

# **ΑΝΑΛΥΣΗ ΑΠΟΔΟΣΗΣ ΣΥΣΤΗΜΑΤΩΝ ΡΟΗΣ ΕΡΓΑΣΙΩΝ (WORKFLOWS)**

**Παναγιώτης Τουρλάκης**

Διπλωματική Εργασία

Επιβλέπων Καθηγητής: Παναγιώτης Κατσαρός

Θεσσαλονίκη 2006

Τμήμα Πληροφορικής

Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης

## ΠΕΡΙΕΧΟΜΕΝΑ

### ΕΝΟΤΗΤΑ 1<sup>η</sup> – ΕΙΣΑΓΩΓΗ

1.1 – Η ροή εργασίας.....	1
1.2 – Συστατικά μέρη των ροών εργασίας.....	1
1.3 – Συστήματα ροών εργασίας.....	2
1.4 – Εφαρμογές ροών εργασίας.....	3

### ΕΝΟΤΗΤΑ 2<sup>η</sup> – ΔΙΑΧΕΙΡΙΣΗ ΡΟΗΣ ΕΡΓΑΣΙΑΣ

2.1 – Εισαγωγή.....	4
2.2 – Διαχείριση της ροής εργασίας και Συστήματα διαχείρισης.....	5
2.3 – Συστήματα ροής εργασίας.....	10

### ΕΝΟΤΗΤΑ 3<sup>η</sup> – ΜΟΝΤΕΛΟΠΟΙΗΣΗ ΡΟΗΣ ΕΡΓΑΣΙΑΣ ΜΕ ΧΡΩΜΑΤΙΣΜΕΝΑ ΔΙΚΤΥΑ PETRI

3.1 – Εισαγωγή.....	13
3.2 – Το εργαλείο CPN-Tools.....	15
3.3 – Μοντέλα ροής εργασιών βασισμένα σε χρωματισμένα δίκτυα Petri...	18
3.3.1 – Μοντελοποίηση των Αρχών χειρισμού περιπτώσεων.....	18
3.3.2 – Μοντελοποίηση μηχανισμού κατανομής εργασίας.....	22
3.3.3 – Μοντελοποίηση συστήματος ροής εργασίας που βασίζεται σε προϊόντα.....	28

### ΕΝΟΤΗΤΑ 4<sup>η</sup> – ΑΝΑΛΥΣΗ ΑΠΟΔΟΣΗΣ ΜΟΝΤΕΛΩΝ ΡΟΗΣ ΕΡΓΑΣΙΑΣ ΜΕ ΤΗ ΧΡΗΣΗ ΧΡΩΜΑΤΙΣΜΕΝΩΝ ΔΙΚΤΥΩΝ PETRI

4.1 – Εισαγωγή.....	32
4.2 – Τεχνικές ανάλυσης απόδοσης.....	32
4.3 – Μέτρα απόδοσης.....	34

4.4 – Εργαλεία παρακολούθησης χρωματισμένων δικτύων Petri.....	35
4.5 – Δομή των εργαλείων παρακολούθησης.....	36
4.5.1 – Εργαλεία παρακολούθησης σημείων διακοπής.....	36
4.5.2 – Εργαλεία παρακολούθησης συλλογής δεδομένων.....	37
4.5.3 – Εργαλεία παρακολούθησης καταγραφής σε αρχείο.....	38
4.6 – Παράδειγμα υλοποίησης εργαλείων παρακολούθησης με το CPN-Tools.....	39
4.6.1 – Υλοποίηση εργαλείου παρακολούθησης σημείου διακοπής.	42
4.6.2 – Υλοποίηση εργαλείου παρακολούθησης συλλογής δεδομένων.....	44
4.6.3 – Υλοποίηση εργαλείου παρακολούθησης καταγραφής σε αρχείο.....	47
4.7 – Αρχεία εξόδου της ανάλυσης απόδοσης.....	49

**ΕΝΟΤΗΤΑ 5<sup>η</sup> – ΑΝΑΛΥΣΗ ΑΠΟΔΟΣΗΣ ΔΙΑΔΙΚΑΣΙΑΣ ΡΟΗΣ ΕΡΓΑΣΙΩΝ ΜΕ ΠΕΡΙΟΡΙΣΜΕΝΟΥΣ ΠΟΡΟΥΣ ΧΡΗΣΙΜΟΠΟΙΩΝΤΑΣ ΧΡΩΜΑΤΙΣΜΕΝΑ ΔΙΚΤΥΑ PETRI**

5.1 – Εισαγωγή.....	53
5.2 – Παρουσίαση του μοντέλου.....	53
5.3 – Ανάλυση της απόδοσης του μοντέλου με εργαλεία παρακολούθησης.....	59
5.3.1 – Μέσος αριθμός απασχολημένων πόρων κατά την προσομοίωση.....	59
5.3.2 – Μέσος χρόνος διεκπεραίωσης περιπτώσεων.....	61
5.3.3 – Μέση χρησιμοποίηση πόρων.....	63
5.4 – Στατιστικά αποτελέσματα προσομοίωσης.....	65

**ΕΝΟΤΗΤΑ 6<sup>η</sup> – ΣΥΜΠΕΡΑΣΜΑΤΑ.....**

**ΒΙΒΛΙΟΓΡΑΦΙΑ.....**

73

# ΕΝΟΤΗΤΑ 1<sup>η</sup>

## ΕΙΣΑΓΩΓΗ

### 1.1 – Η ροή εργασίας

Η ροή εργασίας (workflow) γενικά εκφράζει τη μετακίνηση εγγράφων και εντολών κατά τη διάρκεια μιας εργασίας. Αποτελεί τη λειτουργική άποψη μιας διαδικασίας, εκφράζοντας το πώς είναι δομημένες οι εντολές και οι διεργασίες, ποιοι τις εκτελούν, πώς και με ποια σειρά σχετίζονται μεταξύ τους, πώς συγχρονίζονται, πώς ρέουν οι πληροφορίες που υποστηρίζουν τις διεργασίες, ακόμα και πώς αυτές παρακολουθούνται.

Η πλειοψηφία των πεδίων εφαρμογής που θεωρούνται σήμερα για τις ροές εργασίας ενσωματώνουν σε αυτές και τη διάσταση του χρόνου. Αυτό σημαίνει ότι οι ροές εργασίας αποκτούν την επιπρόσθετη ιδιότητα να εκτιμούν την απόδοση των εργασιών στις οποίες εφαρμόζονται. Με άλλα λόγια, η απόδοση μιας εργασίας θεωρείται ως ξεχωριστό μέτρο.

Τα προβλήματα που εκφράζονται με ροές εργασίας μπορούν να μοντελοποιηθούν και να αναλυθούν χρησιμοποιώντας εργαλεία αναπαράστασης με γράφους, όπως τα δίκτυα Petri.

### 1.2 – Συστατικά μέρη των ροών εργασίας

Οι ροές εργασίας μπορούν, θεωρητικά πάντα, να αναλυθούν σε τρία δομικά μέρη :

- Περιγραφή εισόδου
- Κανόνες μετασχηματισμού / Αλγόριθμοι
- Περιγραφή εξόδου

Τα δομικά αυτά στοιχεία μπορούν να συνδυαστούν αν και μόνο αν:

η έξοδος ενός προηγούμενου στοιχείου / συνόλου στοιχείων είναι ίση με την απαραίτητη είσοδο για το επόμενο στοιχείο / σύνολο στοιχείων.

Κάθε στοιχείο μέσα σε μια ροή εργασίας χαρακτηρίζεται από τις εισόδους και τις εξόδους του, οι οποίες εκφράζονται από διάφορους τύπους δεδομένων και τα σημασιολογικά τους χαρακτηριστικά. Οι κανόνες μετασχηματισμού ή οι αλγόριθμοι χρησιμοποιούνται μόνο όταν κρίνεται απαραίτητο να μετασχηματιστεί ένας τύπος εισόδου σε κάποιον άλλο τύπο εξόδου με διαφορετικά χαρακτηριστικά.

### **1.3 – Συστήματα ροών εργασίας**

Τα συστήματα ροών εργασίας χρησιμοποιούνται κατά βάση από οργανισμούς κι επιχειρήσεις. Είναι συστήματα που προσφέρουν τη δυνατότητα σε κάποιον οργανισμό να διαχειρίζεται τη ροή της εργασίας μέσα στα πλαίσια λειτουργίας του. Πιο συγκεκριμένα, τα συστήματα αυτά χρησιμεύουν στον καθορισμό, την εκτέλεση, την παρακολούθηση και την ανακατεύθυνση των περιπτώσεων εργασίας μέσα σε ένα εργασιακό περιβάλλον.

Τα συστήματα ροών εργασίας περιέχουν δυο βασικά στοιχεία :

- Τη μονάδα μοντελοποίησης της ροής εργασίας
- Τη μονάδα εκτέλεσης της ροής εργασίας

Η μονάδα μοντελοποίησης αναφέρεται στους διαχειριστές του συστήματος. Τους επιτρέπει να ορίζουν δραστηριότητες και διεργασίες, να τις αναλύουν και να τις προσομοιώνουν, καθώς και να τις αναθέτουν σε ανθρώπινο δυναμικό.

Η μονάδα εκτέλεσης παρέχει στον τελικό χρήστη του συστήματος μια διεπαφή εκτέλεσης και προσομοίωσης, ενώ περιέχει και μια μηχανή ροής εργασίας η οποία βοηθά ή/και εκτελεί την καθοδήγηση των διεργασιών και δραστηριοτήτων μέσα στο περιβάλλον.

#### **1.4 – Εφαρμογές ροών εργασίας**

Μια εφαρμογή αποκαλείται εφαρμογή ροής εργασίας (workflow application) όταν διάφοροι άνθρωποι, ή και εφαρμογές, συμμετέχουν στην επεξεργασία δεδομένων της εφαρμογής, ώστε να ολοκληρωθεί μια περίπτωση της διεργασίας που περιγράφει η εφαρμογή. Με άλλα λόγια, κάποια περίπτωση εργασίας είναι απαραίτητο να περάσει από διάφορα στάδια επεξεργασίας, είτε από μια εφαρμογή είτε από ανθρώπινο παράγοντα, προκειμένου να θεωρηθεί ως ένα ολοκληρωμένο στιγμιότυπο της συνολικής διεργασίας.

Μια διεργασία πάντα εμπεριέχει διαρκείς αλλαγές κι ενημερώσεις των στοιχείων που διακινούνται στα πλαίσιά της. Παρ' όλα αυτά, είναι ιδιαίτερα εύκολο να εισαχθεί κάποιος παράγοντας στη λειτουργία μιας εφαρμογής, αφού δε χρειάζεται να γίνουν σημαντικές αλλαγές στο μοντέλο.

## **ΕΝΟΤΗΤΑ 2<sup>η</sup>**

### **ΔΙΑΧΕΙΡΙΣΗ ΡΟΗΣ ΕΡΓΑΣΙΑΣ (WORKFLOW MANAGEMENT)**

#### **2.1 – Εισαγωγή**

Οι σημερινές δραστηριότητες των επιχειρήσεων δημιουργούν την ανάγκη για όσο το δυνατό μεγαλύτερη υποστήριξη από τα πληροφοριακά τους συστήματα. Πλέον δεν είναι αρκετό να επικεντρώνονται απλά σε μεμονωμένες εργασίες. Είναι απαραίτητο τα πληροφοριακά συστήματα να μπορούν να παρακολουθούν, να ελέγχουν και να υποστηρίζουν τις λογιστικές απόψεις μιας επιχειρησιακής διαδικασίας. Η ανάγκη αυτή για τεχνικές, αρχές και εργαλεία υποστήριξης της ροής εργασίας οδήγησε στην καθιέρωση της έννοιας «διαχείριση ροής εργασίας».

Μέχρι πριν ελάχιστα χρόνια δεν υπήρχαν εργαλεία ειδικά για την υποστήριξη της διαχείρισης της ροής εργασίας. Αυτό είχε ως αποτέλεσμα κάποια μέρη της επιχειρησιακής διαδικασίας να προγραμματίζονται ως αυτούσια μέρη μέσα στις εφαρμογές. Για παράδειγμα, μια εφαρμογή που υποστηρίζει την εργασία Α καλεί μια άλλη εφαρμογή που υποστηρίζει την εργασία Β. Αυτό σημαίνει ότι κάθε εφαρμογή γνωρίζει για την ύπαρξη της άλλης, πράγμα μη επιθυμητό αφού κάθε φορά που μεταβάλλεται η επιχειρησιακή διαδικασία είναι ανάγκη να τροποποιηθούν και οι εφαρμογές. Επειδή αυτό μπορεί να συμβαίνει στην πλειοψηφία των εφαρμογών της ροής εργασίας, είναι τελικά δύσκολος ο έλεγχος και η παρακολούθηση της όλης διαδικασίας. Για το λόγο αυτό αναγνωρίστηκε η ανάγκη για συστήματα διαχείρισης της ροής εργασίας (workflow management systems).

## 2.2 – Διαχείριση της ροής εργασίας και Συστήματα διαχείρισης

- Ορισμός 2.1: Ένα σύστημα διαχείρισης ροής εργασίας είναι ένα πακέτο λογισμικού που επιτρέπει τον ορισμό, την εκτέλεση, την καταχώρηση και τον έλεγχο ροών εργασίας.

Ο κύριος στόχος της διαχείρισης της ροής εργασίας είναι να εξασφαλιστεί ότι η κατάλληλη δραστηριότητα εκτελείται από το σωστό άτομο στο σωστό χρόνο. Παρόλο που είναι πιθανό η διαχείριση της ροής εργασίας να γίνει χωρίς να χρησιμοποιηθεί ένα σύστημα διαχείρισης, οι περισσότεροι συγχέουν τη διαχείριση ροής εργασίας με τα συστήματα διαχείρισης ροής εργασίας. Από μια άποψη, αυτό δε θα έπρεπε να συμβαίνει διότι περιορίζει τα πεδία εφαρμογής της διαχείρισης της ροής εργασίας. Από την άλλη, τα συστήματα διαχείρισης της ροής εργασίας δίνουν μια χειροπιαστή μορφή στις ουσιώδεις αρχές, τεχνικές και μεθόδους της διαχείρισης ροής εργασίας.

Η ροή εργασίας είναι βασισμένη σε περιπτώσεις (case-based), δηλαδή κάθε μέρος εργασίας εκτελείται για κάποια συγκεκριμένη περίπτωση. Η διαχείριση της ροής εργασίας αποσκοπεί στον αποτελεσματικότερο χειρισμό των περιπτώσεων αυτών. Ο χειρισμός των περιπτώσεων επιτυγχάνεται εκτελώντας εργασίες, ή καθήκοντα (tasks) με συγκεκριμένη σειρά. Ο ορισμός της διαδικασίας ροής εργασίας (workflow process definition) καθορίζει ποια εργασία πρέπει να εκτελεστεί και με ποια σειρά. Εφόσον οι εργασίες τίθενται με συγκεκριμένη σειρά, είναι χρήσιμο να ορίζονται συνθήκες (conditions) που αντιστοιχούν στις αιτιολογικές σχέσεις μεταξύ τους. Μια συνθήκη μπορεί να είναι αληθής ή ψευδής. Κάθε εργασία έχει προ-συνθήκες (preconditions) και μετα-συνθήκες (postconditions). Οι προ-συνθήκες εξετάζονται πριν την εκτέλεση της εργασίας, ενώ οι μετα-συνθήκες εξετάζονται μετά την εκτέλεσή της. Μπορούμε να χειριστούμε πολλαπλές περιπτώσεις ακολουθώντας τον ίδιο ορισμό διαδικασίας ροής εργασίας.

- Ορισμός 2.2: Μια εργασία που πρέπει να εκτελεστεί για μια συγκεκριμένη περίπτωση αποτελεί ένα αντικείμενο εργασίας (work item).



Η πλειοψηφία των αντικειμένων εργασίας εκτελούνται από κάποιον πόρο (resource).

- Ορισμός 2.3: Πόρος είναι ένας άνθρωπος ή ένα μηχάνημα που μπορεί να εκτελέσει ένα αντικείμενο εργασίας.

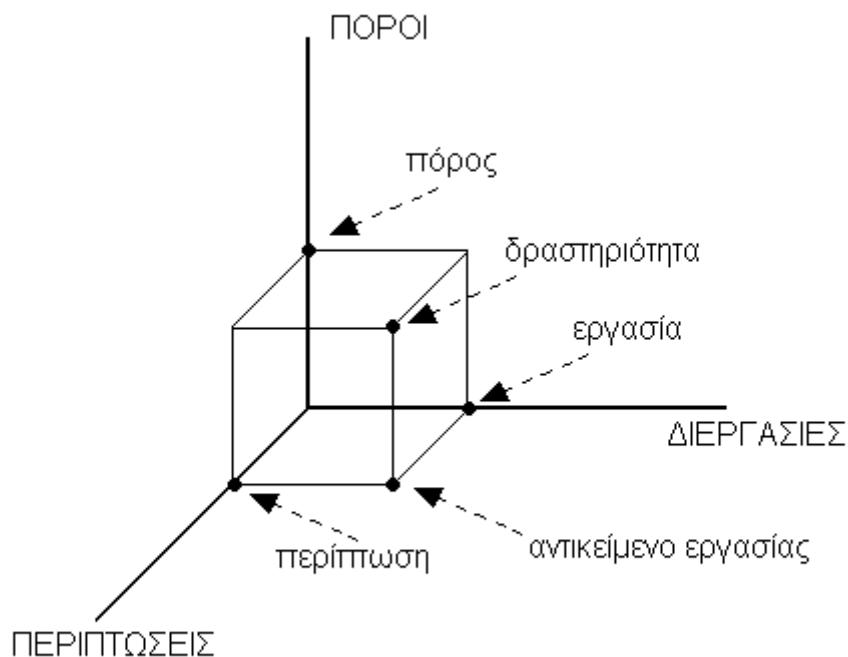
Στις περισσότερες επιχειρήσεις οι πόροι είναι κατά κύριο λόγο άνθρωποι. Οι πόροι χειρίζονται συγκεκριμένα αντικείμενα εργασίας. Για να διευκολυνθεί η ανάθεση αντικειμένων εργασίας σε πόρους, οι πόροι ομαδοποιούνται σε τάξεις (resource classes).

- Ορισμός 2.4: Τάξη πόρων είναι μια ομάδα πόρων με παρόμοια χαρακτηριστικά.

Μπορούν να ανήκουν πολλαπλοί πόροι στην ίδια τάξη και ένας πόρος μπορεί να είναι μέλος πολλαπλών τάξεων.

- Ορισμός 2.5: Εάν μια τάξη βασίζεται στις δυνατότητες των μελών της, αποκαλείται ρόλος (role).
- Ορισμός 2.6: Εάν μια τάξη βασίζεται στη δομή του οργανισμού, τότε αποκαλείται οργανωτική μονάδα (organizational unit).
- Ορισμός 2.7: Ένα αντικείμενο εργασίας που εκτελείται από ένα συγκεκριμένο πόρο αποκαλείται δραστηριότητα (activity).

Αν φανταστούμε τη σχηματική αναπαράσταση μιας ροής εργασίας, παρατηρούμε περιπτώσεις, αντικείμενα εργασίας και δραστηριότητες. Τα αντικείμενα εργασίας συνδέουν τις περιπτώσεις και τις εργασίες, ενώ οι δραστηριότητες συνδέουν τις περιπτώσεις, τις εργασίες και τους πόρους.



Σχήμα 1

Στο Σχήμα 1 φαίνεται η σχηματική αναπαράσταση μιας ροής εργασίας, που αποτελείται από τρεις διαστάσεις:

1. Διάσταση περιπτώσεων: Υποδεικνύει το γεγονός ότι ο χειρισμός των περιπτώσεων γίνεται ξεχωριστά για κάθε περίπτωση. Από την άποψη της ροής εργασίας, οι περιπτώσεις δεν επηρεάζουν άμεσα η μία την άλλη. Οι περιπτώσεις επηρεάζονται έμμεσα μεταξύ τους λόγω του διαμοιρασμού των πόρων και των δεδομένων.
2. Διάσταση διεργασιών: Στη διάσταση αυτή καθορίζεται η διαδικασία της ροής εργασίας, δηλαδή οι εργασίες και η δρομολόγηση μέσω των εργασιών αυτών.
3. Διάσταση πόρων: Οι πόροι ομαδοποιούνται σε ρόλους και οργανωτικές μονάδες.

Είναι εύκολο να απεικονίσουμε μια ροή εργασίας σαν ένα σύνολο σημείων επάνω στην τρισδιάστατη αναπαράσταση που φαίνεται στο Σχήμα 1. Κάθε σημείο αναπαριστά είτε ένα αντικείμενο εργασίας (συνδυασμός περίπτωσης και εργασίας), είτε μια δραστηριότητα (συνδυασμός περίπτωσης, εργασίας και πόρου). Η αναπαράσταση αυτή ουσιαστικά δείχνει ότι η

διαχείριση της ροής εργασίας είναι ο συνδυασμός κρίκος μεταξύ των περιπτώσεων, των εργασιών και του οργανισμού.

Ο ορισμός της διαδικασίας ροής εργασίας καθορίζει τον τρόπο με τον οποίο οι περιπτώσεις δρομολογούνται μέσω των εργασιών που πρέπει να εκτελεστούν. Το Σχήμα 2 δείχνει τους διαφορετικούς τύπους δρομολόγησης που έχουν καθιερωθεί στον τομέα της διαχείρισης της ροής εργασίας.

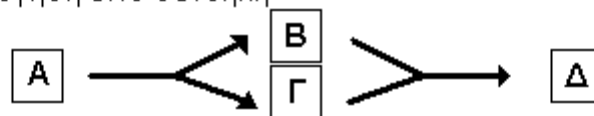
(α) Σειριακή δρομολόγηση



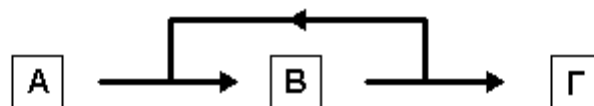
(β) Παράλληλη δρομολόγηση



(γ) Δρομολόγηση υπό συνθήκη



(δ) Επαναληπτική δρομολόγηση



Σχήμα 2

Στο Σχήμα 2 ορίζονται τέσσερις τύποι δρομολόγησης:

1. Σειριακή δρομολόγηση (sequential routing): Οι εργασίες εκτελούνται σειριακά εάν η εκτέλεση μιας εργασίας ακολουθείται από την επόμενη εργασία. Στο Σχήμα 2(α), η εργασία B εκτελείται αφού ολοκληρωθεί η εργασία A και προτού ξεκινήσει η εργασία Γ.
2. Παράλληλη δρομολόγηση (parallel routing): Στο Σχήμα 2(β), οι εργασίες B και Γ εκτελούνται παράλληλα. Αυτό σημαίνει ότι η B και η Γ

εκτελούνται είτε ταυτόχρονα είτε με οποιαδήποτε σειρά. Για να μοντελοποιηθεί η παράλληλη δρομολόγηση, χρησιμοποιούνται δύο δομικά τμήματα:

- Το διαχωριστικό «ΚΑΙ» (AND-split): Στο Σχήμα 2(β), το διαχωριστικό ΚΑΙ επιτρέπει στις εργασίες Β και Γ να εκτελεστούν αφού ολοκληρωθεί η εργασία Α.
  - Το «ΚΑΙ» ένωσης (AND-join): Συγχρονίζει παράλληλες ροές. Στο Σχήμα 2(β), η εργασία Δ μπορεί να εκτελεστεί αφού ολοκληρωθούν οι εργασίες Β και Γ.
3. Δρομολόγηση υπό συνθήκη (conditional routing): Στο Σχήμα 2(γ), εκτελείται είτε η εργασία Β είτε η εργασία Γ (αποκλειστικό OR). Για να μοντελοποιηθεί η επιλογή μεταξύ δύο ή περισσότερων εναλλακτικών εργασιών χρησιμοποιούνται δύο δομικά τμήματα:
- Το διαχωριστικό «Ή» (OR-split): Στο Σχήμα 2(γ), εάν εκτελεστεί η εργασία Α, γίνεται επιλογή μεταξύ των εργασιών Β και Γ.
  - Το «Ή» ένωσης (OR-join): Στο Σχήμα 2(γ), η εργασία Δ μπορεί να ξεκινήσει εφόσον ολοκληρωθεί η εργασία Β ή η εργασία Γ.
4. Επαναληπτική δρομολόγηση (iterative routing): Μερικές φορές είναι απαραίτητο να εκτελεστεί κάποια εργασία πολλαπλές φορές. Στο Σχήμα 2(δ), η εργασία Β εκτελείται μία ή περισσότερες φορές.

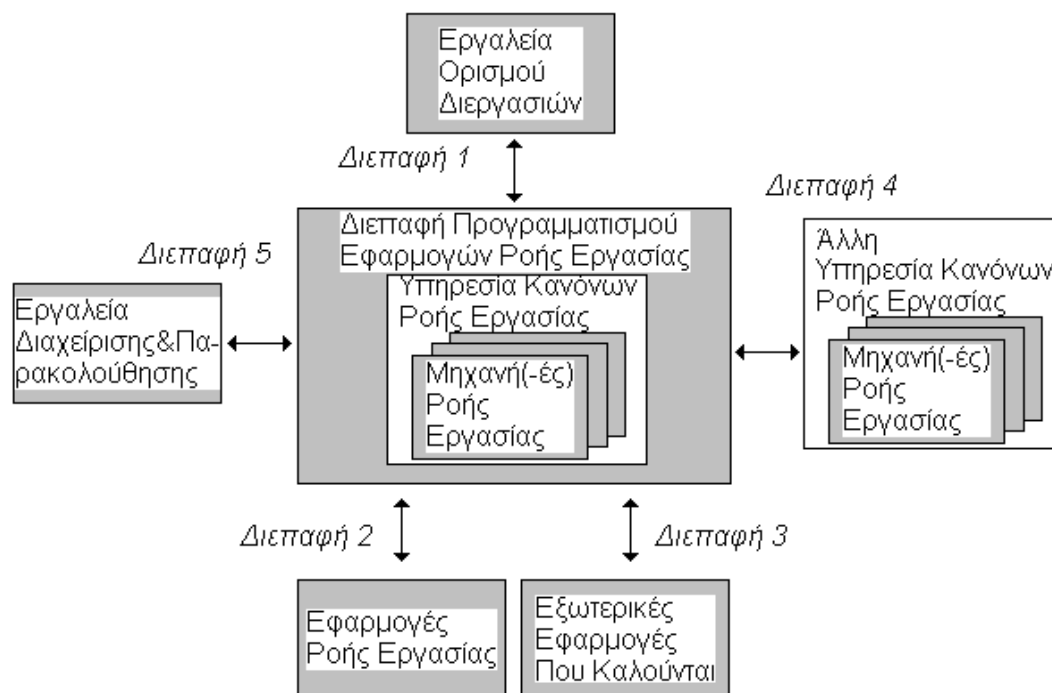
Πρέπει να γίνει ξεκάθαρο ότι κάθε επιχειρησιακή διαδικασία δεν είναι απαραίτητα και διαδικασία ροής εργασίας. Μια διαδικασία ροής εργασίας χαρακτηρίζεται από τρία στοιχεία:

1. Μια διαδικασία ροής εργασίας είναι καθοδηγούμενη από περιπτώσεις (case-driven).
2. Η διαδικασία καθαυτή θεωρείται ότι είναι θεμελιώδης.
3. Η διαδικασία μπορεί να οριστεί με συγκεκριμένο και σαφή τρόπο.

### 2.3 – Συστήματα ροής εργασίας (Workflow systems)

Είναι απαραίτητο να γίνει ένας διαχωρισμός ανάμεσα στους όρους «σύστημα διαχείρισης ροής εργασίας» και «σύστημα ροής εργασίας». Ένα σύστημα διαχείρισης της ροής εργασίας είναι ένα πακέτο λογισμικού που προορίζεται για εφαρμογή σε διάφορους οργανισμούς. Ωστόσο, εάν το σύστημα διαχείρισης δεν εγκατασταθεί, παραμετροποιηθεί και τροφοδοτηθεί με δεδομένα για τον ορισμό διεργασιών και εφαρμογών, τότε δε μπορεί να χρησιμοποιηθεί. Για το λόγο αυτό, χρησιμοποιείται ο όρος «σύστημα ροής εργασιών».

- Ορισμός 2.8: Σύστημα ροής εργασιών είναι το σύνολο του εγκατεστημένου συστήματος διαχείρισης ροής εργασιών, των δεδομένων ορισμού διεργασιών, των οργανωτικών δεδομένων, των εφαρμογών, των δεδομένων των εφαρμογών, του συστήματος διαχείρισης της βάσης δεδομένων, των αρχείων παραμετροποίησης και άλλων στοιχείων λογισμικού που περικλείουν το πραγματικό σύστημα διαχείρισης της ροής εργασιών.



Σχήμα 3

Το Σχήμα 3 παρουσιάζει μια αναπαράσταση των κύριων συστατικών μερών και διεπαφών που εμπεριέχει η αρχιτεκτονική μιας ροής εργασίας. Ο πυρήνας κάθε συστήματος ροής εργασίας είναι η υπηρεσία κανόνων ροής εργασίας (workflow enactment service).

- Ορισμός 2.9: Η υπηρεσία κανόνων ροής εργασίας παρέχει το περιβάλλον εκτέλεσης όπου ελέγχεται και εκτελείται η ροή της εργασίας.

Για λόγους διαχείρισης, η υπηρεσία κανόνων ροής εργασίας μπορεί να χρησιμοποιεί πολλαπλές μηχανές ροής εργασίας (workflow engines).

- Ορισμός 2.10: Μια μηχανή ροής εργασίας χειρίζεται συγκεκριμένα μέρη της ροής της εργασίας, ενώ διαχειρίζεται και επιλεγμένα μέρη των πόρων.
- Ορισμός 2.11: Τα εργαλεία ορισμού διεργασιών (process definition tools) χρησιμοποιούνται για να καθορίζονται και να αναλύονται ορισμοί διαδικασιών ροής εργασίας, ή / και ομαδοποιήσεις πόρων.

Τα εργαλεία αυτά χρησιμοποιούνται κατά τη διαδικασία σχεδιασμού. Ο τελικός χρήστης (end-user) επικοινωνεί με το σύστημα ροής εργασίας μέσω των εφαρμογών ροής εργασίας (workflow client applications).

- Ορισμός 2.12: Οι εφαρμογές ροής εργασίας συνιστούν μια διεπαφή μέσω της οποίας παρέχονται στο χρήστη αντικείμενα εργασίας.

Επιλέγοντας ένα αντικείμενο εργασίας, ο χρήστης μπορεί να εκτελέσει μια εργασία για μια συγκεκριμένη περίπτωση. Αν είναι απαραίτητο, η μηχανή ροής εργασίας καλεί εξωτερικές εφαρμογές μέσω της διεπαφής 3 (Σχήμα 3). Τα εργαλεία διαχείρισης και παρακολούθησης (administration and monitoring tools) χρησιμεύουν στην παρακολούθηση και τον έλεγχο της ροής εργασίας.

- Ορισμός 2.13: Τα εργαλεία διαχείρισης και παρακολούθησης καταχωρούν την πρόοδο των περιπτώσεων, όπως επίσης ανιχνεύουν συμφορήσεις στο σύστημα (bottlenecks). Επιπρόσθετα, θέτουν παραμέτρους, δεσμεύουν και αναθέτουν ανθρώπους σε εργασίες και χειρίζονται δυσλειτουργίες.

Μέσω της διεπαφής 4 (Σχήμα 3), το σύστημα ροής εργασίας έχει τη δυνατότητα να συνδέεται με άλλα συστήματα ροής εργασίας. Το σύνολο των πέντε διεπαφών που παρουσιάζονται στα συστήματα ροής εργασιών έχει καθιερωθεί κοινώς με τον όρο «διεπαφή προγραμματισμού των εφαρμογών ροής εργασίας» (workflow application programming interface).

## ΕΝΟΤΗΤΑ 3<sup>η</sup>

### ΜΟΝΤΕΛΟΠΟΙΗΣΗ ΡΟΗΣ ΕΡΓΑΣΙΑΣ ΜΕ ΧΡΩΜΑΤΙΣΜΕΝΑ ΔΙΚΤΥΑ PETRI

#### 3.1 – Εισαγωγή

Ο βασικός σκοπός ενός συστήματος διαχείρισης ροής εργασιών είναι να υποστηρίξει τον ορισμό, την καταχώρηση, την εκτέλεση και τον έλεγχο των διεργασιών. Ακριβώς επειδή οι διεργασίες είναι από τους επικρατέστερους παράγοντες κατά τη διαχείριση της ροής εργασίας, είναι ιδιαίτερα σημαντικό να χρησιμοποιείται ένα εγκατεστημένο πλαίσιο εργασίας (framework) για τη μοντελοποίηση και την ανάλυση των διαδικασιών ροής εργασίας. Τα δίκτυα Petri (Petri nets) αποτελούν μια ιδιαίτερα διαδεδομένη και εύχρηστη τεχνική μοντελοποίησης διεργασιών, αφού έχουν χρησιμοποιηθεί για τη διαχείριση και ανάλυση ποικίλων επιχειρησιακών διαδικασιών.

Η εφαρμογή των δικτύων Petri στη μοντελοποίηση ροών εργασίας θεωρείται κατάλληλη για αρκετούς λόγους:

- Γραφική αναπαράσταση:  
Τα δίκτυα Petri είναι ουσιαστικά μια γραφική γλώσσα, γι' αυτό και είναι εύκολα στην εκμάθηση και την εφαρμογή. Επιπρόσθετα, η γραφική τους φύση επιτρέπει τη διάδραση με τον τελικό χρήστη.
- Τυπική σημασιολογία:  
Μια διαδικασία ροής εργασίας που καθορίζεται με όρους δικτύων Petri έχει ξεκάθαρο και συγκεκριμένο ορισμό, αφού τα σημασιολογικά χαρακτηριστικά των δικτύων Petri είναι τυπικά ορισμένα.



- **Εκφραστικότητα:**  
Τα δίκτυα Petri υποστηρίζουν όλες τις αρχές που απαιτούνται για τη μοντελοποίηση μιας διαδικασίας ροής εργασίας. Όλοι οι τύποι δρομολόγησης εργασιών που χρησιμοποιούνται σήμερα στα συστήματα διαχείρισης ροής εργασιών μπορούν να μοντελοποιηθούν.
- **Ανάλυση:**  
Τα δίκτυα Petri διαθέτουν ποικίλες τεχνικές ανάλυσης, και ως εκ τούτου είναι ιδανικά για τη διαχείριση και μελέτη διαδικασιών ροής εργασίας. Οι τεχνικές αυτές μπορούν να αξιοποιηθούν ώστε να αποδεικνύονται ιδιότητες διαφόρων συστημάτων, καθώς και να υπολογίζονται διαφορετικά μέτρα απόδοσης. Με τον τρόπο αυτό, είναι δυνατό να αξιολογήσουμε εναλλακτικές ροές εργασίας χρησιμοποιώντας τα ίδια εργαλεία ανάλυσης που βασίζονται σε δίκτυα Petri.
- **Ανεξαρτησία από διανομείς λογισμικού:**  
Τα δίκτυα Petri προσφέρουν ένα πλαίσιο εργασίας που είναι ανεξάρτητο από εκδόσεις λογισμικού ή διανομείς. Δε βασίζονται σε κάποιο συγκεκριμένο πακέτο λογισμικού, ούτε αλλοιώνονται οι ιδιότητές τους εάν κυκλοφορήσει μια νέα έκδοση λογισμικού.

Τα χρωματισμένα δίκτυα Petri (coloured Petri nets , CP-nets) είναι μια γραφική γλώσσα για τη μοντελοποίηση των καταστάσεων ενός συστήματος, καθώς και των γεγονότων που μεταβάλλουν το σύστημα από μια κατάσταση σε κάποια άλλη. Τα χρωματισμένα δίκτυα Petri συνδυάζουν όλες τις δυνατότητες των απλών δικτύων Petri με τις γλώσσες προγραμματισμού. Η τυπική σημασιολογία των δικτύων Petri επιτρέπει τη διαχείριση και τον έλεγχο ταυτόχρονων διεργασιών σε ένα σύστημα ροής εργασίας, ενώ οι γλώσσες προγραμματισμού χρησιμοποιούνται για τον ορισμό διαφόρων τύπων δεδομένων και το χειρισμό των δεδομένων αυτών.

Οι ιδιότητες των χρωματισμένων δικτύων Petri μπορούν να επεκταθούν περαιτέρω με δύο επιπλέον χαρακτηριστικά, το χρόνο και την ιεραρχία:

- Χρωματισμένα δίκτυα Petri με χρόνο (timed coloured Petri nets):  
Κύριο χαρακτηριστικό τους είναι ότι υλοποιούν ένα γενικό ρολόι που μοντελοποιεί το πέρασμα του χρόνου. Η ιδιότητα αυτή είναι ιδιαίτερα χρήσιμη στα συστήματα διαχείρισης ροής εργασιών, αφού ο χρόνος είναι σημαντικός παράγοντας σε όλες τις επιχειρησιακές διεργασίες.
- Ιεραρχικά χρωματισμένα δίκτυα Petri (hierarchical coloured Petri nets):  
Μεγάλα και πολύπλοκα μοντέλα μπορούν να δημιουργηθούν με ιεραρχικό τρόπο, κατά τον οποίο μέρη του μοντέλου υλοποιούνται ως σελίδες (pages), οι οποίες επικοινωνούν μεταξύ τους με σαφώς ορισμένο τρόπο. Δεδομένης της πολυπλοκότητας μιας διαδικασίας ροής εργασίας, η ιδιότητα της ιεραρχίας διευκολύνει τη μοντελοποίηση ροών εργασίας και την κατανόηση των μοντέλων.

### 3.2 – Το εργαλείο CPN-Tools

Το εργαλείο CPN-Tools είναι ένα πακέτο λογισμικού για χρωματισμένα δίκτυα Petri. Υποστηρίζει την κατασκευή, την προσομοίωση, την ανάλυση χώρου καταστάσεων καθώς και την ανάλυση απόδοσης μοντέλων χρωματισμένων δικτύων Petri. Το εργαλείο CPN-Tools αποτελείται από δύο βασικά μέρη: τη γραφική διεπαφή χρήστη (graphical user interface, GUI) και τη γλώσσα μοντελοποίησης χρωματισμένων δικτύων Petri (coloured Petri nets modelling language, CPN ML). Ο συνδυασμός των δύο αυτών στοιχείων έχει ως αποτέλεσμα τη δημιουργία τριών ολοκληρωμένων εργαλείων, που είναι ενσωματωμένα στο εργαλείο CPN-Tools: τον συντάκτη χρωματισμένων δικτύων Petri (CPN editor), τον προσομοιωτή χρωματισμένων δικτύων Petri (CPN simulator), και το εργαλείο χώρου καταστάσεων των χρωματισμένων δικτύων Petri (CPN state space tool).

Η γραφική διεπαφή χρήστη είναι το γραφικό περιβάλλον με το οποίο αλληλεπιδρά ο χρήστης. Η γλώσσα μοντελοποίησης χρωματισμένων δικτύων Petri είναι η γλώσσα προγραμματισμού που χρησιμοποιείται για τις δηλώσεις των μεταβλητών, των τύπων και των επιγραφών (inscriptions) του δικτύου. Επιπρόσθετα, η γλώσσα μοντελοποίησης υλοποιεί τη μηχανή προσομοίωσης, καθώς και τους απαραίτητους αλγορίθμους και δομές δεδομένων που απαιτούνται για τη δημιουργία και διαχείριση των χώρων καταστάσεων. Αυτό σημαίνει ότι η γλώσσα μοντελοποίησης, σαν μέρος του εργαλείου CPN-Tools, είναι υπεύθυνη για τον υπολογισμό των ενεργοποιημένων δεσμευτικών στοιχείων (binding elements) στους διάφορους τρόπους κατά τους οποίους σημαδεύεται (different markings) ένα μοντέλο χρωματισμένου δικτύου Petri.

Η γλώσσα μοντελοποίησης αποτελεί επέκταση της τυπικής γλώσσας μοντελοποίησης (standard modelling language, SML), με κάποια επιπλέον συντακτικά χαρακτηριστικά που απλοποιούν τις δηλώσεις των τύπων και των μεταβλητών. Αυτό προσδίδει κάποια πλεονεκτήματα στη γλώσσα μοντελοποίησης. Πρώτον, η εκφραστικότητα της τυπικής γλώσσας μοντελοποίησης κληρονομείται από το ίδιο το εργαλείο (CPN-Tools). Δεύτερον, η τυπική γλώσσα είναι τόσο τυποποιημένη ώστε πολλά σφάλματα μοντελοποίησης να αναγνωρίζονται έγκαιρα. Επίσης, ο πολυμορφισμός και το συντακτικό της τυπικής γλώσσας μοντελοποίησης επιτρέπει στις επιγραφές του δικτύου να συντάσσονται με μαθηματικό τρόπο. Για τις επιγραφές ενός χρωματισμένου δικτύου Petri χρησιμοποιείται μόνο το καθαρά συναρτησιακό μέρος της τυπικής γλώσσας μοντελοποίησης.

Συνήθως, όταν πρόκειται να χρησιμοποιηθεί το εργαλείο CPN-Tools, αρχικά κατασκευάζεται το μοντέλο χρωματισμένου δικτύου Petri του συστήματος που πρόκειται να μελετηθεί. Ο χρήστης εργάζεται απευθείας επάνω στη γραφική αναπαράσταση του μοντέλου. Για να είναι δυνατό να προσομοιωθεί η λειτουργία του μοντέλου, θα πρέπει αυτό να αποτελεί ένα συντακτικά σωστό δίκτυο Petri. Εφόσον διορθωθούν τα συντακτικά σφάλματα που πιθανώς έχουν παρατηρηθεί, ο χρήστης μπορεί να μεταβεί στον προσομοιωτή χρωματισμένων δικτύων Petri. Κατά τη μετάβαση αυτή δημιουργείται ο απαραίτητος κώδικας για την προσομοίωση του μοντέλου. Ο συντακτικός έλεγχος, καθώς και η μεταγλώττιση (compilation) του μοντέλου, υλοποιούνται από το τμήμα γλώσσας μοντελοποίησης του συντάκτη.

Αφού μεταγλωττιστεί ο κώδικας για την προσομοίωση, ο χρήστης μπορεί να ξεκινήσει να μελετά τη συμπεριφορά του συστήματος μέσω διαδοχικών προσομοιώσεων. Οι πρώτες προσομοιώσεις που συνήθως γίνονται είναι εκτελέσεις βήμα-προς-βήμα (step-by-step) του μοντέλου που δημιουργήθηκε. Η εκτέλεση παρουσιάζεται με μεγάλη λεπτομέρεια στο χρήστη, ο οποίος μπορεί να παρατηρήσει τον τρόπο με τον οποίο μετακινούνται οι ενδείξεις (tokens) ανάμεσα στις θέσεις (places) του δικτύου, να διαπιστώσει το πώς ενεργοποιούνται οι μεταβάσεις (transitions) λόγω των ενδείξεων, καθώς και να επιλέξει τα στοιχεία που θα εκτελεστούν στη συνέχεια. Η διαδικασία προσομοίωσης τυπικά αποκαλύπτει ελαττώματα ή / και σφάλματα που είναι πιθανό να παρατηρηθούν στο μοντέλο, τα οποία στη συνέχεια πρέπει να διορθωθούν. Για το λόγο αυτό, η πρώτη φάση συνήθως αποτελείται από έναν αριθμό επαναλήψεων, μεταβαίνοντας εναλλακτικά στο συντάκτη και τον προσομοιωτή, βελτιώνοντας έτσι σταδιακά το μοντέλο. Η προσομοίωση, ή εκτέλεση, του μοντέλου καθοδηγείται από τη μηχανή προσομοίωσης της γλώσσας μοντελοποίησης χρωματισμένων δικτύων Petri. Χρησιμοποιεί το τμήμα προσομοιωτή της γραφικής διεπαφής χρήστη για την απεικόνιση της λειτουργίας του μοντέλου.

Η επόμενη φάση αποτελείται από μερικές μακροσκελείς προσομοιώσεις ώστε να εξαχθούν πιο λεπτομερή συμπεράσματα για την υλοποίηση του μοντέλου. Στις προσομοιώσεις αυτές, η απεικόνιση της λειτουργίας του μοντέλου αντικαθίσταται από μια πιο γενική αναπαράσταση της λειτουργίας του συστήματος. Το αποτέλεσμα που παρουσιάζεται στο χρήστη, ανάλογα με τον τρόπο υλοποίησης, είναι η κατάσταση του συστήματος μετά από πολλές ενδιάμεσες μεταβάσεις καταστάσεων, ή / και στατιστικά στοιχεία μέχρι την τρέχουσα κατάσταση.

Αφού ολοκληρωθούν οι φάσεις αυτές, ο χρήστης έχει τη δυνατότητα να μεταβεί στο εργαλείο χώρου καταστάσεων για να επιβεβαιώσει την ορθότητα της λειτουργίας του μοντελοποιημένου συστήματος. Αρχικά ελέγχεται η εγκυρότητα του μοντέλου για ανάλυση του χώρου καταστάσεων, ενώ στη συνέχεια παράγεται ο χώρος καταστάσεων ή μέρος αυτού. Η εξαγωγή του χώρου καταστάσεων από ένα μοντέλο υλοποιείται από το τμήμα γλώσσας μοντελοποίησης του εργαλείου χώρου καταστάσεων. Ο χρήστης πλέον μπορεί να θέσει αιτήματα για τη συμπεριφορά του συστήματος,

χρησιμοποιώντας τη διαθέσιμη γλώσσα. Στο σημείο αυτό, χρησιμοποιείται η γραφική διεπαφή χρήστη για την απεικόνιση πληροφοριών για το χώρο καταστάσεων, καθώς και για να παρουσιάσει τα αποτελέσματα των αιτημάτων που έθεσε ο χρήστης.

### **3.3 – Μοντέλα ροής εργασιών βασισμένα σε χρωματισμένα δίκτυα Petri**

Έχει γίνει μεγάλη μελέτη πάνω στις ροές εργασίας και τον τρόπο αναπαράστασής τους με χρωματισμένα δίκτυα Petri. Στη συνέχεια παρουσιάζονται μερικά παραδείγματα μοντέλων ροής εργασίας, με τη γενική περιγραφή τους καθώς και την αναπαράστασή τους ως χρωματισμένα δίκτυα Petri.

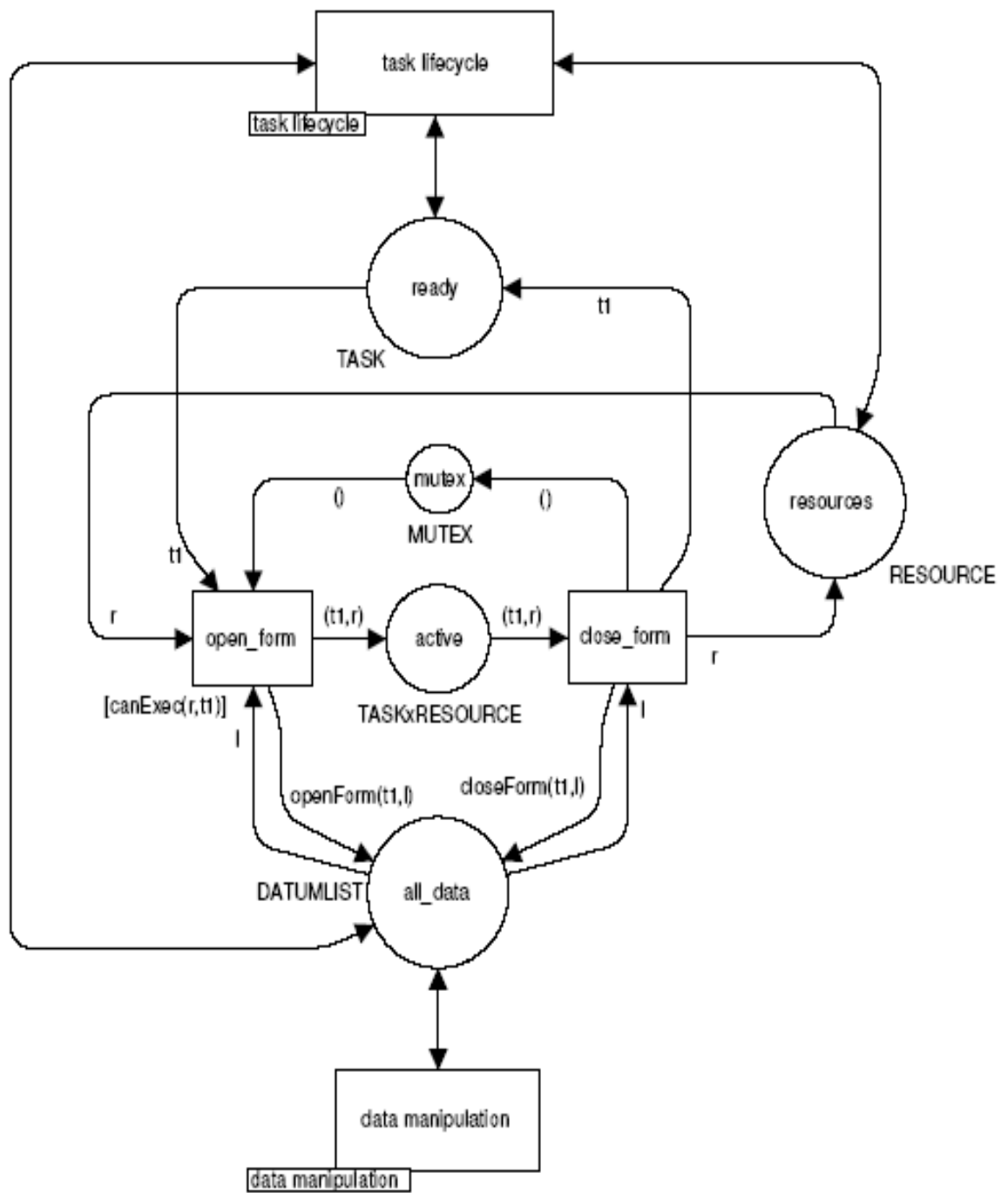
#### 3.3.1 – Μοντελοποίηση των Αρχών χειρισμού περιπτώσεων ([6])

Ο χειρισμός περιπτώσεων (case handling) αποτελεί ένα εναλλακτικό παράδειγμα για την υποστήριξη της ευελιξίας σε επιχειρησιακές διεργασίες προσανατολισμένες στη γνώση. Σε αντίθεση με τη διαχείριση ροής εργασίας, που χρησιμοποιεί προκαθορισμένες δομές ελέγχου για να καθορίσει τι θα έπρεπε να συμβαίνει κατά τη διάρκεια μιας διαδικασίας ροής εργασίας, ο χειρισμός περιπτώσεων εστιάζει στο τι *μπορεί* να γίνει ώστε να επιτευχθεί ένας επιχειρησιακός στόχος.

Τα κύρια στοιχεία του χειρισμού περιπτώσεων είναι:

- Αποφυγή καθορισμού μέσα από συμφραζόμενα, παρέχοντας όλες τις διαθέσιμες πληροφορίες. Αυτό σημαίνει ότι μια περίπτωση παρουσιάζεται ως ένα σύνολο και όχι σαν γενικά μέρη που συνδέονται μεταξύ τους με κάποιο τρόπο.
- Καθορισμός των διεργασιών που ενεργοποιούνται βάσει των διαθέσιμων πληροφοριών και όχι βάσει των διεργασιών που έχουν ήδη εκτελεστεί.
- Σαφής διαχωρισμός της κατανομής εργασίας από την εξουσιοδότηση. Παράλληλα, επιτρέπεται να υπάρχουν διάφοροι τύποι ρόλων, όχι μόνο ο ρόλος εκτέλεσης.
- Επιτρέπεται στους εργαζόμενους να παρατηρούν και να τροποποιούν / προσθέτουν δεδομένα πριν ή μετά την εκτέλεση των αντίστοιχων εργασιών. Με άλλα λόγια, οι πληροφορίες μπορούν να καταχωρηθούν τη στιγμή που γίνονται διαθέσιμες.

Στο Σχήμα 1 παρουσιάζεται το ανώτερο (ιεραρχικά) επίπεδο ενός μοντέλου χειρισμού περιπτώσεων. Στο Σχήμα 2 αναφέρονται οι ορισμοί των δεδομένων που χρησιμοποιούνται στο μοντέλο αυτό. Στα χρωματισμένα δίκτυα Petri, τα δεδομένα ορίζονται ως σύνολα χρωμάτων (color sets). Βάσει των συνόλων χρωμάτων δημιουργούνται οι ενδείξεις που αποθηκεύονται στις θέσεις του δικτύου, οι οποίες, με τη σειρά τους, ενεργοποιούν τις μεταβάσεις ώστε να επιτευχθεί η ροή εκτέλεσης του μοντέλου.



Σχήμα 1

```

color STRINGLIST = list STRING;

color DATUM = product
  (* name *) STRING *
  (* enabled *) BOOL *
  (* isset *) BOOL;
color DATUMLIST = list DATUM;

color ROLE = STRING;

color RESOURCE = product
  (* name *) STRING *
  (* roles *) STRINGLIST;

color MUTEX = unit with null;

color TASKSTATE =
  with initial | enabled | finished;

color TASK = product
  (* name *) STRING *
  (* data *) STRINGLIST *
  (* mandatory *) STRINGLIST *
  (* previous *) STRINGLIST *
  (* successors *) STRINGLIST *
  (* exec role *) STRING *
  (* skip role *) STRING *
  (* redo role *) STRING;

color TASKxRESOURCE =
  product TASK * RESOURCE;

color TASKENTRY = product
  (* name *) STRING *
  (* state *) TASKSTATE;

color TASKLIST = list TASKENTRY;

```

## Σχήμα 2

Η εκτέλεση του μοντέλου ξεκινά από τη θέση *ready*. Όταν μια εργασία είναι έτοιμη για επεξεργασία, τοποθετείται στη θέση αυτή από το σύστημα. Αντίστοιχα, όταν οι προϋποθέσεις μιας εργασίας έχουν εκπληρωθεί, το σύστημα αφαιρεί την εργασία από τη θέση αυτή. Στο σημείο αυτό ο χρήστης μπορεί να ξεκινήσει το χειρισμό μιας εργασίας που βρίσκεται στη θέση *ready*, ανοίγοντας την αντίστοιχη φόρμα διαχείρισης δεδομένων. Αυτό επιτυγχάνεται εκτελώντας τη μετάβαση *open\_form* που καταναλώνει μια εργασία από τη θέση *ready* και ένα πόρο από τη θέση *resources*. Και οι δύο αυτές ενδείξεις αποθηκεύονται εμμέσως στη θέση *active*. Όταν εκτελεστεί η μετάβαση *open\_form* θα καταναλώσει μια ένδειξη από τη θέση *all\_data*, θέτοντας όλα τα στοιχεία δεδομένων (data elements) στην τρέχουσα εργασία καθώς ενεργοποιήθηκε. Στη συνέχεια, εκτελώντας τη μετάβαση *close\_form* θα απενεργοποιηθούν τα στοιχεία δεδομένων που ορίζονται από την πλέον ολοκληρωμένη εργασία, ενώ η ίδια η εργασία θα μεταφερθεί πίσω στη θέση *ready* και ο πόρος που χρησιμοποιήθηκε θα τοποθετηθεί πίσω στη θέση *resources* ως διαθέσιμος για επόμενη εργασία.

Πρέπει να σημειωθεί ότι τα συστήματα χειρισμού περιπτώσεων δεν επιτρέπουν συνήθως τον ταυτόχρονο χειρισμό δεδομένων. Εάν πολλαπλοί χρήστες ανοίξουν παράλληλα διαφορετικές φόρμες για την ίδια περίπτωση, το



αποτέλεσμα είναι απρόβλεπτο αφού θα μπορούσαν να εισάγουν αλληλοσυγκρουόμενες τιμές για το ίδιο δεδομένο στις διαφορετικές φόρμες. Η συμπεριφορά αυτή μοντελοποιείται στη θέση *mutex*, που φέρει μόνο μία ένδειξη. Όταν η ένδειξη αυτή καταναλώνεται από τη μετάβαση *open\_form*, καμία άλλη φόρμα δε μπορεί να ανοιχθεί έως ότου εκτελεστεί η μετάβαση *close\_form* που θα τροφοδοτήσει τη θέση *mutex* με μια νέα ένδειξη.

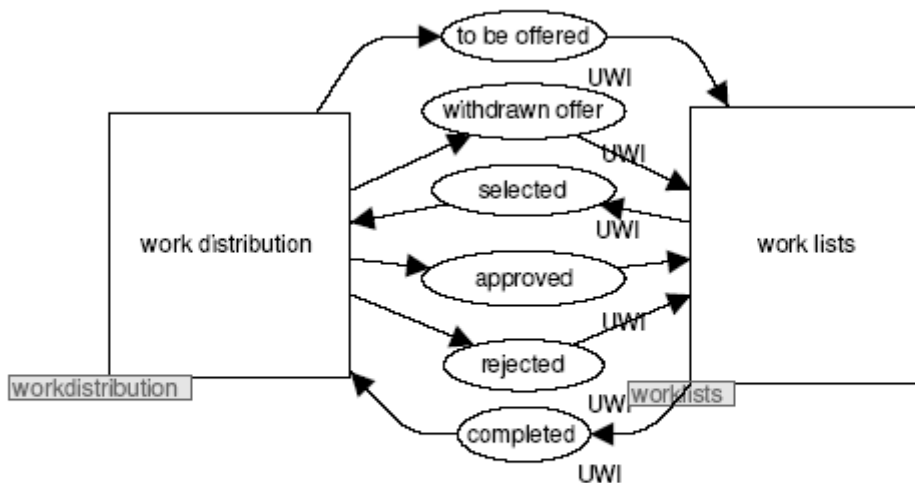
Στα συστήματα χειρισμού περιπτώσεων είναι επίσης δυνατό να καθορίζονται φόρμες που συνδέονται άμεσα με τις περιπτώσεις, σε αντίθεση με τις φόρμες που συνδέονται με εργασίες. Στο μοντέλο αυτό, οι φόρμες περιπτώσεων (*case forms*) αναπαρίστανται από εργασίες που δεν έχουν προγενέστερες και μεταγενέστερες εργασίες, όπως και καθόλου υποχρεωτικά στοιχεία δεδομένων. Είναι τοποθετημένες στη θέση *ready* και δεν την εγκαταλείπουν ακριβώς επειδή δεν είναι συνδεδεμένες με άλλες εργασίες.

Οι συγγραφείς του μοντέλου το δημιούργησαν για να ελέγξουν την ορθότητα ενός τέτοιου συστήματος και την εναρμόνισή του με τις αρχές χειρισμού περιπτώσεων που προορίζονταν να υλοποιηθούν. Για να το επιβεβαιώσουν αυτό, πραγματοποίησαν μια ανάλυση του χώρου καταστάσεων για το μοντέλο αυτό, από την οποία εξάχθηκαν διάφορα συμπεράσματα για τη λειτουργία του συστήματος και τη διατήρηση των αρχών χειρισμού περιπτώσεων που υλοποιούνται στο μοντέλο. Η αναλυτική περιγραφή των αποτελεσμάτων παρουσιάζεται στην εργασία [6].

### 3.3.2 – Μοντελοποίηση μηχανισμού κατανομής εργασίας ([8])

Όταν αναφερόμαστε σε ένα μηχανισμό κατανομής εργασίας (*work distribution mechanism*) παρατηρούμε τη ροή της εργασίας από την οπτική γωνία των πόρων. Αυτό σημαίνει ότι ασχολούμαστε με τον τρόπο κατά τον οποίο το σύστημα κατανέμει την εργασία βάσει της δομής του οργανισμού και των δυνατοτήτων του ανθρώπινου δυναμικού. Τα σύγχρονα συστήματα διαχείρισης ροής εργασίας προσφέρουν ποικίλους μηχανισμούς υποστήριξης των επιχειρησιακών διεργασιών από άποψη πόρων. Η μελέτη πάνω στον

τομέα αυτό δεν έχει ως στόχο τη διαπίστωση και σύγκριση του *τι* κάνουν τα διάφορα συστήματα, αλλά την κατανόηση του *πως* το κάνουν.



Σχήμα 3

Το Σχήμα 3 απεικονίζει το ανώτερο επίπεδο ενός μοντέλου ροής εργασίας που διαθέτει μηχανισμό κατανομής εργασίας. Το βασικό αυτό μέρος του μοντέλου αποτελείται από δύο υπο-συστήματα (sub-systems) ή υποσελίδες (sub-pages), το *work distribution* και το *work lists*. Τα δύο υποσυστήματα επικοινωνούν μεταξύ τους ανταλλάσσοντας μηνύματα μέσω των έξι θέσεων που παρεμβάλλονται ανάμεσά τους. Τα μηνύματα αυτά περιέχουν πληροφορίες για κάποιο χρήστη και για κάποιο αντικείμενο εργασίας, άρα ορίζονται ως συνδυασμός χρήστη – αντικείμενου εργασίας (*color UWI = product User \* WI*).

Η σημασιολογία των μηνυμάτων που ανταλλάσσονται ανάλογα με τη θέση στην οποία τοποθετούνται είναι η εξής:

- to be offered: ένα αντικείμενο εργασίας προσφέρεται στο χρήστη

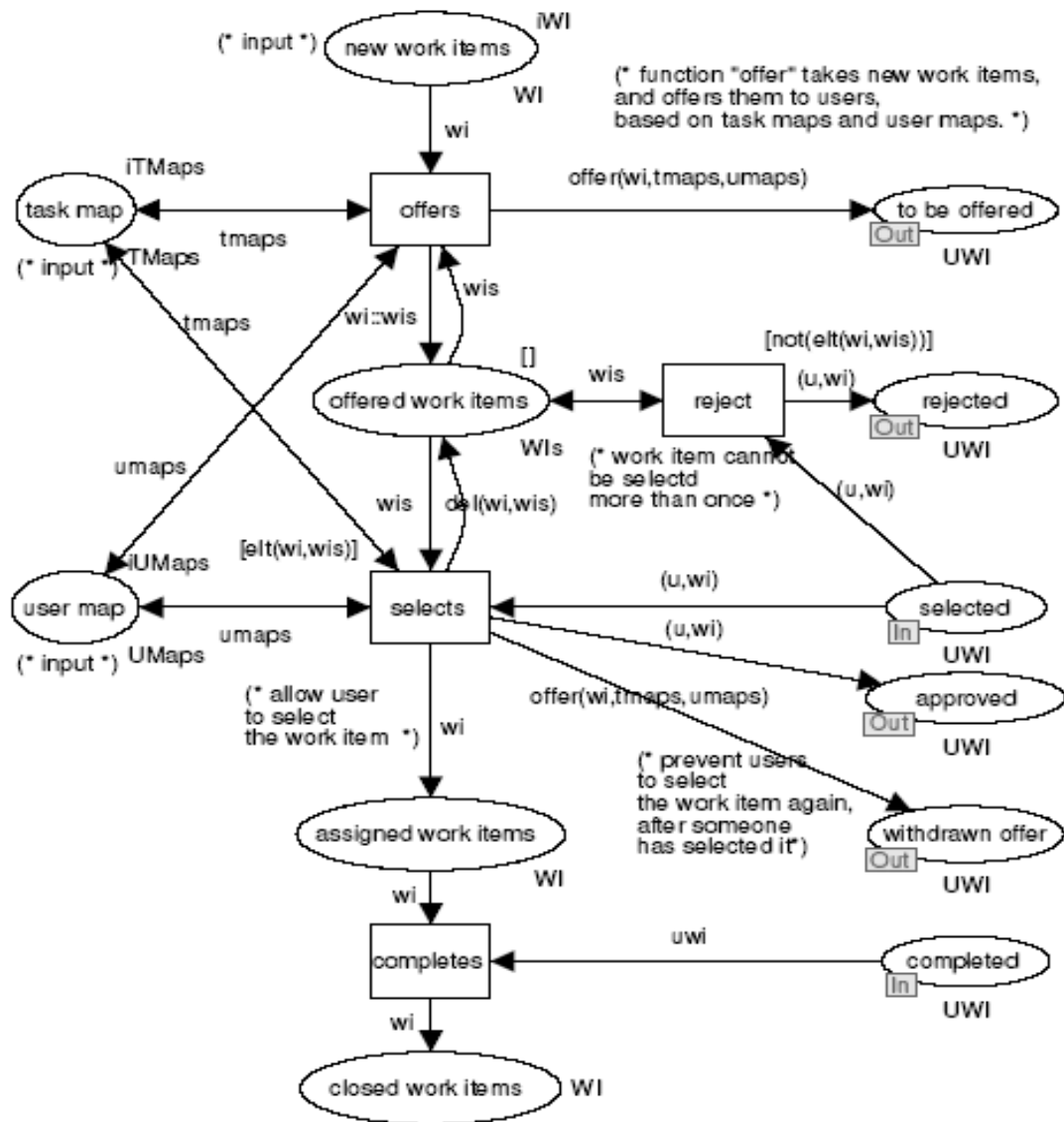
- withdrawn offer: απομακρύνεται το αντικείμενο εργασίας από το χρήστη στον οποίο προσφέρθηκε
- selected: ο χρήστης στέλνει αίτημα για να επιλέξει το αντικείμενο εργασίας
- approved: επιτρέπεται στο χρήστη να επιλέξει το αντικείμενο εργασίας
- rejected: δεν επιτρέπεται στο χρήστη να επιλέξει το αντικείμενο εργασίας
- completed: ο χρήστης ολοκλήρωσε την εκτέλεση του αντικειμένου εργασίας

Στο Σχήμα 4 παρατίθενται οι ορισμοί των δεδομένων που χρησιμοποιούνται στο μοντέλο.

```
color Task = string;
color Case = int;
color WI = product Case * Task;
color User = string;
color Role = string;
color Group = string;
```

```
color WI = product
Case * Task;
color Users = list
User;
color TMap = prod-
uct Task * Role *
Group;
color UMap = prod-
uct User * Roles *
Groups;
```

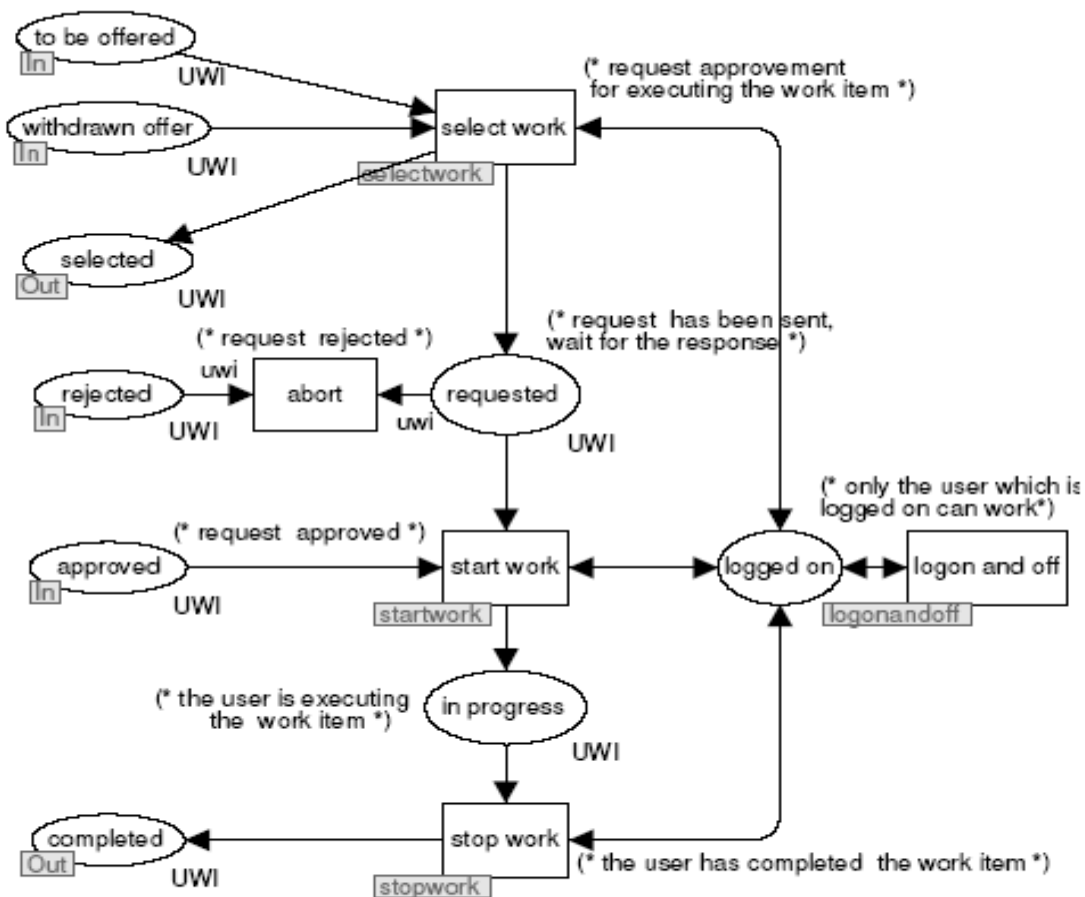
Σχήμα 4



Σχήμα 5

Το Σχήμα 5 παρουσιάζει την υπο-σελίδα στην οποία υλοποιείται ο μηχανισμός κατανομής της εργασίας (*work distribution*). Στο τόξο από τη μετάβαση *offers* στη θέση *to be offered* υπάρχει η συνάρτηση *offer*. Η συνάρτηση αυτή περιέχει κανόνες διανομής του συγκεκριμένου μηχανισμού κατανομής εργασίας. Τα αντικείμενα εργασίας που προσφέρονται στους χρήστες αποθηκεύονται στη θέση *offered work items*. Εφόσον ληφθεί ένα αίτημα από το χρήστη να επιλέξει ένα αντικείμενο εργασίας, λαμβάνεται απόφαση για το αν θα του επιτραπεί να επιλέξει το αντικείμενο (οπότε και να το εκτελέσει) ή αν θα απορριφθεί το αίτημά του. Η απόφαση αυτή λαμβάνεται

βάσει της υπόθεσης ότι σε μια συγκεκριμένη χρονική στιγμή μόνο ένας χρήστης μπορεί να εργάζεται πάνω στο συγκεκριμένο αντικείμενο εργασίας. Εάν το αντικείμενο εργασίας έχει ήδη επιλεγθεί (δηλαδή δε βρίσκεται στη θέση *offered work items*), τότε το μοντέλο απορρίπτει το αίτημα. Εάν κανένας χρήστης δεν έχει επιλέξει το συγκεκριμένο αντικείμενο εργασίας, τότε αποστέλλεται μια έγκριση στο χρήστη και το αντικείμενο εργασίας μετακινείται στη θέση *assigned work items*. Τα αντικείμενα εργασίας που μετακινούνται στη θέση αυτή δεν είναι δυνατό να επιλεγούν ξανά.



Σχήμα 6

Στο Σχήμα 6 φαίνεται η υπο-σελίδα *work lists*. Το υπο-σύστημα αυτό λαμβάνει μηνύματα από τη μονάδα κατανομής εργασίας σχετικά με το ποια

αντικείμενα εργασίας πρόκειται να διατεθούν και σε ποιους χρήστες. Τα βασικά στοιχεία του υπο-συστήματος *work lists* αντιστοιχούν στις βασικές λειτουργίες που μπορεί να εκτελέσει ο χρήστης: να πραγματοποιήσει είσοδο / έξοδο από το σύστημα (μετάβαση *logon and off*), να επιλέξει εργασία (μετάβαση *select work*), να ξεκινήσει να εργάζεται (μετάβαση *start work*) και να σταματά να εργάζεται (μετάβαση *stop work*). Όταν προσφέρεται ένα αντικείμενο εργασίας στους χρήστες (*offered*), αυτοί μπορούν να το επιλέξουν (*selected*). Όταν κάποιος χρήστης επιλέξει το αντικείμενο εργασίας, αποστέλλεται το αίτημα στη μονάδα κατανομής εργασίας. Εάν το αίτημα απορριφθεί (*rejected*) η λειτουργία απορρίπτεται (*abort*). Εάν το αίτημα γίνει αποδεκτό (*approved*) ο χρήστης μπορεί να ξεκινήσει να εργάζεται πάνω στο αντικείμενο. Όσο ο χρήστης ασχολείται με το συγκεκριμένο αντικείμενο εργασίας, αυτό θεωρείται ότι βρίσκεται υπό επεξεργασία (*in progress*). Στη συνέχεια, ο χρήστης μπορεί να σταματήσει να εργάζεται, οπότε το αντικείμενο εργασίας θεωρείται ολοκληρωμένο (*completed*). Για να έχει τη δυνατότητα ο χρήστης να εκτελέσει όλη ή ένα τμήμα της παραπάνω διαδικασίας, θα πρέπει να έχει εισέλθει στο σύστημα (*logged on*).

Το μοντέλο αυτό αποτελεί μια γενική εικόνα όλων των συστημάτων διαχείρισης ροής εργασίας. Ενσωματώνει τους βασικότερους όρους των συστημάτων αυτών, δηλαδή τις εργασίες, τις περιπτώσεις, τους χρήστες, τα αντικείμενα εργασίας, τους ρόλους και τις ομάδες, ενώ παράλληλα δείχνει τον τρόπο με τον οποίο όλα αυτά τα στοιχεία αλληλοσυσχετίζονται. Ο βασικός στόχος των συγγραφέων ήταν να δημιουργήσουν ένα κατανοητό μοντέλο αναφοράς για συστήματα κατανομής εργασίας. Επιπρόσθετα, επέκτειναν το μοντέλο για τρεις διαφορετικές εφαρμογές (*Staffware, FileNet, FLOWer*), ώστε να ερευνήσουν τις ομοιότητες και διαφορές μεταξύ διαφορετικών μηχανισμών κατανομής εργασίας. Περισσότερες λεπτομέρειες για τις επεκτάσεις του μοντέλου και τη μελέτη του με τις προαναφερθείσες εφαρμογές βρίσκονται στην εργασία [8].

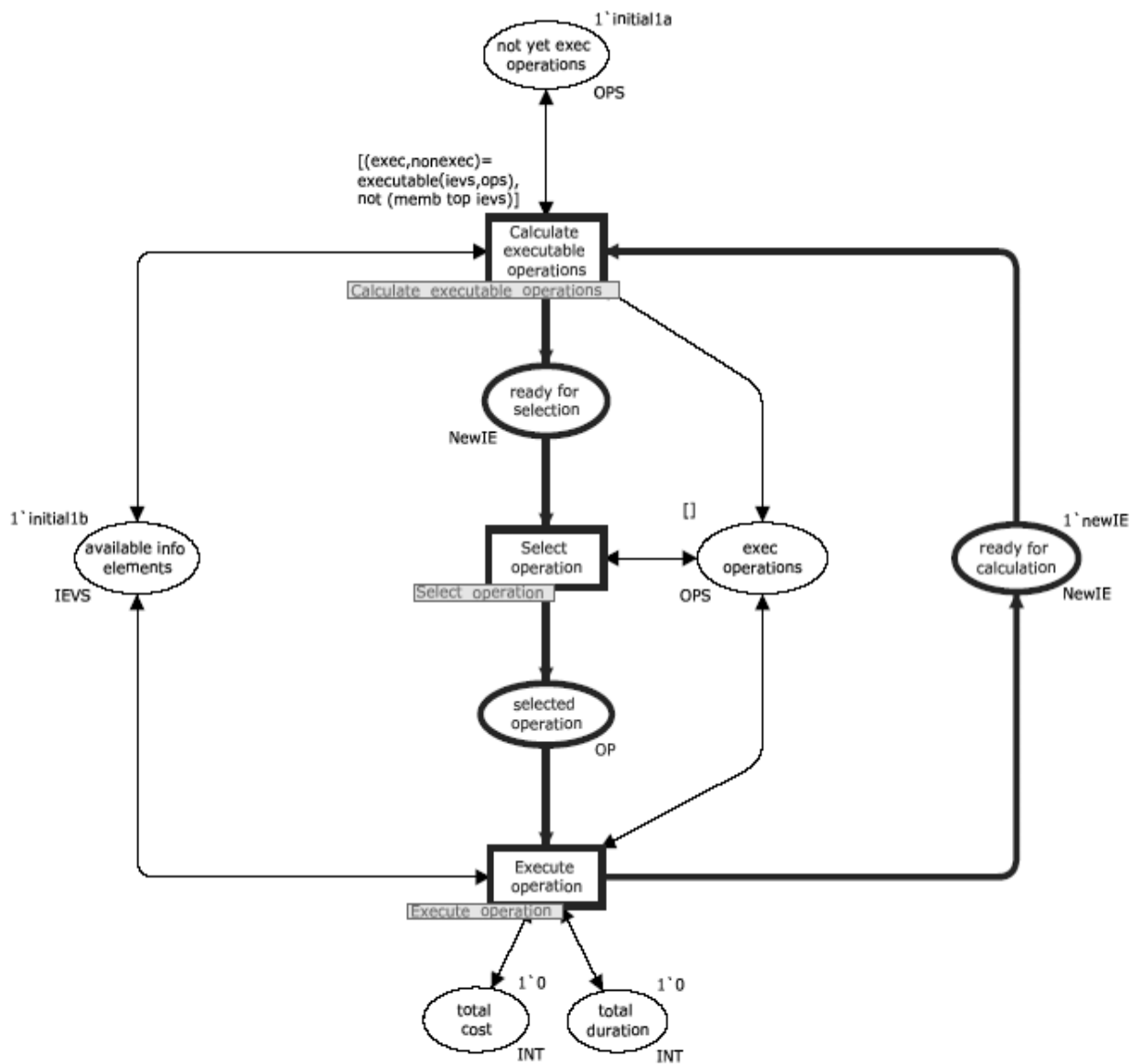
### 3.3.3 – Μοντελοποίηση συστήματος ροής εργασίας που βασίζεται σε προϊόντα ([7])

Μια νέα προσέγγιση στο σχεδιασμό διεργασιών ροής εργασίας είναι η χρήση της θεωρίας σχεδιασμού ροών εργασίας που βασίζονται σε προϊόντα (product-based workflows). Η θεωρία αυτή εστιάζει στο προϊόν μιας ροής εργασίας ως το κύριο αντικείμενο ανάλυσης, ενώ παράγει ένα μοντέλο δεδομένων της επεξεργασίας πληροφοριών στη διαδικασία ροής εργασίας.

Ο σχεδιασμός ροών εργασίας που βασίζονται σε προϊόντα θεωρείται μια επαναστατική προσέγγιση (revolutionary approach). Αυτό σημαίνει ότι ο σχεδιασμός ολόκληρης της διαδικασίας γίνεται ξεκινώντας από το μηδέν (from scratch). Η χρήση της μεθόδου αυτής έχει αρκετά πλεονεκτήματα:

- Το γεγονός ότι ο σχεδιασμός ξεκινά από το μηδέν προσδίδει αρκετό χώρο για βελτιώσεις απόδοσης (radicalism). Προσεγγίσεις που χρησιμοποιούν ήδη υπάρχουσες διαδικασίες αποσκοπούν στην επέκταση αυτών των διαδικασιών, υπερκαλύπτοντας σφάλματα ή ανεπιθύμητες κατασκευαστικές μεθόδους.
- Ο σχεδιασμός ροών εργασίας που βασίζονται σε προϊόντα είναι αντικειμενικός (objective). Επειδή οι προδιαγραφές του προϊόντος αποτελούν τη βάση για το σχεδιασμό μιας ροής εργασίας, κάθε πληροφορία και κανόνας παραγωγής μπορεί να συσχετιστεί και να επαληθευθεί βάσει των προδιαγραφών αυτών. Ως αποτέλεσμα, στο προκύπτον μοντέλο δεν υπάρχουν περιττές εργασίες. Επίσης, η ταξινόμηση των κανόνων παραγωγής καθοδηγείται εξ' ολοκλήρου από τους στόχους απόδοσης της όλης προσπάθειας.
- Η αναλυτική (analytical) προσέγγιση του σχεδιασμού με βάση το προϊόν αποδίδει λεπτομερή στοιχεία, κατάλληλα για να χρησιμοποιηθούν για σκοπούς ανάπτυξης συστημάτων.
- Το γεγονός ότι η μέθοδος εστιάζει στο προϊόν δίνει μια ξεκάθαρη και αντικειμενική αναπαράσταση του προϊόντος ροής εργασίας που βοηθά

στην τεκμηρίωση της διαδικασίας. Επειδή το μοντέλο δεδομένων προϊόντος περιέχει λιγότερες πληροφορίες από ένα μοντέλο διεργασίας, είναι λιγότερο πολύπλοκο στο σχεδιασμό, την κατανόηση και τη διατήρηση.



Σχήμα 7

Στο Σχήμα 7 παρουσιάζεται το ανώτερο επίπεδο ενός μοντέλου ροής εργασίας που βασίζεται σε προϊόν. Η ένδειξη στη θέση *ready for calculation* δεν περιέχει ουσιαστικές πληροφορίες για τη διαμόρφωση του προϊόντος,



αλλά υπάρχει για να διαβεβαιώνει ότι η μετάβαση *calculate executable operations* δε θα εκτελεστεί παρά μόνο όταν έχει εκτελεστεί η λειτουργία που επιλέχθηκε στην προηγούμενη εκτέλεση της κύριας ροής. Η θέση *available info elements* περιέχει μια λίστα από διαθέσιμα στοιχεία δεδομένων, μαζί με τις αντίστοιχες τιμές τους. Τα στοιχεία της λίστας μεταβάλλονται με το πέρασμα του χρόνου εκτέλεσης. Η θέση *not yet executable operations* περιλαμβάνει μια λίστα από λειτουργίες που δε μπορούν να εκτελεστούν ακόμα. Ομοίως, η θέση *executable operations* περιέχει μια λίστα με λειτουργίες που μπορούν να εκτελεστούν. Οι θέσεις *total cost* και *total duration* περιέχουν από μία ακέραια τιμή που αντιπροσωπεύει το συνολικό κόστος και το συνολικό χρόνο εκτέλεσης αντίστοιχα. Οι τιμές αυτές ανανεώνονται με την εκτέλεση κάθε λειτουργίας. Η θέση *ready for calculation* διαβεβαιώνει ότι η μετάβαση *select operation* ενεργοποιείται μόνο όταν έχει γίνει επανυπολογισμός των εκτελέσιμων λειτουργιών. Η θέση *selected operation* χρησιμοποιείται για να αποθηκεύει τις λειτουργίες που έχουν επιλεγθεί.

Η εκτέλεση ξεκινά ενεργοποιώντας τη μετάβαση *calculate executable operations*. Όταν εκτελείται, όλες οι διαθέσιμες λειτουργίες για εκτέλεση καθορίζονται από τη λίστα στη θέση *not yet executable operations*. Αφού καθοριστούν οι εκτελέσιμες λειτουργίες αποθηκεύονται στη θέση *executable operations*. Στη συνέχεια, επιλέγεται μία από τις εκτελέσιμες λειτουργίες και τελικά εκτελείται, προσθέτοντας το παραγόμενο στοιχείο στη λίστα διαθέσιμων στοιχείων δεδομένων και καθορίζεται η τιμή του στοιχείου αυτού. Εν συνεχεία, η ροή ξεκινά πάλι: οι εκτελέσιμες λειτουργίες επανυπολογίζονται βάσει των νέων πληροφοριών, μια λειτουργία μπορεί να επιλεγθεί, και ακολουθεί η ίδια διαδικασία έως ότου παραχθεί το τελικό προϊόν της διεργασίας. Πιο συγκεκριμένα, αυτό σημαίνει ότι το ανώτερο στοιχείο δεδομένων (δηλαδή αυτό που αποτελείται από το συνδυασμό των μικρότερων στοιχείων δεδομένων) έχει τοποθετηθεί στη λίστα των διαθέσιμων στοιχείων δεδομένων.

Το μοντέλο αυτό αποτελεί ένα παράδειγμα για συστήματα ροής εργασίας τα οποία επικεντρώνονται στο προϊόν της διαδικασίας και όχι στον τρόπο με τον οποίο εκτελείται η διαδικασία. Οι συγγραφείς του μοντέλου το δημιούργησαν για να μελετήσουν το χρονικό και το χρηματικό κόστος από την

εφαρμογή ενός τέτοιου συστήματος στον πραγματικό κόσμο. Για να εξαχθούν αξιόπιστα συμπεράσματα, υλοποίησαν πέντε διαφορετικές στρατηγικές επιλογής λειτουργίας για εκτέλεση, ενώ για να επιβεβαιωθεί η ορθότητα του συστήματος πραγματοποίησαν και μια ανάλυση του χώρου καταστάσεων για το μοντέλο. Τα αποτελέσματα από την εφαρμογή κάθε στρατηγικής επιλογής, καθώς και αυτά της ανάλυσης του χώρου καταστάσεων, παρουσιάζονται λεπτομερώς στην εργασία [7].

## ΕΝΟΤΗΤΑ 4<sup>η</sup>

### ΑΝΑΛΥΣΗ ΑΠΟΔΟΣΗΣ ΜΟΝΤΕΛΩΝ ΡΟΗΣ ΕΡΓΑΣΙΑΣ ΜΕ ΤΗ ΧΡΗΣΗ ΧΡΩΜΑΤΙΣΜΕΝΩΝ ΔΙΚΤΥΩΝ PETRI

#### 4.1 – Εισαγωγή

Η απόδοση αποτελεί συνήθως σημαντικό παράγοντα στο σχεδιασμό, την ανάπτυξη και την παραμετροποίηση συστημάτων. Η βεβαιότητα ότι κάποιο σύστημα λειτουργεί σωστά δεν είναι πάντα αρκετή, γιατί ένα σύστημα πρέπει να λειτουργεί και αποτελεσματικά. Για το λόγο αυτό διεξάγονται μελέτες ανάλυσης απόδοσης (performance analysis), ώστε να αξιολογούνται τα ήδη υπάρχοντα και τα υπό σχεδίαση συστήματα, να συγκρίνονται διαφορετικές παράμετροι, ή ακόμη και να βρίσκεται η βέλτιστη παραμετροποίηση κάποιου συστήματος.

#### 4.2 – Τεχνικές ανάλυσης απόδοσης

Υπάρχουν τρεις εναλλακτικές τεχνικές για την ανάλυση της απόδοσης ενός συστήματος:

- η μέτρηση (measurement)
- τα αναλυτικά μοντέλα (analytical models)
- τα μοντέλα προσομοίωσης (simulation models)

Η μέτρηση της απόδοσης ενός συστήματος μπορεί να εξάγει ακριβή συμπεράσματα όσον αφορά την απόδοση του συστήματος. Όλες οι

πληροφορίες και λεπτομέρειες λαμβάνονται υπ' όψιν, ενώ δε γίνονται υποθέσεις που απλοποιούν τη συμπεριφορά του συστήματος. Με άλλα λόγια, διεξάγεται μια απευθείας μελέτη του συστήματος. Παρ' όλα αυτά, η μέτρηση είναι προαιρετική εάν το υπό μελέτη σύστημα ήδη υπάρχει. Οι μετρήσεις που γίνονται μπορεί να είναι ακριβείς, ή να μην είναι, ανάλογα με την παρούσα κατάσταση του συστήματος.

Τα αναλυτικά μοντέλα μπορούν να παρέχουν ακριβή αποτελέσματα όταν μελετάται η απόδοση ενός συστήματος. Τα αποτελέσματα αυτά είναι ακριβή, με την έννοια ότι δεν αποτελούν εκτιμήσεις της απόδοσης του συστήματος. Παρ' όλα αυτά, τα αποτελέσματα που εξάγονται από τα αναλυτικά μοντέλα μπορεί να είναι ακριβή αλλά και ανακριβή, ανάλογα με τις υποθέσεις που έγιναν ώστε να δημιουργηθεί το μοντέλο. Συνήθως είναι αρκετά δύσκολο να μοντελοποιηθούν με ακρίβεια επιχειρησιακά συστήματα χρησιμοποιώντας αναλυτικά μοντέλα. Αυτό συμβαίνει επειδή η αναλυτική μοντελοποίηση προϋποθέτει τόσο πολλές υποθέσεις και απλοποιήσεις, ώστε να είναι αβέβαια τα αποτελέσματα της απόδοσης ενός τέτοιου συστήματος.

Η μοντελοποίηση προσομοίωσης διαφέρει αρκετά από τις αναλυτικές τεχνικές μοντελοποίησης. Η συγκεκριμένη τεχνική σπάνια εξάγει ακριβείς απαντήσεις για την απόδοση κάποιου συστήματος, αλλά είναι πιθανό να εκτιμηθεί το πόσο ακριβείς είναι οι εκτιμήσεις. Επιπλέον, είναι δυνατό να δημιουργηθούν και να αναλυθούν πολύπλοκα μοντέλα χωρίς να γίνονται περιοριστικές υποθέσεις για το σύστημα. Παρ' όλα αυτά, μπορεί να αποδειχθεί αρκετά χρονοβόρα η εκτέλεση των απαραίτητων προσομοιώσεων, ενώ είναι σχετικά δύσκολο να παραχθούν αρκετά ακριβή αποτελέσματα. Η ανάλυση της απόδοσης ενός μοντέλου που βασίζεται σε προσομοίωση περιλαμβάνει στατιστική έρευνα των εξαγόμενων δεδομένων, την εξερεύνηση μεγάλων συνόλων δεδομένων, την κατάλληλη οπτική αναπαράσταση, καθώς και την επαλήθευση και εξακρίβωση πειραμάτων προσομοίωσης.

Στο υπόλοιπο της εργασίας αυτής χρησιμοποιείται η τεχνική της μοντελοποίησης προσομοίωσης.

### 4.3 – Μέτρα απόδοσης

Κατά τη διάρκεια της προσομοίωσης, ένα χρωματισμένο δίκτυο Petri μπορεί να περιέχει και να παράγει ένα μεγάλο αριθμό από ποσοτικές πληροφορίες που αφορούν την απόδοση του συστήματος. Όταν αναλύουμε ένα σύστημα ροής εργασιών, υπάρχουν αρκετά μέτρα απόδοσης που μας ενδιαφέρει να υπολογίσουμε. Ενδεικτικά, μερικά από αυτά είναι:

- Ο χρόνος διεκπεραίωσης μιας εργασίας
- Ο συνολικός μέσος χρόνος ολοκλήρωσης των εργασιών
- Ο μέσος χρόνος αναμονής των εργασιών στο σύστημα
- Ο αριθμός των εργασιών που βρίσκονται υπό επεξεργασία σε μια συγκεκριμένη χρονική στιγμή
- Ο συνολικός χρόνος επεξεργασίας του συστήματος για συγκεκριμένο φόρτο εργασίας
- Η μέση χρησιμοποίηση των διαθέσιμων πόρων
- Η αναλογία κατανομής των διαθέσιμων πόρων σε εργασίες, ανάλογα με τη μέθοδο επιλογής πόρων του συστήματος
- Ο μέσος χρόνος επεξεργασίας της ίδιας εργασίας από πόρους με διαφορετικούς ρόλους

Τα μέτρα απόδοσης που υπολογίζονται κατά τη διάρκεια μιας προσομοίωσης αποτελούν μόνο εκτιμήσεις των πραγματικών μέτρων απόδοσης. Για το λόγο αυτό μπορούν να χρησιμοποιηθούν διαστήματα εμπιστοσύνης (confidence intervals) ώστε να αποδειχθεί το πόσο ακριβείς είναι οι εκτιμήσεις ενός μέτρου απόδοσης. Για να υπολογιστούν σωστά διαστήματα εμπιστοσύνης, πρέπει να είναι δυνατή η συλλογή εκτιμήσεων που είναι ανεξάρτητες και ομοιόμορφα κατανομημένες (independent and identically distributed, IID). Αυτό σημαίνει ότι οι εκτιμήσεις των μέτρων απόδοσης πρέπει να είναι ανεξάρτητες μεταξύ τους, ενώ παράλληλα έχουν την ίδια πιθανότητα κατανομής στο σύστημα.

#### 4.4 – Εργαλεία παρακολούθησης χρωματισμένων δικτύων Petri

Σε ένα σύστημα που έχει μοντελοποιηθεί ως χρωματισμένο δίκτυο Petri, τα μέτρα απόδοσης υπολογίζονται παρατηρώντας και εξάγοντας δεδομένα από τις καταστάσεις και τα γεγονότα που συμβαίνουν στο δίκτυο κατά τη διάρκεια μιας προσομοίωσης. Προφανώς, οι απαραίτητες πληροφορίες πρέπει να περιέχονται στο μοντέλο ώστε να παρατηρηθούν. Ωστόσο, ο χρήστης χρειάζεται ένα μέσο ώστε να έχει τη δυνατότητα να εξάγει δεδομένα κατά την προσομοίωση. Τα χρωματισμένα δίκτυα Petri διαθέτουν, για το σκοπό αυτό, ένα σύνολο εργαλείων που επιτρέπουν στο χρήστη να αναλύει με διάφορους τρόπους την απόδοση ενός συστήματος.

Τα εργαλεία παρακολούθησης (monitors) είναι μηχανισμοί που μπορούν να παρακολουθήσουν τις καταστάσεις και τα γεγονότα που συμβαίνουν σε ένα χρωματισμένο δίκτυο Petri κατά την προσομοίωση, ενώ έχουν τη δυνατότητα να εκτελέσουν κατάλληλες λειτουργίες βάσει των παρατηρήσεων. Τα εργαλεία παρακολούθησης μπορούν να χρησιμοποιηθούν για την παρακολούθηση αλλά και τον έλεγχο μιας προσομοίωσης.

Υπάρχουν τρεις βασικοί τύποι εργαλείων παρακολούθησης:

- Τα εργαλεία παρακολούθησης σημείων διακοπής (breakpoint monitors)
- Τα εργαλεία παρακολούθησης συλλογής δεδομένων (data collection monitors)
- Τα εργαλεία παρακολούθησης καταγραφής σε αρχείο (write-in-file monitors)

Τα εργαλεία παρακολούθησης σημείων διακοπής ελέγχουν τη διαδικασία της προσομοίωσης και τη διακόπτουν όταν ικανοποιηθεί κάποια προϋπόθεση. Συνήθως χρησιμοποιούνται για να σταματά η προσομοίωση όταν έχει ολοκληρωθεί ένας συγκεκριμένος αριθμός βημάτων, ή όταν έχει περάσει ένα ορισμένο χρονικό διάστημα προσομοίωσης.

Τα εργαλεία παρακολούθησης συλλογής δεδομένων εξάγουν αριθμητικά δεδομένα κατά τη διάρκεια προσομοιώσεων. Τα δεδομένα που συλλέγονται χρησιμοποιούνται στη συνέχεια για στατιστική ανάλυση των μέτρων απόδοσης που υπολογίζονται.

Τα εργαλεία παρακολούθησης καταγραφής σε αρχείο δημιουργούν αρχεία στα οποία καταγράφονται πληροφορίες για τα δεδομένα που παρατηρούνται κατά την προσομοίωση. Κύρια λειτουργία τους είναι η ανανέωση των αρχείων καταγραφής κάθε φορά που μεταβάλλεται η κατάσταση του υπό ανάλυση μοντέλου.

#### **4.5 – Δομή των εργαλείων παρακολούθησης**

Στα χρωματισμένα δίκτυα Petri όλα τα εργαλεία παρακολούθησης έχουν συγκεκριμένη δομή. Ανάλογα με τον τύπο εργαλείου παρακολούθησης, υπάρχουν συγκεκριμένες συναρτήσεις που διαμορφώνουν τη δομή και τη λειτουργικότητά του. Οι συναρτήσεις αυτές αποκαλούνται συναρτήσεις παρακολούθησης (monitoring functions), ενώ καθεμία παίζει διαφορετικό ρόλο στη λειτουργία του εργαλείου παρακολούθησης του οποίου είναι μέρος.

##### **4.5.1 - Εργαλεία παρακολούθησης σημείων διακοπής**

Τα εργαλεία παρακολούθησης αυτού του τύπου περιλαμβάνουν τις εξής συναρτήσεις παρακολούθησης:

- Συνάρτηση κριτηρίου (predicate function)
- Συνάρτηση δράσης (action function)

Η συνάρτηση κριτηρίου περιέχει την προϋπόθεση διακοπής (stop criterion) της προσομοίωσης. Κατά τη διάρκεια της προσομοίωσης, όταν το κριτήριο αυτό ικανοποιηθεί τότε η συνάρτηση δράσης θα ενημερώσει τον προσομοιωτή ότι βρέθηκε ένα σημείο διακοπής (breakpoint) και η διαδικασία θα διακοπεί.

#### 4.5.2 – Εργαλεία παρακολούθησης συλλογής δεδομένων

Τα εργαλεία παρακολούθησης του τύπου αυτού περιλαμβάνουν τις εξής συναρτήσεις παρακολούθησης:

- Συνάρτηση αρχικοποίησης (initialization function)
- Συνάρτηση κριτηρίου (predicate function)
- Συνάρτηση παρατήρησης (observation function)
- Συνάρτηση δράσης (action function)
- Συνάρτηση διακοπής (stop function)

Η συνάρτηση αρχικοποίησης καλείται μόνο μία φορά προτού ξεκινήσει η διαδικασία προσομοίωσης. Η συνάρτηση αυτή μπορεί να περιέχει μια προαιρετική αριθμητική τιμή. Εάν η τιμή αυτή δεν οριστεί τότε δεν θα εξαχθούν δεδομένα από το μοντέλο πριν την προσομοίωση. Εάν η τιμή οριστεί, τότε θα εξαχθεί από το μοντέλο κάποια αριθμητική τιμή που θα χρησιμοποιηθεί για να ανανεωθούν τα στατιστικά στοιχεία κατά την προσομοίωση. Η συνάρτηση κριτηρίου περιέχει ένα κριτήριο, το οποίο όταν ικανοποιηθεί καλούνται οι συναρτήσεις παρατήρησης και δράσης. Η συνάρτηση παρατήρησης εξετάζει τους κόμβους του μοντέλου που παρακολουθούνται και επιστρέφει μια αριθμητική τιμή. Η συνάρτηση δράσης χρησιμοποιεί τις αριθμητικές τιμές που επιστρέφονται από τη συνάρτηση παρατήρησης και στη συνέχεια ενημερώνει τα στατιστικά στοιχεία της παρακολούθησης. Η λειτουργία της συνάρτησης διακοπής είναι ανάλογη της



συνάρτησης αρχικοποίησης, με τη διαφορά ότι καλείται όταν ικανοποιηθεί κάποιο κριτήριο διακοπής. Η διακοπή της προσομοίωσης συμβαίνει όταν μία ή περισσότερες από τις παρακάτω προϋποθέσεις ικανοποιούνται:

- Δεν υπάρχουν άλλες ενεργοποιημένες μεταβάσεις στο δίκτυο
- Ο αριθμός βημάτων που έχουν οριστεί για την προσομοίωση έχει ολοκληρωθεί
- Ο χρήστης διακόπτει την προσομοίωση
- Έχει ικανοποιηθεί το κριτήριο διακοπής ενός εργαλείου παρακολούθησης σημείου διακοπής (εάν περιέχεται στο μοντέλο)

#### 4.5.3 – Εργαλεία παρακολούθησης καταγραφής σε αρχείο

Τα εργαλεία παρακολούθησης καταγραφής σε αρχείο περιέχουν τις εξής συναρτήσεις παρακολούθησης:

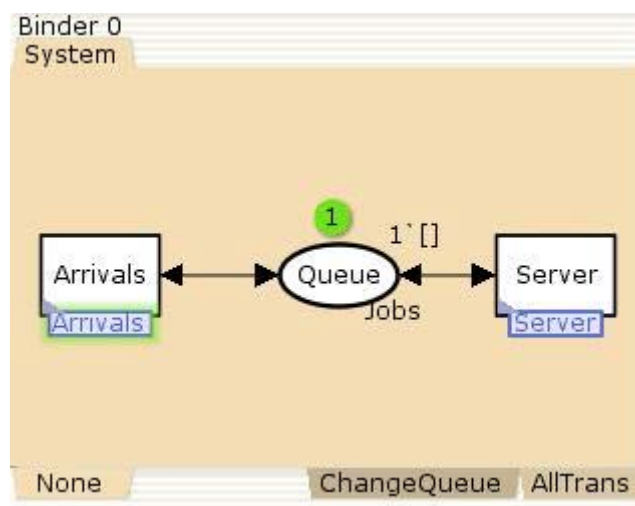
- Συνάρτηση αρχικοποίησης (initialization function)
- Συνάρτηση κριτηρίου (predicate function)
- Συνάρτηση παρατήρησης (observation function)
- Συνάρτηση δράσης (action function)
- Συνάρτηση διακοπής (stop function)

Η συνάρτηση αρχικοποίησης καλείται μία φορά πριν ξεκινήσει η προσομοίωση του μοντέλου. Περιέχει μια αλφαριθμητική τιμή (string value) που προστίθεται στο αρχείο καταγραφής πριν την προσομοίωση. Η συνάρτηση κριτηρίου περιέχει μια προϋπόθεση, η οποία όταν εκπληρωθεί καλούνται οι συναρτήσεις παρατήρησης και δράσης. Η συνάρτηση παρατήρησης εξετάζει τους κόμβους του μοντέλου που παρακολουθούνται και επιστρέφει μια αλφαριθμητική τιμή. Η συνάρτηση δράσης λαμβάνει το αλφαριθμητικό που επιστράφηκε από τη συνάρτηση παρακολούθησης και το

προσθέτει στο αρχείο καταγραφής. Η συνάρτηση διακοπής καλείται όταν εκπληρωθεί κάποιο από τα κριτήρια διακοπής. Προσθέτει στο αρχείο καταγραφής ένα αλφαριθμητικό που δηλώνει τη διακοπή της διαδικασίας προσομοίωσης.

#### 4.6 – Παράδειγμα υλοποίησης εργαλείων παρακολούθησης με το CPN-Tools

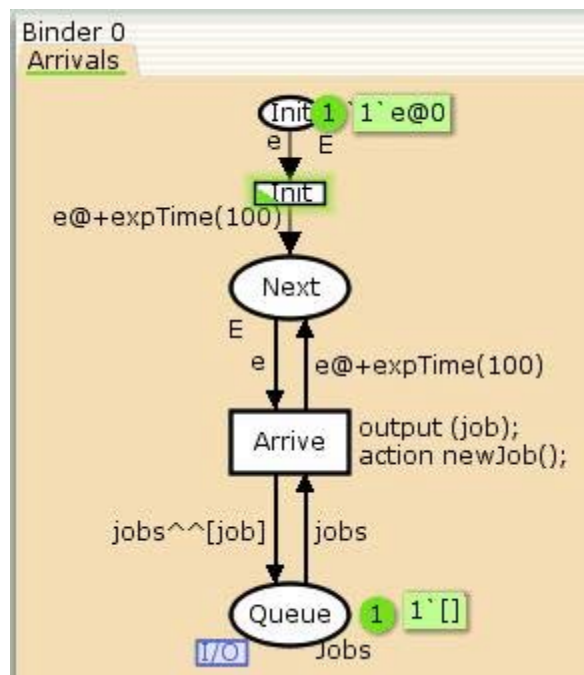
Στο σημείο αυτό πρέπει να σημειωθεί ότι για τη μελέτη των εργαλείων παρακολούθησης χρησιμοποιήθηκε η έκδοση 2.0.0 του CPN-Tools. Στην έκδοση αυτή έχουν προστεθεί λειτουργίες που παράγουν έτοιμες γενικευμένες φόρμες (generic templates) για εργαλεία παρακολούθησης, γεγονός το οποίο απλοποιεί κατά ένα μεγάλο μέρος τη δημιουργία τους.



Σχήμα 1

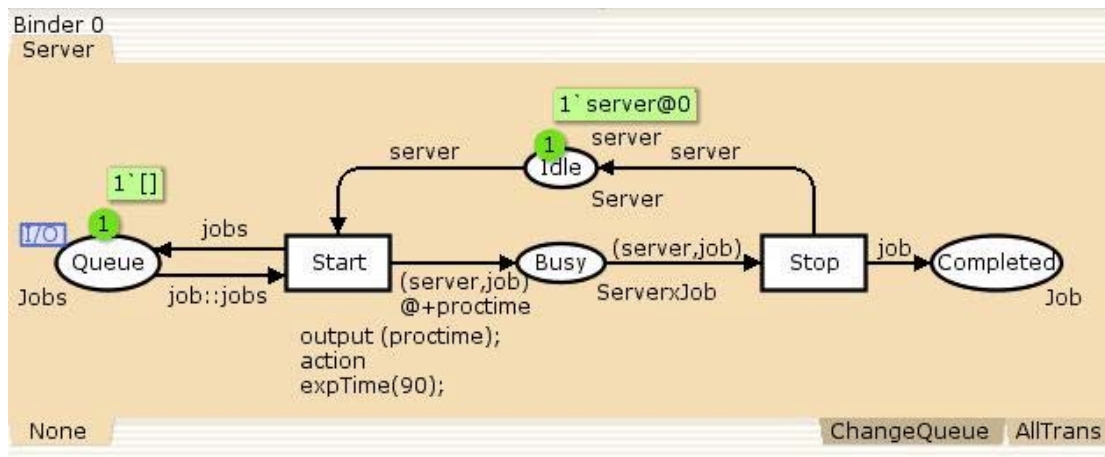
Στο Σχήμα 1 παρουσιάζεται το ανώτερο επίπεδο ενός απλού μοντέλου ροής εργασιών (*Queue System*, [10]). Στο σύστημα αυτό ένας εξυπηρέτης (*server*) επεξεργάζεται εργασίες που φτάνουν περιοδικά στο σύστημα. Εάν ο εξυπηρέτης είναι απασχολημένος όταν φτάνει μια νέα εργασία τότε αυτή

προστίθεται στο τέλος μιας ουράς (queue). Εάν ο εξυπηρέτης δεν είναι απασχολημένος όταν φτάνει κάποια εργασία τότε αυτή περνά απευθείας στον εξυπηρέτη και ξεκινά η επεξεργασία της. Όταν ολοκληρώνεται η επεξεργασία μιας εργασίας ο εξυπηρέτης γίνεται και πάλι διαθέσιμος (idle state). Όταν ο εξυπηρέτης δεν είναι απασχολημένος και υπάρχουν εργασίες στην ουρά αναμονής, τότε θα επιλέξει την εργασία που βρίσκεται στην κεφαλή της ουράς και θα αρχίσει να την επεξεργάζεται.



Σχήμα 2

Στο Σχήμα 2 απεικονίζεται η υπο-σελίδα των αφίξεων (arrivals) των εργασιών. Μια ένδειξη στη θέση *Next* χρησιμοποιείται για να καθορίζει την άφιξη νέων εργασιών. Η μετάβαση *Init* είναι η μόνη που αρχικά ενεργοποιείται και χρησιμοποιείται για να τοποθετεί ένδειξεις στη θέση *Next*. Η μετάβαση *Arrive* ενεργοποιείται μόνο όταν η σφραγίδα χρόνου (time stamp) της ένδειξης στη θέση *Next* είναι ίση με τον τρέχων χρόνο του μοντέλου. Όταν εκτελείται η μετάβαση *Arrive* δημιουργείται μια νέα εργασία, ενώ στη συνέχεια τοποθετείται στο τέλος της ουράς αναμονής μέσω της θέσης *Queue*.



Σχήμα 3

Στο Σχήμα 3 φαίνεται η υπο-σελίδα του εξυπηρέτη. Οι θέσεις *Idle* και *Busy* χρησιμοποιούνται για να χαρακτηρίζουν την κατάσταση του εξυπηρέτη. Μια ένδειξη στη θέση *Idle* δηλώνει ότι ο εξυπηρέτης είναι διαθέσιμος για μια νέα εργασία, ενώ μια ένδειξη στη θέση *Busy* δηλώνει ότι ο εξυπηρέτης είναι απασχολημένος. Όταν εκτελείται η μετάβαση *Start*, καθορίζεται ο χρόνος επεξεργασίας της εργασίας που αφαιρείται από την κεφαλή της ουράς εργασιών, ενώ παράλληλα θα μεταφερθεί μια ένδειξη στη θέση *Busy* δηλώνοντας ότι ο εξυπηρέτης είναι απασχολημένος. Όταν η σφραγίδα χρόνου της ένδειξης αυτής είναι ίση με τον τρέχων χρόνο του μοντέλου, ενεργοποιείται η μετάβαση *Stop* που μεταφέρει την ολοκληρωμένη εργασία στη θέση *Completed* και τοποθετεί μια ένδειξη στη θέση *Idle* ώστε να τεθεί ο εξυπηρέτης και πάλι σε διαθεσιμότητα.

```

colset E = with e timed;
colset INT = int;
colset Server = with server timed;
colset JobType = with A | B;
colset Job = record jobType : JobType *
                AT : INT;
colset Jobs = list Job;
colset ServerxJob = product Server * Job timed;

var proctime : INT;
var job: Job;
var jobs: Jobs;

```

Σχήμα 4

```

fun expTime (mean: int) =
  let
    val realMean = Real.fromInt mean
    val rv = exponential((1.0/realMean))
  in
    floor (rv+0.5)
  end;

fun intTime() = IntInf.toInt (time());

fun newJob() = {jobType = JobType.ran(),
                AT      = intTime()}

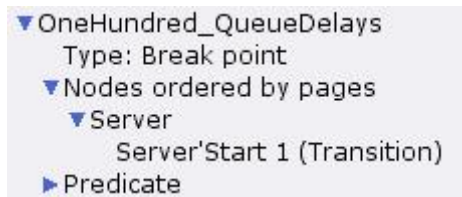
```

Σχήμα 5

Στα Σχήματα 4 και 5 παρουσιάζονται οι δηλώσεις των δεδομένων και των συναρτήσεων που χρησιμοποιούνται στο μοντέλο.

#### 4.6.1 – Υλοποίηση εργαλείου παρακολούθησης σημείου διακοπής

Το εργαλείο παρακολούθησης αυτό δημιουργείται για να διακόπτει τη διαδικασία προσομοίωσης όταν 100 εργασίες έχουν περάσει από την ουρά αναμονής του συστήματος.



Σχήμα 6

Στο Σχήμα 6 φαίνεται η γενική μορφή του εργαλείου παρακολούθησης. Όπως φαίνεται και από το Σχήμα, η παρακολούθηση εφαρμόζεται για τη μετάβαση *Start* που βρίσκεται στην υπο-σελίδα του εξυπηρέτη (*Server*).

```
fun pred (bindelem) =  
  let  
    fun predBindElem (Server'Start (1, {jobs, job,proctime})) =  
      Queue_Delay.count()=100  
      | predBindElem _ = false  
  
  in  
    predBindElem bindelem  
  end
```

Σχήμα 7

Το Σχήμα 7 δείχνει τη μορφή της συνάρτησης κριτηρίου. Ο γενικευμένος έτοιμος κώδικας που παράγεται αυτόματα περιέχει όλες τις πληροφορίες που χρειάζεται ώστε να παρακολουθείται η μετάβαση *Start* κάθε φορά που αυτή ενεργοποιείται. Η μόνη αλλαγή που απαιτείται από το χρήστη είναι να τροποποιήσει τη συνθήκη ικανοποίησης του κριτηρίου. Αυτό πραγματοποιείται με τον εξής τρόπο: όταν δημιουργείται ο έτοιμος κώδικας, η έκφραση αληθείας του κριτηρίου παρουσιάζεται ως

```
fun predBindElem (Server'Start (1, {jobs, job,proctime})) = true  
  | predBindElem _ = false.
```

Η έκφραση αυτή δηλώνει ότι το εργαλείο παρακολούθησης θα ενεργοποιείται κάθε φορά που ενεργοποιείται η μετάβαση *Start*, ενώ σε κάθε άλλη περίπτωση το κριτήριο δεν θα ικανοποιείται. Το κριτήριο θα πρέπει να

ικανοποιείται όταν ο μετρητής που έχει οριστεί για την ουρά αναμονής φτάσει στο 100. Οπότε η επιθυμητή μορφή του κριτηρίου θα είναι

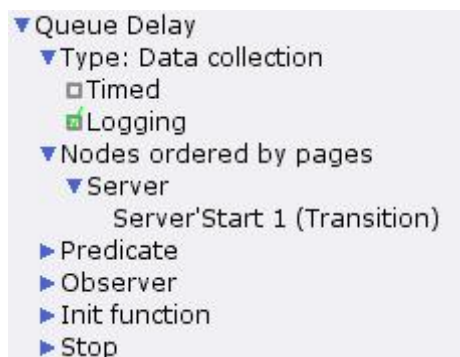
```
fun predBindElem (Server'Start (1, {jobs, job,proctime})) =  
    Queue_Delay.count()=100  
| predBindElem _ = false.
```

Με τον τρόπο αυτό, όταν θα έχουν περάσει 100 εργασίες από την ουρά αναμονής του συστήματος, θα ενεργοποιηθεί το εργαλείο παρακολούθησης και θα διακόψει την προσομοίωση στο σημείο αυτό.

Στα εργαλεία παρακολούθησης σημείων διακοπής ο χρήστης έχει τη δυνατότητα να τροποποιήσει μόνο τη συνάρτηση κριτηρίου. Η συνάρτηση δράσης δημιουργείται και εκτελείται αυτόματα από το εργαλείο παρακολούθησης. Για το λόγο αυτό δεν είναι ορατή από το χρήστη και δε χρειάζεται κάποια τροποποίηση από μέρους του.

#### 4.6.2 – Υλοποίηση εργαλείου παρακολούθησης συλλογής δεδομένων

Αυτό το εργαλείο παρακολούθησης δημιουργείται για να υπολογίζεται ο χρόνος αναμονής των εργασιών στην ουρά του συστήματος.



Σχήμα 8

Το Σχήμα 8 δίνει τη γενική μορφή του εργαλείου παρακολούθησης. Η παρακολούθηση που υλοποιείται εδώ συσχετίζεται με τη μετάβαση *Start*, που βρίσκεται στην υπο-σελίδα του εξυπηρέτη (*Server*).

Στο μοντέλο, κάθε εργασία ορίζεται ως μια εγγραφή (*record*) η οποία περιλαμβάνει ένα πεδίο με το όνομα *AT*. Το πεδίο αυτό αναπαριστά το χρόνο που η εν λόγω εργασία προστίθεται στην ουρά αναμονής. Όταν εκτελείται η μετάβαση *Start*, αφαιρείται η εργασία που βρίσκεται στην κεφαλή της ουράς, ενώ η καθυστέρηση της εργασίας στην ουρά μπορεί να υπολογιστεί αφαιρώντας την τιμή του πεδίου *AT* της εργασίας από τον τρέχων χρόνο του μοντέλου.

Το εργαλείο παρακολούθησης πρέπει να ενεργοποιείται κάθε φορά που εκτελείται η μετάβαση *Start*. Για το λόγο αυτό, η συνάρτηση κριτηρίου θα έχει τη μορφή που φαίνεται στο Σχήμα 9:

```
fun pred (bindelem) =  
  let  
    fun predBindElem (Server'Start (1, {jobs, job,proctime})) = true  
      | predBindElem _ = false  
  in  
    predBindElem bindelem  
  end
```

Σχήμα 9

Ο χρήστης δε χρειάζεται να κάνει καμία τροποποίηση στον έτοιμο κώδικα που παράγεται για τη συνάρτηση κριτηρίου.

Η συνάρτηση παρατήρησης καλείται κάθε φορά που ικανοποιείται η συνάρτηση κριτηρίου, δηλαδή κάθε φορά που εκτελείται η μετάβαση *Start*. Όταν συμβαίνει αυτό, η συνάρτηση παρατήρησης πρέπει να επιστρέψει (ως αριθμητική τιμή) το χρόνο αναμονής της εργασίας που βρισκόταν στην κεφαλή της ουράς. Ο χρόνος αυτός μπορεί να υπολογιστεί με την έκφραση

`(intTime()-(#AT job)).`



Η συνάρτηση `intTime()` επιστρέφει τον τρέχων χρόνο του μοντέλου ως ακέραια τιμή. Ο τελεστής «#» χρησιμοποιείται για να είναι δυνατή η προσπέλαση του πεδίου `AT` της εγγραφής μιας εργασίας. Συνεπώς, η παραπάνω έκφραση υπολογίζει το χρόνο αναμονής της εργασίας στην ουρά του συστήματος. Η τελική μορφή της συνάρτησης παρατήρησης θα είναι αυτή που φαίνεται στο Σχήμα 10:

```
fun obs (bindelem) =  
  let  
    fun obsBindElem (Server'Start (1, {jobs, job, proctime})) =  
      (intTime() - (#AT job))  
      | obsBindElem _ = ~1  
  in  
    obsBindElem bindelem  
  end
```

Σχήμα 10

Το εργαλείο παρακολούθησης δε χρειάζεται να συλλέξει δεδομένα πριν την προσομοίωση, αλλά ούτε και αφού αυτή διακοπεί. Συνεπώς, οι συναρτήσεις αρχικοποίησης και διακοπής θα έχουν τη μορφή που φαίνεται στο Σχήμα 11:

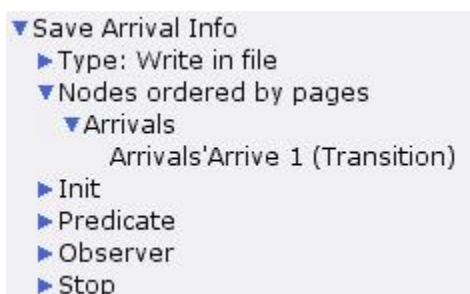
```
fun init () = NONE  
fun stop () = NONE
```

Σχήμα 11

Στα εργαλεία παρακολούθησης συλλογής δεδομένων, η συνάρτηση δράσης δεν είναι ορατή από το χρήστη και για το λόγο αυτό δεν απαιτείται τροποποίηση από μέρους του.

#### 4.6.3 – Υλοποίηση εργαλείου παρακολούθησης καταγραφής σε αρχείο

Αυτό το εργαλείο παρακολούθησης υλοποιείται για να δημιουργήσει ένα αρχείο, στο οποίο θα καταγράφονται πληροφορίες για τις εργασίες που φτάνουν στο σύστημα κατά τη διάρκεια μιας προσομοίωσης.



Σχήμα 12

Στο Σχήμα 12 παρουσιάζεται η γενική μορφή του εργαλείου παρακολούθησης, που εφαρμόζεται στη μετάβαση *Arrive* της υπο-σελίδας *Arrivals*.

Υποτίθεται ότι ο χρήστης επιθυμεί να εμφανίζεται μια επικεφαλίδα περιγραφής στο αρχείο πληροφοριών, η οποία θα περιέχει το κείμενο:

```
Job arrivals
=====
```

Αυτό επιτυγχάνεται τροποποιώντας ανάλογα τη συνάρτηση αρχικοποίησης, της οποίας η μορφή φαίνεται στο Σχήμα 13:

```
fun init () =
  "Job arrivals \n===== \n"
```

Σχήμα 13

Το εργαλείο παρακολούθησης πρέπει να ενεργοποιείται κάθε φορά που πραγματοποιείται μια άφιξη εργασίας, δηλαδή κάθε φορά που εκτελείται η μετάβαση *Arrive*. Οπότε, η συνάρτηση κριτηρίου θα έχει τη μορφή που φαίνεται στο Σχήμα 14:

```

fun pred (bindelem) =
  let
    fun predBindElem (Arrivals'Arrive (1, {jobs, job})) = true
      | predBindElem _ = false
  in
    predBindElem bindelem
  end

```

Σχήμα 14

Ο χρήστης δεν είναι απαραίτητο να τροποποιήσει το γενικευμένο κώδικα που παράγεται αυτόματα για τη συνάρτηση κριτηρίου.

Κάθε φορά που φτάνει μια νέα εργασία στο σύστημα, η συνάρτηση παρατήρησης θα πρέπει να προσθέτει πληροφορίες στο αρχείο καταγραφής.

```

fun obs (bindelem) =
  let
    fun obsBindElem (Arrivals'Arrive (1, {jobs, job})) =
      Int.toString(Count_trans_occur_Arrivals'Arrive_1.count()) ^
      " " ^
      (Job.mkstr job) ^ "\n"
    | obsBindElem _ = ""
  in
    obsBindElem bindelem
  end

```

Σχήμα 15

Το Σχήμα 15 παρουσιάζει τη μορφή που πρέπει να έχει η συνάρτηση παρατήρησης. Για κάθε νέα εργασία, το πρώτο τμήμα της (αλφαριθμητικής) πληροφορίας που καταγράφεται στο αρχείο είναι ένας μετρητής που δείχνει τον αριθμό εργασιών που έχουν φτάσει μέχρι στιγμής στο σύστημα. Η πληροφορία αυτή εξάγεται από τη συνάρτηση *Count\_trans\_occur\_Arrivals'Arrive\_1* που έχει οριστεί στις δηλώσεις του μοντέλου. Η συνάρτηση *Int.toString()* μετατρέπει μια ακέραια αριθμητική τιμή σε αλφαριθμητικό. Το δεύτερο μέρος της πληροφορίας που καταγράφεται είναι ο (αλφαριθμητικός) χαρακτηρισμός της νέας εργασίας, όπως αυτός έχει οριστεί επίσης στις δηλώσεις του μοντέλου. Η συνάρτηση *.mkstr* χρησιμοποιείται για να εκφράσει το χαρακτηρισμό της εργασίας ως αλφαριθμητική έκφραση.

Όταν διακοπεί η προσομοίωση, υποτίθεται ότι ο χρήστης επιθυμεί να προσθέσει μερικές πληροφορίες στο αρχείο ως συμπέρασμα. Αυτό επιτυγχάνεται τροποποιώντας τη συνάρτηση διακοπής του εργαλείου παρακολούθησης. Η μορφή που μπορεί να έχει η συνάρτηση αυτή φαίνεται στο Σχήμα 16:

```
fun stop () =  
  "\n\nΑ total of "  
  Int.toString(Count_trans_occur_Arrivals'Arrive_1.count()) ^  
  " jobs arrived during the "^(Int.toString(step())) ^  
  " steps of this simulation.\n"
```

Σχήμα 16

Οι πληροφορίες που καταγράφονται από την παραπάνω συνάρτηση διακοπής παρουσιάζουν το συνολικό αριθμό εργασιών που έφτασαν στο σύστημα κατά τη διάρκεια της προσομοίωσης, όπως επίσης και το συνολικό αριθμό βημάτων προσομοίωσης που εκτελέστηκαν. Η συνάρτηση *step()* είναι μια από τις συναρτήσεις του προσομοιωτή και χρησιμοποιείται για να εξάγονται αριθμητικές πληροφορίες σχετικά με τα βήματα προσομοίωσης.

Στα εργαλεία παρακολούθησης καταγραφής σε αρχείο, η συνάρτηση δράσης δημιουργείται και καλείται αυτόματα από το σύστημα. Συνεπώς δεν είναι ορατή από το χρήστη και δεν απαιτείται η τροποποίησή της.

#### 4.7 – Αρχεία εξόδου της ανάλυσης απόδοσης

Όταν πραγματοποιείται ανάλυση της απόδοσης ενός συστήματος με το εργαλείο CPN-Tools, δημιουργούνται διάφορα αρχεία εξόδου που περιέχουν τα στατιστικά στοιχεία της ανάλυσης:

- Αρχεία καταγραφής εργαλείων παρακολούθησης συλλογής δεδομένων
- Αναφορές απόδοσης προσομοίωσης

- Αναφορές απόδοσης επαναληπτικών προσομοιώσεων
- Αρχεία καταγραφής επαναληπτικών προσομοιώσεων
- Αρχεία διαστημάτων εμπιστοσύνης

Τα αρχεία αυτά αποθηκεύονται σε ένα φάκελο με την ονομασία “output”. Ο φάκελος αυτός δημιουργείται αυτόματα στην ίδια τοποθεσία με τα αρχεία του μοντέλου, εκτός κι αν οριστεί διαφορετικά από το χρήστη.

Εάν σε ένα μοντέλο έχουν υλοποιηθεί εργαλεία παρακολούθησης συλλογής δεδομένων, τότε είναι δυνατό οι αριθμητικές τιμές που εξάγονται από τα εργαλεία αυτά να αποθηκεύονται σε αρχεία καταγραφής (data collector log files). Για να συμβεί αυτό, θα πρέπει μέσα σε κάθε εργαλείο παρακολούθησης να είναι ενεργοποιημένη η επιλογή καταγραφής (Logging). Το περιεχόμενο των αρχείων αυτών είναι οργανωμένο σε τέσσερις στήλες δεδομένων. Η πρώτη στήλη περιλαμβάνει τις τιμές δεδομένων που παρατηρήθηκαν από το αντίστοιχο εργαλείο παρακολούθησης. Η δεύτερη στήλη είναι ένας μετρητής. Η τρίτη στήλη παρουσιάζει τα βήματα της προσομοίωσης στα οποία συλλέχθηκαν τα δεδομένα, ενώ στην τέταρτη στήλη καταγράφονται οι αντίστοιχες χρονικές τιμές του μοντέλου στις οποίες παρατηρήθηκαν τα δεδομένα αυτά. Τα αρχεία καταγραφής των εργαλείων παρακολούθησης συλλογής δεδομένων αποθηκεύονται σε ένα φάκελο με την ονομασία “logfiles”, ο οποίος περιέχεται μέσα στο φάκελο “output” της προσομοίωσης.

Οι αναφορές απόδοσης της προσομοίωσης (simulation performance reports) περιλαμβάνουν μια περίληψη των στατιστικών που υπολογίζονται από τα εργαλεία παρακολούθησης συλλογής δεδομένων (εάν έχουν υλοποιηθεί) κατά την προσομοίωση. Οι αναφορές αυτές αποθηκεύονται μόλις κληθούν οι συναρτήσεις διακοπής των αντίστοιχων εργαλείων παρακολούθησης. Τα αρχεία αναφοράς απόδοσης της προσομοίωσης είναι τύπου HTML και έχουν την ονομασία “PerfReport.html”.

Όταν εκτελούνται επαναληπτικές προσομοιώσεις (simulation replications) ενός μοντέλου που περιέχει εργαλεία παρακολούθησης συλλογής δεδομένων, τότε μερικά από τα στατιστικά που υπολογίζονται στο τέλος κάθε προσομοίωσης χρησιμοποιούνται για να εξαχθούν πιο

αξιόπιστα στατιστικά στοιχεία, τα οποία βασίζονται στα δεδομένα των ανεξάρτητων επαναληπτικών προσομοιώσεων. Αυτά τα στοιχεία τοποθετούνται σε αρχεία καταγραφής επαναληπτικών προσομοιώσεων (replication log files). Για κάθε εργαλείο παρακολούθησης συλλογής δεδομένων καταγράφονται οι εξής τιμές:

- μετρητής (count)
- ελάχιστη τιμή (minimum)
- μέγιστη τιμή (maximum)
- μέση τιμή (average)
- άθροισμα (sum) (μόνο για στατιστικά που δεν περιέχουν χρόνο)

Για τις επαναληπτικές προσομοιώσεις δημιουργείται ένας φάκελος με την ονομασία “gers” μέσα στο φάκελο “output”. Τα αρχεία καταγραφής των επαναληπτικών προσομοιώσεων τοποθετούνται σε ένα φάκελο ονόματι “logfiles”, ο οποίος περιέχεται μέσα στο φάκελο “gers”.

Οι αναφορές απόδοσης επαναληπτικών προσομοιώσεων (replication performance reports) περιέχουν στατιστικά που υπολογίζονται για τα δεδομένα που βρίσκονται στα αρχεία καταγραφής επαναληπτικών προσομοιώσεων. Τα διαστήματα εμπιστοσύνης που περιλαμβάνονται στις αναφορές απόδοσης επαναληπτικών προσομοιώσεων είναι συνήθως πιο ακριβή από τα αντίστοιχα των αναφορών απόδοσης προσομοίωσης. Τα αρχεία αναφορών απόδοσης των επαναληπτικών προσομοιώσεων είναι τύπου HTML, έχουν την ονομασία “PerfReportIID.html” και τοποθετούνται μέσα στο φάκελο “gers”.

Τα διαστήματα εμπιστοσύνης, που μπορούν να βρεθούν στις αναφορές απόδοσης επαναληπτικών προσομοιώσεων, αποθηκεύονται επίσης ως απλά αρχεία κειμένου (confidence interval files). Μπορούν να υπολογιστούν τρία διαφορετικά διαστήματα εμπιστοσύνης: 90%, 95% και 99%. Το περιεχόμενο των αρχείων αυτών είναι οργανωμένο σε στήλες. Η πρώτη στήλη περιλαμβάνει την ονομασία του εργαλείου παρακολούθησης συλλογής δεδομένων και το στατιστικό μέτρο για το οποίο υπολογίστηκε το διάστημα εμπιστοσύνης. Η δεύτερη στήλη εκφράζει το επίπεδο του διαστήματος εμπιστοσύνης, δηλαδή 90, 95, ή 99. Η τρίτη στήλη παρουσιάζει το πλήθος των τιμών για τις οποίες υπολογίστηκαν η μέση τιμή και το διάστημα εμπιστοσύνης. Η τέταρτη στήλη είναι το μέγεθος του μισού του

διαστήματος εμπιστοσύνης. Οι δύο τελευταίες στήλες εκφράζουν αντίστοιχα το κατώτερο και το ανώτερο όριο του διαστήματος εμπιστοσύνης που υπολογίστηκε. Τα αρχεία των διαστημάτων εμπιστοσύνης έχουν την ονομασία “confidenceintervalsX.txt”, όπου το X είναι ίσο με 90, 95, ή 99, ανάλογα με το επίπεδο του αντίστοιχου διαστήματος εμπιστοσύνης. Τα αρχεία αυτά αποθηκεύονται μέσα στο φάκελο “gers”.

## ΕΝΟΤΗΤΑ 5<sup>η</sup>

### ΑΝΑΛΥΣΗ ΑΠΟΔΟΣΗΣ ΔΙΑΔΙΚΑΣΙΑΣ ΡΟΗΣ ΕΡΓΑΣΙΩΝ ΜΕ ΠΕΡΙΟΡΙΣΜΕΝΟΥΣ ΠΟΡΟΥΣ ΧΡΗΣΙΜΟΠΟΙΩΝΤΑΣ ΧΡΩΜΑΤΙΣΜΕΝΑ ΔΙΚΤΥΑ ΡΕΤΡΙ

#### 5.1 – Εισαγωγή

Στην Ενότητα αυτή παρουσιάζεται η αφαιρετική αναπαράσταση μιας πραγματικής επιχειρησιακής διαδικασίας. Η διαδικασία αυτή θεωρείται ως ένα σύνολο από αλληλοσυσχετιζόμενες εργασίες, όπου ένας περιορισμένος αριθμός πόρων είναι διαθέσιμος να συμμετέχει στην εκτέλεση της συνολικής διεργασίας. Ενώ μπορεί να υπάρχουν μερικοί πόροι ικανοί να εκτελέσουν ένα δοθέν αντικείμενο εργασίας, η επεξεργασία του αντικειμένου εργασίας ανατίθεται σε ακριβώς έναν από αυτούς. Οι πόροι είναι ομαδοποιημένοι σε ρόλους, ώστε να διευκολύνεται η ανάθεσή τους σε αντικείμενα εργασίας.

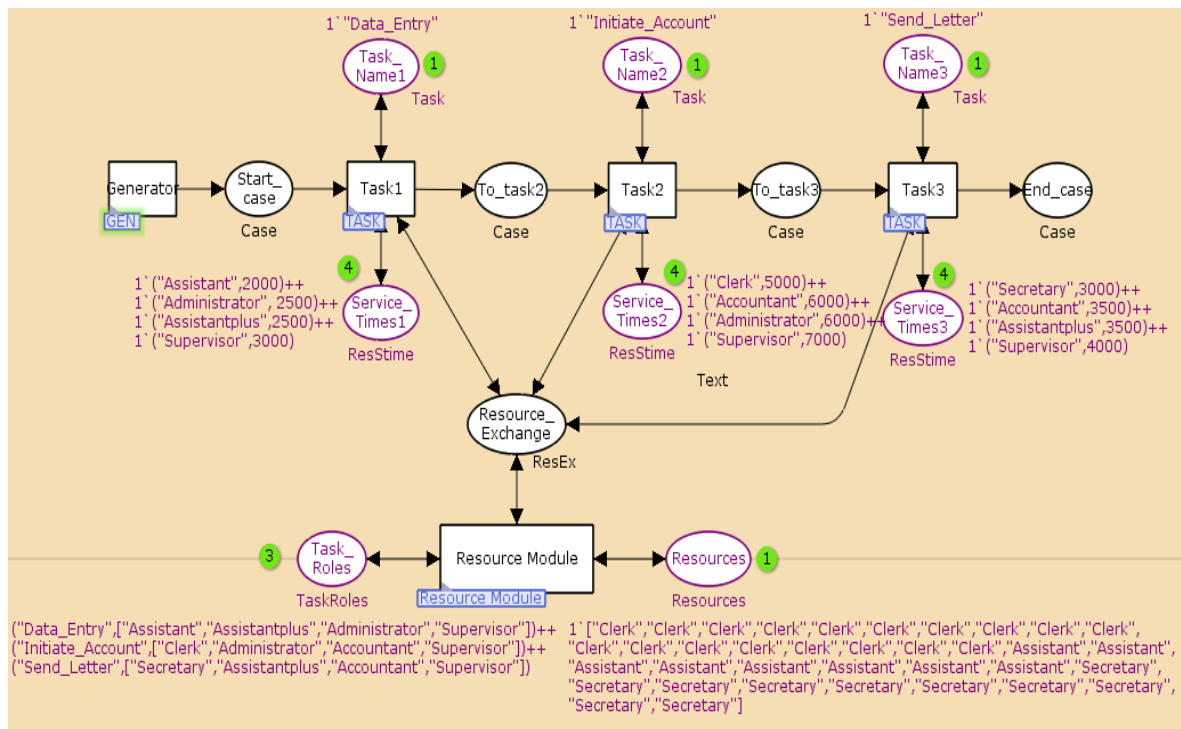
Σκοπός της Ενότητας αυτής είναι να παρουσιάσει μια προσέγγιση για την ανάλυση μιας τέτοιας διεργασίας που περιορίζεται από πόρους, χρησιμοποιώντας εργαλεία παρακολούθησης. Το μοντέλο που παρουσιάζεται και αναλύεται στη συνέχεια, καθώς και τα εργαλεία παρακολούθησης που υλοποιούνται, δημιουργήθηκαν με το εργαλείο CPN-Tools.

#### 5.2 – Παρουσίαση του μοντέλου

Στο Σχήμα 1 παρουσιάζεται το ανώτερο επίπεδο του μοντέλου ροής εργασιών ([5]). Τα διαφορετικά μέρη που αποτελούν τη διεργασία είναι η μονάδα δημιουργίας περιπτώσεων (*Generator*), η διαδικασία που αποτελείται

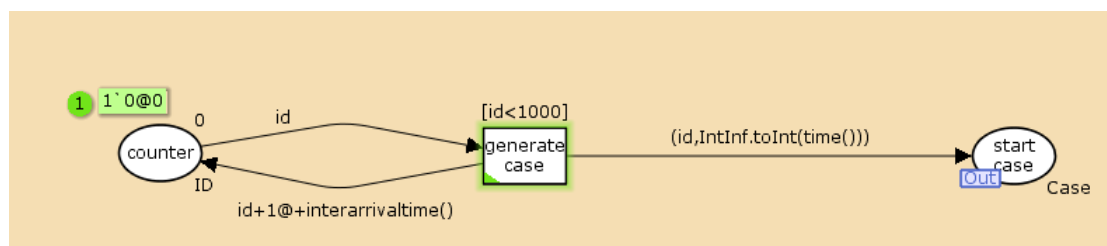


από τρία σειριακά στάδια (*Task1*, *Task2* και *Task3*) και η μονάδα διαχείρισης των πόρων (*Resource Module*) που κατανέμει τους πόρους στη διεργασία.



Σχήμα 1

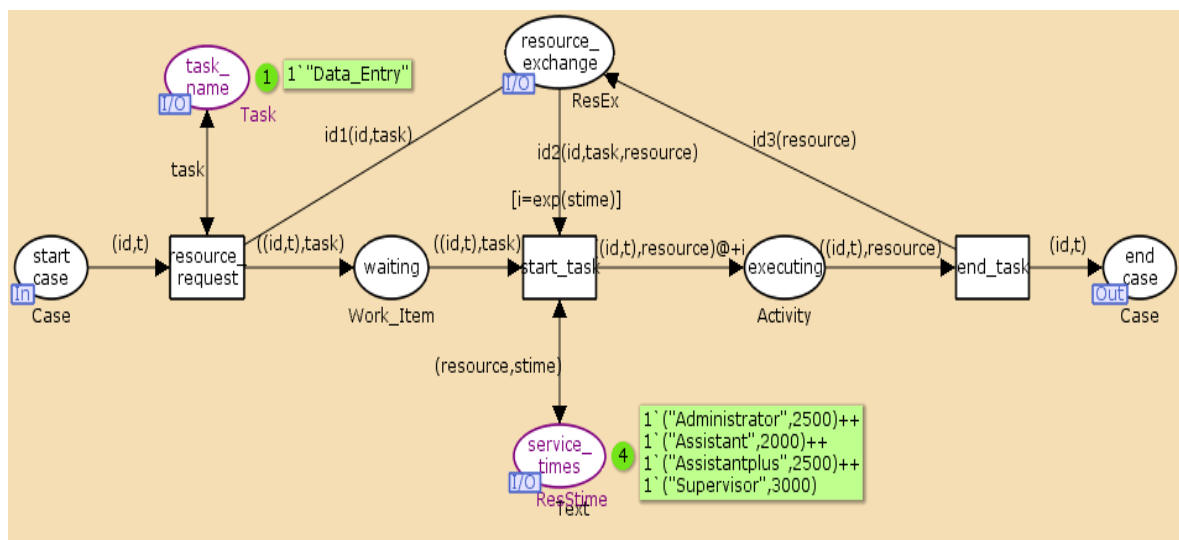
Η μονάδα περιπτώσεων παράγει περιπτώσεις με τις οποίες τροφοδοτείται η διαδικασία. Κάθε περίπτωση που παράγεται αποτελείται από έναν αύξοντα αναγνωριστικό αριθμό (*id*) και μια ακέραια τιμή που χαρακτηρίζει το χρόνο που αυτή εισήλθε στο σύστημα. Στο Σχήμα 2 φαίνεται η υπο-σελίδα της μονάδας δημιουργίας των περιπτώσεων.



Σχήμα 2

Η διαδικασία αποτελείται από τρία στάδια (εργασίες) που δημιουργήθηκαν έτσι ώστε η επεξεργασία να μπορεί να γίνεται σειριακά αλλά και παράλληλα. Για τις τρεις εργασίες έχει υλοποιηθεί ένα ενιαίο υποσύστημα, το οποίο προσδίδει σε κάθε εργασία την ίδια δομή και λειτουργικότητα. Η διαφοροποίηση μεταξύ των εργασιών γίνεται στο ανώτερο επίπεδο του μοντέλου, όπου ορίζονται ξεχωριστές παράμετροι (μεταβλητές). Για κάθε εργασία το όνομα προστίθεται ως ένδειξη στην αντίστοιχη θέση *Task\_Name*, ενώ οι χρόνοι επεξεργασίας (ένας για κάθε ρόλο που μπορεί να εκτελέσει την εργασία) προστίθενται ως ένδειξη στην αντίστοιχη θέση *Service\_Times*. Τα υπο-συστήματα των εργασιών και της μονάδας διαχείρισης πόρων επικοινωνούν και ανταλλάσσουν πόρους μεταξύ τους μέσω της θέσης *Resource\_Exchange*. Οι ρόλοι που σχετίζονται με κάθε εργασία καθορίζονται στη θέση *Task\_Roles*, ενώ οι διαθέσιμοι ρόλοι τοποθετούνται στη θέση *Resources*. Οι πόροι ονομάζονται ανάλογα με το ρόλο τους, αφού ο ρόλος είναι το μοναδικό χαρακτηριστικό που καθορίζει την ανάθεση πόρων σε εργασίες.

Όταν ένα αντικείμενο εργασίας ρέει μέσα στη διαδικασία, οι χρόνοι εξυπηρέτησης μπορεί να διαφέρουν ανά εργασία ενώ εξαρτώνται και από τον τύπο του πόρου που εκτελεί την εργασία. Καθεμία από τις εργασίες έχει την ίδια λειτουργία με τις υπόλοιπες, με τη διαφορά ότι χρησιμοποιεί διαφορετικές τιμές χρόνων εξυπηρέτησης και διαφορετικούς τύπους πόρων.

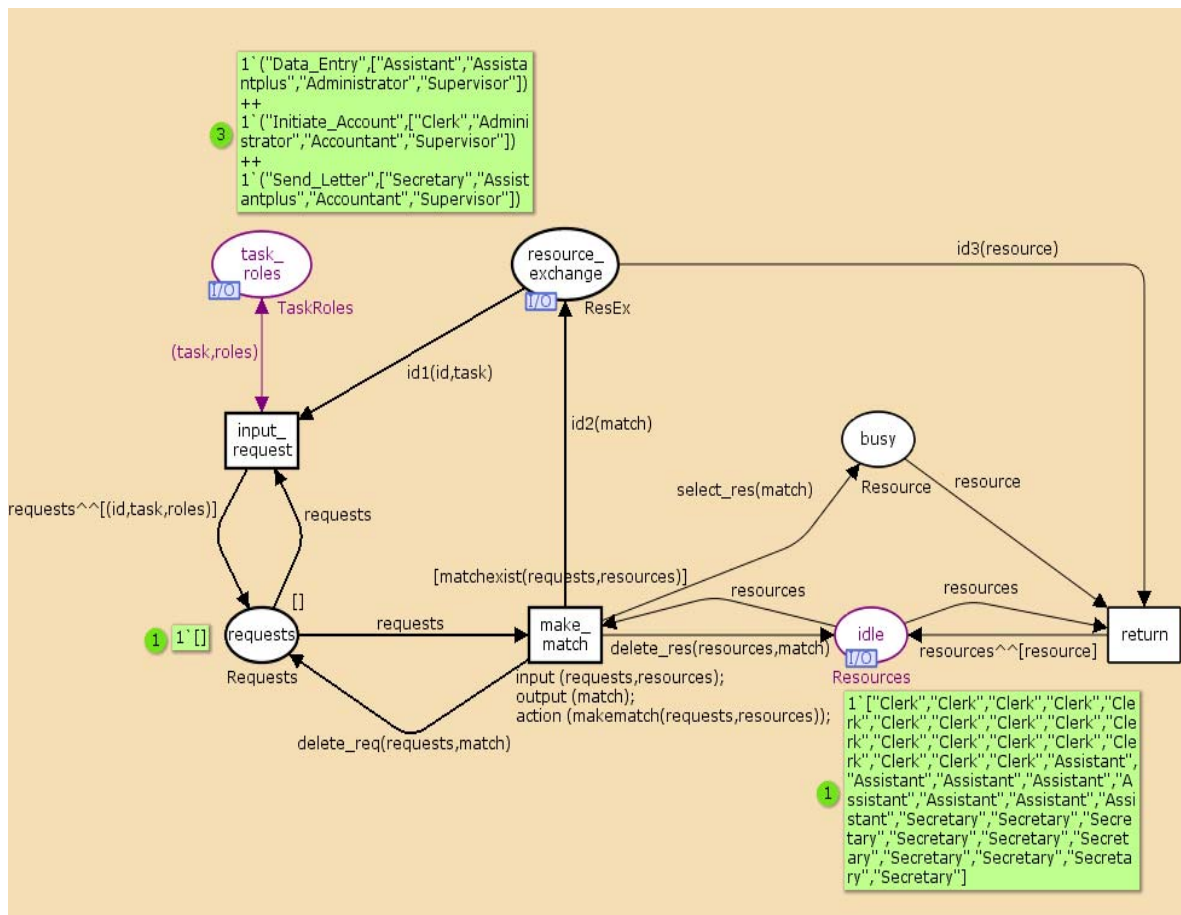


Σχήμα 3

Στο Σχήμα 3 φαίνεται η υπο-σελίδα μιας εργασίας. Η επικοινωνία της εργασίας με τη μονάδα διαχείρισης πόρων πραγματοποιείται μέσω της θέσης *resource\_exchange*. Ο σκοπός της επικοινωνίας αυτής είναι να δεσμευθεί ένας κατάλληλος πόρος για να επεξεργαστεί ένα συγκεκριμένο αντικείμενο εργασίας. Μετά την ολοκλήρωση της επεξεργασίας, ο πόρος επιστρέφει στη μονάδα διαχείρισης πόρων. Η διαδικασία που ακολουθείται μέσα σε μια εργασία ξεκινά με την είσοδο μιας νέας περίπτωσης στη θέση *start\_case*. Το όνομα της εργασίας, που είναι αποθηκευμένο στη θέση *task\_name*, προστίθεται στα χαρακτηριστικά της περίπτωσης. Αυτό γίνεται επειδή το όνομα της εργασίας είναι απαραίτητο για την απαίτηση ενός πόρου. Όταν προστίθεται το όνομα της εργασίας, η περίπτωση μετατρέπεται σε αντικείμενο εργασίας που πρέπει να εκτελεστεί. Ένα αίτημα για πόρο, που περιλαμβάνει το αναγνωριστικό της περίπτωσης και το όνομα της εργασίας, αποστέλλεται στη μονάδα διαχείρισης πόρων μέσω της θέσης *resource\_exchange*. Το αντικείμενο εργασίας αναμένει στη θέση *waiting* μέχρι ένα αντικείμενο εργασίας με το αναγνωριστικό περίπτωσης και το ίδιο όνομα εργασίας (καθώς και το δεσμευμένο πόρο) να γίνει διαθέσιμο στη θέση *resource\_exchange*. Ο μέσος χρόνος που απαιτείται από τον πόρο για να ολοκληρώσει την επεξεργασία του αντικειμένου εργασίας λαμβάνεται από τη θέση *service\_times*. Όταν ο χρόνος αυτός παρέλθει, ο πόρος επιστρέφει στη μονάδα διαχείρισης πόρων (μέσω της θέσης *resource\_exchange*) και η περίπτωση εγκαταλείπει την εργασία μέσω της θέσης *end\_case*.

Η διαδικασία που ακολουθείται μέσα στη μονάδα διαχείρισης πόρων ξεκινά όταν γίνεται αίτημα για κάποιο πόρο. Τα αιτήματα για πόρους λαμβάνονται στη θέση *resource\_exchange* και περιέχουν πληροφορίες για τα αναγνωριστικά των περιπτώσεων και τα ονόματα των εργασιών, αφού τα στοιχεία αυτά είναι απαραίτητα για να επιστραφούν τα κατάλληλα αιτήματα στα κατάλληλα αντικείμενα εργασίας. Στη μετάβαση *input\_request* προστίθενται στο αίτημα οι ρόλοι που μπορούν να εκτελέσουν το αντικείμενο εργασίας. Στη συνέχεια, το αίτημα τοποθετείται στο τέλος μιας λίστας αιτημάτων. Από εκεί, τα αιτήματα αντιστοιχίζονται με τους διαθέσιμους πόρους ξεκινώντας από την κορυφή της λίστας των αιτημάτων. Η αντιστοίχιση αυτή γίνεται βάσει του συνδυασμού του αναγνωριστικού της περίπτωσης και του ονόματος της εργασίας. Οι διαθέσιμοι πόροι είναι τοποθετημένοι στη θέση

*idle*. Εάν μπορεί να γίνει αντιστοίχιση, τότε ενεργοποιείται η μετάβαση *make\_match* και γίνεται προσπάθεια να αντιστοιχιστεί το πρώτο αίτημα της λίστας αιτημάτων με ένα διαθέσιμο πόρο. Όταν γίνει η αντιστοίχιση, ο κατάλληλος πόρος μεταφέρεται στη θέση *resource\_exchange* και στη συνέχεια στην αντίστοιχη εργασία. Το αίτημα και ο πόρος που αντιστοιχίστηκαν διαγράφονται από τη λίστα αιτημάτων και τη λίστα πόρων αντίστοιχα. Οι απασχολημένοι πόροι τοποθετούνται στη θέση *busy\_res*. Από την κορυφή έως το τέλος, η λίστα αιτημάτων επαληθεύεται και γίνονται οι ανάλογες αντιστοιχίσεις. Αιτήματα για τα οποία δεν υπάρχει διαθέσιμος πόρος παραμένουν στη λίστα αιτημάτων, ενώ όταν ένας πόρος ολοκληρώσει μια εργασία τοποθετείται πίσω στη λίστα πόρων μέσω της θέσης *resource\_exchange* και της μετάβασης *return*. Στο Σχήμα 4 φαίνεται η υποσελίδα της μονάδας διαχείρισης πόρων.



Σχήμα 4

Το Σχήμα 5 παρουσιάζει τις δηλώσεις των συνόλων χρωμάτων και συναρτήσεων του μοντέλου.

```

▼ Declarations
  ▼ Standard declarations
    ▼ colset E = with e;
    ▼ colset ET = E timed;
    ▼ colset I = int;
    ▼ colset TI = I timed;
    ▼ colset BOOL= bool;
    ▼ colset S = string;
  ▼ Initial declarations
    ▼ colset ID = int timed;
    ▼ var id:ID;
    ▼ colset T = int;
    ▼ colset Case = product ID * T timed;
    ▼ var i:I;
    ▼ var t:T;
    ▼ var c:Case;
    ▼ var s:S;
  ▼ Initial settings
    ▼ fun interarrivaltime() =
      round(exponential(1.0/375.0));
  ▼ Resource&Task declarations
    ▼ colset Task = S; (*taskName*)
    ▼ var task:Task;
    ▼ colset Work_Item = product Case * Task;
    ▼ colset Stime = I;
    ▼ var stime:Stime;
    ▼ colset Prerequisite = product ID * Task;
    ▼ colset Role = S;
    ▼ colset Roles = list Role;
    ▼ var roles:Roles;
    ▼ colset TaskRoles = product Task * Roles;
    ▼ colset Request = product ID * Task * Roles;
    ▼ var request:Request;
    ▼ colset Requests = list Request;
    ▼ var requests:Requests;
    ▼ colset Resource = Role;
    ▼ var resource:Resource;
    ▼ colset Resources = list Resource;
    ▼ var resources:Resources;
    ▼ colset Match = product ID * Task * Resource;
    ▼ var match:Match;
    ▼ colset ResStime = product Resource * Stime;
    ▼ colset Activity= product Case * Resource timed;
    ▼ colset ResEx = union id1:Prerequisite+id2:Match+id3:Resource;

  ▼ Resource&Task functions
    ▼ fun select_role (id:ID,task:Task,role:Role, [:Resources) = [ ]
      select_role(id:ID,task:Task,role:Role,hd::tl:Resources) =
      if hd=role then (id,task,hd:Resource)::nil else select_role(id,task,role,tl);
    ▼ fun select_req(id:ID,task:Task,[:Roles,resources:Resources) = [ ]
      select_req(id:ID,task:Task,hd::tl:Roles,resources:Resources) =
      if select_role(id,task,hd,resources) <> [ ]
      then select_role(id,task,hd,resources) else select_req(id,task,tl,resources);
    ▼ fun select(:Requests,resources:Resources) = [ ]
      select (hd::tl,resources:Resources) =
      if select_req(#1 hd, #2 hd, #3 hd,resources) <> [ ]
      then select_req(#1 hd,#2 hd, #3 hd,resources) else select (tl,resources);
    ▼ fun matchexist(req,res) = select(req,res)<>[ ];
    ▼ fun makematch(req,res) = hd(select(req,res));
    ▼ fun select_res(match:Match)
      = #3 match;
    ▼ fun delete_res(:Resources,match:Match) = [ ]
      delete_res(hd::tl:Resources,match:Match) =
      if hd = #3 match then tl else hd::delete_res(tl,match);
    ▼ fun delete_req(:Requests,match:Match) = [ ]
      delete_req(hd::tl:Requests,match:Match) =
      if #1 hd = #1 match then tl else hd::delete_req(tl,match);
    ▼ fun exp(stime) =round(exponential(1.0/Real.fromInt(stime)));
  
```

Σχήμα 5

Η εφαρμογή πάνω στην οποία βασίζεται το μοντέλο αφορά το άνοιγμα ενός τραπεζικού λογαριασμού. Μια σύντομη περιγραφή της τραπεζικής αυτής διαδικασίας είναι η εξής: όταν γίνεται μια αίτηση για άνοιγμα ενός νέου λογαριασμού, τα δεδομένα εισάγονται στο σύστημα. Στη συνέχεια, ο νέος λογαριασμός ενεργοποιείται και, τέλος, μια επιστολή ειδοποίησης αποστέλλεται στον πελάτη.

Στο βασικό μοντέλο (Σχήμα 1), κάθε εργασία διαθέτει δύο θέσεις που καθορίζουν τις σχετικές μεταβλητές που τη χαρακτηρίζουν. Το όνομα κάθε εργασίας αναγράφεται ως ένδειξη στην αντίστοιχη θέση *Task\_Name*, ενώ οι

χρόνοι εξυπηρέτησης τοποθετούνται στις αντίστοιχες θέσεις *Service\_Times*. Όπως φαίνεται, κάθε ρόλος χρειάζεται διαφορετικό χρονικό διάστημα για να ολοκληρώσει μια εργασία. Η μονάδα διαχείρισης πόρων διαθέτει επίσης δύο καθοριστικές θέσεις. Η θέση *Task\_Roles* περιγράφει για κάθε εργασία ποιό ρόλοι μπορούν να την εκτελέσουν. Οι ρόλοι είναι ταξινομημένοι από τους εξειδικευμένους προς τους πιο γενικευμένους. Ένας εξειδικευμένος πόρος μπορεί να εκτελέσει μια συγκεκριμένη εργασία, ενώ ένας γενικευμένος πόρος έχει τη δυνατότητα να εκτελέσει περισσότερες εργασίες. Η θέση *Resources* περιέχει τους πόρους που είναι διαθέσιμοι για επεξεργασία. Στη συγκεκριμένη υλοποίηση του μοντέλου, χρησιμοποιούνται 36 εξειδικευμένοι πόροι: 18 υπάλληλοι (*clerks*), 8 βοηθοί (*assistants*) και 10 γραμματείς (*secretaries*).

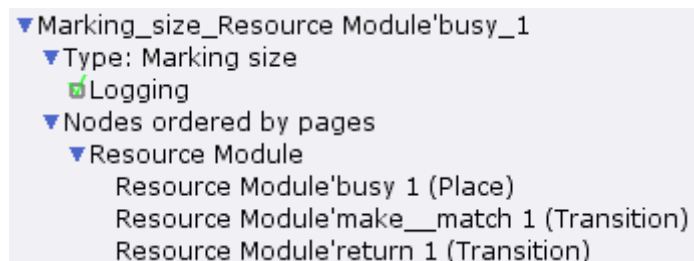
### **5.3 – Ανάλυση της απόδοσης του μοντέλου με εργαλεία παρακολούθησης**

Στο σημείο αυτό αναπτύσσονται τρία διαφορετικά εργαλεία παρακολούθησης, τα οποία εξάγουν πολύτιμες πληροφορίες για την ανάλυση της απόδοσης του συστήματος.

#### **5.3.1 – Μέσος αριθμός απασχολημένων πόρων κατά την προσομοίωση**

Το εργαλείο παρακολούθησης που αναπτύσσεται εδώ χρησιμοποιείται για να υπολογίζει το μέσο αριθμό πόρων που είναι ταυτόχρονα απασχολημένοι εκτελώντας κάποια εργασία κατά τη διάρκεια της προσομοίωσης. Αυτό το εργαλείο παρακολούθησης είναι τύπου μεγέθους ενδείξεων (*marking size monitor*) και αποτελεί μια υποκατηγορία των εργαλείων παρακολούθησης συλλογής δεδομένων. Τα εργαλεία παρακολούθησης του τύπου αυτού παρακολουθούν τη ροή ενδείξεων στις

θέσεις όπου εφαρμόζονται και εξαγουν δεδομένα σχετικά με το πλήθος των ενδείξεων που βρίσκονται στις θέσεις αυτές.



Σχήμα 6

Στο Σχήμα 6 δίνεται η μορφή του εργαλείου παρακολούθησης. Επειδή το μέτρο απόδοσης που πρέπει να υπολογιστεί αφορά τους απασχολημένους πόρους, είναι φανερό ότι το εργαλείο παρακολούθησης θα εφαρμοστεί στη θέση *busy* που βρίσκεται στην υπο-σελίδα της μονάδας διαχείρισης πόρων (*Resource Module*). Στο Σχήμα 6, πέρα από τη θέση *busy*, το εργαλείο παρακολούθησης φαίνεται ότι έχει συσχετιστεί και με τις μεταβάσεις *make\_match* και *return*, που είναι γειτονικές της θέσης *busy*. Αυτό συμβαίνει για τον εξής λόγο: αν το εργαλείο παρακολούθησης είχε σχετιστεί μόνο με τη θέση *busy*, τότε σε κάθε βήμα προσομοίωσης για τη θέση αυτή θα εκτελούνταν ένας υπολογισμός του μεγέθους των ενδείξεων. Αυτό θα ήταν άσκοπο, αφού θα εκτελούνταν πολλοί περιττοί υπολογισμοί. Προσθέτοντας στο εργαλείο παρακολούθησης τις γειτονικές μεταβάσεις της θέσης *busy*, ο χρήστης εξασφαλίζει ότι υπολογισμοί του μεγέθους των ενδείξεων θα εκτελούνται μόνο όταν εκτελείται κάποια από τις δύο μεταβάσεις *make\_match* και *return*. Με άλλα λόγια, θα εκτελείται ένας υπολογισμός των ενδείξεων μόνο όταν θα συμβαίνει κάποια αλλαγή στις ενδείξεις αυτές.

Στα εργαλεία παρακολούθησης μεγέθους ενδείξεων, καμία από τις συναρτήσεις παρακολούθησης δεν είναι ορατή από το χρήστη. Αυτό συμβαίνει επειδή ο ρόλος τους είναι πολύ συγκεκριμένος και δεν απαιτείται καμία τροποποίηση από το μέρος του χρήστη. Η μόνη ενέργεια που απαιτείται από το χρήστη είναι να δημιουργήσει ένα εργαλείο

παρακολούθησης του τύπου αυτού και απλά να το εφαρμόσει στο επιθυμητό μέρος του μοντέλου.

### 5.3.2 – Μέσος χρόνος διεκπεραίωσης περιπτώσεων

Σκοπός αυτού του εργαλείου παρακολούθησης είναι να υπολογίζει το μέσο χρόνο ολοκλήρωσης των περιπτώσεων (throughput time), από τη στιγμή που εισάγονται στο σύστημα μέχρι τη στιγμή που ολοκληρώνεται η επεξεργασία τους. Το εργαλείο παρακολούθησης που αναπτύσσεται είναι τύπου συλλογής δεδομένων και η δομή του παρουσιάζεται στο Σχήμα 7.

```
▼ Flow time
  ▼ Type: Data collection
     Timed
     Logging
  ▼ Nodes ordered by pages
    ▼ TASK
      TASK'end_task 1 (Transition)
    ▼ TASK
      TASK'end_task 2 (Transition)
    ▼ TASK
      TASK'end_task 3 (Transition)
  ▼ Predicate
    fun pred (bindelem) =
    let
      fun predBindElem (TASK'end_task (3, {id, resource, t})) = true
        | predBindElem _ = false
    in
      predBindElem bindelem
    end
  ▼ Observer
    fun obs (bindelem) =
    let
      fun obsBindElem (TASK'end_task (3, {id, resource, t})) = IntInf.toInt(time())-t
        | obsBindElem _ = ~1
    in
      obsBindElem bindelem
    end
  ▼ Init function
    fun init () =
      NONE
  ▼ Stop
    fun stop x = NONE
```

Σχήμα 7



Το μέτρο απόδοσης που υπολογίζεται στην περίπτωση αυτή είναι ο μέσος χρόνος στον οποίο ολοκληρώνεται η επεξεργασία μιας περίπτωσης. Είναι φανερό, λοιπόν, ότι το εργαλείο παρακολούθησης πρέπει να εφαρμοστεί στα μέρη του μοντέλου που υποδηλώνουν την ολοκλήρωση επεξεργασίας μιας περίπτωσης. Συνεπώς, η παρακολούθηση θα πραγματοποιηθεί στις μεταβάσεις *end\_task*, που βρίσκονται στις αντίστοιχες υπο-σελίδες *Task(1)*, *Task(2)* και *Task(3)* των εργασιών της διαδικασίας.

Η παρακολούθηση είναι απαραίτητο να πραγματοποιείται κάθε φορά που ολοκληρώνεται η επεξεργασία μιας περίπτωσης, δηλαδή κάθε φορά που εκτελείται μια από τις μεταβάσεις *end\_task*. Άρα, η συνάρτηση κριτηρίου που παράγεται αυτόματα κατά τη δημιουργία του εργαλείου παρακολούθησης δε χρειάζεται κάποια τροποποίηση, αφού το κριτήριο ικανοποιείται κάθε φορά που ενεργοποιείται κάποια από τις μεταβάσεις *end\_task*.

Εφόσον ικανοποιηθεί η συνάρτηση κριτηρίου, καλείται η συνάρτηση παρατήρησης, η οποία πρέπει να εξάγει τα απαραίτητα δεδομένα. Στην προκειμένη περίπτωση, πρέπει να υπολογιστεί ο χρόνος ολοκλήρωσης μιας περίπτωσης. Αυτό επιτυγχάνεται υπολογίζοντας τη διαφορά

$$\text{IntInf.toInt}(\text{time}())-t.$$

Η συνάρτηση *time()* επιστρέφει τον τρέχων χρόνο του μοντέλου, ενώ η συνάρτηση *IntInf.toInt(time())* μετατρέπει τον χρόνο αυτό σε μια ακέραια τιμή. Η μεταβλητή *t*, που ορίζεται στις δηλώσεις του μοντέλου, δίνει τον (ακέραιο) χρόνο κατά τον οποίο η περίπτωση εισήλθε στο σύστημα. Συνεπώς, η παραπάνω διαφορά υπολογίζει το χρόνο ολοκλήρωσης της περίπτωσης μέσα σε κάθε εργασία.

Πριν από την προσομοίωση, αλλά και αφού αυτή διακοπεί, δεν είναι απαραίτητο να εξαχθούν δεδομένα από το μοντέλο. Άρα, οι συναρτήσεις αρχικοποίησης και διακοπής δεν είναι ανάγκη να τροποποιηθούν.

### 5.3.3 – Μέση χρησιμοποίηση πόρων

Αυτό το εργαλείο παρακολούθησης υλοποιείται για να υπολογίζει τη μέση χρησιμοποίηση πόρων για την επεξεργασία περιπτώσεων. Ο τύπος του εργαλείου παρακολούθησης είναι συλλογής δεδομένων και η δομή του παρουσιάζεται στο Σχήμα 8.

```
▼ Resource Rates
  ▼ Assistant Rate
    ▼ Type: Data collection
       Timed
       Logging
    ▼ Nodes ordered by pages
      ▼ Resource Module
        Resource Module'busy_1 (Place)
        Resource Module'make__match 1 (Transition)
      ▼ Predicate
        fun pred (bindelem,
                  Resource_Module'busy_1_mark : Resource ms) =
          let
            fun predBindElem (Resource_Module'make__match (1,{match,requests,resources})) =
              if (#3 match) = "Assistant" then true else false
              | predBindElem _ = false
          in
            predBindElem bindelem
          end
      ▼ Observer
        fun obs (bindelem,
                 Resource_Module'busy_1_mark : Resource ms) =
          let
            fun obsBindElem (Resource_Module'make__match (1,{match,requests,resources})) =
              size (filter (fn (resource) => resource="Assistant") Resource_Module'busy_1_mark)
              | obsBindElem _ = ~1
          in
            obsBindElem bindelem
          end
      ▼ Init function
        fun init (Resource_Module'busy_1_mark : Resource ms) =
          NONE
      ▼ Stop
        fun stop (Resource_Module'busy_1_mark : Resource ms) =
          NONE
```

Σχήμα 8

Το μέτρο απόδοσης που υπολογίζεται σε αυτή την περίπτωση είναι ο μέσος αριθμός πόρων που χρησιμοποιούνται για την επεξεργασία των περιπτώσεων. Ο υπολογισμός δεν αφορά το σύνολο των πόρων, αλλά εκτελείται ξεχωριστά για κάθε είδος πόρου που χρησιμοποιείται στο μοντέλο.

Επειδή ένα εργαλείο παρακολούθησης συλλογής δεδομένων εξάγει αποτελέσματα μόνο για ένα μέτρο απόδοσης, είναι απαραίτητο να δημιουργηθεί ένα εργαλείο παρακολούθησης για κάθε είδος πόρου που χρησιμοποιείται. Παρ' όλα αυτά, εδώ παρουσιάζεται η μορφή του εργαλείου παρακολούθησης για ένα είδος πόρου (*Assistant*). Η υλοποίηση για τα υπόλοιπα είδη είναι ακριβώς ίδια, με μόνη διαφορά την αλλαγή της ονομασίας του πόρου μέσα στις συναρτήσεις κριτηρίου και παρατήρησης.

Το μέρος του μοντέλου στο οποίο θα εφαρμοστεί το εργαλείο παρακολούθησης είναι η θέση *busy* και η μετάβαση *make\_match* που βρίσκονται στην υπο-σελίδα της μονάδας διαχείρισης των πόρων (*Resource Module*). Η θέση *busy* διατηρεί τους πόρους οι οποίοι είναι απασχολημένοι πάνω σε διάφορες περιπτώσεις, ενώ η μετάβαση *make\_match* ουσιαστικά τροφοδοτεί τη θέση *busy* παρέχοντας τις αντιστοιχίσεις (*matches*) των πόρων σε περιπτώσεις.

Η συνάρτηση κριτηρίου θα πρέπει να ικανοποιείται κάθε φορά που ενεργοποιείται η μετάβαση *make\_match*. Όταν συμβαίνει αυτό, μία αντιστοίχιση πραγματοποιείται και ένας πόρος αποστέλλεται στη θέση *busy*. Η αντιστοίχιση, σύμφωνα με τον ορισμό του μοντέλου, αναπαρίσταται από ένα συνδυασμό των εξής στοιχείων: του αναγνωριστικού της περίπτωσης, του ονόματος της εργασίας και του πόρου που επιλέχθηκε (*product ID\*Task\*Resource*). Η έκφραση (*#3 match*) μας επιτρέπει να αναφερθούμε στο τρίτο στοιχείο του συνδυασμού μιας αντιστοίχισης, δηλαδή στην ονομασία του πόρου που επιλέχθηκε. Στην προκειμένη περίπτωση, το κριτήριο παρακολούθησης θα πρέπει να ικανοποιείται όταν το τρίτο στοιχείο του συνδυασμού μιας αντιστοίχισης είναι *Assistant*. Άρα, η συνάρτηση κριτηρίου θα ικανοποιείται με τον τρόπο αυτό εάν προστεθεί η έκφραση

*if (#3 match)= "Assistant" then true else false.*

Όταν ικανοποιηθεί η συνάρτηση κριτηρίου, θα κληθεί η συνάρτηση παρατήρησης. Το μέγεθος που πρέπει να υπολογιστεί αφορά τον αριθμό των πόρων που βρίσκονται στη θέση *busy* και που είναι τύπου *Assistant*. Όμως, στη θέση *busy* τοποθετούνται πόροι όλων των τύπων. Οπότε, είναι αναγκαίο να φιλτράρονται οι ενδείξεις στη θέση *busy* ώστε να λαμβάνονται υπ' όψιν μόνο οι πόροι τύπου *Assistant*. Με την έκφραση

*size (filter (fn (resource) => resource = "Assistant") Resource\_Module ` busy \_1\_mark)*

επιτυγχάνεται ακριβώς το επιθυμητό αποτέλεσμα. Η έκφραση *(fn (resource) => resource = "Assistant")* είναι μια συνάρτηση προϋπόθεσης, που αληθεύει μόνο όταν το στοιχείο *resource* είναι ίσο με "Assistant". Όταν εφαρμοστεί η έκφραση *(filter (fn (resource) => resource = "Assistant") Resource\_Module ` busy \_1\_mark)*, τότε η προϋπόθεση θα εφαρμοστεί σε όλα τα στοιχεία της θέσης *busy* και θα τα φιλτράρει σύμφωνα με την προϋπόθεση. Τέλος, η συνάρτηση *size()* θα επιστρέψει τον αριθμό των στοιχείων που βρέθηκαν βάσει του φίλτρου "Assistant".

Προτού ξεκινήσει η διαδικασία της προσομοίωσης δεν είναι απαραίτητο να εξαχθούν αριθμητικά δεδομένα από το μοντέλο. Το ίδιο ισχύει και κατά τη διακοπή της προσομοίωσης. Κατά συνέπεια, οι συναρτήσεις αρχικοποίησης και διακοπής δε χρειάζονται τροποποίηση.

#### **5.4 – Στατιστικά αποτελέσματα προσομοίωσης**

Έχοντας υλοποιήσει τα απαραίτητα εργαλεία παρακολούθησης, το επόμενο βήμα για την ανάλυση της απόδοσης του συστήματος είναι η προσομοίωσή του.

Όπως έχει προαναφερθεί, η εκτέλεση της προσομοίωσης βήμα προς βήμα είναι εξαιρετικά χρονοβόρα και χρησιμεύει περισσότερο στην εύρεση σφαλμάτων κατά την κατασκευή του μοντέλου. Για το λόγο αυτό, η προσομοίωση του συστήματος πραγματοποιείται εκτελώντας πολλαπλά συνεχόμενα βήματα. Επίσης, όταν πρόκειται να μελετηθούν τα αποτελέσματα από την ανάλυση απόδοσης ενός συστήματος, μια μοναδική προσομοίωση δεν είναι αρκετή. Η συμπεριφορά του συστήματος είναι διαφορετική κάθε φορά που εκτελείται, δηλαδή κάθε φορά παράγει διαφορετικά αποτελέσματα. Αυτό σημαίνει ότι τα αριθμητικά δεδομένα που εξάγονται από μια μοναδική προσομοίωση του συστήματος δεν επαρκούν για να εκτιμηθεί σωστά η απόδοσή του. Συνεπώς, εκτελούνται πολλαπλές προσομοιώσεις του

μοντέλου ώστε να επιτευχθεί μια ακριβέστερη εκτίμηση των μέτρων απόδοσης, συνδυάζοντας τα αποτελέσματα των ξεχωριστών προσομοιώσεων.

Για να παραχθούν τα αποτελέσματα από την ανάλυση απόδοσης του συστήματος της τραπεζικής διαδικασίας που παρουσιάστηκε, εκτελούνται 100 ξεχωριστές προσομοιώσεις του μοντέλου. Κάθε προσομοίωση ολοκληρώνεται όταν επεξεργαστούν 1000 περιπτώσεις. Το εργαλείο CPN-Tools διαθέτει μια εντολή που δίνει τη δυνατότητα εκτέλεσης πολλαπλών προσομοιώσεων σειριακά. Η εντολή αυτή είναι η *CPN`Replications.nreplications 100*.

<b>Μέσος αριθμός απασχολημένων πόρων</b>	<b>Μέση τιμή</b>	<b>Κατώτερο όριο</b>	<b>Ανώτερο όριο</b>
<b>Διάστημα εμπιστοσύνης 90%</b>	25.044098	24.908136	25.180060
<b>Διάστημα εμπιστοσύνης 95%</b>	25.044098	24.881433	25.206763
<b>Διάστημα εμπιστοσύνης 99%</b>	25.044098	24.828273	25.259923

Σχήμα 9

Το Σχήμα 9 παρουσιάζει τα αποτελέσματα της ανάλυσης απόδοσης του συστήματος, όσον αφορά το μέσο αριθμό πόρων που απασχολούνται ταυτόχρονα. Όπως είναι φανερό, από τους 36 συνολικά διαθέσιμους πόρους του συστήματος, ταυτόχρονα απασχολούνται περίπου 25 από αυτούς. Οι διαφορές των στατιστικών μεταξύ των διαστημάτων εμπιστοσύνης είναι ασήμαντες.

Στη συγκεκριμένη υλοποίηση του μοντέλου χρησιμοποιούνται μόνο τρεις διαφορετικοί τύποι πόρων, ένας για κάθε εργασία. Κατά συνέπεια, δεν υπάρχουν ιδιαίτερα κριτήρια επιλογής για την αντιστοίχιση πόρων σε εργασίες, άρα ο μέσος αριθμός πόρων που απασχολούνται ταυτόχρονα εξαρτάται κυρίως από τις αφίξεις των εργασιών. Θα μπορούσαν να προστεθούν στο μοντέλο περισσότεροι τύποι πόρων για οποιαδήποτε, ή και για κάθε, εργασία. Γενικότερα, η ανάλυση απόδοσης του συστήματος χρησιμοποιώντας κάθε φορά διαφορετικούς τύπους πόρων είναι ιδιαίτερα χρήσιμη, αφού μπορεί να γίνει σύγκριση των αποτελεσμάτων κάθε παραμετροποίησης των πόρων ώστε να βρεθεί μια βέλτιστη λύση για μέγιστη χρησιμοποίησή τους.

<b>Μέσος χρόνος ολοκλήρωσης εργασιών</b>	<b>Μέση τιμή</b>	<b>Κατώτερο όριο</b>	<b>Ανώτερο όριο</b>
<b>Διάστημα εμπιστοσύνης 90%</b>	10909.364840	10823.191065	10995.538615
<b>Διάστημα εμπιστοσύνης 95%</b>	10909.364840	10806.266846	11012.462834
<b>Διάστημα εμπιστοσύνης 99%</b>	10909.364840	10772.573676	11046.156004

Σχήμα 10

Στο Σχήμα 10 καταγράφονται τα στατιστικά αποτελέσματα της ανάλυσης απόδοσης του συστήματος για το μέσο χρόνο στον οποίο ολοκληρώνονται οι περιπτώσεις. Όπως φαίνεται από την ανάλυση αυτή, κάθε εργασία ολοκληρώνεται κατά μέσο όρο σε περίπου 10909 μονάδες χρόνου.

Οι χρόνοι ολοκλήρωσης των εργασιών που παρουσιάζονται εκφράζονται ως ακέραιες τιμές χρόνου βάσει του χρόνου του μοντέλου.

Στη συγκεκριμένη υλοποίηση του μοντέλου, οι αφίξεις των περιπτώσεων πραγματοποιούνται εκθετικά. Σημαντικές διαφορές θα μπορούσαν να υπάρχουν στα στατιστικά αποτελέσματα της ανάλυσης εάν χρησιμοποιούνταν εναλλακτικές μέθοδοι κατανομής των αφίξεων. Επιπρόσθετα, για την επεξεργασία των περιπτώσεων μέσα σε κάθε εργασία χρησιμοποιούνται μόνο τρεις τύποι πόρων. Πιο συγκεκριμένα, υπάρχει ένας τύπος πόρων που ανταποκρίνεται σε καθεμία από τις τρεις εργασίες, ενώ παράλληλα οι χρόνοι επεξεργασίας είναι σταθεροί για κάθε τύπο πόρου. Μια αξιοσημείωτη διαφορά στο μέσο χρόνο ολοκλήρωσης κάθε περίπτωσης θα προέκυπτε εάν στο μοντέλο υπήρχαν περισσότεροι τύποι πόρων για κάθε εργασία, γεγονός που συνεπάγεται ότι θα υπήρχε και μεγαλύτερη ποικιλία στους χρόνους επεξεργασίας των περιπτώσεων.

<b>Μέση χρησιμοποίηση Βοηθών (Assistants)</b>	<b>Μέση τιμή</b>	<b>Κατώτερο όριο</b>	<b>Ανώτερο όριο</b>
<b>Διάστημα εμπιστοσύνης 90%</b>	6.067100	6.032625	6.101575
<b>Διάστημα εμπιστοσύνης 95%</b>	6.067100	6.025854	6.108346
<b>Διάστημα εμπιστοσύνης 99%</b>	6.067100	6.012374	6.121826

Σχήμα 11

<b>Μέση χρησιμοποίηση Υπαλλήλων (Clerks)</b>	<b>Μέση τιμή</b>	<b>Κατώτερο όριο</b>	<b>Ανώτερο όριο</b>
<b>Διάστημα εμπιστοσύνης 90%</b>	13.965540	13.884654	14.046426
<b>Διάστημα εμπιστοσύνης 95%</b>	13.965540	13.868769	14.062311
<b>Διάστημα εμπιστοσύνης 99%</b>	13.965540	13.837143	14.093937

Σχήμα 12

<b>Μέση χρησιμοποίηση Γραμματέων (Secretaries)</b>	<b>Μέση τιμή</b>	<b>Κατώτερο όριο</b>	<b>Ανώτερο όριο</b>
<b>Διάστημα εμπιστοσύνης 90%</b>	8.439750	8.397130	8.482370
<b>Διάστημα εμπιστοσύνης 95%</b>	8.439750	8.388760	8.490740
<b>Διάστημα εμπιστοσύνης 99%</b>	8.439750	8.372096	8.507404

Σχήμα 13



Τα Σχήματα 11, 12 και 13 παρουσιάζουν τα αποτελέσματα της ανάλυσης για τη μέση χρησιμοποίηση των πόρων του συστήματος, που αντίστοιχα είναι οι βοηθοί, οι υπάλληλοι και οι γραμματείς. Σύμφωνα με τα στατιστικά αποτελέσματα, από τους συνολικά 8 βοηθούς οι 6 ήταν απασχολημένοι, από τους συνολικά 18 υπαλλήλους περίπου 14 απασχολούνταν, ενώ περίπου 8 από συνολικά 10 γραμματείς ήταν επίσης απασχολημένοι με περιπτώσεις. Τα αντίστοιχα ποσοστά μέσης χρησιμοποίησης είναι: 75% για τους βοηθούς, περίπου 78% για τους υπαλλήλους και 80% για τους γραμματείς.

Στο μοντέλο χρησιμοποιούνται μόνο τρεις διαφορετικοί τύποι πόρων. Οι συγκεκριμένοι πόροι είναι εξειδικευμένοι (specialists), δηλαδή μπορούν να εκτελέσουν μόνο τις συγκεκριμένες περιπτώσεις στις οποίες εξειδικεύονται. Ο ορισμός του μοντέλου επιτρέπει να προστεθούν επίσης πιο γενικευμένοι πόροι (generalists), δηλαδή πόροι που μπορούν να επεξεργαστούν περισσότερους από έναν τύπους περιπτώσεων. Οι χρόνοι επεξεργασίας των περιπτώσεων αυξάνονται όσο πιο γενικευμένος είναι ένας πόρος, ενώ παράλληλα η προτεραιότητά του για επεξεργασία είναι μικρότερη. Η ανάλυση της απόδοσης του συστήματος θα εμφάνιζε ενδιαφέρουσες διαφορές εάν συμμετείχαν στη διαδικασία και αρκετοί πιο γενικευμένοι πόροι. Η σημαντικότερη, όμως, μεταβολή στα αποτελέσματα της ανάλυσης θα συνέβαινε εάν επιλεγόταν μια διαφορετική κατανομή των πόρων. Στη συγκεκριμένη υλοποίηση του μοντέλου, η επιλογή των πόρων γίνεται βάσει των προτεραιοτήτων μεταξύ τους. Αυτό σημαίνει ότι για μια νέα περίπτωση επιλέγεται πάντα ο πιο εξειδικευμένος πόρος, ενώ αν δεν υπάρχουν διαθέσιμοι εξειδικευμένοι πόροι τότε επιλέγεται ο αμέσως πιο γενικευμένος. Θα μπορούσαν να χρησιμοποιηθούν πολλοί εξειδικευμένοι πόροι και λιγότεροι γενικευμένοι, ή το αντίστροφο. Η ανάλυση τέτοιων περιπτώσεων και η σύγκριση των αποτελεσμάτων τους θα μπορούσαν να οδηγήσουν στην εύρεση μιας βέλτιστης αναλογίας πόρων, με την οποία θα επιδιώκεται η μέγιστη χρησιμοποίηση κάθε τύπου πόρων. Παράλληλα, μια βέλτιστη αναλογία πόρων θα πρέπει να έχει ως στόχο και τη μείωση του μέσου χρόνου

εξυπηρέτησης του συστήματος, αφού η επεξεργασία των περιπτώσεων εξαρτάται άμεσα από τη διαθεσιμότητα κατάλληλων πόρων.

## ΕΝΟΤΗΤΑ 6<sup>η</sup>

### ΣΥΜΠΕΡΑΣΜΑΤΑ

Στην εργασία αυτή παρουσιάστηκε μια προσέγγιση για την ανάλυση της απόδοσης διαδικασιών ροής εργασίας χρησιμοποιώντας χρωματισμένα δίκτυα Petri. Τα χρωματισμένα δίκτυα Petri προσφέρουν ένα πλαίσιο εργασίας ιδιαίτερα κατάλληλο για τη μοντελοποίηση και την ανάλυση συστημάτων ροής εργασίας, ενώ η εφαρμογή τους είναι αρκετά εύκολη. Με τη χρήση του εργαλείου CPN-Tools, αποδείχθηκε το πόσο εύκολο είναι να υλοποιηθούν διάφοροι τύποι εργαλείων παρακολούθησης για την υποστήριξη της διαδικασίας ανάλυσης. Η ελαστικότητα των εργαλείων παρακολούθησης καθιστά δυνατή τη χρήση τους για την ανάλυση απόδοσης ενός μοντέλου ροής εργασίας, ακόμα και όταν στο ίδιο μοντέλο εφαρμόζονται διαφορετικές περιπτώσεις ανά προσομοίωση. Στην έκδοση 2.0.0 του εργαλείου CPN-Tools έχουν προστεθεί νέες δυνατότητες για την υποστήριξη των εργαλείων παρακολούθησης, γεγονός το οποίο διευκολύνει σε μεγάλο βαθμό τη δημιουργία τους.

Έχει πραγματοποιηθεί μεγάλη μελέτη πάνω στις διαδικασίες ροής εργασίας. Τα συστήματα ροής εργασιών με περιορισμένους πόρους αποτελούν ένα ρεαλιστικό πρόβλημα, για το οποίο έχουν προταθεί διάφορες λύσεις. Οι στρατηγικές οργάνωσης που μπορούν να εφαρμοστούν στα συστήματα αυτά ποικίλουν, ενώ η ανάλυση διάφορων παραμετροποιήσεών τους κρίνεται απαραίτητη για την καλύτερη λειτουργία τους. Το σίγουρο είναι ότι στον τομέα των συστημάτων ροής εργασίας με περιορισμό πόρων οι μελέτες θα συνεχιστούν μελλοντικά, αφού τα συστήματα αυτά εκφράζουν ένα μεγάλο μέρος των πραγματικών επιχειρήσεων.

## BIBΛΙΟΓΡΑΦΙΑ

1. Lars M. Kristensen, Soren Christensen, Kurt Jensen, “The practitioner’s guide to coloured Petri nets”, CPN Group, Department of Computer Science, University of Aarhus, Denmark; 1998 Springer-Verlag
2. W.M.P. van der Aalst, “The Application of Petri Nets to Workflow Management”, Department of Mathematics and Computing Science, Eindhoven University of Technology
3. Dongsheng Liu, Jianmin Wang, Stephen C.F. Chan, Jianguang Sun, Li Zhang, “Modeling workflow processes with colored Petri nets”, Department of Computer Science and Technology, Tsinghua University, Department of Computing, The Hong Kong Polytechnic University, 2002 – Computers In Industry 49 – Elsevier
4. Lisa Wells, “Performance Analysis using Coloured Petri Nets”, PhD Dissertation, 2002, Department of Computer Science, University of Aarhus, Denmark
5. Mariska Netjes, Wil M.P. van der Aalst, Hajo A. Reijers, “Analysis of resource-constrained processes with Colored Petri Nets”, Eindhoven University of Technology
6. Christian W. Guenther, Wil M.P. van der Aalst, “Modeling The Case Handling Principles with Colored Petri Nets”, Eindhoven University of Technology
7. Irene Vanderfeesten, Wil van der Aalst, Hajo A. Reijers, “Modelling a product based workflow system in CPN tools”, Eindhoven University of Technology
8. M. Pesic, W.M.P. van der Aalst, “Modeling Work Distribution Mechanisms Using Colored Petri Nets”, Department of Technology Management, Eindhoven University of Technology

9. CPN Tools Homepage, <http://wiki.daimi.au.dk/cpntools/cpntools.wiki>
10. CPN Tools Help Files (version 2.0.0)
11. WikiPedia – The Free Online Encyclopedia, <http://www.wikipedia.org/>