντος) από θέση σε θέση.

Με τη χρήση όμως των έγχρωμων δικτύων Petri, η παραπάνω δομή μπορεί να αναπαρασταθεί με ένα σύνολο από μάρκες ενός συγκεκριμένου σύνθετου τύπου που μπορεί να οριστεί με τη βοήθεια της γλώσσας περιγραφής SML.

Η προσέγγιση αυτή υιοθετείται και στην εργασία [20] για να μοντελοποιηθεί ένα σύστημα ροής εργασιών (workflow), στην οποία προτείνεται ένα μοντέλο προϊόντος (product data model) όπου κάθε μάρκα έχει τύπο της μορφής:

$$("Operationid", product, [item_1, item_2, item_x], (attributes)) \quad (4.1)$$

Με τον τρόπο αυτό, κάθε μάρκα αναπαριστά τα στοιχεία ενός επεξεργαστικού σταδίου για την επίτευξη ενός συγκεκριμένου στόχου/προϊόντος. Μαζί με τον χαρακτηρισμό (ονομασία) της ενέργειας καθορίζεται το προϊόν-αποτέλεσμα της συγκεκριμένης ενέργειας, μια λίστα από προαπαιτούμενα στοιχεία για την έναρξη της

διαδικασίας καθώς επίσης και οι ιδιότητες της ενέργειας ως προς τη χρονική διάρκειά της, το κόστος της κ.λπ. Μια ενέργεια μπορεί να «εκτελεστεί» μόνο αν όλα τα προαπαιτούμενα στοιχεία είναι διαθέσιμα, ενώ ένα τελικό προϊόν παράγεται όταν όλες οι ενέργειες που το αφορούν (το θέτουν ως αποτέλεσμα) «εκτελεστούν».

Επιπλέον, με τον τρόπο αυτό υπάρχει η δυνατότητα να οριστούν μέθοδοι (με τη μορφή συναρτήσεων) που να υπολογίζουν ποιές ενέργειες είναι δυνατόν να εκτελεστούν βάσει των διαθέσιμων στοιχείων και των ποσοτήτων και άρα ποια άλλα στοιχεία (ενδιάμεσα προϊόντα) είναι δυνατόν να παραχθούν. Ακόμη, σε περίπτωση που πολλές ενέργειες είναι δυνατόν να εκτελεστούν, μπορεί να τεθεί προτεραιότητα σε κάποιες ή να εισαχθεί λογική βελτιστοποίησης της σειράς των ενεργειών βάσει της ελαχιστοποίησης του κόστους ή της χρονικής καθυστέρησης της παραγωγής. Η σειρά των ενεργειών με στοιχεία όπως το όνομα της ενέργειας, η διάρκειά της και το κόστος της, μπορούν να καταγραφούν ώστε να χρησιμοποιηθούν ως αντικείμενο μελέτης κατά την εκτέλεση προσομοιώσεων για την συμπεριφορά και επίδοση του μοντέλου. Τέλος με αυτόν τον τρόπο, ένα προϊόν μπορεί να παράγεται με πολλές διαφορετικές δρομολογήσεις καθεμιά από τις οποίες μπορεί να εμπλέκει διαφορετικές διεργασίες ή/και πρώτες ύλες.

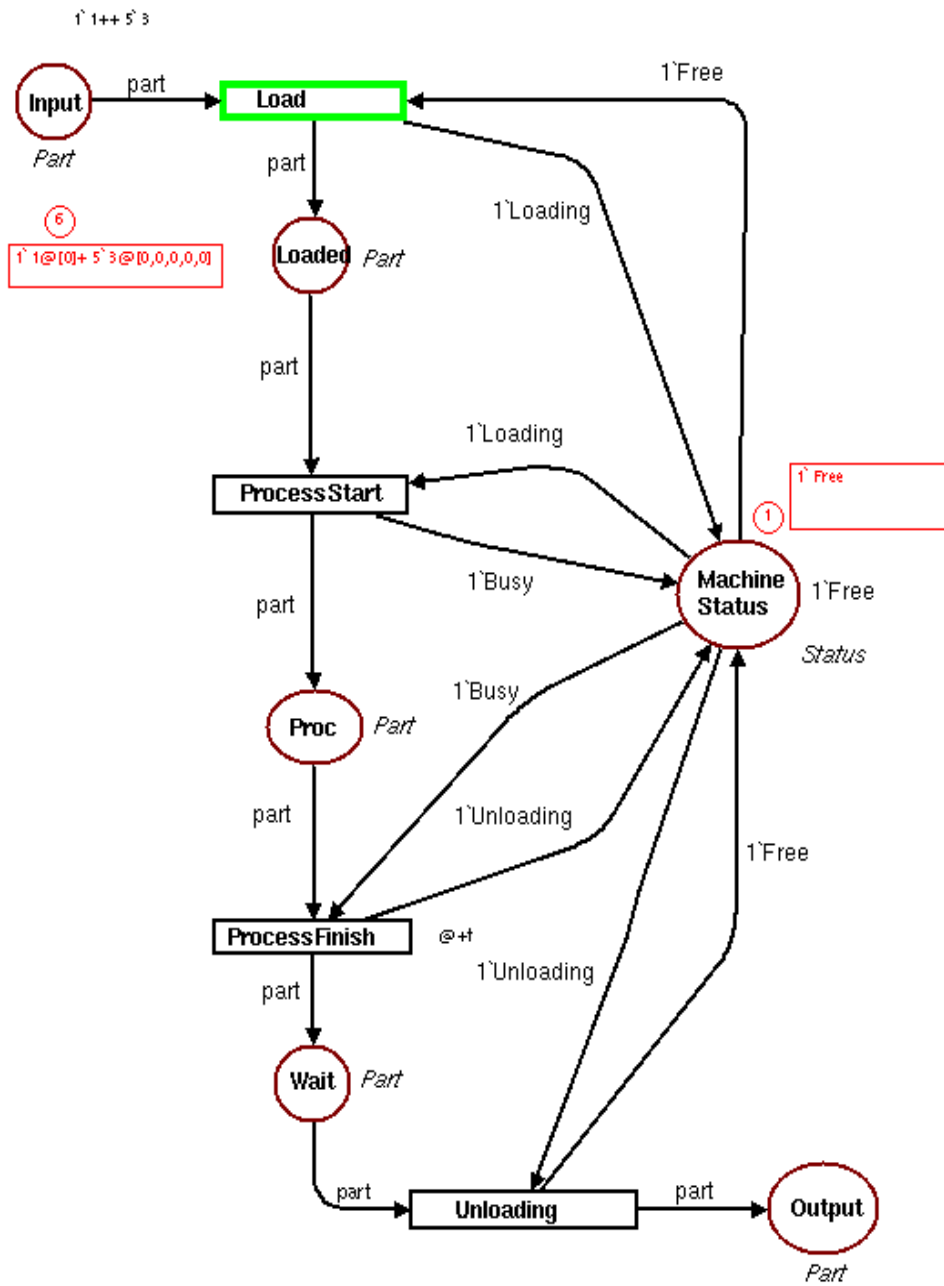
4.2 Μοντελοποίηση μηχανών

4.2.1 Αποκλειστικές μηχανές

Συχνά παρατηρείται η ανάγκη αποκλειστικής ενασχόλησης μιας μηχανής με ένα συγκεκριμένο τύπο προϊόντος λόγω π.χ. της αυξημένης ζήτησής του.

Στο σχήμα 4.3 απεικονίζεται το μοντέλο μιας τέτοιας μηχανής. Ένα μόνο προϊόν μπορεί κάθε φορά να φορτωθεί και να οδηγηθεί προς επεξεργασία. Η κατεργασία στην οποία υπόκειται διαρκεί τον ίδιο χρόνο κάθε φορά (στη συγκεκριμένη περίπτωση 5 χρονικές μονάδες). Η μάρκα στη θέση **Machine Status** αντικατοπτρίζει κάθε χρονική στιγμή την κατάσταση της μηχανής (που μπορεί είτε να είναι ανενεργή (**Free**), σε φάση φόρτωσης (**Loading**), ξεφορτώματος (**Unloading**) ή να είναι απασχολημένη (**Busy**) με την επεξεργασία κάποιου προϊόντος.

Στο ίδιο σχήμα διακρίνονται ακόμη θέσεις που αναπαριστούν buffers προσωρινής αποθήκευσης του προϊόντος κατά την είσοδο (**Loaded**), επεξεργασία (**Proc**) και



Σχήμα 4.3: Μοντέλο αποκλειστικής μηχανής

```
use "/CPN_EXAMPLES/tod.sml";

color Int = int;
color Part = Int timed;
color Status = with Loading|Busy|Unloading|Free;

val t = 5;

var part: Part;
```

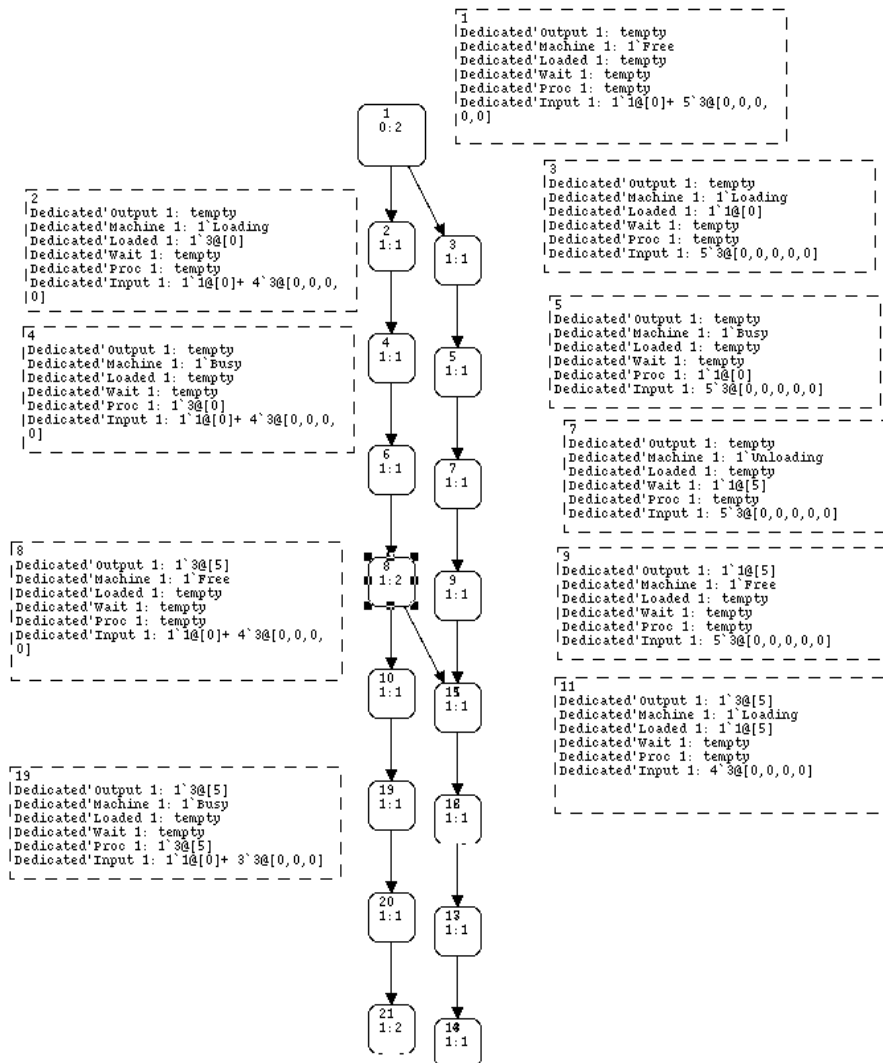
Σχήμα 4.4: Ορισμοί για το μοντέλο της αποκλειστικής μηχανής

αναμονή για την έξοδο από τη μηχανή (*Wait*), μεταβάσεις που αναπαριστούν τα διάφορα στάδια επεξεργασίας (*ProcessStart*, *ProcessFinish*) καθώς επίσης και η σύνδεση της τελευταίας με μια συγκεκριμένη χρονική διάρκεια *t*.

Στόχος της συγκεκριμένης μηχανής είναι η αποκλειστική ενασχόληση με το συγκεκριμένο προϊόν, γι' αυτό και δεν επιτρέπεται η είσοδος άλλου προϊόντος όσο αυτή επεξεργάζεται κάποιο (δηλαδή δεν είναι σε κατάσταση διαθεσιμότητας), όπως αυτό δηλώνεται με την απαίτηση ύπαρξης μιας μάρκας τύπου *Status* με τιμή *Free* για την ενεργοποίηση της διαδικασίας φόρτωσης ενός νέου προϊόντος προς επεξεργασία.

Ανάλυση χώρου καταστάσεων

Όπως φαίνεται και από το μέγιστο αριθμό μαρκών στις θέσεις *Loaded*, *Proc* στην αναφορά 4.2.1, η ενασχόληση της μηχανής είναι όντως αποκλειστική σε ένα προϊόν. Η θέση της κατάστασης περιέχει πάντα μια μάρκα όπως είναι αναμενόμενο. Οι νεκρές σημάνσεις αφορούν τις τελικές καταστάσεις όπου όλα τα προϊόντα έχουν επεξεργαστεί και τα τελικά προϊόντα βρίσκονται στην έξοδο της μηχανής. Αν και οι τελικές αυτές καταστάσεις αντιστοιχούν στα ίδια τελικά προϊόντα, η αρίθμησή τους είναι διαφορετική λόγω της διαφορετικής χρονικής στιγμής άφιξης των προϊόντων στην έξοδο της μηχανής ανάλογα με τη σειρά που επιλέχθηκαν για επεξεργασία.



Σχήμα 4.5: Τμήμα του γράφου καλυψιμότητας του μοντέλου αποκλειστικής μηχανής

4.2.2 Μηχανές με πολλαπλές λειτουργίες

Το προηγούμενο μοντέλο δεν καλύπτει την περίπτωση των μηχανών που μπορούν να επιτελούν περισσότερες από μια λειτουργίες. Για να αντιμετωπιστεί αυτό, προτείνεται το μοντέλο του σχήματος 4.7, όπου η μηχανή μπορεί να επιτελέσει πολλές διαφορετικές λειτουργίες (π.χ. P_1 , P_2 κλπ.) αλλά όχι ταυτόχρονα.

Ο ρόλος των θέσεων και των μεταβάσεων είναι παρόμοιος με το προηγούμενο μοντέλο. Εξακολουθεί η μηχανή να επεξεργάζεται μόνο ένα προϊόν κάθε φορά. Στο

```

Statistics
-----
Occurrence Graph
Nodes: 105
Arcs: 104
Secs: 0
Status: Full

Scc Graph
Nodes: 105
Arcs: 104
Secs: 0

Boundedness Properties
-----
Best Integers Bounds      Upper      Lower
New'Input 1              6          0
New'Loaded 1             1          0
New'Machine 1           1          1
New'Output 1            6          0
New'Proc 1               1          0
New'Wait 1              1          0

Best Upper Multi-set Bounds
New'Input 1             1'1++ 5'3
New'Loaded 1           1'1++ 1'3
New'Machine Status 1   1>Loading++ 1'Busy++ 1'Unloading++ 1'Free
New'Output 1           1'1++ 5'3
New'Proc 1             1'1++ 1'3
New'Wait 1             1'1++ 1'3

Best Lower Multi-set Bounds
New'Input 1            empty
New'Loaded 1           empty
New'Machine 1         empty
New'Output 1          empty
New'Proc 1            empty
New'Wait 1            empty

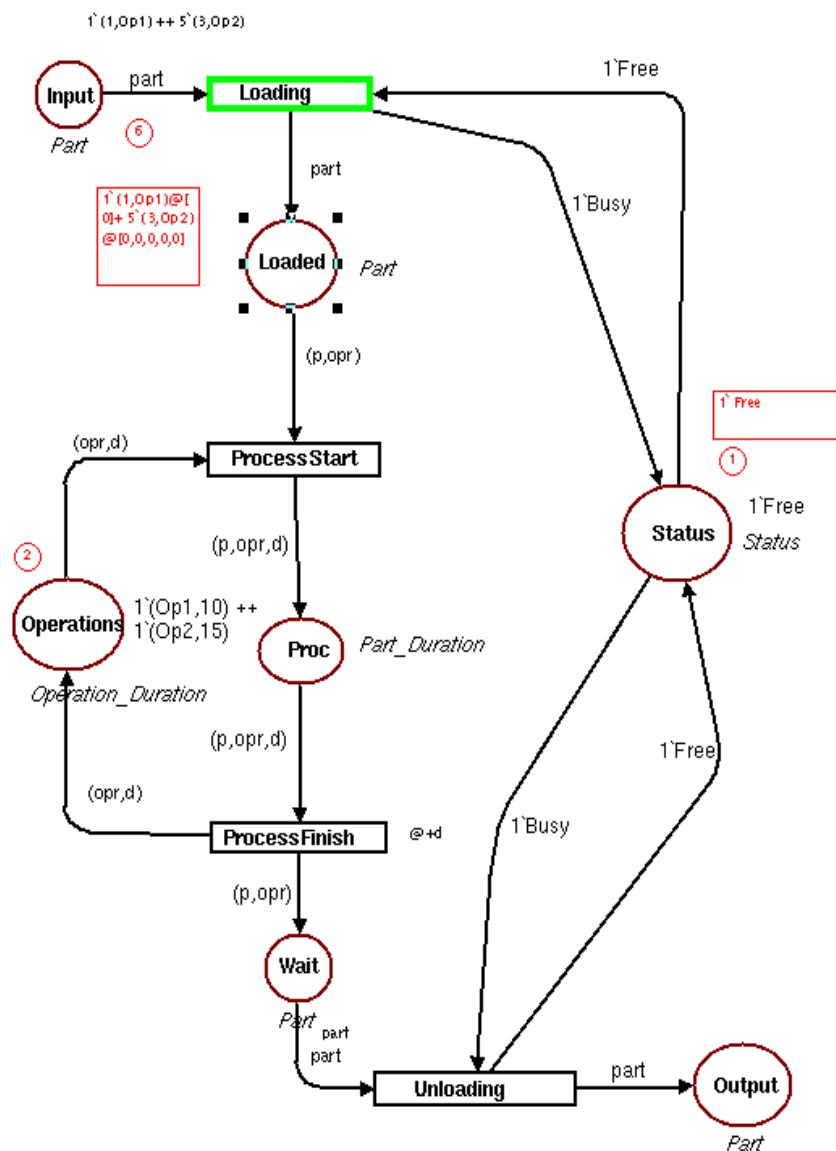
Home Properties
-----
Home Markings: Initial Marking is not a home marking

Liveness Properties
-----
Dead Markings: 6 [97,96,89,88,105,...]
Dead Transitions Instances: None

```

Σχήμα 4.6: Αναφορά της ανάλυσης του χώρου καταστάσεων για το μοντέλο αποκλειστικής μηχανής

συγκεκριμένο μοντέλο έχει οριστεί μια συγκεκριμένη σειρά ενεργειών, όμως στη γενικότερη περίπτωση το ποια ενέργεια θα εκτελεστεί πρώτα μπορεί να ειπωθεί σαν ένα πρόβλημα (χρονο)προγραμματισμού που κάποιο άλλο σύστημα σε υψηλότερο



Σχήμα 4.7: Μοντέλο μηχανής πολλαπλών λειτουργιών

επίπεδο πρέπει να διευθετήσει (σε συνεργασία με το μοντέλο παραγωγής προϊόντων) με στόχο τη βέλτιστη συμπεριφορά της μηχανής ως προς το συνολικό χρόνο παραγωγής ή άλλες παραμέτρους.

Οι λειτουργίες που μπορεί να επιτελέσει η μηχανή περιγράφονται από τις μάρκες

```

use "/CPN_EXAMPLES/tod.sml";

color Int = int;
color Part_Type = Int timed;
color Duration = Int;
color Status = with Busy|Free;
color Operation = with Op1|Op2|Op3;
color Operation_Duration = product Operation * Duration;
color Part = product Part_Type * Operation;
color Parts = list Part;
color Part_Duration= product Part_Type * Operation * Duration;

var part: Part;
var opr:Operation;
var p:Part_Type;
var l:Parts;
var d:Duration;

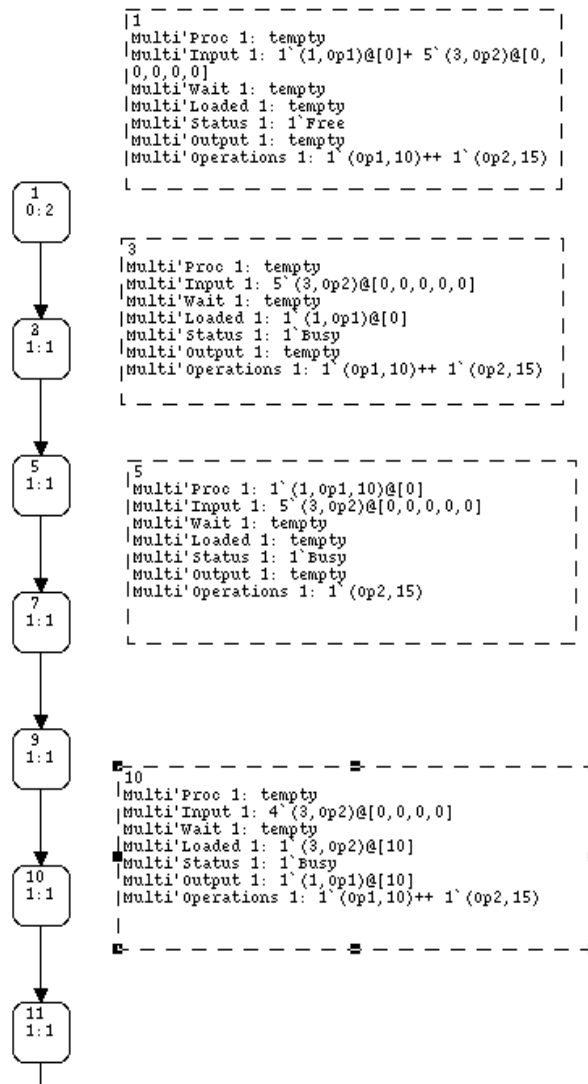
```

Σχήμα 4.8: Ορισμοί για το μοντέλο της μηχανής πολλαπλών λειτουργιών

που είναι στη θέση `Operations` και καθεμιά από αυτές περιέχει και πληροφορία για τη χρονική διάρκεια της αντίστοιχης επεξεργασίας. Για κάθε προϊόν που εμφανίζεται στην είσοδο της μηχανής στη θέση (`Input`), θα πρέπει να υπάρχει πληροφορία για το ποια επεξεργασία θα υποστεί. Βάσει αυτής (και εφόσον η μηχανή υποστηρίζει τη λειτουργία αυτή και είναι βεβαίως ελεύθερη) αποφασίζεται η φόρτωση του κομματιού προς επεξεργασία.

Όπως τονίστηκε και σε προηγούμενα κεφάλαια, η ανάλυση του χώρου καταστάσεων εξαρτάται από την αρχική σήμανση. Για να είναι όσο το δυνατόν πληρέστερη η ανάλυση αυτή θα πρέπει να δοθεί προσοχή ώστε η αρχική σήμανση να είναι αντιπροσωπευτική της λειτουργίας του μοντέλου υπό πραγματικές συνθήκες. Στο μοντέλο του σχήματος 4.7, για παράδειγμα, αν φτάσει ένα κομμάτι για να υποστεί μια επεξεργασία που δεν υποστηρίζεται από τη συγκεκριμένη μηχανή, θα πρέπει να ληφθεί μέριμνα πριν φορτωθεί στη μηχανή μιας και αν φορτωθεί το σύστημα θα μπλοκάρει. Ακόμη θα πρέπει να προβλέπεται διαδρομή αναδρομολόγησης των προϊόντων αυτών σε άλλες μηχανές ώστε να μην παραμένουν διαρκώς στη θέση εισόδου της συγκεκριμένης μηχανής. Ένα ολοκληρωμένο μοντέλο θα πρέπει να αναμένει όλες αυτές

τις καταστάσεις και να προβλέπει μια σωστή «αντίδραση».



Σχήμα 4.9: Τμήμα του γράφου καλυψιμότητας του μοντέλου μηχανής πολλαπλών λειτουργιών

Η βασική διαφορά σε σχέση με το προηγούμενο μοντέλο της αποκλειστικής μηχανής φαίνεται από τους διαφορετικούς τύπους ενεργειών και άρα διαφορετικών μαρκών που μπορεί να βρεθούν στη θέση Proc, όπως καταγράφεται στην αναφορά του σχήματος 4.2.2. Η θέση της κατάστασης (Status) περιέχει πάντα μια μάρ-

```

Statistics
-----
Occurrence Graph
Nodes: 105
Arcs: 104
Secs: 0
Status: Full

Scc Graph
Nodes: 105
Arcs: 104
Secs: 0

Boundedness Properties
-----
Best Integers Bounds      Upper      Lower
New'Input 1               6          0
New'Loaded 1              1          0
New'Operations 1          2          1
New'Output 1              6          0
New'Proc 1                 1          0
New'Status 1              1          1
New'Wait 1                 1          0

Best Upper Multi-set Bounds
New'Input 1               1'(1,0p1)++ 5'(3,0p2)
New'Loaded 1              1'(1,0p1)++ 1'(3,0p2)
New'Operations 1          1'(0p1,10)++ 1'(0p2,15)
New'Output 1              1'(1,0p1)++ 5'(3,0p2)
New'Proc 1                 1'(1,0p1,10)++ 1'(3,0p2,15)
New'Status 1              1'Busy++ 1'Free
New'Wait 1                 1'(1,0p1)++ 1'(3,0p2)

Best Lower Multi-set Bounds
New'Input 1               empty
New'Loaded 1              empty
New'Operations 1          empty
New'Output 1              empty
New'Proc 1                 empty
New'Status 1              empty
New'Wait 1                 empty

Home Properties
-----
Home Markings: Initial Marking is not a home marking

Liveness Properties
-----
Dead Markings: 6 [97,96,89,88,105,...]
Dead Transitions Instances: None

```

Σχήμα 4.10: Αναφορά της ανάλυσης του χώρου καταστάσεων για το μοντέλο μηχανής πολλαπλών λειτουργιών

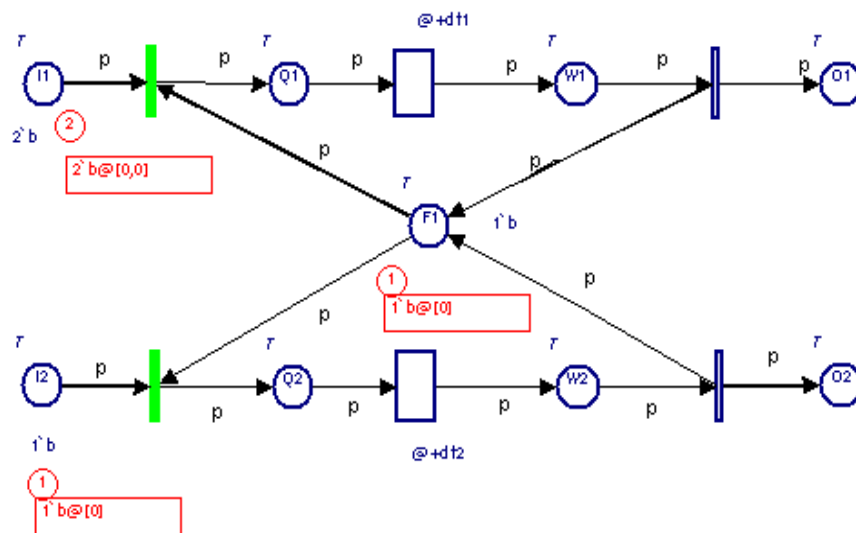
κα όπως είναι επιθυμητό. Κι εδώ οι νεκρές σημάνσεις αντιστοιχούν στην τελική

κατάσταση του μοντέλου όπου όλα τα προϊόντα έχουν επεξεργαστεί, με κάθε διαφορετική νεκρή σήμανση να αντιστοιχεί στη διαφορετική χρονική σειρά με την οποία επιλέγονται οι ενέργειες προς διεκπεραίωση.

Μοντελοποίηση με απλά δίκτυα Petri

Στο σχήμα 4.11 παρουσιάζεται το μοντέλο μηχανής πολλαπλών λειτουργιών με χρήση απλών δικτύων Petri. Όπως μπορεί να παρατηρήσει κανείς και από το τμήμα δηλώσεων του δικτύου 4.2.2 υπάρχουν μάρκες μόνο ενός τύπου. Έτσι, για την αναπαράσταση της διαδικασίας φόρτωσης, επεξεργασίας και εκφόρτωσης των διαφόρων προϊόντων απαιτούνται διαφορετικές θέσεις για κάθε τύπο προϊόντος (εδώ παρουσιάζονται μόνο δύο τύποι προϊόντων). Ο περιορισμός αυτός εισάγει ζήτημα εύκολης κλιμάκωσης (scaling) του μοντέλου για την κάλυψη πιο γενικών περιπτώσεων.

Από την αναφορά του σχήματος 4.2.2 παρατηρείται εύκολα ο μικρός αριθμός των κόμβων του γράφου καλυψιμότητας, παρόλο που ο αριθμός των κόμβων του ίδιου του μοντέλου είναι στα επίπεδα αυτού του αντίστοιχου έγχρωμου.



Σχήμα 4.11: Μοντέλο μηχανής πολλαπλών λειτουργιών με απλά δίκτυα Petri

```

use "/CPN_EXAMPLES/tod.sml";

color T = with b timed;
var p:T;

val dt1=2;
val dt2=3;

```

Σχήμα 4.12: Ορισμοί για το μοντέλο της μηχανής πολλαπλών λειτουργιών με χρήση απλών δικτύων Petri

```

Statistics
-----
Occurrence Graph
Nodes: 25
Arcs: 24
Secs: 0
Status: Full

(...)

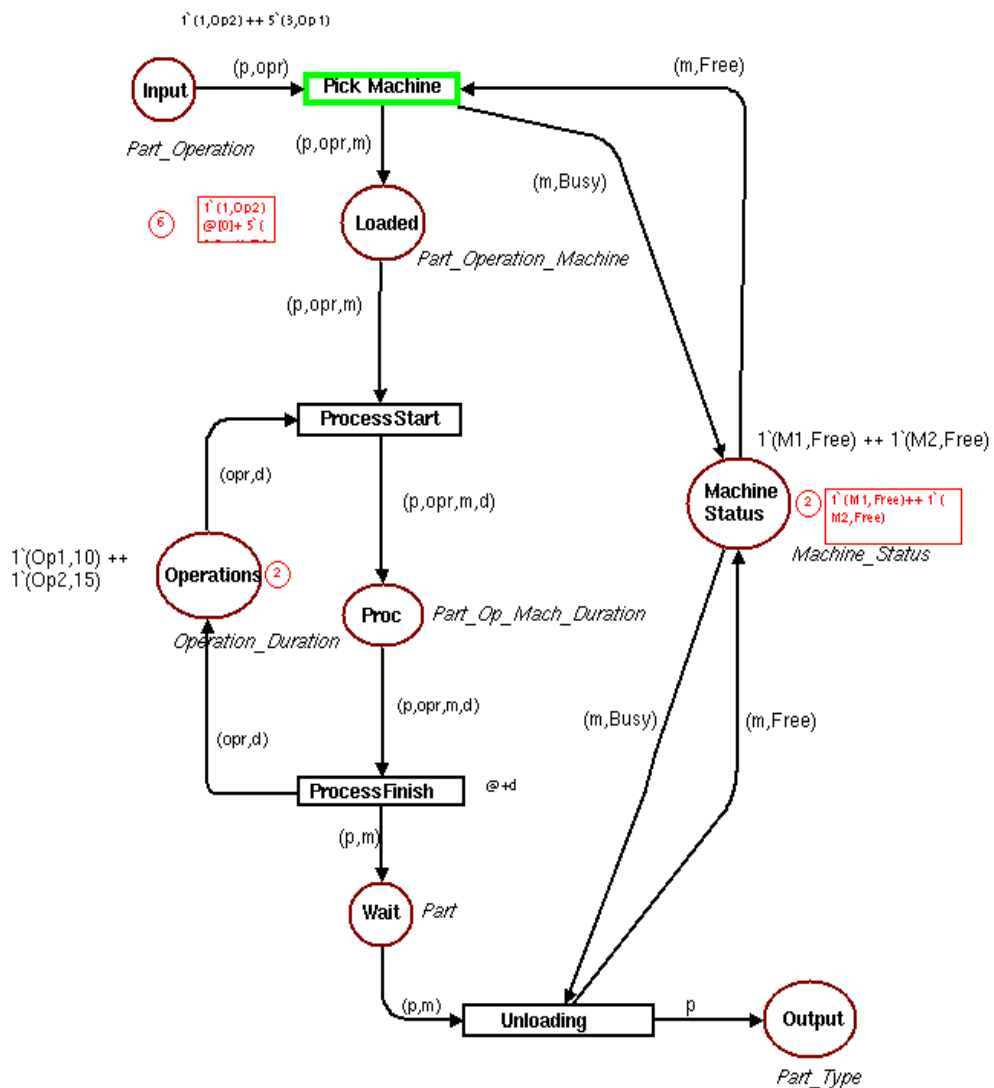
```

Σχήμα 4.13: Τμήμα της αναφοράς της ανάλυσης του χώρου καταστάσεων για το μοντέλο μηχανής πολλαπλών λειτουργιών με απλά δίκτυα Petri

4.2.3 Μηχανές με παράλληλες λειτουργίες

Μια ακόμη πιο ρεαλιστική ανάγκη που πρέπει να καλυφθεί είναι και η περίπτωση των πολλαπλών μηχανών που εργάζονται παράλληλα. Εδώ μπορεί να τεθεί ένας διαχωρισμός αν οι μηχανές είναι ίδιου τύπου ή όχι. Στην πρώτη περίπτωση που απεικονίζεται στο σχήμα 4.14, ο αριθμός των μαρκών που βρίσκονται στη θέση Proc κάθε στιγμή αναπαριστά τις ταυτόχρονες κατεργασίες.

Κάθε μάρκα περιλαμβάνει πληροφορία και για το ποια μηχανή έχει αναλάβει ποια κατεργασία και τίνος προϊόντος (ανάθεση εργασιών). Καθεμιά μηχανή εκτελεί μια κατεργασία κάθε χρονική στιγμή. Η κατάσταση τους αναπαριστάται από την αντίστοιχη μάρκα στη θέση Machine Status. Η θέση αυτή περιλαμβάνει τόσες μάρκες όσες είναι και οι μηχανές του συστήματος.



Σχήμα 4.14: Μοντέλο παράλληλων μηχανών ίδιου τύπου

Ανάλυση χώρου καταστάσεων

Η βασική διαφορά σε σχέση με το μοντέλο της αποκλειστικής μηχανής φαίνεται στο μέγιστο αριθμό μαρκών που μπορεί να βρεθούν στη θέση Proc στην αναφορά 4.2.3, ο οποίος είναι μεγαλύτερος της μονάδας. Η θέση της κατάστασης περιέχει πάντα μια μάρκα όπως είναι επιθυμητό.

```

use "/CPN_EXAMPLES/tod.sml";

color Int = int;
color Part_Type = Int timed;
color Duration = Int;

color Status = with Busy|Free;
color Machine = with M1|M2|M3;
color Machine_Status = product Machine * Status;
color Operation = with Op1|Op2|Op3;

color Part_Operation = product Part_Type * Operation;
color Part_Operation_Machine = product Part_Type * Operation * Machine;
color Part_Op_Mach_Duration = product Part_Type * Operation * Machine * Duration;
color Part = product Part_Type * Machine;
color Operation_Duration = product Operation * Duration;

var m: Machine;
var p: Part_Type;
var opr: Operation;
var d: Duration;

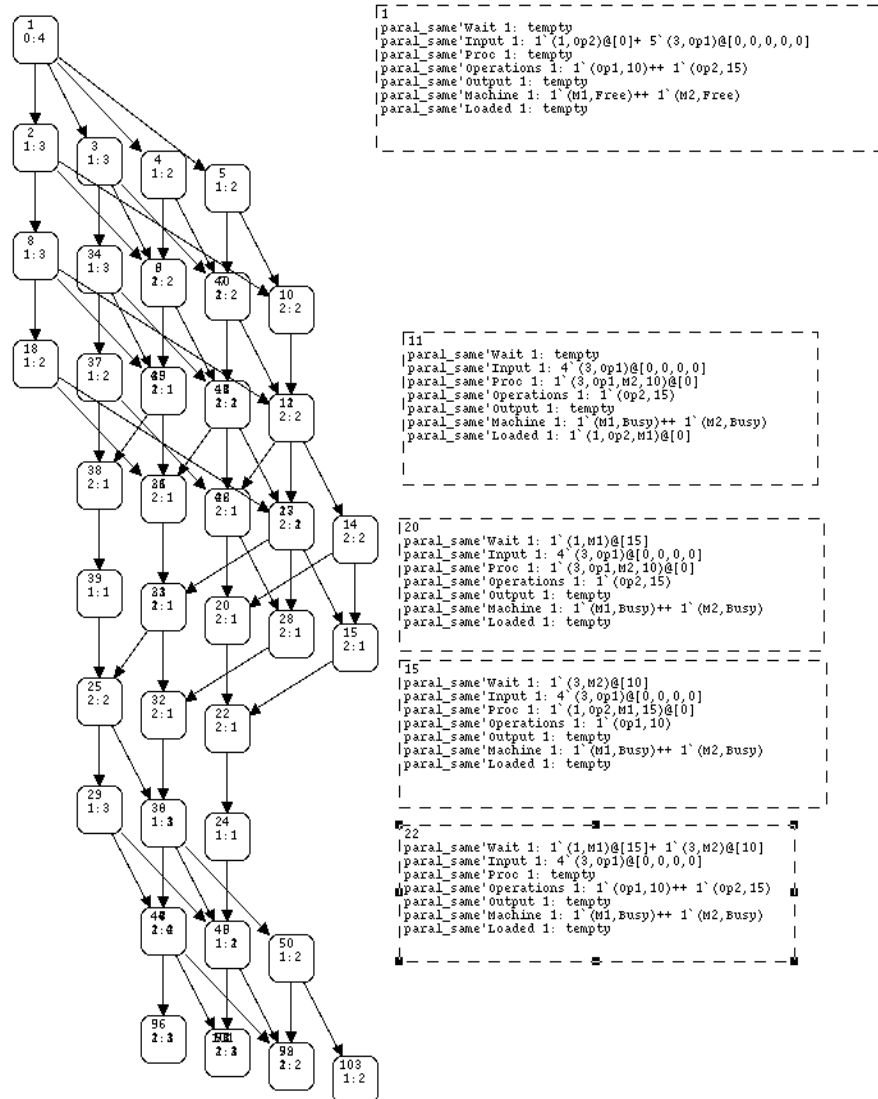
```

Σχήμα 4.15: Ορισμοί για το μοντέλο παράλληλων μηχανών ίδιου τύπου

Στη δεύτερη περίπτωση που έχουμε μηχανές διαφορετικού τύπου ή κατάστασης (π.χ. παλαιότητας), είναι λογικό οι χρόνοι διεκπεραίωσης κάθε ενέργειας να είναι διαφορετικοί για κάθε μηχανή. Στη θέση `Operations` οι μάρκες που αναπαριστούν τη λειτουργικότητα κάθε μηχανής πλέον αναφέρουν και τον τύπο της μηχανής (π.χ. M_1 , M_2).

Όπως εύκολα παρατηρεί κανείς, μπορεί εύκολα ένα μοντέλο τέτοιου τύπου να προβλέψει ένα μεγάλο αριθμό μηχανών χωρίς να χρειαστεί δραματική αύξηση των θέσεων και των μεταβάσεων, φαινόμενο που παρατηρείται στην περίπτωση των μοντέλων με απλά δίκτυα Petri.

Όταν κάποιο προϊόν φτάσει στο σύστημα των μηχανών επιλέγεται με τυχαίο τρόπο κάποια από τις ελεύθερες μηχανές (που έχει την απαραίτητη λειτουργικότητα ώστε να εκτελέσει τη ζητούμενη κατεργασία). Κι εδώ όμως μπορεί να εισαχθεί κάποιο σύστημα υψηλότερου επιπέδου που θα μπορεί να καθορίσει το ποια μηχανή θα αναλάβει λύνοντας το αντίστοιχο πρόβλημα διαχείρισης (management problem).



Σχήμα 4.16: Τμήμα του γράφου καλυψιμότητας του μοντέλου παράλληλων μηχανών ίδιου τύπου

Ανάλυση χώρου καταστάσεων

Η βασική διαφορά σε σχέση με το προηγούμενο μοντέλο των παράλληλων μηχανών ίδιου τύπου φαίνεται από τους διαφορετικούς τύπους ενεργειών και άρα διαφορετικών μαρκών που μπορεί να βρεθούν στη θέση Proc. Ο μέγιστος αριθμός τους βέβαια μπορεί να υπερβεί τη μονάδα γι' αυτό και αναφερόμαστε σε παράλληλες μηχανές. Η θέση της κατάστασης περιέχει πάντα μια μάρκα όπως είναι επιθυμητό.

```

Statistics
-----
Occurrence Graph
Nodes: 194
Arcs: 288
Secs: 0
Status: Full

Scc Graph
Nodes: 194
Arcs: 288
Secs: 0

Boundedness Properties
-----
Best Integers Bounds      Upper      Lower
New'Input 1                6          0
New'Loaded 1               2          0
New'Machine 1              2          2
New'Operations 1           2          0
New'Output 1               6          0
New'Proc 1                 2          0
New'Wait 1                 2          0

Best Upper Multi-set Bounds
New'Input 1                1'(1,Op2)+ 5'(3,Op1)
New'Loaded 1               1'(1,Op2,M1)+ 1'(1,Op2,M2)+ 1'(3,Op1,M1)+ 1'(3,Op1,M2)
New'Machine 1              1'(M1,Busy)+ 1'(M1,Free)+ 1'(M2,Busy)+ 1'(M2,Free)
New'Operations 1           1'(Op1,10)+ 1'(Op2,15)
New'Output 1               1'1+ 5'3
New'Proc 1                 1'(1,Op2,M1,15)+ 1'(1,Op2,M2,15)+ 1'(3,Op1,M1,10)+ 1'(3,Op1,M2,10)
New'Wait 1                 1'(1,M1)+ 1'(1,M2)+ 1'(3,M1)+ 1'(3,M2)

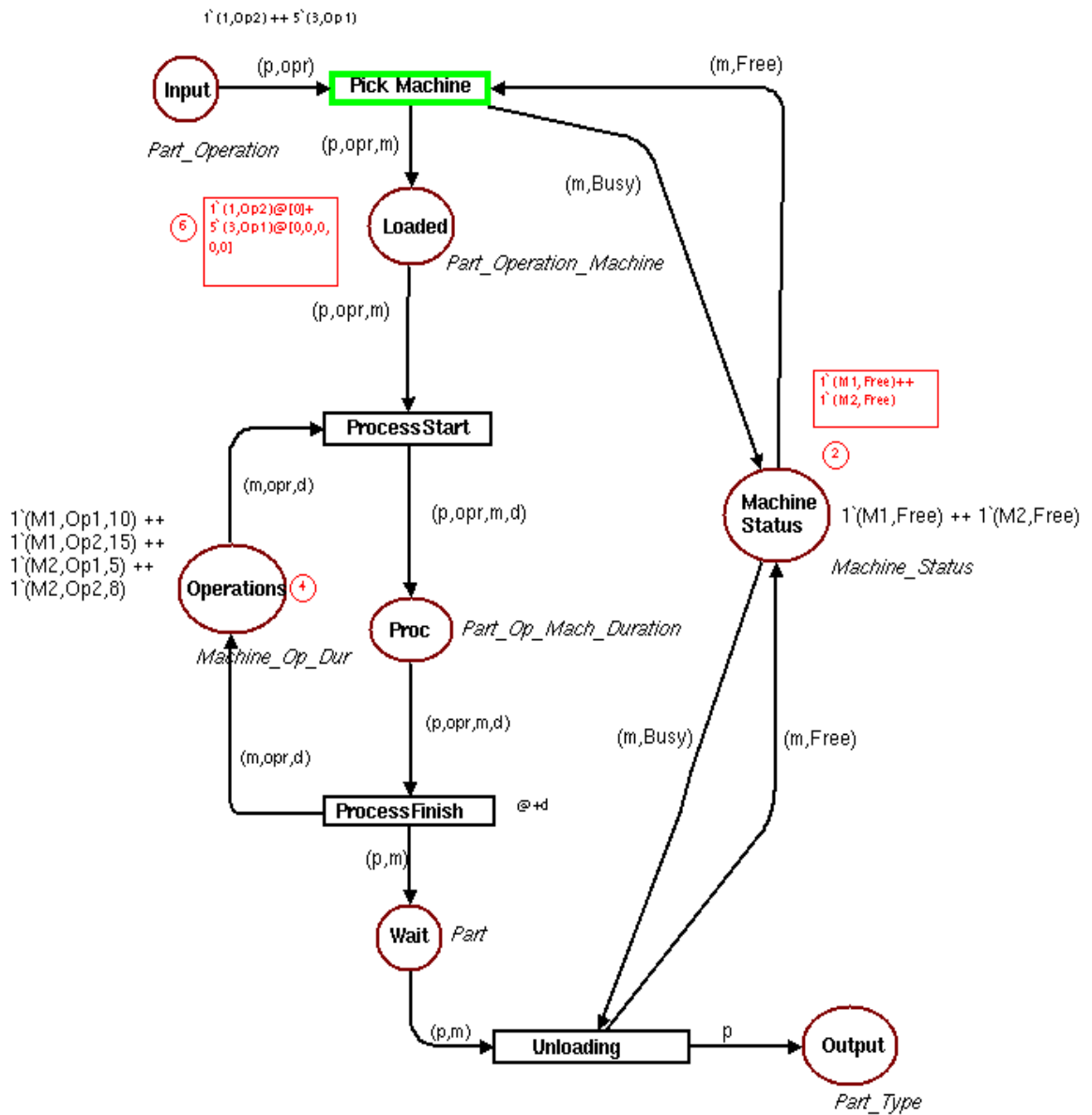
Best Lower Multi-set Bounds
New'Input 1                empty
New'Loaded 1               empty
New'Machine 1              empty
New'Operations 1           empty
New'Output 1               empty
New'Proc 1                 empty
New'Wait 1                 empty

Home Properties
-----
Home Markings: Initial Marking is not a home marking

Liveness Properties
-----
Dead Markings: [192,193,194]
Dead Transitions Instances: None

```

Σχήμα 4.17: Αναφορά της ανάλυσης του χώρου καταστάσεων για το μοντέλο παράλληλων μηχανών ίδιου τύπου



Σχήμα 4.18: Μοντέλο παράλληλων μηχανών διαφορετικού τύπου

Στον πίνακα 4.1 φαίνεται η συμπεριφορά του δικτύου ως προς το μέγεθος του γράφου καλυψιμότητας όταν αυξάνεται ο αριθμός των μηχανών που καλείται να επεξεργαστεί το συγκεκριμένο αριθμό προϊόντων εισόδου:

```

use "/CPN_EXAMPLES/tod.sml";

color Int = int;
color Part_Type = Int timed;
color Duration = Int;

color Status = with Busy|Free;
color Machine = with M1|M2|M3;
color Machine_Status = product Machine * Status;
color Operation = with Op1|Op2|Op3;

color Part_Operation = product Part_Type * Operation;
color Part_Operation_Machine = product Part_Type * Operation * Machine;
color Part_Op_Mach_Duration = product Part_Type * Operation * Machine * Duration;
color Part = product Part_Type * Machine;
color Machine_Op_Dur = product Machine * Operation * Duration;

var m: Machine;
var p: Part_Type;
var opr: Operation;
var d: Duration;

```

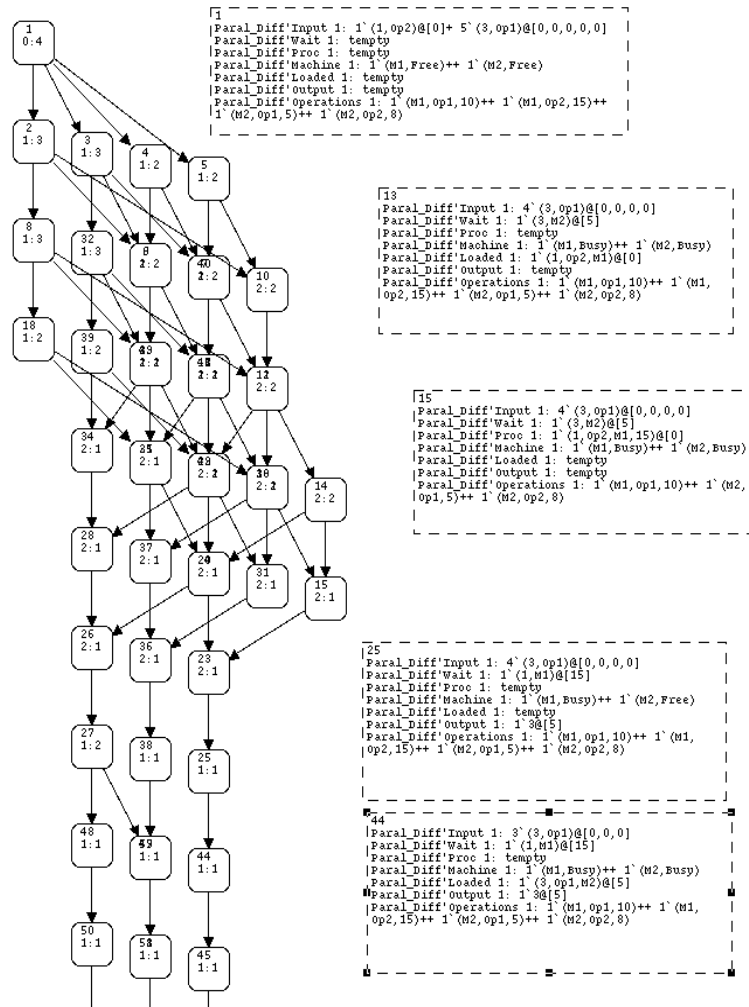
Σχήμα 4.19: Ορισμοί για το μοντέλο παράλληλων μηχανών διαφορετικού τύπου

Μοντελοποίηση με απλά δίκτυα Petri

Στο σχήμα 4.22 παρουσιάζεται το μοντέλο παράλληλων μηχανών ίδιου τύπου με χρήση απλών δικτύων Petri ενώ στο σχήμα 4.23 το μοντέλο παράλληλων μηχανών διαφορετικού τύπου. Το τμήμα ορισμών και δηλώσεων του δικτύου (4.2.3) παραμένει πάντα απλό σε σχέση με αυτό των έγχρωμων δικτύων μιας και υπάρχουν μάρκες μόνο ενός τύπου.

Κι εδώ ο περιορισμός της ύπαρξης μαρκών μόνο ενός απλού τύπου ($b=(black)$) επιβάλλει διαφορετικές σειρές θέσεων για κάθε τύπο προϊόντος κάτι που προφανώς δυσκολεύει την κάλυψη περιπτώσεων με περισσότερους τύπους προϊόντων ή μηχανών.

Μπορεί να παρατηρήσει κανείς ότι τα μοντέλα των θεμελιωδών υποσυστημάτων με χρήση των απλών δικτύων Petri στηρίζονται σε κοινές αρχές, έχουν τις ίδιες ιδιότητες και συντίθενται από τα ίδια γεγονότα. Οι όποιες διαφορές τους είναι κα-



Σχήμα 4.20: Τμήμα του γράφου καλυψιμότητας του μοντέλου παράλληλων μηχανών διαφορετικού τύπου

θαρά λειτουργικές και επιβάλλονται από τις ανάγκες αναπαράστασης των διαφόρων γεγονότων.

Ένα κοινό σημείο των θεμελιωδών μοντέλων με απλά δίκτυα Petri αλλά και των μοντέλων των σύνθετων συστημάτων που δημιουργούνται από αυτά, είναι πως αν εξαιρεθούν οι θέσεις που αντιστοιχούν σε αποθήκες, όλες οι υπόλοιπες θέσεις τους είναι δομικά ασφαλείς, δηλαδή περιέχουν το πολύ μια μάρκα με οποιαδήποτε αρχική σήμανση. Αυτό από τη μια μπορεί να βοηθήσει στον εύκολο εντοπισμό λανθασμένων λειτουργιών στο μοντέλο αλλά από την άλλη, όπως τονίστηκε και προηγουμένως,

Statistics			

Occurrence Graph			
Nodes:	185		
Arcs:	267		
Secs:	0		
Status:	Full		
Scc Graph			
Nodes:	185		
Arcs:	267		
Secs:	0		
Boundedness Properties			

Best Integers Bounds	Upper	Lower	
New'Input 1	6	0	
New'Loaded 1	2	0	
New'Machine 1	2	2	
New'Operations 1	4	2	
New'Output 1	6	0	
New'Proc 1	2	0	
New'Wait 1	2	0	
Best Upper Multi-set Bounds			
New'Input 1	1'(1,Op2)+ 5'(3,Op1)		
New'Loaded 1	1'(1,Op2,M1)+ 1'(1,Op2,M2)+ 1'(3,Op1,M1)+ 1'(3,Op1,M2)		
New'Machine 1	1'(M1,Busy)+ 1'(M1,Free)+ 1'(M2,Busy)+ 1'(M2,Free)		
New'Operations 1	1'(M1,Op1,10)+ 1'(M1,Op2,15)+ 1'(M2,Op1,5)+ 1'(M2,Op2,8)		
New'Output 1	1'1+ 5'3		
New'Proc 1	1'(1,Op2,M1,15)+ 1'(1,Op2,M2,8)+ 1'(3,Op1,M1,10)+ 1'(3,Op1,M2,5)		
New'Wait 1	1'(1,M1)+ 1'(1,M2)+ 1'(3,M1)+ 1'(3,M2)		
Best Lower Multi-set Bounds			
New'Input 1	empty		
New'Loaded 1	empty		
New'Machine 1	empty		
New'Operations 1	empty		
New'Output 1	empty		
New'Proc 1	empty		
New'Wait 1	empty		
Home Properties			

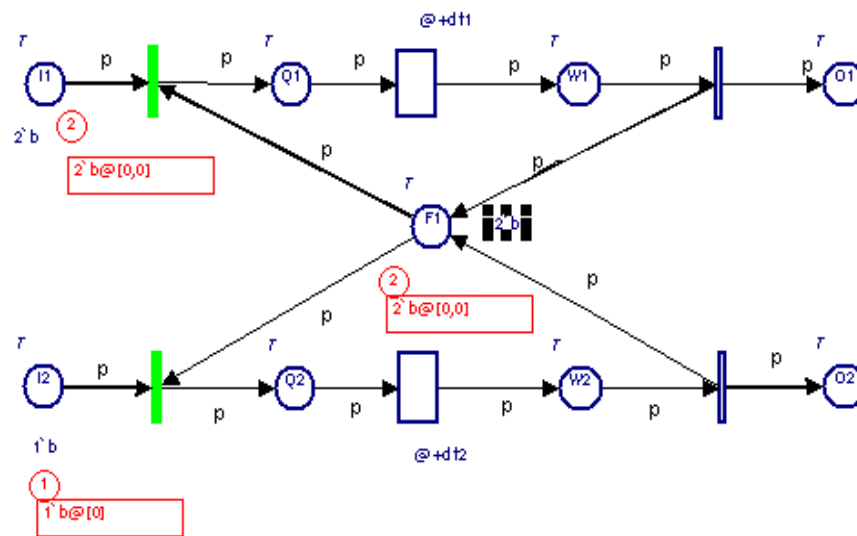
Home Markings: Initial Marking is not a home marking			
Liveness Properties			

Dead Markings: 6 [185,184,183,182,181,...]			
Dead Transitions Instances: None			

Σχήμα 4.21: Αναφορά της ανάλυσης του χώρου καταστάσεων για το μοντέλο παράλληλων μηχανών διαφορετικού τύπου

Πίνακας 4.1: Μέγεθος γράφου καλυψιμότητας ως προς τον αριθμό των παράλληλων μηχανών

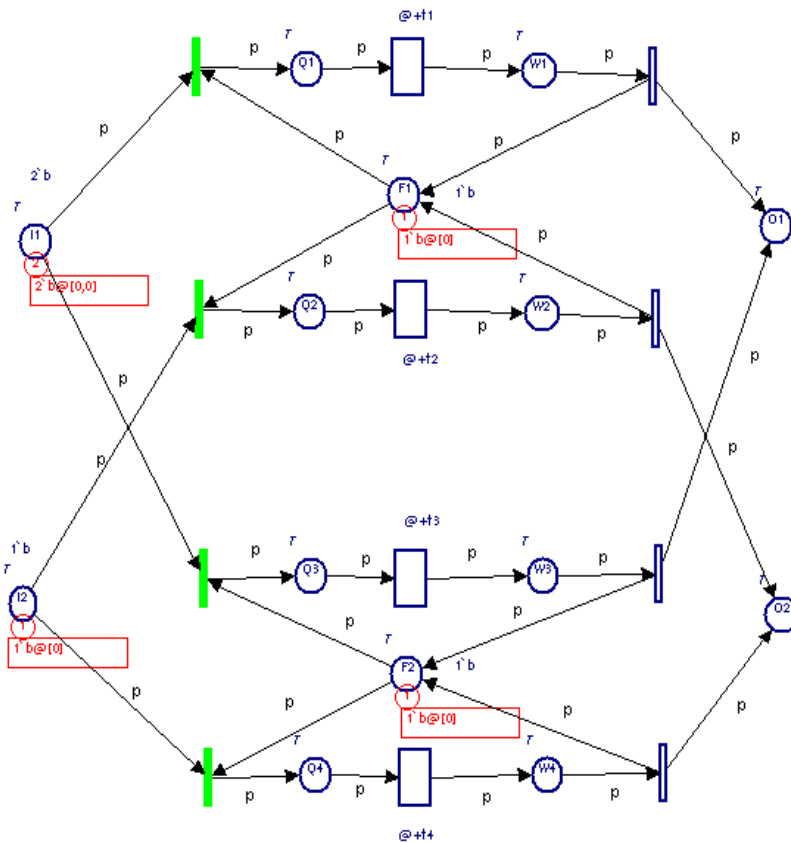
Αριθμός μηχανών	Αριθμός κόμβων OGG	Αριθμός τόξων
2	185	267
3	231	401
4	317	731



Σχήμα 4.22: Μοντέλο παράλληλων μηχανών ίδιου τύπου με απλά δίκτυα Petri

αυξάνει σημαντικά τον συνολικό αριθμό θέσεων στο σύστημα και κατ' επέκταση και τον γράφο καλυψιμότητας όσο το μοντέλο επιδιώκει να καλύψει πιο πολύπλοκες καταστάσεις.

Στα παραπάνω θα πρέπει να προστεθεί ότι σε όλα τα θεμελιώδη υποσυστήματα εκτός τις εισόδους των προϊόντων πρέπει να εισαχθούν επιπλέον είσοδοι που θα είναι υπεύθυνες για την εμφάνιση των βλαβών, οι οποίες είναι γενικά ιδιαίτερα χρήσιμες σε online χρήσεις των μοντέλων, αλλά μπορούν να δημιουργήσουν προβλήματα σε offline χρήσεις και προσομοιώσεις λειτουργίας.



Σχήμα 4.23: Μοντέλο παράλληλων μηχανών διαφορετικού τύπου με απλά δίκτυα Petri

4.2.4 Μηχανές συναρμολόγησης

Οι μηχανές αυτές συναρμολογούν κάποιο προϊόν χρησιμοποιώντας δύο ή περισσότερες πρώτες ύλες ή άλλα ενδιάμεσα προϊόντα. Βασική τους διαφορά σε σχέση με τις απλές μηχανές είναι ότι απαιτούν πολλαπλά στοιχεία στην είσοδό τους για να ξεκινήσει η λειτουργία τους. Το προϊόν της διαδικασίας μπορεί να είναι είτε το τελικό προϊόν μιας διαδικασίας είτε κάποιο ενδιάμεσο προϊόν που με τη σειρά του να αποτελεί είσοδο για μια άλλη μηχανή συναρμολόγησης. Η τελευταία ιδιότητα χρησιμοποιείται πολύ συχνά στη μοντελοποίηση με απλά δίκτυα Petri στις περιπτώσεις που έχουμε μηχανές που δέχονται περισσότερες από δύο εισόδους για να παράγουν το τελικό προϊόν, κάτι που αντιμετωπίζεται ως διαδοχική τοποθέτηση πολλών

```
use "/CPN_EXAMPLES/tod.sml";

color T = with b timed;
var p:T;

val dt1=2;
val dt2=3;
val t1=2;
val t2=3;
val t3=3;
val t4=1;
```

Σχήμα 4.24: Ορισμοί για το μοντέλο της παράλληλης μηχανής χρήση απλών δικτύων Petri

μηχανών συναρμολόγησης με δύο εισόδους.

Στη γενική περίπτωση, οι μηχανές αυτές πρέπει με κάποιο τρόπο να είναι συνδεδεμένες με το μοντέλο παραγωγής προϊόντων όπου περιγράφονται οι εξαρτήσεις και οι απαιτήσεις των διαφόρων ενδιάμεσων σταδίων παραγωγής σε πρώτες ύλες και άλλα ενδιάμεσα προϊόντα, όπως παρουσιάστηκε στην παράγραφο 4.1.

Η αναφορά στο μηχανισμό αυτό είναι προτιμότερο να γίνει λίγο πριν ξεκινήσει η διαδικασία φορτώματος μιας μηχανής με κάποιο προϊόν ώστε να εξασφαλίζεται η παρουσία όλων των απαιτούμενων υλικών για να ξεκινήσει και η διαδικασία συναρμολόγησης/παραγωγής. Με τον τρόπο αυτό εξασφαλίζεται ότι δε θα χαθεί πολύτιμος χρόνος κάτι που συμβαίνει στην περίπτωση της εκ των υστέρων διαπίστωσης της έλλειψης κάποιου προαπαιτούμενου στοιχείου. Σε μια τέτοια περίπτωση θα πρέπει να προβλέπεται μια διαδικασία ξεφορτώματος της μηχανής ώστε να είναι διαθέσιμη για μια άλλη ενέργεια (στην περίπτωση βέβαια που η μηχανή δεν είναι αφιερωμένη στη συγκεκριμένη λειτουργία ή στο συγκεκριμένο στάδιο παραγωγής).

Εναλλακτικά, ο μηχανισμός ελέγχου της παρουσίας των απαιτούμενων προϊόντων μπορεί να ανατρέξει στο μοντέλο προϊόντος αφού φορτώσει τα όποια διαθέσιμα στοιχεία και να μπλοκάρει μέχρι να εμφανιστούν τα υπόλοιπα σε κάποια κεντρική αποθήκη αποθεμάτων.

4.2.5 Μηχανές αποσυναρμολόγησης

Οι μηχανές αυτές αποσυνθέτουν κάποιο προϊόν σε δύο ή περισσότερα άλλα ενδιαμέσα προϊόντα. Εδώ δεν υπάρχει η απαίτηση του ελέγχου της ύπαρξης πολλαπλών στοιχείων στην είσοδο πριν την έναρξη της παραγωγικής διαδικασίας. Αν όμως η μηχανή δεν είναι αποκλειστικά αφιερωμένη σε έναν τύπο προϊόντος, χρειάζεται να καταφύγει στο κεντρικό μοντέλο προϊόντος ώστε να έχει πληροφορία για τη διαδικασία καθώς και τα παραγόμενα προϊόντα της διαδικασίας αποσυναρμολόγησης.

4.2.6 Μηχανές που είναι δυνατόν να χαλάσουν

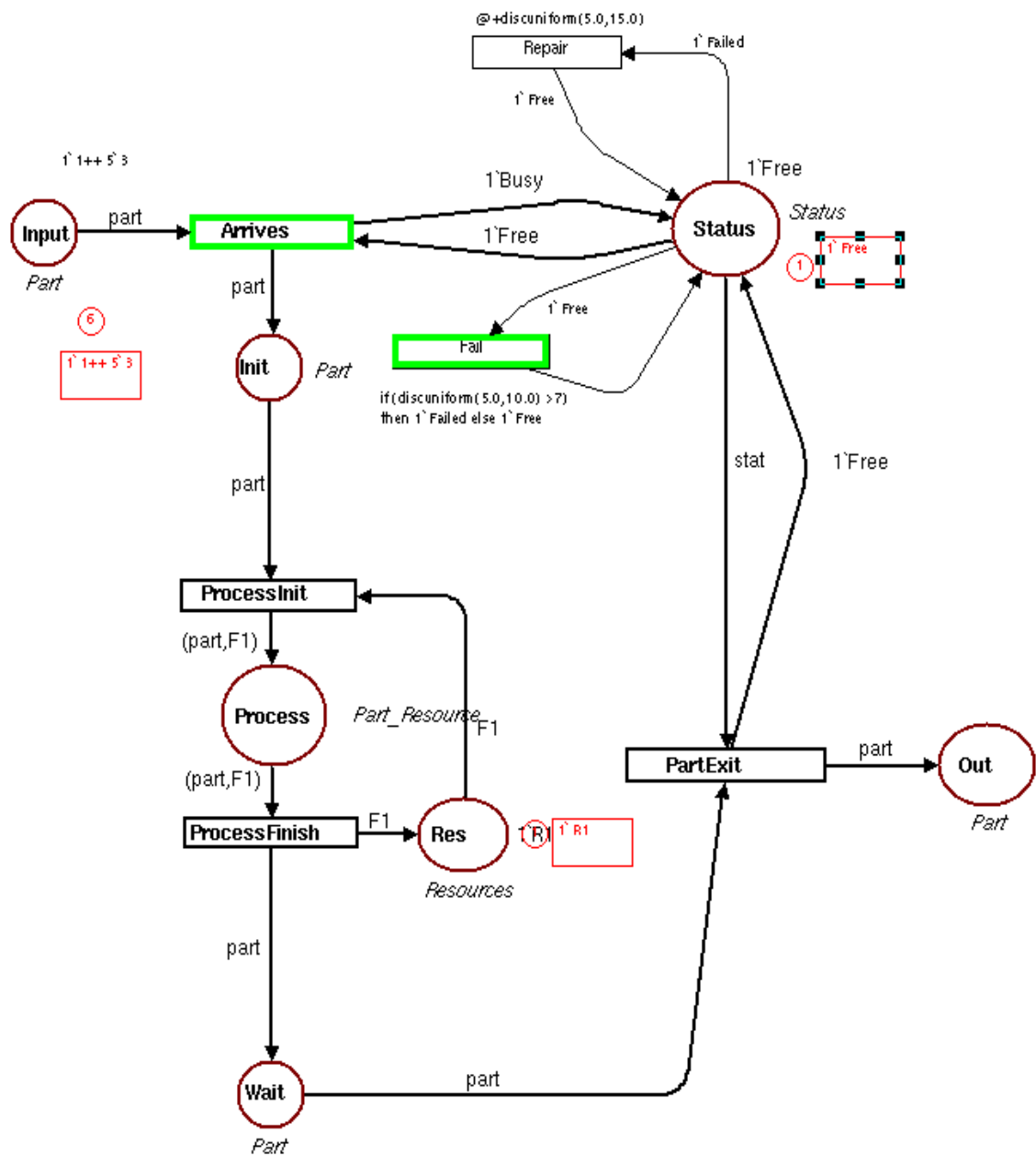
Για τη μοντελοποίηση της περίπτωσης να χαλάσει μια μηχανή και να τεθεί σε διαδικασία επισκευής μέχρι να διατεθεί πάλι πίσω στην παραγωγική διαδικασία, καταρχήν εμπλουτίζουμε τις τιμές που μπορεί να λάβει η μάρκα που υποδηλώνει την κατάσταση της μηχανής ώστε να περιλαμβάνει και την τιμή Failed. Η μηχανή του σχήματος 4.25 μπορεί να διαπιστωθεί ότι έχει χαλάσει αφού τελειώσει η κατεργασία ενός προϊόντος (και έχει τεθεί σε κατάσταση διαθεσιμότητας). Αν συμβεί κάτι τέτοιο, υπόκειται σε μια διαδικασία επισκευής (η οποία στη συγκεκριμένη περίπτωση μπορεί να διαρκέσει ένα τυχαίο χρονικό διάστημα μεταξύ 5 και 15 χρονικών μονάδων, όπως φαίνεται από το τμήμα κώδικα που είναι συσχετισμένο με την αντίστοιχη μετάβαση) και δίνεται μετά και πάλι στην παραγωγή.

4.3 Μοντελοποίηση συστημάτων μεταφοράς

Τα συστήματα μεταφοράς είναι υπεύθυνα για τη μεταφορά των προϊόντων από τη μια μηχανή σε μια άλλη, από μια μηχανή στο χώρο αποθήκευσης ή αντίστροφα από το χώρο προσωρινής αποθήκευσης προς μια μηχανή για περαιτέρω επεξεργασία. Τα πιο συχνά χρησιμοποιούμενα τέτοια συστήματα είναι οι ιμάντες μεταφοράς (conveyors), τα αυτόματα καθοδηγούμενα οχήματα (AGV) καθώς και οι ρομποτικές διατάξεις.

4.3.1 Οχήματα μεταφοράς

Τα οχήματα (AGV-ΟΑΠ) ανήκουν στα πιο ευέλικτα συστήματα μεταφοράς. Επιτρέπουν τη μεταφορά ενός ή περισσότερων προϊόντων από κάποιο σημείο σε ένα



Σχήμα 4.25: Μοντέλο μηχανής που μπορεί να χαλάσει

οποιοδήποτε άλλο.

Οι ενέργειες φόρτωσης και εκφόρτωσης μπορεί να γίνονται από μηχανές ενώ

```

use "/CPN_EXAMPLES/tod.sml";

color Int = int;
color Part = Int;

color E= with e;
color Buf = list Part;
color Data = Part;

color Resources = with R1|R2|R3|R4|R5|R6|R7;
color List_Resources = list Resources;
color Part_Resource = product Part*Resources;
color Status = with Busy|Free|Failed;

var stat,sti,sto: Status;
var pf: Part_Resource;
var F,F1: Resources;
var L,L1: List_Resources;
var part: Part;

fun discuniform (x,y) = round (uniform(x,y));

```

Σχήμα 4.26: Ορισμοί για το μοντέλο της μηχανής που μπορεί να χαλάσει

γενικά πρέπει να ληφθεί υπόψιν ο χρόνος που απαιτείται για αυτές τις διαδικασίες ¹.

Σε ένα σύστημα μπορούν να υπάρχουν περισσότερα του ενός οχήματα AGV και οποιοδήποτε είναι ελεύθερο μπορεί να αναλάβει μια συγκεκριμένη διαδρομή. Η διαδρομή καθορίζεται από την αρχή και το τέλος (από το σημείο ανάληψης και παράδοσης των προϊόντων).

Στην περίπτωση που μεταφέρονται περισσότερα του ενός προϊόντος θα πρέπει να καθοριστεί και η σειρά με την οποία τοποθετούνται πάνω στο όχημα κάτι που έχει συνέπεια στο πώς ξεφορτώνεται το όχημα. Στο μοντέλο ενός τέτοιου οχήματος στο σχήμα 4.27 υιοθετείται η προσέγγιση του να φορτώνεται το ένα πάνω στο άλλο. Κατά συνέπεια θα ξεφορτώνεται πρώτο αυτό που φορτώθηκε τελευταίο (στρατηγική

¹ Στο επόμενο κεφάλαιο θα δούμε πώς οι συγκεκριμένες μεταβάσεις που αναπαριστούν τις διαδικασίες αυτές μπορούν να αντικατασταθούν από μοντέλα μηχανών ώστε να γίνει πιο ρεαλιστική η μοντελοποίηση.

τύπου LIFO). Το όχημα έχει περιορισμένη χωρητικότητα (C) και η θέση Counter έχει αριθμό μαρκών ίσο με τον αριθμό των κομματιών που είναι φορτωμένα ήδη στο όχημα μεταφοράς. Το όχημα παραμένει στο ίδιο σημείο μέχρι να φορτωθεί πλήρως και αφού γίνει αυτό μετακινείται στην τελική θέση όπου θα γίνει η εκφόρτωση. Η κίνηση αυτή έχει μια διάρκεια $Moue_Dur$ μονάδες χρόνου. Αφού τελειώσει η εκφόρτωση (στο μοντέλο αυτό τόσο η διαδικασία της φόρτωσης όσο και της εκφόρτωσης θεωρούνται σύντομες διαδικασίες), το όχημα είναι έτοιμο να επιστρέψει στην αρχική του θέση. Η διαδικασία επιστροφής μπορεί να πάρει λιγότερο χρόνο (Ret_Dur χρονικές μονάδες) μιας και το όχημα είναι πλέον χωρίς φορτίο.

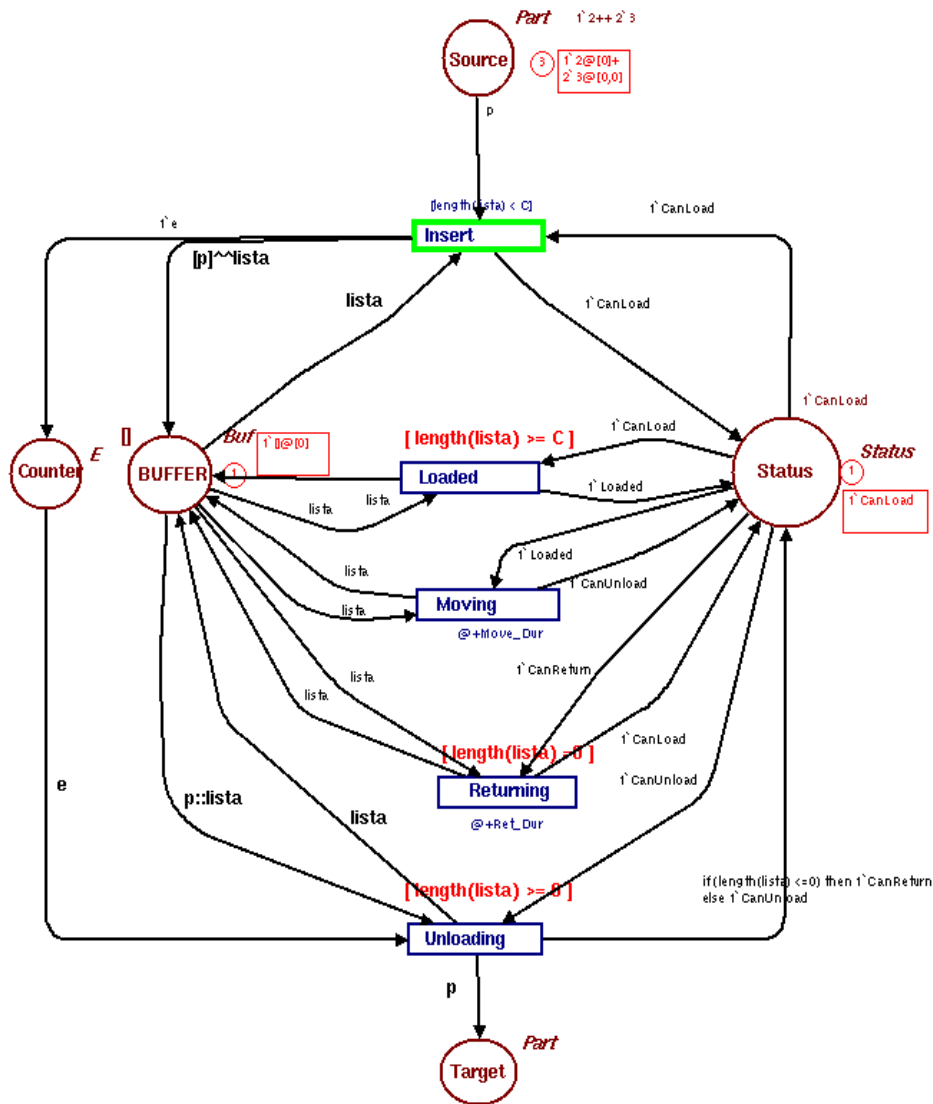
Γενικά, τα μοντέλα αυτού του τύπου μπορούν εύκολα να επεκταθούν ώστε να χρησιμοποιηθούν για να μοντελοποιήσουν και συστήματα ανύψωσης (γερανούς) που μπορεί να κινούνται είτε σε μία είτε σε δύο διαστάσεις.

Αναφορά γράφου καλυψιμότητας

Από την αναφορά 4.3.1 της ανάλυσης του χώρου των καταστάσεων για τη συγκεκριμένη είσοδο προκύπτει ότι τηρείται η μέγιστη χωρητικότητα του οχήματος μεταφοράς μιας και ο μέγιστος αριθμός μαρκών στη θέση Counter είναι δύο (ενώ ο ελάχιστος είναι μηδέν μιας και μπορεί να αδειάσει). Η θέση Status πάντα περιέχει μια μάρκα που αναπαριστά την κατάσταση κάτι που είναι το επιθυμητό. Ως προς τις ιδιότητες της ζωτικότητας, οι σημάνσεις που αναφέρονται ως νεκρές είναι τελικές για τις συγκεκριμένες αρχικές συνθήκες. Πρόκειται για καταστάσεις όπου στο όχημα έχει φορτωθεί η μια μάρκα που έχει απομείνει και αναμένεται μια ακόμη ώστε να γεμίσει και να ξεκινήσει την κίνησή του προς τον προορισμό του.

4.3.2 Ιμάντες (ή ταινίες) μεταφοράς

Οι ιμάντες μεταφοράς χρησιμοποιούνται συνήθως για τη μεταφορά ογκωδών ενδιάμεσων προϊόντων πάνω σε προκαθορισμένες διαδρομές. Μια λωρίδα μπορεί να χωριστεί σε τμήματα καθένα από τα οποία μπορεί να μεταφέρει ένα προϊόν κάθε φορά. Αν δεν υπάρχει καποιος μηχανισμός ρομποτικού βραχίονα που μπορεί να επιτρέψει την κατ' επιλογή ανάκτηση ενός προϊόντος (random access), τότε τα προϊόντα εξέρχονται του συστήματος μεταφοράς με τη σειρά με την οποία μπήκαν στο σύστημα (στρατηγική FIFO).



Σχήμα 4.27: Μοντέλο οχήματος μεταφοράς

Ο χρόνος μετατόπισης από το ένα τμήμα στο επόμενο (και άρα ο χρόνος κύλησης του μάντα κατά μια θέση) πρέπει να ληφθεί κι αυτός υπόψη. Συχνά χρησιμοποιούνται κυκλικές ταινίες μεταφοράς πάνω στις οποίες τα προϊόντα μετακινούνται αναμένοντας μέχρι να ελευθερωθεί κάποιος ενδιαμέσος χώρος αποθήκευσης ώστε να μεταφερθούν εκεί. Στο μοντέλο του σχήματος 4.31 δεν υπάρχει τέτοιος περιορισμός στη θέση που αναπαριστά την έξοδο από το σύστημα οπότε αν κάποιο προϊόν

```

use "/CPN_EXAMPLES/tod.sml";
color Int = int;

color Part = Int timed;

color E= with e;
color Buf = list Part;

var lista:Buf;
var p:Part;

val C=2;

val Move_Dur = 4;
val Ret_Dur = 2;

color Status = with CanLoad|Loaded|Moving|CanUnload|CanReturn;

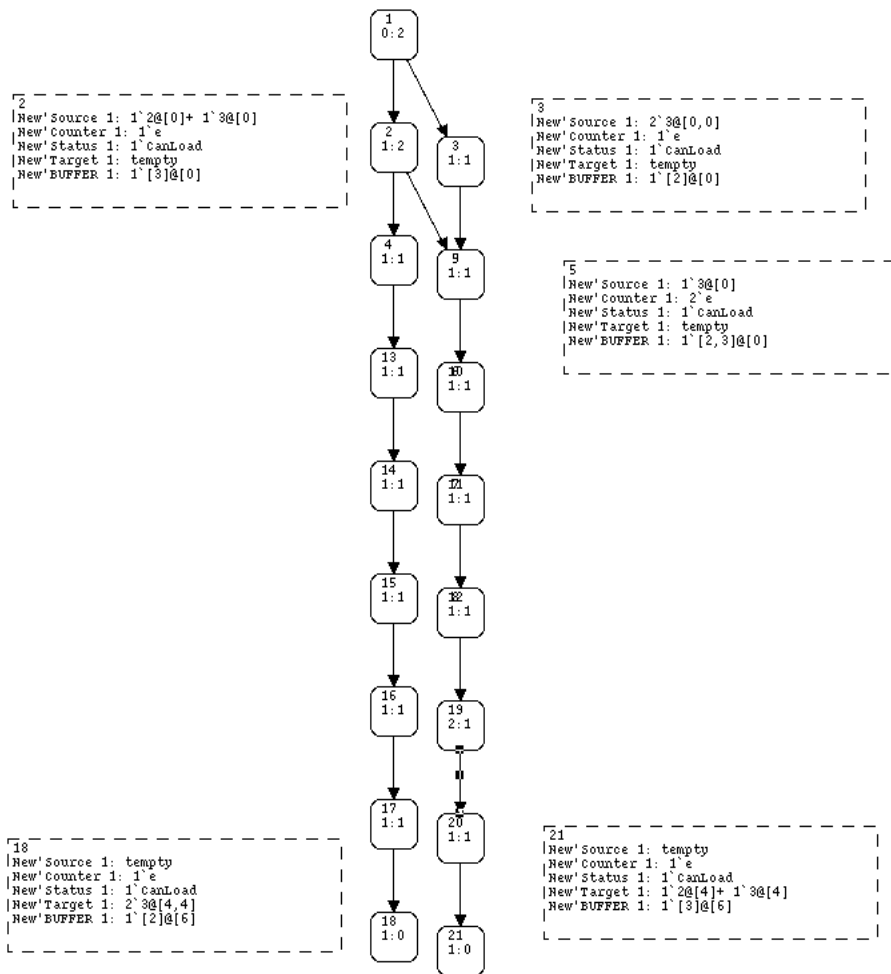
```

Σχήμα 4.28: Ορισμοί για το μοντέλο οχήματος μεταφοράς

βρίσκεται στο τέλος του ιμάντα θα μπει στον ενδιάμεσο χώρο αποθήκευσης. Όπως και στα προηγούμενα μοντέλα, κάθε προϊόν χαρακτηρίζεται από ένα ακέραιο αριθμό (ως κωδικός). Ο ιμάντας έχει περιορισμένη χωρητικότητα (όση και το μέγεθος της λίστας που βρίσκεται στη θέση BUFFER - στη συγκεκριμένη περίπτωση είναι ίση με 3). Αν δεν υπάρχει κάποιο προϊόν στην είσοδο του υποσυστήματος για μεταφορά (στη θέση Entry), τότε η αντίστοιχη θέση «σημειώνεται» με τον κωδικό 0 και βέβαια για να φορτωθεί το επόμενο προϊόν (αν υπάρχει) πρέπει να περάσει ο χρόνος δ για να κυλήσει ο ιμάντας. Η θέση E κρατά ανα πάσα στιγμή τόσες μάρκες όσα είναι και τα προϊόντα που βρίσκονται πάνω στον ιμάντα μεταφοράς. Το προϊόν που έχει φορτωθεί στον ιμάντα θα ξεφορτωθεί όταν κυλήσει πλήρως ο ιμάντας (εδώ μετά από 3 χρονικές περιόδους διάρκειας δ) και η θέση τότε μαρκάρεται με τον κωδικό 0.

Αναφορά γράφου καλυψιμότητας

Η ανάλυση του χώρου των καταστάσεων για το συγκεκριμένο μοντέλο περιορίστηκε στους 400 κόμβους κι αυτό γιατί ο χώρος αυξάνεται διαρκώς αφού έχει γίνει θεώρηση ενός κυκλικού ιμάντα μεταφοράς που διαρκώς κινείται, ανεξάρτητα από το αν υπάρχει κάποιο προϊόν στην είσοδο του συστήματος για μεταφορά ή όχι. Ο με-



Σχήμα 4.29: Τμήμα του γράφου καλυψιμότητας του μοντέλου οχήματος μεταφοράς

γάλος αριθμός νεκρών σημάνσεων αναφέρεται σ' αυτή ακριβώς την περίπτωση όπου δεν υπάρχουν άλλα προϊόντα προς μεταφορά. Όπως φαίνεται από τη μέγιστη και την ελάχιστη τιμή της θέσης Counter στην αναφορά 4.3.2, ο αριθμός των κομματιών πάνω στον ιμάντα περιορίζεται στη συγκεκριμένη χωρητικότητα που έχει τεθεί.

4.3.3 Ρομποτικοί βραχίονες

Οι ρομποτικοί βραχίονες συνήθως χρησιμοποιούνται για μετακινήσεις ελαφριών ενδιάμεσων προϊόντων και μάλιστα σε μικρές αποστάσεις. Λόγω της αποκλειστικής

```

Statistics
-----
Occurrence Graph
Nodes: 12
Arcs: 11
Secs: 0
Status: Full

Boundedness Properties
-----
Best Integers Bounds
Upper      Lower
AGV'BUFFER 1      1      1
AGV'Counter 1     2      0
AGV'Source 1     3      1
AGV'Status 1     1      1
AGV'Target 1     0      0

Best Upper Multi-set Bounds
AGV'BUFFER 1     1' []++ 1'[2]++ 1'[2,3]++ 1'[3]++ 1'[3,2]++ 1'[3,3]
AGV'Counter 1     2'e
AGV'Source 1     1'2++ 2'3
AGV'Status 1     1'CanLoad++ 1'Loaded++ 1'CanReturn
AGV'Target 1     empty

Best Lower Multi-set Bounds
AGV'BUFFER 1     empty
AGV'Counter 1     empty
AGV'Source 1     empty
AGV'Status 1     empty
AGV'Target 1     empty

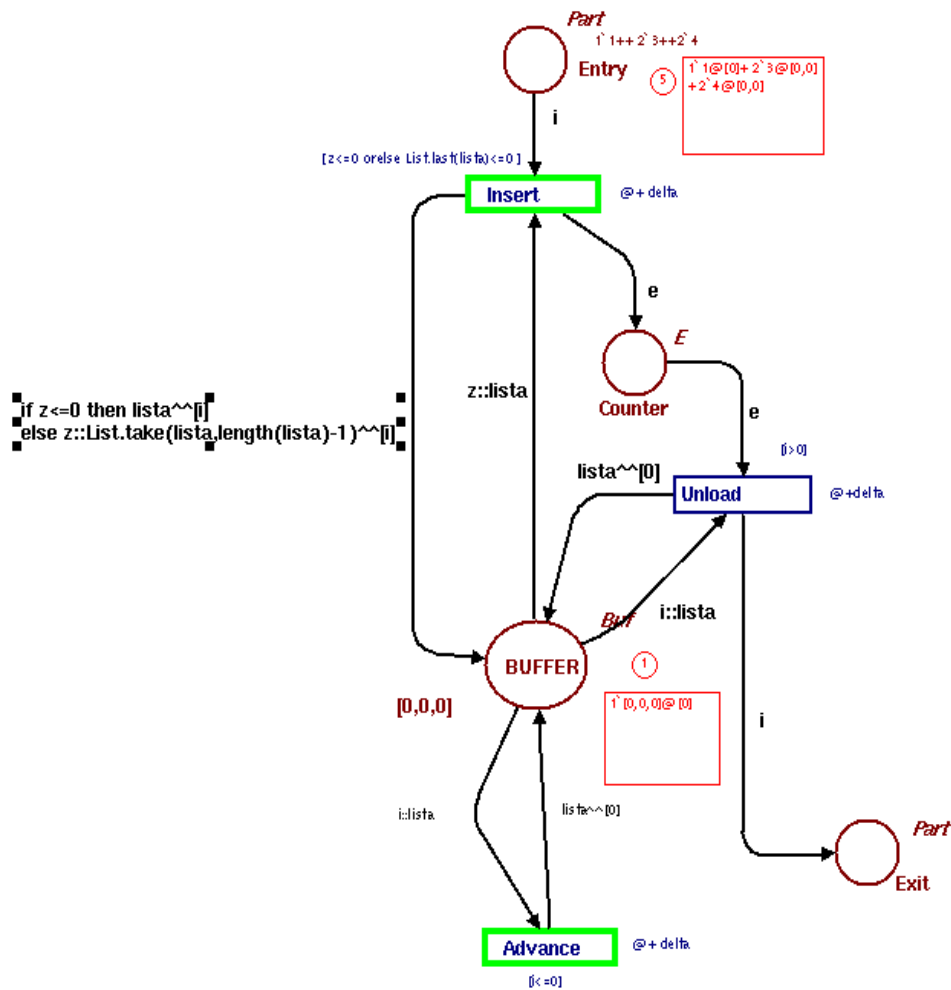
Liveness Properties
-----
Dead Markings: [7,11,12]
Dead Transitions Instances:

AGV'Returning 1
AGV'Unloading 1

```

Σχήμα 4.30: Αναφορά της ανάλυσης του χώρου καταστάσεων για το μοντέλο οχήματος μεταφοράς

ενασχόλησης με ένα προϊόν κάθε φορά μπορούν να μοντελοποιηθούν με χρήση των μοντέλων αποκλειστικών μηχανών ενώ για καλύτερη μοντελοποίηση πρέπει να εισαχθεί και η περίπτωση βλάβης.



Σχήμα 4.31: Μοντέλο μάντα μεταφοράς

4.4 Μοντελοποίηση συστημάτων αποθήκευσης

Τα συστήματα αποθήκευσης (είτε προσωρινής είτε πιο μόνιμης) χρησιμοποιούνται για να μπορέσει να ανταποκριθεί το σύστημα σε τυχαία γεγονότα όπως οι βλάβες κάποιων μηχανών ή μη αναμενόμενες απαιτήσεις. Στην κατηγορία αυτή ανήκουν οι αποκλειστικοί ενδιάμεσοι χώροι αποθήκευσης (dedicated buffers), τα διαμοιραζόμενα συστήματα αποθήκευσης καθώς και τα συστήματα αποθήκευσης που λειτουργούν βάσει της στρατηγικής FIFO.

```
use "/CPN_EXAMPLES/tod.sml";
color Int = int;

color Part = Int timed;

color E= with e;
color Buf = list Part;

val delta =2;

var lista:Buf;
var i:Part;
var z:Part;
```

Σχήμα 4.32: Ορισμοί για το μοντέλο μάντα μεταφοράς

Στους αποκλειστικούς χώρους αποθήκευσης μπορούν να μείνουν προϊόντα ενός συγκεκριμένου τύπου ενώ στους διαμοιραζόμενους μπορούν να αποθηκευτούν οποιοδήποτε τύπου προϊόντα.

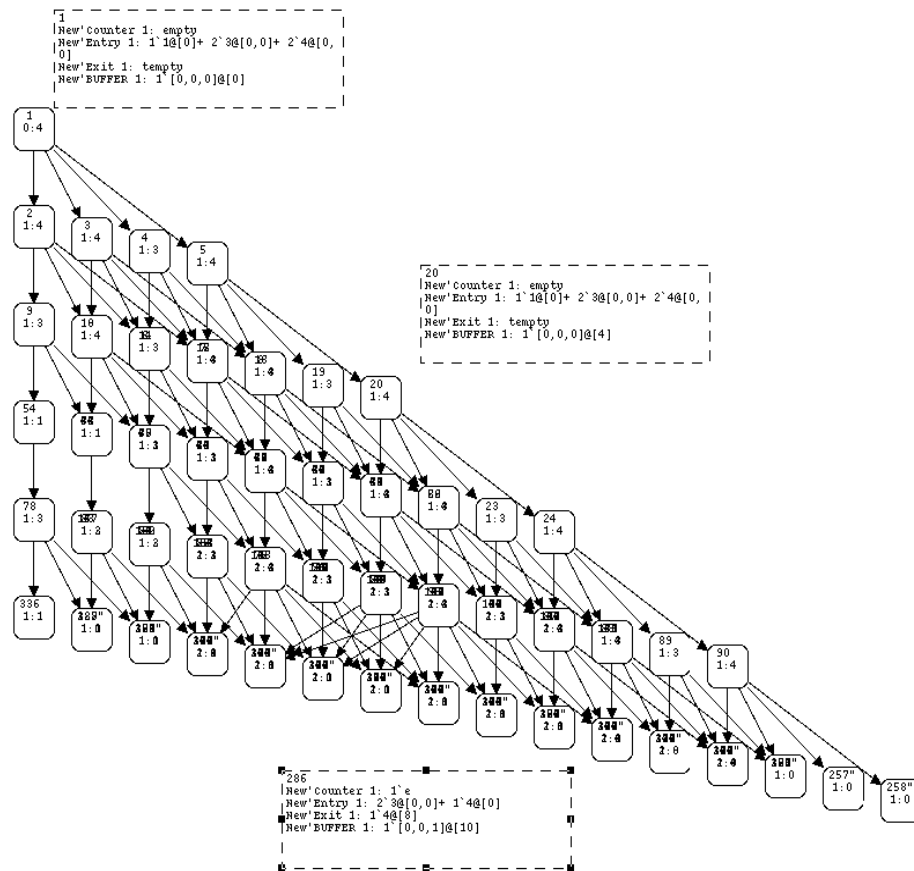
Η χωρητικότητά τους στην πράξη είναι περιορισμένη ενώ είναι χρήσιμο ανά πάσα στιγμή να είναι γνωστός ο ελεύθερος χώρος που έχει απομείνει αλλά και ο αριθμός των προϊόντων κάθε τύπου που έχουν αποθηκευτεί.

Στα συστήματα αποθήκευσης FIFO μπορεί να θεωρηθεί ότι ανήκει και ο μάντας μεταφοράς όπου τα προϊόντα που εισάγονται πρώτα είναι και αυτά που βγαίνουν πρώτα.

4.5 Μοντελοποίηση εργαλειοθηκών

Τα εργαλεία που απαιτούνται από τις μηχανές ή από τους εργάτες για την διεκπεραίωση των διαφόρων εργασιών μπορεί να είναι είτε αποκλειστικής χρήσης από μια συγκεκριμένη μηχανή (ή εργάτη) ή να διαμοιράζονται ανάμεσα σε όλους τους ενδιαφερόμενους και μετά την χρήση τους για κάποια ενέργεια να τίθενται και πάλι στη διάθεση όλων. Η τελευταία περίπτωση για λόγους κόστους εμφανίζεται πιο συχνά.

Υπόψιν στην τελευταία περίπτωση πρέπει να ληφθεί και ο χρόνος που κάνει η μηχανή είτε για να λάβει το εργαλείο από την κεντρική αποθήκη εργαλείων είτε για



Σχήμα 4.33: Τμήμα του γράφου καλυψιμότητας του μοντέλου ιμάντα μεταφοράς

να αλλάξει κάποιο εργαλείο ακόμα κι αν αυτό έχει ανατεθεί σ' αυτήν αποκλειστικά. Επίσης πρέπει να προβλέπεται και η επιστροφή του εργαλείου στην αποθήκη για επαναχρησιμοποίησή του από την ίδια ή άλλες μηχανές ενώ μπορεί επίσης να προβλέπεται η έννοια της καταστροφής ενός εργαλείου κατά τη χρήση του.

Για τη μοντελοποίηση της όλης αυτής διαδικασίας μπορεί να χρησιμοποιηθεί μια κεντρική αποθήκη (θέση) η οποία θα μπορεί να διαμοιράζεται μεταξύ πολλών μηχανών. Η διαθεσιμότητα της αποθήκης αυτής μπορεί να είναι τοπική, να απευθύνεται σε μια συγκεκριμένη ομάδα ή να είναι καθολική ανάλογα με τον αριθμό των μεταβάσεων που την συνδέουν με τις διαδικασίες αυτές.

```

Statistics
-----
Occurrence Graph
Nodes: 401
Arcs: 470
Secs: 0
Status: Partial

Scc Graph
Nodes: 401
Arcs: 470
Secs: 0

Boundedness Properties
-----
Best Integers Bounds
Upper Lower
New'BUFFER 1 1 1
New'Counter 1 3 0
New'Entry 1 5 1
New'Exit 1 3 0

Best Upper Multi-set Bounds
New'BUFFER 1 1'[0,0,0]++ 1'[0,0,1]++ 1'[0,0,3]++ 1'[0,0,4]++ 1'[0,1,0]++
1'[0,1,3]++ 1'[0,1,4]++ 1'[0,3,0]++ 1'[0,3,1]++ 1'[0,3,3]++ 1'[0,3,4]++ 1'[0,4,0]++
1'[0,4,1]++ 1'[0,4,3]++ 1'[0,4,4]++ 1'[1,0,0]++ 1'[1,0,3]++ 1'[1,0,4]++ 1'[1,3,0]++
1'[1,3,3]++ 1'[1,3,4]++ 1'[1,4,0]++ 1'[1,4,3]++ 1'[1,4,4]++ 1'[3,0,0]++ 1'[3,0,1]++
1'[3,0,3]++ 1'[3,0,4]++ 1'[3,1,0]++ 1'[3,1,3]++ 1'[3,1,4]++ 1'[3,3,0]++ 1'[3,3,1]++
1'[3,3,4]++ 1'[3,4,0]++ 1'[3,4,1]++ 1'[3,4,3]++ 1'[3,4,4]++ 1'[4,0,0]++ 1'[4,0,1]++
1'[4,0,3]++ 1'[4,0,4]++ 1'[4,1,0]++ 1'[4,1,3]++ 1'[4,1,4]++ 1'[4,3,0]++ 1'[4,3,1]++
1'[4,3,3]++ 1'[4,3,4]++ 1'[4,4,0]++ 1'[4,4,1]++ 1'[4,4,3]
New'Counter 1 3^e
New'Entry 1 1'1++ 2'3++ 2'4
New'Exit 1 1'1++ 2'3++ 2'4

Best Lower Multi-set Bounds
New'BUFFER 1 empty
New'Counter 1 empty
New'Entry 1 empty
New'Exit 1 empty

Liveness Properties
-----
Dead Markings: 209 [401,400,399,398,397,...]

```

Σχήμα 4.34: Αναφορά της ανάλυσης του χώρου καταστάσεων για το μοντέλο ιμάντα μεταφοράς

Κεφάλαιο 5

Μοντελοποίηση σύνθετων συστημάτων

5.1 Ιεραρχική αποσύνθεση

Η βασική ιδέα πίσω από τα ιεραρχικά δίκτυα CP και τη διαδικασία της *ιεραρχικής αποσύνθεσης* (hierarchical decomposition) είναι να επιτραπεί η δημιουργία ενός πολύπλοκου μοντέλου επαναχρησιμοποιώντας άλλα μικρότερα δίκτυα τα οποία «συνεργάζονται» με έναν καλά ορισμένο τρόπο. Με τον τρόπο αυτό αποφεύγεται η μονολιθική κατασκευή και η δυσκολία επέκτασης και ανάλυσης του σύνθετου μοντέλου.

Η ιδέα αυτή είναι αντίστοιχη με τις modular προσεγγίσεις που συναντά κανείς στις σύγχρονες γλώσσες προγραμματισμού. Στο υψηλό επίπεδο επιδιώκεται η απλουστευμένη περιγραφή της διαδικασίας που μοντελοποιείται χωρίς να υπάρχουν λεπτομερείς καταγραφές του τρόπου με τον οποίο αυτή λειτουργεί εσωτερικά. Τα χαμηλότερα επίπεδα καθορίζουν τις εσωτερικές λειτουργικές λεπτομέρειες.

5.1.1 Ιεραρχικά έγχρωμα δίκτυα Petri

Σε ένα ιεραρχικό έγχρωμο δίκτυο Petri (HCPN) είναι δυνατή η συσχέτιση μιας μετάβασης (και βέβαια των άμεσα σχετιζόμενων τόξων και θέσεων) με ένα ξεχωριστό δίκτυο CP, που μπορεί να παρέχει μια πιο ακριβή και λεπτομερή περιγραφή της λειτουργίας που αναπαριστά η συγκεκριμένη μετάβαση ([12]). Τόσο η γενική συμπεριφορά όσο και οι επιμέρους θα πρέπει να είναι κατάλληλα ολοκληρωμένες (integrated) ώστε να γίνεται εφικτή η αναμενόμενη συνολική συμπεριφορά του συστήματος.

Οι μεταβάσεις που συνδέονται με άλλα δίκτυα CP ονομάζονται *μεταβάσεις αντικατάστασης* (substitution transitions). Το άλλο αυτό δίκτυο καταλαμβάνει πλέον τη δική του υποσελίδα (subpage) και κάθε τέτοιο δίκτυο έχει τη δική του σήμανση που είναι ανεξάρτητη από τη σήμανση του δικτύου του υψηλότερου επιπέδου (που καταλαμβάνει την *κύρια σελίδα* (prime page)).

Κάθε υποσελίδα έχει έναν αριθμό από θέσεις που χαρακτηρίζονται ως θέσεις εισόδου (In), εξόδου (Out) ή και διπλού ρόλου (I/O). Οι θέσεις αυτές αναφέρονται ως *θύρες* (ports) και ουσιαστικά αποτελούν το μέσο επικοινωνίας του υποδικτύου με τα δίκτυα που το εμπλέκουν.

Μέσω των θέσεων εισόδου, το υποδίκτυο λαμβάνει μάρκες ενώ παραδίδει μάρκες στα «γειτονικά» δίκτυα μέσω των θέσεων που χαρακτηρίζονται ως θύρες εξόδου.

Μια θέση με το χαρακτηρισμό I/O μπορεί να παίξει και τους δύο ρόλους ταυτόχρονα.

Στο δίκτυο του αμέσως υψηλότερου επιπέδου που εμπλέκει το υποδίκτυο, οι θέσεις που σχετίζονται με τη μετάβαση αντικατάστασης χαρακτηρίζονται ως υποδοχείς (sockets) και μπορεί να είναι είτε υποδοχείς εισόδου είτε υποδοχείς εξόδου. Για να καθοριστεί η ακριβής σχέση του υποδικτύου με το αμέσως υψηλότερου επιπέδου δίκτυο θα πρέπει να περιγραφεί η διασύνδεση των θέσεων-υποδοχέων με τις θέσεις-θύρες μέσω της διαδικασίας *ανάθεσης θυρών* (port assignment). Όταν μια θέση-θύρα ανατίθεται σε μια θέση-υποδοχή τότε γίνονται ταυτόσημες δηλαδή αποτελούν διαφορετικές αναπαραστάσεις μιας μοναδικής θέσης. Αυτό πρακτικά σημαίνει ότι έχουν πάντα την ίδια σήμανση. Όταν μια υποδοχή εισόδου δέχεται μια μάρκα αυτή αυτομάτως γίνεται διαθέσιμη και στην θύρα εισόδου του υποδικτύου και άρα μπορεί να χρησιμοποιηθεί για να ενεργοποιήσει μεταβάσεις του υποδικτύου. Κατ' αναλογία, το υποδίκτυο μπορεί να παράγει μάρκες σε μια θύρα εξόδου του. Αυτόματα οι μάρκες γίνονται διαθέσιμες στις αντίστοιχες υποδοχές εξόδου και μπορούν πλέον να χρησιμοποιηθούν από μεταβάσεις του υψηλότερου επιπέδου δικτύου.

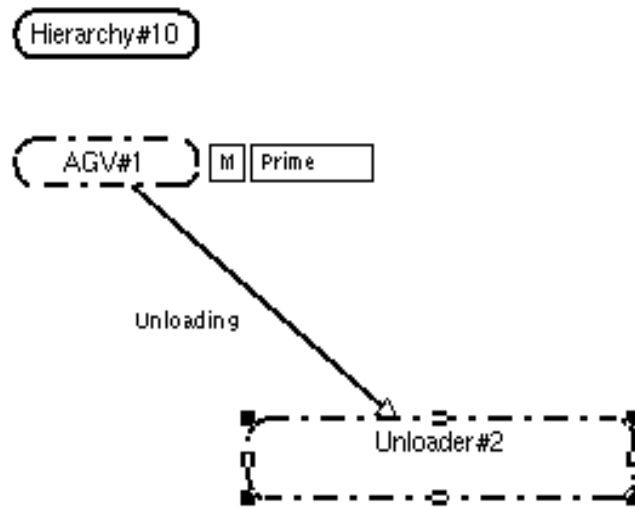
Τα υποδίκτυα σχηματίζουν επίπεδα ιεραρχίας μιας και με τη σειρά του κάθε υποδίκτυο μπορεί να περιέχει μεταβάσεις αντικατάστασης και να επικοινωνεί έτσι με δικά του υποδίκτυα. Για να περιγραφεί η ιεραρχία αυτή χρησιμοποιείται συχνά ένας *γράφος ιεραρχίας* (page hierarchy graph), όπως αυτός του σχήματος 5.1.

Στα ιεραρχικά δίκτυα υπάρχει ακόμη η έννοια των *θέσεων συγχώνευσης* (fusion places) που επιτρέπει στον μελετητή να ορίσει ένα σύνολο από θέσεις ως ταυτόσημες (δηλαδή αναπαριστούν μια μοναδική θέση και άρα έχουν την ίδια σήμανση) παρόλο που μπορεί να βρίσκονται σε διαφορετικά σημεία (ακόμη και σε διαφορετικές σελίδες) στην ιεραρχία των δικτύων.

Στην επόμενη παράγραφο υιοθετούνται οι έννοιες των ιεραρχικών δικτύων με στόχο να εμπλουτιστεί η λειτουργικότητα ενός θεμελιώδους μοντέλου μέσω της επαναχρησιμοποίησης ενός άλλου θεμελιώδους μοντέλου.

5.1.2 Παράδειγμα ιεραρχικού δικτύου

Η εκφόρτωση του οχήματος που μοντελοποιείται με το δίκτυο του σχήματος 4.27 έχει θεωρηθεί ότι είναι μια ακαριαία διαδικασία και μάλιστα ότι ο μηχανισμός εκφόρτωσης δεν παθαίνει ποτέ βλάβη, κάτι που προφανώς απέχει πολύ από την πραγματι-

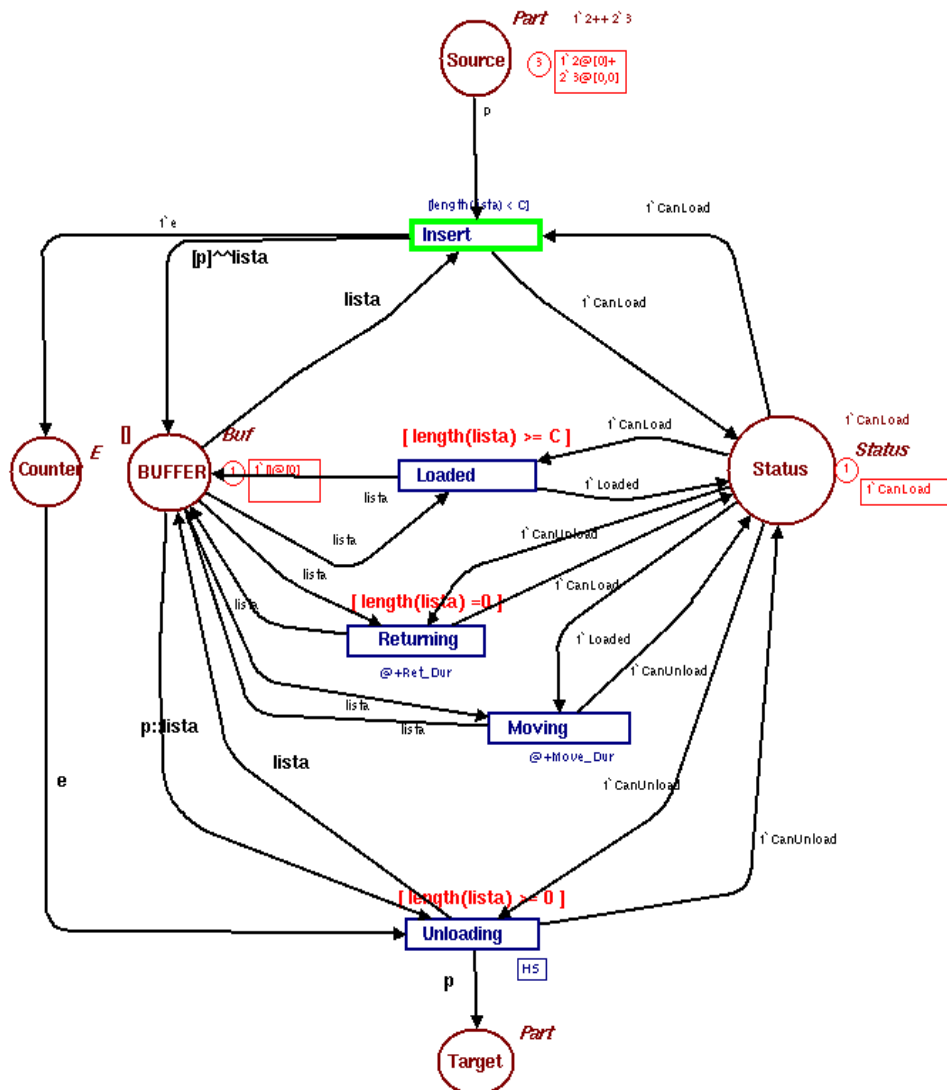


Σχήμα 5.1: Ενδεικτικός γράφος ιεραρχίας

κότητα. Για μια πιο ρεαλιστική αναπαράσταση της όλης διαδικασίας, η μετάβαση που αναπαριστά τη διαδικασία εκφόρτωσης θα αντικατασταθεί από ένα υποδίκτυο που θα αναπαριστά μια μηχανή εκφόρτωσης. Για την αναπαράσταση της μηχανής αυτής μπορεί να χρησιμοποιηθεί το μοντέλο του σχήματος 4.25 που αναπαριστά μια μηχανή που μπορεί να χαλάσει. Το προϊόν στην περίπτωση αυτή δεν υπόκειται κάποια κατεργασία αλλά απλά μεταφέρεται από τον αποθηκευτικό χώρο του οχήματος στον αποθηκευτικό χώρο του προορισμού του οχήματος. Έτσι, η μετάβαση Unloading του σχήματος 5.2 χαρακτηρίζεται με την ετικέτα [HS].¹

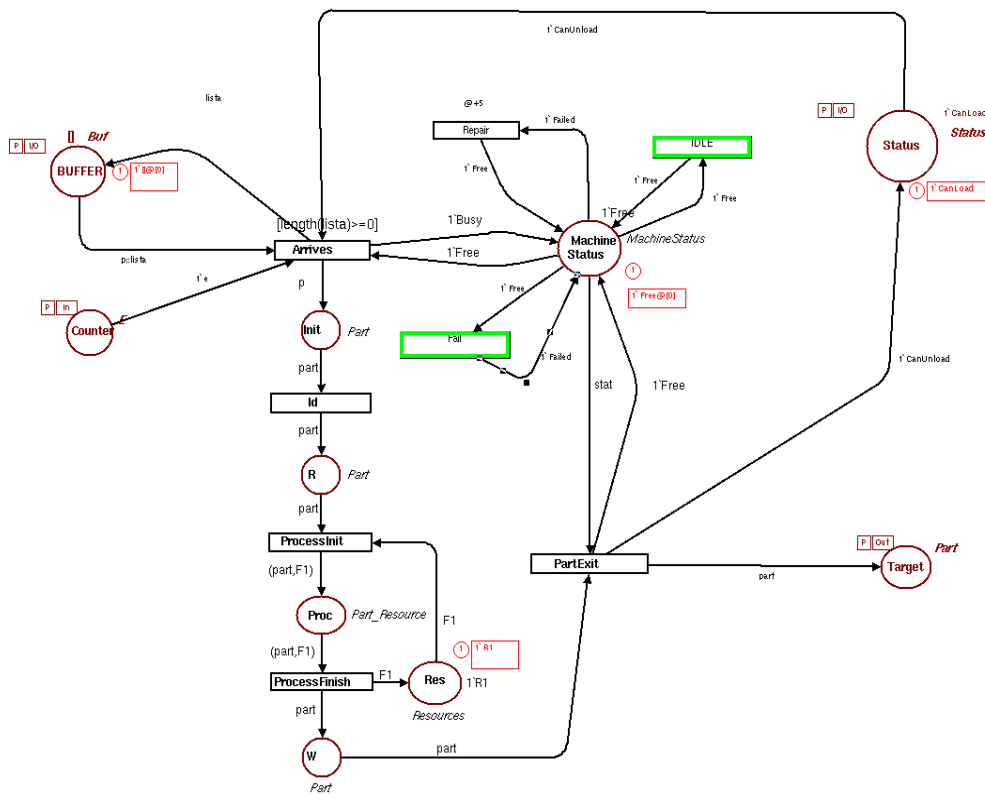
Το δίκτυο που αναπαριστά το μηχανισμό εκφόρτωσης αναπτύσσεται σε μια διαφορετική «σελίδα» του εργαλείου Design/CPN και βέβαια μπορεί να επαναχρησιμοποιηθεί απευθείας το μοντέλο του σχήματος 4.25 μέσω των διαδικασιών επιλογής, αντιγραφής και επικόλλησης που προσφέρονται από το εργαλείο σχεδιασμού. Το δίκτυο αυτό, όπως φαίνεται στο σχήμα 5.3, είναι εμπλουτισμένο με τις θέσεις που χρειάζονται για την επικοινωνία με το κυρίως δίκτυο (BUFFER, Counter, Status, Target)

¹Στο παράρτημα Α' αναφέρονται αναλυτικά οι ενέργειες για να γίνει αυτό με τη βοήθεια του εργαλείου Design/CPN.



Σχήμα 5.2: Κύριο δίκτυο του σύνθετου μοντέλου οχήματος μεταφοράς

και οι οποίες έχουν χαρακτηριστεί με τις ετικέτες [In], [Out] ή [I/O] ανάλογα με τη λειτουργία τους.



Σχήμα 5.3: Υποδίκτυο του σύνθετου μοντέλου οχήματος μεταφοράς που αναπαριστά το μηχανισμό εκφόρτωσης

```
use "/CPN_EXAMPLES/tod.sml";
color Int = int;
color Part = Int timed;
color Status = with CanLoad|Loaded|Moving|CanUnload;
color Data = Part;
color Resources = with R1|R2;
color List_Resources = list Resources;
color Part_Resource = product Part*Resources;
color MachineStatus = with Busy|Free|Failed timed;

color E= with e;
color Buf = list Part;

var lista:Buf;
var p:Part;

val C=2;
val Move_Dur = 4;
val Ret_Dur = 2;

var stat: MachineStatus;
var pf: Part_Resource;
var F,F1: Resources;
var L,L1: List_Resources;
var part: Part;
```

Σχήμα 5.4: Ορισμοί για το σύνθετο μοντέλο οχήματος μεταφοράς

Κεφάλαιο 6

Συμπεράσματα και μελλοντικές επεκτάσεις

Στα πλαίσια της εργασίας αυτής αρχικά εντοπίστηκαν τα θεμελιώδη στοιχεία ενός συστήματος παραγωγής και στη συνέχεια αναπτύχθηκαν τα αντίστοιχα μοντέλα με χρήση *έγχρωμων* χρονικών δικτύων Petri. Έτσι, υλοποιήθηκαν και μελετήθηκαν μοντέλα για μηχανές διαφόρων τύπων (π.χ. παράλληλες, αφοσιωμένες, πολλαπλών λειτουργιών, σύνθεσης, αποσύνθεσης κ.ά.), συστήματα μεταφοράς (π.χ. μάντες και οχήματα) και συστήματα προσωρινής αποθήκευσης ενώ καλύφθηκαν και θέματα διαδικασίας παραγωγής και δρομολόγησης.

Συχνά, αντίστοιχες μοντελοποιήσεις με χρήση απλών (συνήθων) δικτύων Petri περιορίζονται σε «αφιερωμένα» συστήματα παραγωγής, δηλαδή συστήματα τα οποία επεξεργάζονται ή παράγουν ένα τύπο προϊόντος σε όλο το χρόνο λειτουργίας τους. Όπως φάνηκε και από την ενδεικτική παρουσίαση των μοντέλων με χρήση απλών δικτύων, ο περιορισμός αυτός προκύπτει από την έλλειψη δυνατοτήτων κλιμάκωσης των μοντέλων αυτών ώστε να καλύψουν πιο σύνθετες περιπτώσεις με ταυτόχρονη διατήρηση μιας συμπαγούς αναπαράστασης. Αυτόν ακριβώς τον περιορισμό έρχονται να άρουν τα *έγχρωμα* δίκτυα Petri που συνδυάζουν τη δύναμη αναπαράστασης των απλών δικτύων με τις δυνατότητες των γλωσσών προγραμματισμού αναπτύσσοντας μια γλώσσα μοντελοποίησης τόσο θεωρητικά θεμελιωμένης όσο και πρακτικά ευέλικτης ώστε να χρησιμοποιηθεί για τα συστήματα του μεγέθους και της πολυπλοκότητας που συναντώνται στη βιομηχανία. Έτσι, η χρήση των *έγχρωμων* δικτύων Petri δίνει τη δυνατότητα να επεκταθεί η συμπαγής περιγραφή της μοντελοποίησης σε *εέλικτα* συστήματα παραγωγής πολλών και διαφορετικών προϊόντων αλλά και να εξεταστεί η δυνατότητα μοντελοποίησης της παραγωγής προϊόντων με διαφορετικές δρομολογήσεις μέσα σε ένα σύστημα.

Ακόμη, μέσα από την υιοθέτηση των *ιεραρχικών έγχρωμων* δικτύων Petri και τη διαδικασία της *ιεραρχικής αποσύνθεσης* παρουσιάστηκε η δυνατότητα να δημιουργηθεί ένα πολύπλοκο μοντέλο επαναχρησιμοποιώντας τα θεμελιώδη μοντέλα τα οποία συνεργάζονται με ένα καλά ορισμένο τρόπο. Με τον τρόπο αυτό αποφεύγεται η μονολιθική κατασκευή και η δυσκολία επέκτασης, ανάλυσης και αποσφαλμάτωσης του σύνθετου μοντέλου.

Σύμφωνα με τα παραπάνω, ο πρωταρχικός στόχος της ανάπτυξης των θεμελιωδών μοντέλων συστημάτων παραγωγής στην εργασία αυτή ήταν η μοντελοποίηση σύνθετων συστημάτων παραγωγής και δευτερευόντως η ποσοτική αξιολόγηση των συστημάτων και η ανάλυση των επιδόσεών τους. Σε ένα επόμενο βήμα θα μπορούσε

σαν στόχος να τεθεί, αρχικά, ο εμπλουτισμός των συγκεκριμένων μοντέλων τόσο ως προς τη λειτουργικότητά τους όσο και ως προς τις δυνατότητες παραμετροποίησης τους από κάποιον κεντρικό μηχανισμό ελέγχου. Απώτερος σκοπός θα ήταν η δημιουργία μιας βιβλιοθήκης από στοιχεία έτοιμα προς επαναχρησιμοποίηση με όσο το δυνατόν πιο απλή και αυτοματοποιημένη διαδικασία προσαρμογής.

Μια άλλη προέκταση της εργασίας αυτής θα μπορούσε να κάνει χρήση των μοντέλων για την κατασκευή πιο σύνθετων (ιεραρχικών) συστημάτων που θα μπορούσαν να καλύψουν εφαρμογές ελέγχου της συμπεριφοράς των συστημάτων σε πραγματικό χρόνο αλλά και αξιολόγησης της απόδοσής τους. Με εργαλείο τη δυνατότητα καθορισμού ιεραρχικών μοντέλων με αυξανόμενο επίπεδο λεπτομέρειας όσο αυξάνεται και ο αριθμός του επιπέδου αναπαράστασης, ένας ακόμη στόχος θα μπορούσε να είναι και ο καθορισμός ενός γενικού μοντέλου-πλαίσιου (framework) για την αναπαράσταση ενός ευέλικτου συστήματος παραγωγής που θα μπορεί εύκολα να εμπλέξει ως στοιχεία του τα θεμελιώδη μοντέλα των συστημάτων παραγωγής. Στη συνέχεια, θα μπορούσαν να μελετηθούν διαδικασίες και τεχνικές βελτιστοποίησης μιας σειράς από παραμέτρους των μοντέλων με στόχο μια συγκεκριμένη επιθυμητή συμπεριφορά ως προς τις χρονικές επιδόσεις των αναπαραστούμενων διαδικασιών. Κάτι τέτοιο υποβοηθάται από το πλήρες μαθηματικό υπόβαθρο των δικτύων Petri και από τις ιδιότητες που φέρουν τα θεμελιώδη μοντέλα.

Για την επιβεβαίωση των βασικών ιδιοτήτων των θεμελιωδών μοντέλων της εργασίας αυτής αλλά και της σωστής λειτουργίας τους χρησιμοποιήθηκε η τεχνική της προσομοίωσης της λειτουργίας τους για μεγάλο αριθμό βημάτων μέσα από την εξαγωγή του γράφου καλυψιμότητας για συγκεκριμένες στοιχειώδεις αρχικές σημάνσεις. Η αυστηρή απόδειξη των ιδιοτήτων ή της συμπεριφοράς αυτής με μαθηματικό τρόπο, όμως, είναι πρακτικά αδύνατη λόγω της πολυπλοκότητας που εισάγεται. Για το λόγο αυτό θα πρέπει να εξεταστούν και να υιοθετηθούν κι άλλες τεχνικές επιβεβαίωσης της σωστής λειτουργίας τόσο των θεμελιωδών όσο και των πιο σύνθετων μοντέλων. Το τελευταίο γίνεται ακόμη πιο σημαντικό όταν τα διάφορα στοιχειώδη μοντέλα επαναχρησιμοποιούνται στα πλαίσια ενός γενικότερου framework οπότε η σωστή τους λειτουργία θα πρέπει να επιβεβαιωθεί σε σχέση με το νέο τους περιβάλλον κάθε φορά. Ενδεικτικά αναφέρεται στο σημείο αυτό π.χ. η χρήση κανόνων χρονικής λογικής (temporal logic) [5].

Τέλος, ως προς την βελτίωση του βαθμού αυτοματοποίησης των διαδικασιών

μοντελοποίησης, επαναχρησιμοποίησης, προσομοίωσης κλπ. θα ήταν ενδιαφέρον να εξεταστεί η επικοινωνία του εργαλείου Design/CPN με άλλα εργαλεία ή γλώσσες προγραμματισμού με στόχο τον «προγραμματιστικό» έλεγχο των συστημάτων. Προς την κατεύθυνση αυτή μπορεί να εξεταστεί σε μια μελλοντική επέκταση της εργασίας η επικοινωνία των θεμελιωδών μοντέλων ή ενός πιο σύνθετου μοντέλου με εφαρμογές Java [14] με δυνατότητες όπως δυναμική ανάθεση τιμών στις διάφορες μεταβλητές του συστήματος, παρακολούθηση και αναφορά της εξέλιξης της λειτουργίας του κ.ά.

Παράρτημα Α΄

Χρήση του εργαλείου υλοποίησης δικτύων

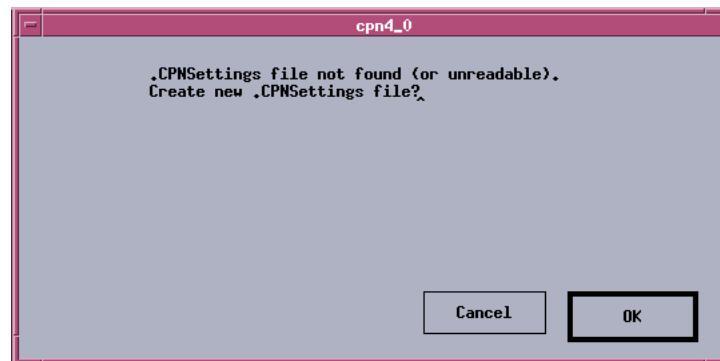
Την επίβλεψη για την ανάπτυξη του εργαλείου Design/CPN την έχει η σχετική ομάδα CPN Group του Πανεπιστημίου του Aarhus στη Δανία. Η πιο πρόσφατη έκδοση του εργαλείου ονομάζεται CPNTools και υποστηρίζει μόνο το περιβάλλον των Windows. Στις επόμενες παραγράφους θα ασχοληθούμε με το εργαλείο Design/CPN όπως αυτό εγκαταστάθηκε και χρησιμοποιήθηκε σε περιβάλλον Linux.

Στον σχετικό ιστοτόπο [4] μπορεί να βρει κανείς τόσο πιο λεπτομερείς οδηγίες χρήσης όσο και πληθώρα άλλου υλικού σχετικά με τα δίκτυα CP και την εξέλιξη της σχετικής έρευνας.

Α΄.1 Εγκατάσταση και ρυθμίσεις του εργαλείου Design/CPN

Με την εγκατάσταση του πακέτου δημιουργείται στην επιφάνεια εργασίας ένα εικονίδιο με την ονομασία CPN. Αν δεν δημιουργηθεί το εικονίδιο, για την εκκίνηση του εργαλείου εκτελείται η εφαρμογή είτε μέσω του shell script με το όνομα cpn στον κατάλογο που έχει εγκατασταθεί η εφαρμογή είτε μέσω της εκτέλεσης της ίδιας της εφαρμογής (το αρχείο με όνομα cpn4_0 στην περίπτωσή μας).

Την πρώτη φορά εκτέλεσης ένα pop-up παράθυρο εμφανίζεται σχετικά με το αρχείο αποθήκευσης των ρυθμίσεων του προγράμματος:



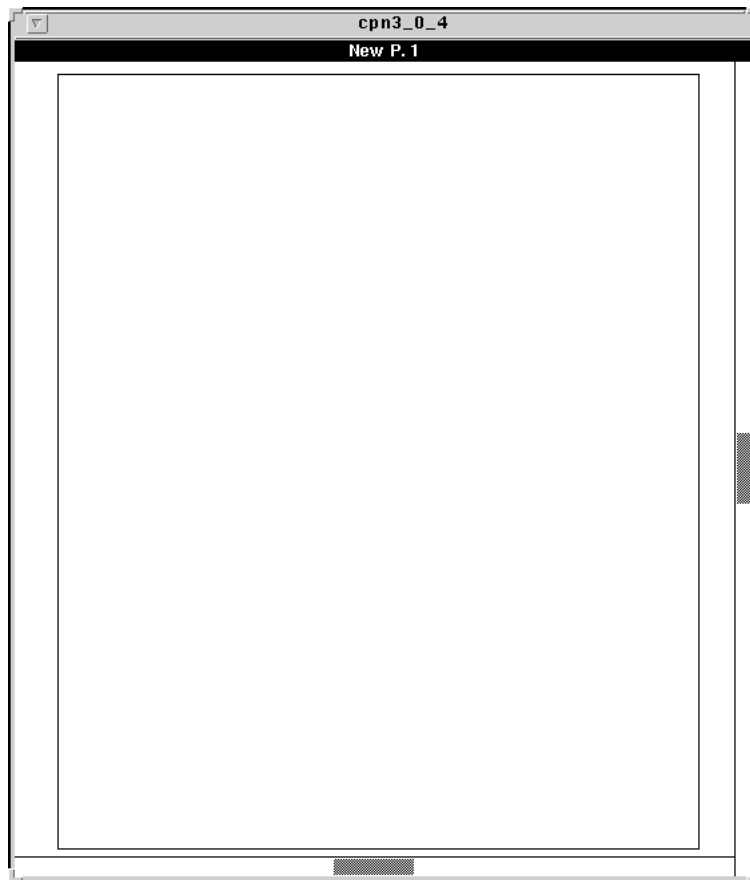
Επιλέγοντας **OK** δημιουργείται το αρχείο ρυθμίσεων και εμφανίζεται η ακόλουθη μπάρα εντολών στην οθόνη:



Α΄.1.1 Δημιουργία νέου δικτύου (page)

Για τη δημιουργία ενός δικτύου CPN εξαρχής, επιλέγουμε **New** στο μενού **File**. Το παράθυρο με τίτλο **New** που εμφανίζεται

Α'.1. ΕΓΚΑΤΑΣΤΑΣΗ ΚΑΙ ΡΥΘΜΙΣΕΙΣ ΤΟΥ ΕΡΓΑΛΕΙΟΥ ΔΕΣΙΓΝ/CPN95



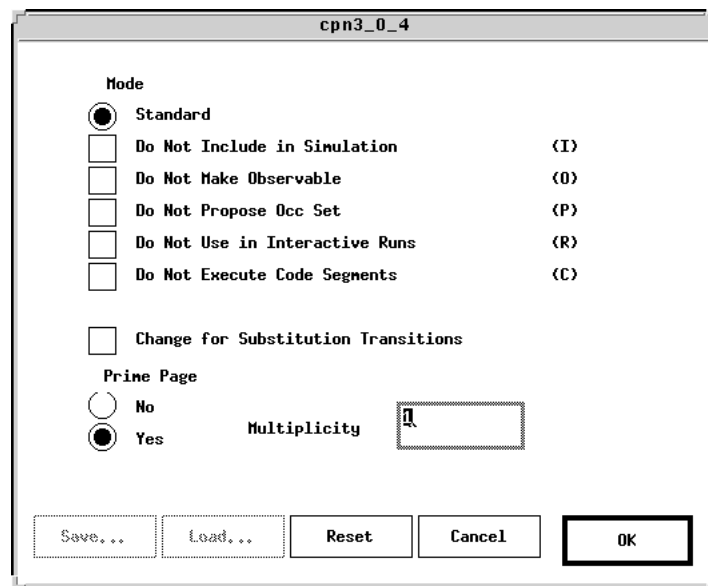
είναι η βασική επιφάνεια εργασίας του εργαλείου CPN Editor όπου μπορούμε να σχεδιάσουμε και να τροποποιήσουμε ένα δίκτυο CP.

Α'.1.2 Καθορισμός κύριας σελίδας

Στο περιβάλλον του Design/CPN, κατ' αντιστοιχία με τη γλώσσα προγραμματισμού C όπου η εκτέλεση του προγράμματος ξεκινά από τη διαδικασία *main*, η εκτέλεση ξεκινά από την κύρια σελίδα (prime page). Από το μενού **Page** επιλέγοντας **Open Page** εμφανίζεται ένα παράθυρο με -αρχικά- δύο κόμβους που αναπαριστούν την ιεραρχία και τις υπάρχουσες σελίδες του δικτύου.



Επιλέγοντας τον κόμβο **New** (κάνοντας κλικ μια φορά πάνω του), και στη συνέχεια επιλέγοντας **Mode Attributes** από το μενού **Set** εμφανίζεται το ακόλουθο παράθυρο:



Πλέον μπορεί να ξεκινήσει ο σχεδιασμός του δικτύου CP μέσω της γραφικής απεικόνισής του στην ενεργή σελίδα.

Α΄.1.3 Το μενού CPN

Ένα διάγραμμα ενός δικτύου CP αποτελείται από τμήματα δηλώσεων (declaration nodes), θέσεις, μεταβάσεις και τόξα. Τα «αντικείμενα» αυτά επιλέγονται μέσα από το μενού CPN.

Α'.1.4 Δημιουργία του κεντρικού σημείου δηλώσεων

Αφού επιλέξουμε τη σελίδα **New**, επιλέγουμε **Declaration Node** κάτω από το μενού **CPN**. Στο παράθυρο που εμφανίζεται επιλέγουμε **Global**.

CPN	Aux	Set	Makeup	Page	Gro
Place				Alt+E	
Transition				Alt+B	
Arc				Alt+K	
CPN Region...				Alt+R	
Declaration Node...					
<hr/>					
Chart...					
<hr/>					
Move to Subpage					
Replace by Subpage...					
<hr/>					
Substitution Transition					
Fusion Place...					
Port Place...					
Port Assignment...					
<hr/>					
Syntax Check				Alt+quoteright	
<hr/>					
Remove Sim Regions...					

Με τη βοήθεια του ποντικιού, σχεδιάζουμε ένα πλαίσιο μέσα στη σελίδα με τίτλο **New**. Το πλαίσιο αυτό θα περιλαμβάνει τις δηλώσεις σχετικά με τους διάφορους τύπους, καθώς και τον τύπο και την ονομασία κάθε μεταβλητής που εμφανίζεται στο δίκτυο. Ακόμη, εδώ συνήθως ορίζονται και συναρτήσεις στη γλώσσα **SML**.

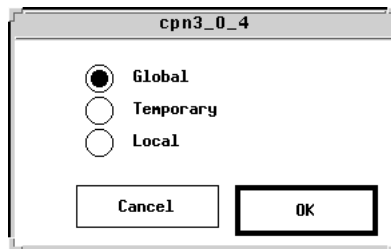
Πατώντας το πλήκτρο **Escape** μπορούμε να τελειώσουμε την εισαγωγή των δηλώσεων.

Α'.1.5 Προσθήκη θέσεων και μεταβάσεων

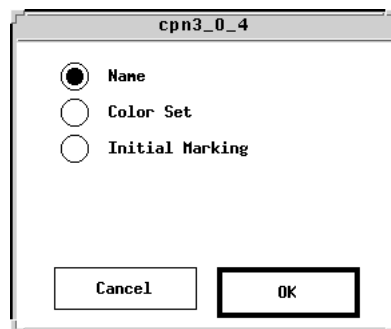
Κάτω από το μενού **CPN** επιλέγουμε **Place**. Με τη βοήθεια και πάλι του ποντικιού, σχεδιάζουμε τη θέση στη σελίδα (με αριστερό κλικ και διαρκώς πατημένο).

Αμέσως μετά το σχεδιασμό της έλλειψης (ή του κύκλου) που αναπαριστά μια θέση του δικτύου, καλούμαστε να την ονομάσουμε εισάγοντας το όνομά της στο σχετικό πλαίσιο κειμένου που εμφανίζεται.

Εναλλακτικά, η ονομασία μπορεί να τεθεί για κάθε στοιχείο του δικτύου μέσω της επιλογής **CPN Region** κάτω από το μενού **CPN** οπότε εμφανίζεται το παράθυρο



Μέσω της ίδιας διαδικασίας και επιλέγοντας αυτή τη φορά **Color Set** μπορεί να εισαχθεί ο τύπος των μαρκών που μπορεί να φιλοξενήσει μια θέση. Ο καθορισμός του τύπου (χρώματος) κάθε θέσης είναι υποχρεωτικός. Πλέον η θέση θα έχει μορφή παρόμοια με το ακόλουθο σχήμα

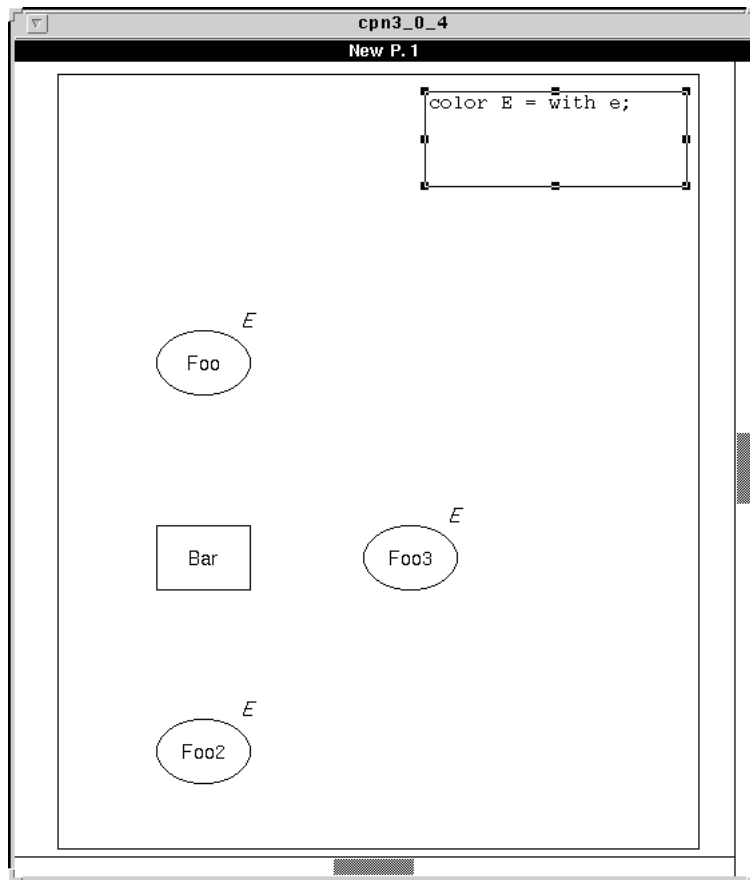


Οι μεταβάσεις μπορούν να δημιουργηθούν με παρόμοια διαδικασία. Αντίθετα από τις θέσεις, δεν χρειάζεται ο καθορισμός χρώματος. Με την προσθήκη μιας μετάβασης και δυο ακόμη θέσεων η σελίδα του δικτύου μπορεί να διαμορφωθεί όπως στο παρακάτω σχήμα

Όπως και πολλά άλλα γραφικά εργαλεία, το περιβάλλον δημιουργίας και τροποποίησης (editor) του εργαλείου Design/CPN έχει πολλές άλλες λειτουργίες όπως δυνατότητες μαζικής επιλογής, αποκοπής, αντιγραφής και επικόλλησης (μέσω του μενού **Edit** ή ακόμη και λειτουργίες στοίχισης των θέσεων και των μεταβάσεων (στο μενού **Align**).

Οποιαδήποτε σημείο του δικτύου που περιέχει κείμενο μπορεί να αλλαχθεί (αφού πρώτα επιλεγεί με ένα απλό κλικ) μέσω της επιλογής **Turn On Text** του μενού **Text**. Αφού γίνει η όποια αλλαγή κειμένου, με την επιλογή **Turn Off Text** επανερχόμαστε στην κατάσταση εισαγωγής στοιχείων του δικτύου.

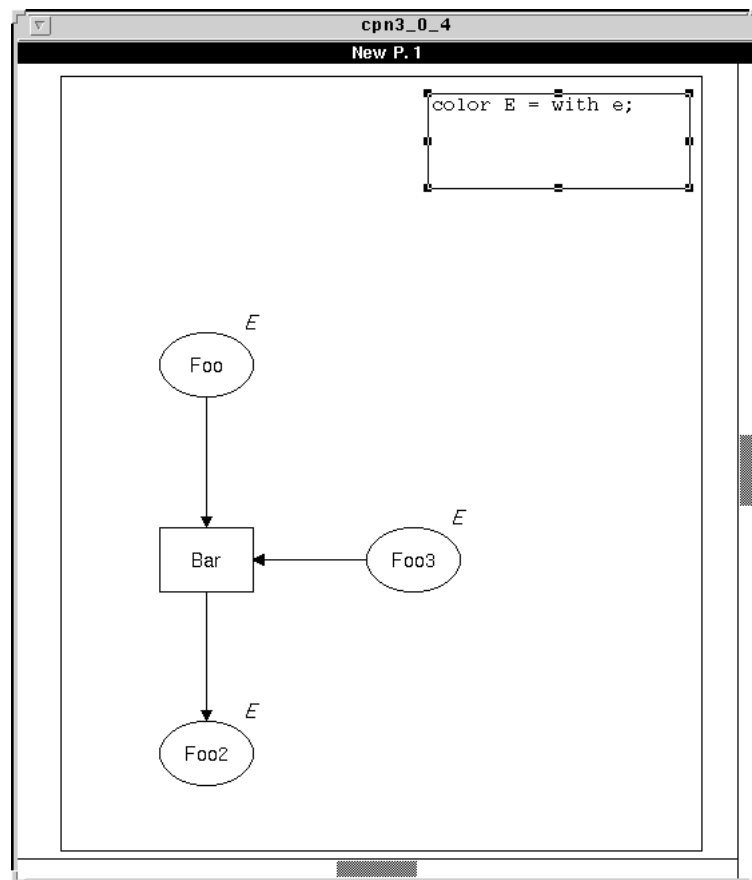
Α'.1. ΕΓΚΑΤΑΣΤΑΣΗ ΚΑΙ ΡΥΘΜΙΣΕΙΣ ΤΟΥ ΕΡΓΑΛΕΙΟΥ ΔΕΣΙΓΝ/CPN99



Α'.1.6 Εισαγωγή τόξων

Επιλέγοντας **Arc** από το μενού **CPN** και με τη χρήση του ποντικιού (κάνοντας κλικ πρώτα στην αρχική θέση (ή μετάβαση), κρατώντας πατημένο το δεξί πλήκτρο, οδηγώντας τον δείκτη (pointer) στην τελική μετάβαση (θέση) και αφήνοντάς το, εισάγεται ένα τόξο που πλέον συνδέει τη θέση με τη μετάβαση (ή τη μετάβαση με τη θέση).

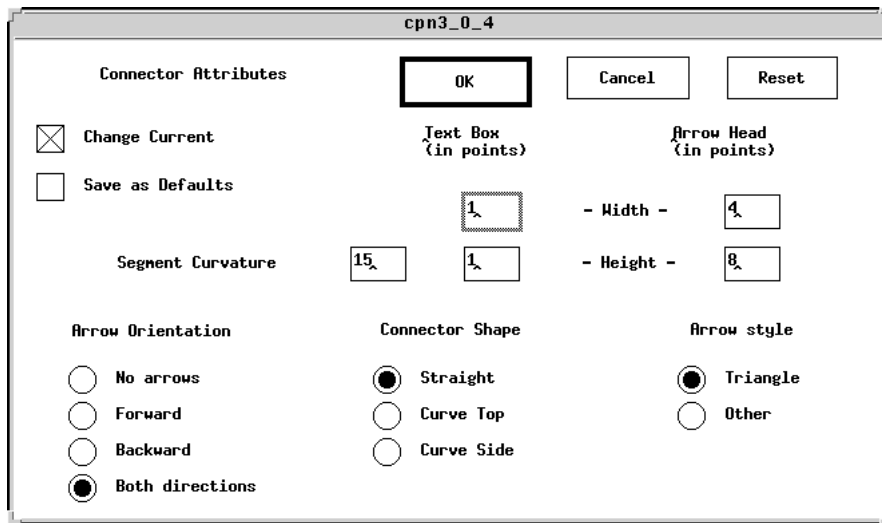
Επαναλαμβάνοντας τη διαδικασία και για άλλα τόξα το δίκτυο πλέον μπορεί να πάρει μια μορφή όπως παρακάτω:



Τα τόξα μπορούν πλέον να εμπλουτιστούν με λειτουργικότητα. Επιλέγοντας ένα τόξο με αριστερό κλικ του ποντικιού, επιλέγοντας **CPN Region** από το μενού **CPN** και στη συνέχεια κάνοντας αριστερό κλικ κοντά στο τόξο, μπορεί να εισαχθεί το κείμενο που θα περιλαμβάνει ονόματα μεταβλητών κλπ.

Ένα τόξο μπορεί να γίνει και διπλής κατεύθυνσης. Αυτό γίνεται επιλέγοντας το συγκεκριμένο τόξο και στη συνέχεια επιλέγοντας **Shape Attributes** από το μενού **Set** οπότε εμφανίζεται το μενού

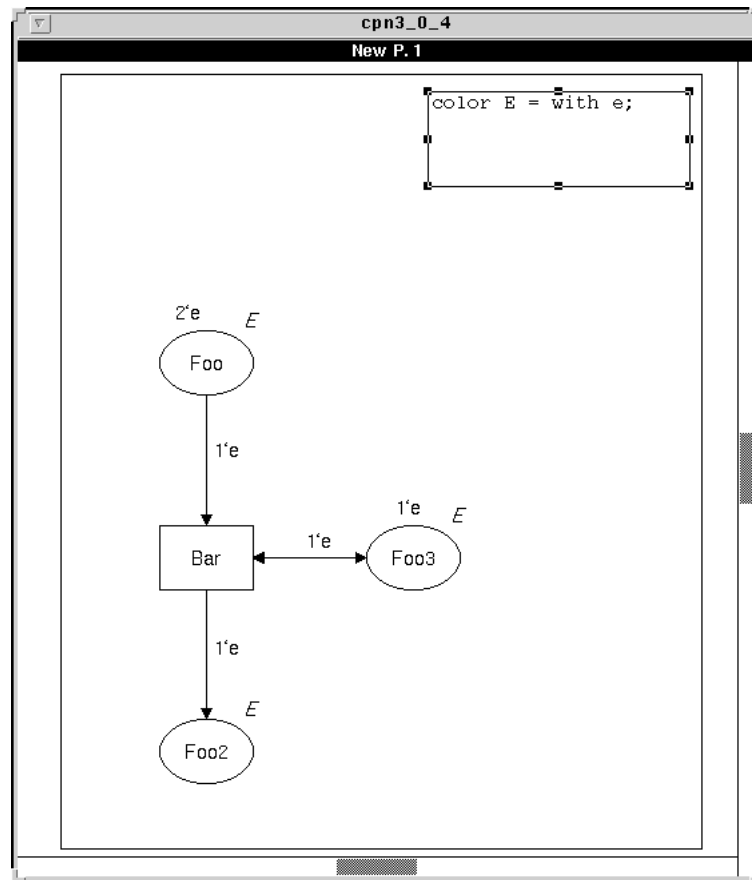
Α'.1. ΕΓΚΑΤΑΣΤΑΣΗ ΚΑΙ ΡΥΘΜΙΣΕΙΣ ΤΟΥ ΕΡΓΑΛΕΙΟΥ ΔΕΣΙΓΝ/ΎΠΝ101



από όπου επιλέγοντας **Both directions** το τόξο γίνεται διπλής κατεύθυνσης.

Α'.1.7 Αρχική σήμανση

Οι θέσεις χρειάζονται κάποια αρχική σήμανση. Ο καθορισμός της γίνεται επιλέγοντας τη θέση (κάνοντας αριστερό κλικ πάνω της) και στη συνέχεια επιλέγοντας **CPN Region** από το μενού **CPN**, μέσω της επιλογής **Initial Marking** και κάνοντας αριστερό κλικ κοντά στη συγκεκριμένη θέση. Η αρχική σήμανση ((2^e) μπορεί να εισαχθεί. Αυτό σημαίνει ότι αρχικά στη συγκεκριμένη θέση βρίσκονται δύο μάρκες με τιμή e.



Α΄.1.8 Έλεγχος σύνταξης

Όταν ο σχεδιασμός του δικτύου ολοκληρωθεί ακολουθεί ο συντακτικός έλεγχος μέσω της επιλογής **Syntax Check** του μενού **CPN**. Τα όποια λάθη εμφανίζονται σε πλαίσια κειμένου στη σελίδα διόρθωσης του δικτύου. Αφού γίνουν οι όποιες διορθώσεις, το μοντέλο πρέπει να σωθεί.

Α΄.1.9 Προσομοίωση του δικτύου

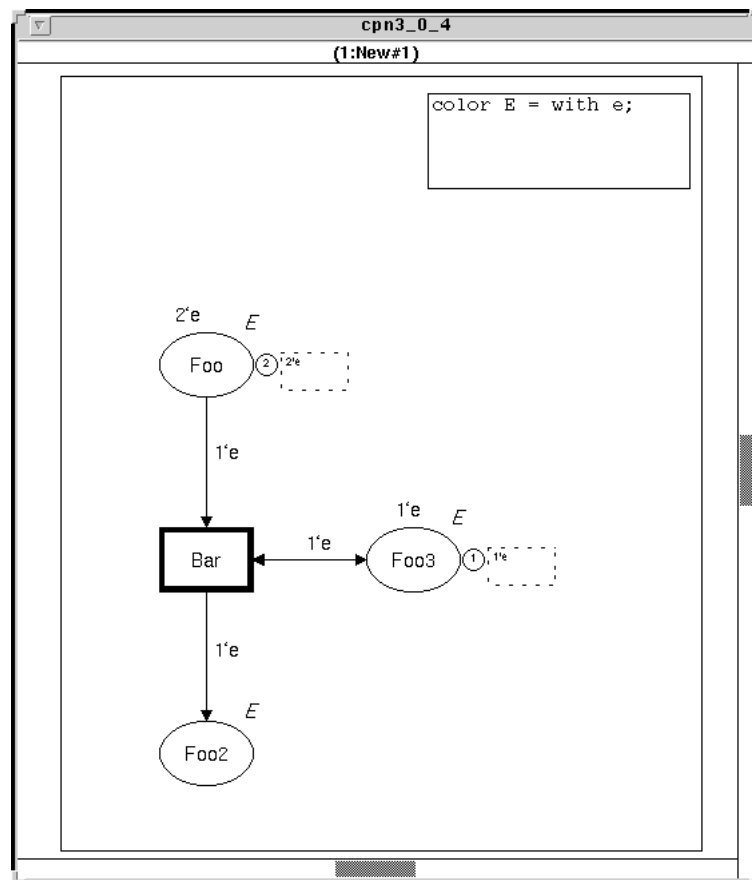
Από το μενού **File** επιλέγουμε **Enter Simulator**. Μετά από λίγο, αλλάζει η εμφάνιση (view) του περιβάλλοντος εργασίας μιας και έχει ξεκινήσει η διαδικασία προσομοίωσης.

Προσομοίωση βήμα - προς -βήμα

Αρχικά επιλέγουμε **Interactive Simulation Options** από το μενού **Set**. Επιλέγοντας στη συνέχεια **Breakpoints Between Steps**, η προσομοίωση σταματά σε κάθε βήμα. Αν δεν είναι ενεργοποιημένη η επιλογή αυτή τότε η προσομοίωση προχωρά πιο γρήγορα χωρίς παρεμβάσεις από το χρήστη. Από το μενού **Sim** επιλέγοντας **Interactive Run**, το δίκτυο θα εκτελεστεί βάσει του αλγορίθμου που έχει περιγραφεί στην παράγραφο (2.2.1) και θα σταματά σε κάθε βήμα.

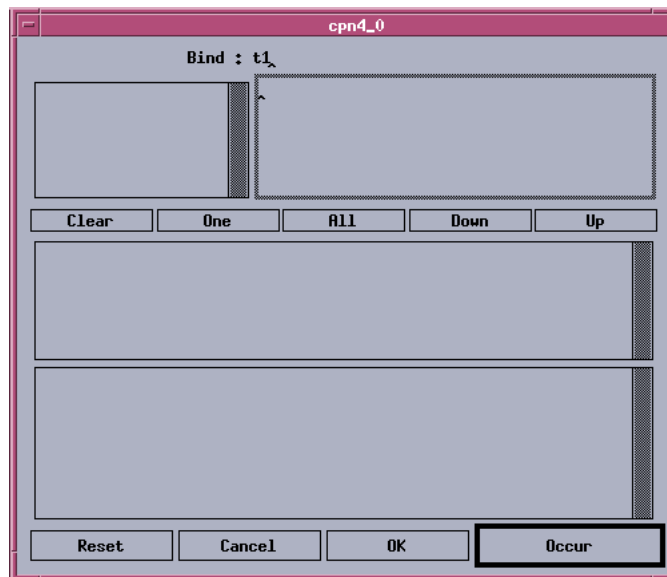
Εκτελώντας συγκεκριμένες μεταβάσεις και επιλέγοντας συσχετίσεις

Τόσο η βήμα- προς -βήμα προσομοίωση όσο και η αυτόματη προσομοίωση επιλέγουν τυχαία τα στοιχεία που τελικά θα εκτελεστούν (ανάμεσα βέβαια στα ενεργοποιημένα). Αν θέλουμε να ελέγξουμε συγκεκριμένες ακολουθίες καταστάσεων, μπορούμε να επιλέξουμε τη μετάβαση που θα εκτελεστεί στο επόμενο βήμα της βήμα-προς-βήμα προσομοίωσης μέσω της εντολής **Bind** κάτω από το μενού **Sim**:



Επιλέγοντας **All** θα εμφανιστούν όλες οι ενεργές συσχετίσεις. Για να εκτελεστεί πρώτη μια συγκεκριμένη μετάβαση, αρχικά επιλέγεται, έπειτα με τη βοήθεια του πλήκτρου **Up** μεταφέρεται στην πρώτη θέση και στη συνέχεια με το πλήκτρο **Occur** εκτελείται.

Α'.1. ΕΓΚΑΤΑΣΤΑΣΗ ΚΑΙ ΡΥΘΜΙΣΕΙΣ ΤΟΥ ΕΡΓΑΛΕΙΟΥ ΔΕΣΙΓΝ/ΨΠΝ105



Α'.1.10 Επαναφορά στην αρχική κατάσταση

Μετά από την επισκόπηση της συμπεριφοράς του μοντέλου μέσω της προσομοίωσης, το δίκτυο μπορεί να επανέλθει στην αρχική κατάσταση (σήμανση) μέσω της επιλογής **Initial state** του μενού **Sim**. Η επαναφορά αυτή είναι ιδιαίτερα σημαντική όταν γίνεται η ανάλυση του χώρου καταστάσεων και η εξαγωγή ολόκληρου του γράφου καλυψιμότητας.

Α'.1.11 Κατάσταση εξαγωγής γράφου καλυψιμότητας

Για τη μετάβαση στην κατάσταση της εξαγωγής γράφου καλυψιμότητας (Occurrence Graph) επιλέγουμε **Enter Occ Graph** από το μενού **File**.

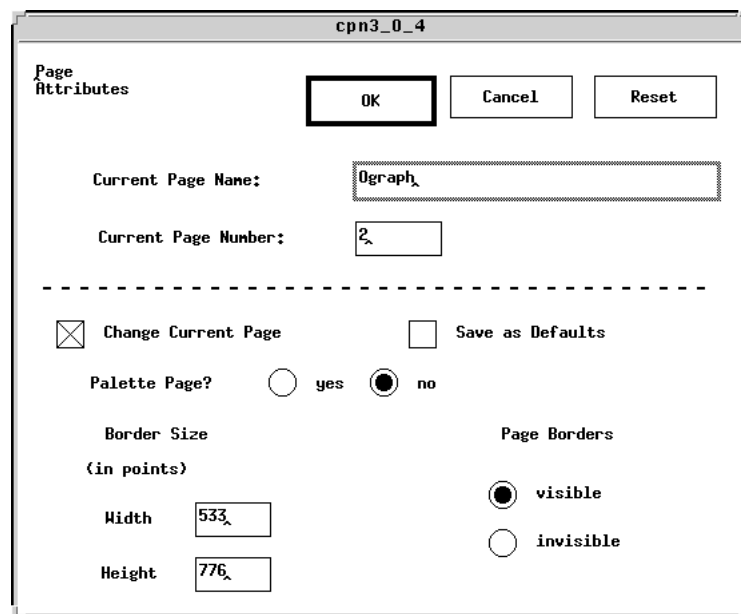
Υπολογισμός γράφου καλυψιμότητας

Από το μενού **Occ** επιλέγουμε **Calculate Occ Graph**. Στη συνέχεια πάλι από το μενού **Occ** επιλέγουμε **Show Statistics**.

Σχεδίαση γράφου καλυψιμότητας

Αρχικά δημιουργούμε μια νέα σελίδα μέσω της επιλογής **New Page** του μενού **Page** όπου θα δημιουργηθεί ο γράφος. Αφού εμφανιστεί η νέα σελίδα, επιλέγουμε

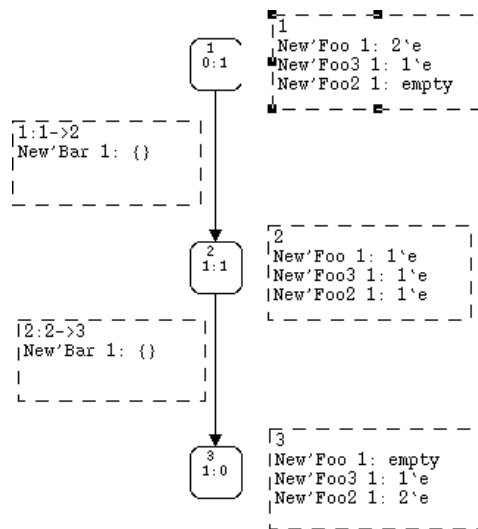
Set Attributes στο μενού **Set**. Στο παράθυρο που εμφανίζεται μπορούμε να αλλάξουμε το όνομα της νέας σελίδας ή άλλα χαρακτηριστικά όπως φαίνεται και στο ακόλουθο σχήμα:



Αφού κάνουμε το παράθυρο της νέας σελίδας ενεργό (με κλικ πάνω του), από το μενού **Occ** επιλέγουμε **Display Node**. Στο παράθυρο που εμφανίζεται μας ζητάται να εισάγουμε τον αριθμό του κόμβου που θα εμφανιστεί (ο κόμβος με αριθμό 1 αντιστοιχεί στην αρχική σήμανση). Επιλέγουμε **OK**. Στη συνέχεια με διαδοχικές εκτελέσεις της διαδικασίας: μενού **Occ** - επιλογή **Display Successors** μπορούν να εμφανιστούν οι διαδοχικές καταστάσεις.

Με διπλό κλικ στον κάθε κόμβο του γράφου αλλά και στα τόξα που τους συνδέουν εμφανίζονται τμήματα κειμένου. Το τμήμα κειμένου κάθε τόξου στο σημείο αυτό αντιστοιχεί στη μετάβαση που εκτελέστηκε ενώ το κείμενο δίπλα σε κάθε κόμβο αναπαριστά τη σήμανση του δικτύου σε κάθε βήμα.

Α'.1. ΕΓΚΑΤΑΣΤΑΣΗ ΚΑΙ ΡΥΘΜΙΣΕΙΣ ΤΟΥ ΕΡΓΑΛΕΙΟΥ ΔΕΣΙΓΝ/ΨΠΝ107



Αποθήκευση αναφοράς γράφου καλυψιμότητας

Επιλέγοντας **Save Report** από το μενού **Occ** εμφανίζεται το παράθυρο όπου μπορεί κανείς να επιλέξει τον τύπο των στατιστικών και των υπόλοιπων στοιχείων που θα εμφανίζονται στην τελική αναφορά. Η αναφορά έχει μορφή παρόμοια με την ακόλουθη:

Α'.1.12 Αποθήκευση του μοντέλου CPN

Με την επιλογή **Save As** του μενού **File** δίνεται η δυνατότητα να αποθηκευτεί το διάγραμμα σε ένα αρχείο με όνομα που μπορεί να καθορίσει ο χρήστης.

Για να αποθηκευτεί ολόκληρη η τρέχουσα κατάσταση του μοντέλου, από το μενού **File** επιλέγουμε **Save State**. (Οι επιλογές κατά την προσομοίωση ή την εξαγωγή του δέντρου καλυψιμότητας **Save** και **Save As** είναι απενεργοποιημένες). Με τον τρόπο αυτό αποθηκεύονται πληροφορίες τόσο για την προσομοίωση όσο και τον γράφο καλυψιμότητας. Έτσι δεν είναι απαραίτητο να ξεκινά ο χρήστης από την αρχή τις -συνήθως- χρονοβόρες διαδικασίες υπολογισμού του γράφου καλυψιμότητας

Κάθε μοντέλο είναι συσχετισμένο με τρία αρχεία π.χ. φoo, φoo.ΔB, φoo.ML

Το αρχείο με την κατάληξη **.ML** καταλαμβάνει αρκετό χώρο στο δίσκο μιας

```

Statistics
----- Occurrence
Graph
Nodes: 3
Arcs: 2Secs: 0
Status: Full

Boundedness Properties
----- Best
Integers Bounds Upper\tab Lower
New'Foo 1 2\tab 0
New'Foo2 1\tab 2\tab 0
New'Foo3 1\tab 1\tab 1

Best Upper Multi-set Bounds New'Foo 1 2\{e New'Foo2 1 2\{e New'Foo3 1 1\{e

Best Lower Multi-set Bounds New'Foo 1 empty New'Foo2 1 empty New'Foo3 1 1\{e

Liveness Properties
-----
Dead Markings: [3]
Dead Transitions Instances: None

```

Σχήμα Α΄.1: Παράδειγμα αναφοράς γράφου καλυψιμότητας

και σ'αυτό αποθηκεύονται τα στοιχεία των προσομοιώσεων ή ο γράφος καλυψιμότητας. Για λόγους εξοικονόμησης χώρου μπορεί να διαγραφεί χωρίς να επηρεάσει τα υπόλοιπα στοιχεία του μοντέλου (που αποθηκεύονται στα άλλα δύο αρχεία).

Μετατροπή των μοντέλων σε άλλα format για χρήση από άλλα εργαλεία

Η τάση που επικρατεί στο χώρο των δικτύων Πετρι και των σχετικών εργαλείων είναι να υιοθετηθεί ένα κοινό format με στόχο την υποστήριξη όλων των διαφορετικών

A'.1. ΕΓΚΑΤΑΣΤΑΣΗ ΚΑΙ ΡΥΘΜΙΣΕΙΣ ΤΟΥ ΕΡΓΑΛΕΙΟΥ ΔΕΣΙΓΝ/ΎΠΝ109

τύπων Petri αλλά κυρίως και των νέων τύπων δικτύων που μπορεί να εισαχθούν στην πορεία ώστε να είναι εύκολος ο τρόπος συνεργασίας των διαφόρων εργαλείων.

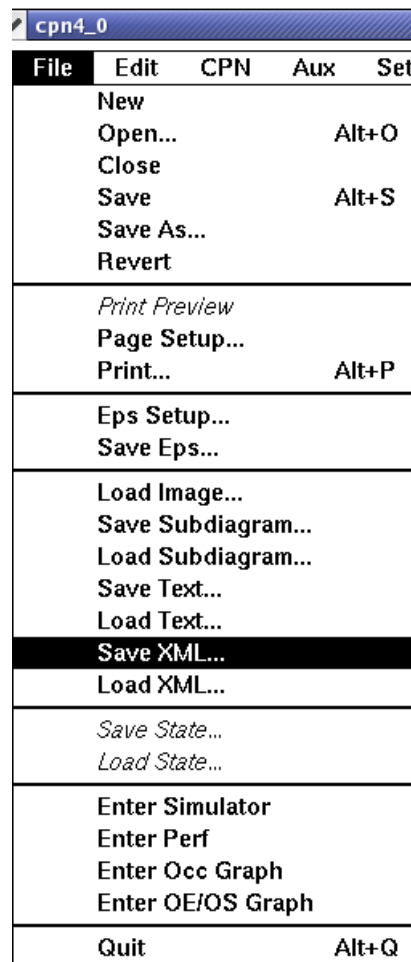
Πρόσφατα προτάθηκε η χρήση μιας γλώσσας περιγραφής της δομής ενός δικτύου Petri που είναι βασισμένη στην XML. Η γλώσσα περιγραφής PNML ([2]) επιτρέπει τον ορισμό τύπων δικτύων Petri ενώ ακόμη προτείνει τρόπους για να υπάρχει ενιαία αντιμετώπιση και των αποτελεσμάτων της ανάλυσης των δικτύων (όπως π.χ. των γράφων καλυψιμότητας).

Τόσο το εργαλείο Design/CPN όσο και το CPN Tools υποστηρίζουν ένα format αποθήκευσης βασισμένο σε XML αν και το πρώτο όχι εγγενώς αλλά μέσω ενός εξωτερικού μετατροπέα.

Τα μοντέλα που σχεδιάζονται με τη βοήθεια του εργαλείου Design/CPN μπορούν να μεταφερθούν και στο εργαλείο CPN Tools ακολουθώντας τα ακόλουθα βήματα:

1. Αποθήκευση του μοντέλου σε XML format

Αυτό γίνεται μέσω της επιλογής **Save XML** κάτω από το μενού **File**. Απαιτείται αυξημένη προσοχή μιας και μόνο οι σελίδες που είναι επιλεγμένες εκείνη τη στιγμή θα αποθηκευτούν. Η σελίδα της ιεραρχίας μπορεί να χρησιμοποιηθεί ώστε να επιλεγούν με εύκολο τρόπο περισσότερες από μια σελίδες προς αποθήκευση.



2. Μετατροπή του XML αρχείου στο format που υποστηρίζεται από το CPN Tools

Η εφαρμογή που αναλαμβάνει τη μετατροπή συνοδεύει τη διανομή της εφαρμογής CPNTools και μπορεί να εκτελεστεί από το μενού Start των Windows π.χ. Start Menu → *Programs* → *CPNTools* → *Utils* → *converter*.

Από το παράθυρο του διαλόγου που εμφανίζεται επιλέγουμε το αρχείο που θέλουμε να μετατραπεί.

3. Φόρτωση του νέου XML αρχείου από το εργαλείο CPN Tools

Πλέον το αρχείο που παράχθηκε από την παραπάνω διαδικασία μπορεί να χρησιμοποιηθεί ως είσοδος στο εργαλείο CPN Tools και να συνεχισθεί εκεί η περαιτέρω επεξεργασία και ανάλυση του μοντέλου.

Bibliography

- [1] Tomaz C. Barros and Angelo Perkusich. Design of supervisors for agile manufacturing systems using colored petri nets.
- [2] J. Billington, S. Christensen, K. van Hee, E. Kindler, O. Kummer, L. Petrucci, R. Post, C. Stehno, and M. Weber. The Petri Net Markup Language: Concepts, Technology, and Tools. In *Proceedings ICATPN*, volume 2679, pages 483–505. Springer, 2003.
- [3] University of Aarhus CPN Group. <http://www.daimi.au.dk/cpnets/intro>. Industrial Use of CP-nets.
- [4] University of Aarhus CPN Group. <http://www.daimi.au.dk/designcpn/>. Web site.
- [5] Leandro Dias da Silva and Angelo Perkusich. Formal verification of component-based software systems. In *NDDL/VVEIS*, pages 113–124, 2003.
- [6] Leandro Dias da Silva and Angelo Perkusich. A Systematic and Formal Approach to the Specification of Flexible Manufacturing Systems Reusing Coloured Petri Nets Models. In *Proceedings INCOM 04*, 2004.
- [7] Desrochers and Jaar. *Petri-net Modeling and Real-Time Controllers*. , 1995.
- [8] C. Ghezzi, D. Mandrioli, S. Morasca, and M. Pezze. A general way to put time in petri nets. In *IWSSD '89: Proceedings of the 5th international workshop on Software specification and design*, pages 60–67, New York, NY, USA, 1989. ACM Press.

- [9] Sheng-Wei Guan, Hsiao-Yeh Yu, and Jen-Shun Yang. A prioritized petri net model and its application in distributed multimedia systems. *IEEE Trans. Comput.*, 47(4):477–481, 1998.
- [10] M. R. Hansen and H. Rischel. *Introduction to Programming using SML*. Addison-Wesley, 1999.
- [11] L.E. Holloway, B. H. Krogh, and A. Giua. A Survey of Petri Net Methods for Controlled Discrete Event Systems. *Discrete Event Dynamic Systems: Theory and Applications*, 7:151–190, 1997.
- [12] K. Jensen. *Coloured Petri Nets. Basic Concepts, Analysis Methods and Practical Use*. Springer-Verlag, 1992.
- [13] Kurt Jensen. An Introduction to the Practical Use of Coloured Petri Nets. Tutorial, Department of Computer Science, University of Aarhus.
- [14] Olaf Kummer, Daniel Moldt, and Frank Wienberh. A framework for interacting design/cpn and java processes. In *ICATPN*, pages 484–493, 2004.
- [15] R. Milner, R. Harper, and M. Tofte. *The Definition of Standard ML*. MIT Press, 1990.
- [16] K.H. Mortensen. Tool Demonstrations. Tutorial, Department of Computer Science, University of Aarhus, June 2000.
- [17] Carl Adam Petri. Kommunikation mit automaten. Dissertation, 1962.
- [18] J.M. Proth and X. Xie. *Petri Nets: A Tool for Design and Management of Manufacturing Systems*. Wiley, 1996.
- [19] G.J. Tsinarakis, N.C. Tsourveloudis, and K.P. Valavanis. Modular Petri Net based modeling, analysis, synthesis and performance evaluation of random topology dedicated production systems. *Journal of Intelligent Manufacturing*, 16:67–92, 2005.
- [20] I. Vanderveesten, W. van der Alst, and H.A. Reijers. Modelling a product based workflow system in CPN Tools. In *Proceedings of the 6th Workshop*

and Tutorial on Practical Use of Coloured Petri Nets and the CPN Tools, pages 99–118, October 2005.

- [21] M. Zhou. *Petri Nets in Flexible and Agile Automation*. Kluwer Academic Publishers, 1995.
- [22] Włodzimierz M. Zuberek. Performance evaluation of concurrent systems using timed petri nets. In *CSC '85: Proceedings of the 1985 ACM thirteenth annual conference on Computer Science*, pages 326–329, New York, NY, USA, 1985. ACM Press.