

## 8.2 Petri Nets

Τα Petri Nets έχουν τα εξής πλεονεκτήματα για τη λεπτομερή περιγραφή των προδιαγραφών ενός συστήματος:

- Έχουν μια αυστηρή δομή και σύνταξη που δεν επιτρέπει παρεμπηνείς.
- Είναι γραφικά και μπορούν να σχεδιαστούν σχετικά εύκολα ειδικά αν χρησιμοποιηθεί κάποιο εργαλείο.
- Είναι απλά (έχουν πολύ απλή σύνταξη και δομή) και μπορούν να γίνουν κατανοητά και από τους περισσότερους χρήστες.
- Οι προδιαγραφές που περιγράφονται με τη χρήση Petri Nets μπορούν να ελεγχθούν.

Κατά συνέπεια, για λογισμικό μικρού μεγέθους μπορούν να χρησιμοποιηθούν ως μέρος μιας διαδικασίας ποιότητας στη φάση της ανάλυσης απαιτήσεων.

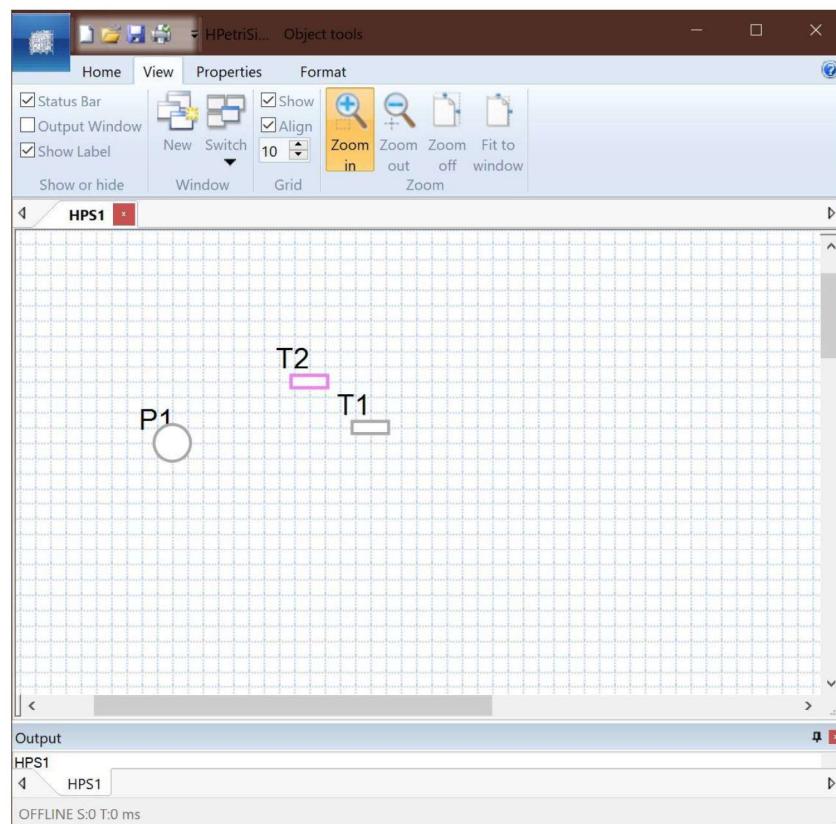
Η διαδικασία συνήθως ξεκινά με τα εξής βήματα:

1. Γίνεται μια πρώτη συζήτηση με τους χρήστες και καταγράφονται οι προδιαγραφές του συστήματος από τον αναλυτή.
2. Ακολουθεί επιβεβαίωση των προδιαγραφών από τους χρήστες (σε αυτή τη φάση μπορούν να χρησιμοποιηθούν και mock-up οθόνες για να είναι ξεκάθαρο στους χρήστες πώς θα λειτουργεί το λογισμικό).
3. Ο αναλυτής στη συνέχεια, βασισμένος στη λεκτική περιγραφή, προσπαθεί να σχεδιάσει ένα (για κάθε περίπτωση χρήσης) Petri Net που να περιγράφει αυστηρά αυτό που καταγράφηκε στη λεκτική περιγραφή.
4. Όπου η περιγραφή έχει ασάφειες που δεν μπορούν να σχεδιαστούν στο Petri Net ο αναλυτής επαναλαμβάνει τα βήματα 1 και 2 μέχρι να μπορεί να οδηγηθεί από την περιγραφή σε ένα μοναδικό Petri Net που να περιγράφει το σύστημα.
5. Το Petri Net ελέγχεται διεξοδικά για να επιβεβαιωθεί η συνέπεια με τις προδιαγραφές και επιβεβαιώνεται από τους χρήστες.
6. Η ομάδα ανάπτυξης ξεκινά την υλοποίηση βασισμένη στο τελικό Petri Net.

Σε αυτή τη λογική, τα Petri Nets χρησιμοποιούνται για να εντοπιστούν λάθη στις αρχικές ασαφείς λεκτικές προδιαγραφές. Στα πλαίσια του βιβλίου

και στα παραδείγματα που δίνουμε οι λεκτικές προδιαγραφές είναι συνήθως αρκετά σαφείς και λεπτομερείς για να μπορέσουμε να σχεδιάσουμε ορθά το αντίστοιχο δίκτυο, αλλά συνήθως αυτό δεν είναι ο κανόνας. Αντίθετα σπάνια από την αρχική περιγραφή μπορούμε να σχεδιάσουμε το δίκτυο και αυτό μας αναγκάζει να πάμε πίσω στον χρήστη και να τον υποχρεώσουμε να σκεφτεί περισσότερο την περιγραφή του λογισμικού που μας έδωσε και να καταλήξουμε σε κάτι που να μπορεί να περιγράφεται αυστηρά.

Δυστυχώς η εμπειρία στην ανάλυση απαιτήσεων έχει διδάξει ότι πολλές φορές οι χρήστες δεν θα μας δώσουν σωστή περιγραφή του λογισμικού που θέλουν να αναπτύξουμε, εκτός αν τους πιέσουμε να σκεφτούν σημεία που δεν είναι ξεκάθαρα και σε αυτό το πρόβλημα τέτοιες διαδικασίες βοηθούν σημαντικά.



**Σχήμα 8.1** Το λογισμικό HPetriSim (έκδοση 0.9)

### 8.2.1 Λογισμικό για σχεδίαση Petri Nets

Το λογισμικό που χρησιμοποιήσαμε σε αυτό το βιβλίο για τη σχεδίαση των Petri Nets είναι το HPetriSim 0.9 beta που μπορεί να βρεθεί στο GitHub τόσο το εκτελέσιμο αρχείο όσο και ο κώδικας.

Είναι ένα εργαλείο που έχει κάποια προβλήματα στη χρήση του (π.χ. η απουσία undo στην έκδοση που χρησιμοποιήσαμε είναι μάλλον το πιο σοβαρό), αλλά που δίνει τη δυνατότητα σε κάποιον να σχεδιάσει εύκολα και γρήγορα δίκτυα και να τα “τρέξει” είτε βήμα προς βήμα (ιδιαίτερα χρήσιμο αν κάποιος ψάχνει το λάθος), είτε σε διάφορες ταχύτητες και να μελετήσει τα αποτελέσματα.

Στα παραδείγματα που ακολουθούν έχουμε χρησιμοποιήσει το συγκεκριμένο εργαλείο για τη σχεδίαση τους, το UI του φαίνεται στο σχήμα 8.1.

### 8.2.2 Συστατικά των Petri Nets

Τα Petri Nets είναι ιδιαίτερα απλά, γιατί χρησιμοποιούν μόνο 4 συστατικά για να σχεδιάσουμε οποιοδήποτε δίκτυο:

- Θέση (συμβολίζεται με ένα κύκλο)
- Μετάβαση (συμβολίζεται με ένα τετράγωνο)
- Σύνδεσμος (συμβολίζεται με ένα βέλος)
- Τεκμήριο (συμβολίζεται με μια μαύρη κουκίδα)

Οι θέσεις συνδέονται με τις μεταβάσεις με βέλος ή το αντίθετο, αλλά ποτέ δεν μπορούν να συνδεθούν δύο θέσεις μεταξύ τους ή δύο μεταβάσεις μεταξύ τους.

Μια θέση μπορεί να έχει ένα ή περισσότερα τεκμήρια ανάλογα με τον τύπο του Petri Net.

Μια θέση μπορεί να έχει χωρητικότητα ανάλογα με τον τύπο του δικτύου Petri Net (αν δεν αναγράφεται χωρητικότητα τότε θεωρείται ως 1).

Ένα βέλος μπορεί να έχει βάρος ανάλογα με τον τύπο του δικτύου (αν δεν αναγράφεται βάρος τότε θεωρείται ως 1).

Η κατάσταση (state) ενός δικτύου μια συγκεκριμένη στιγμή περιγράφεται από τα τεκμήρια που υπάρχουν σε κάθε θέση τη συγκεκριμένη στιγμή.

Υπάρχουν και άλλα είδη Petri Nets που έχουν περισσότερες ιδιότητες (όπως βέλη inhibitor και τεκμήρια πολλαπλών τύπων που συνήθως αναπαρίστανται με διαφορετικά χρώματα), αλλά εμείς θα μιλήσουμε μόνο για δύο είδη Petri Nets που αρκούν για την περιγραφή των περισσότερων προδιαγραφών που θα ασχοληθούμε στα παραδείγματά μας.

Ακολούθως θα μιλήσουμε για αυτές τις δύο κατηγορίες Petri Nets:

- Petri Nets συνθήκης-γεγονότος (condition-event).
- Petri Nets θέσης-μετάβασης (place-transition).

### 8.2.3 Petri Nets συνθήκης-γεγονότος

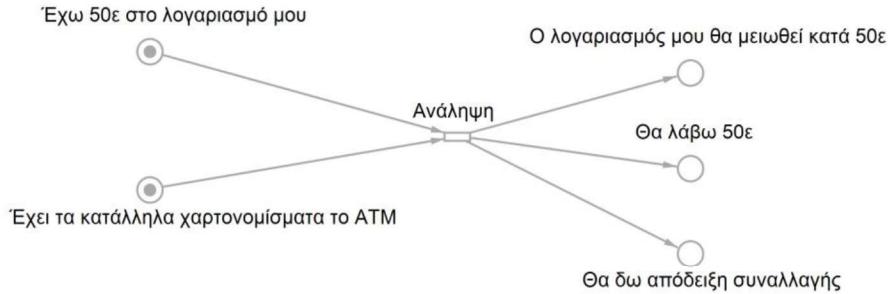
Η ιδιαιτερότητα των συγκεκριμένων δικτύων είναι ότι μπορούν να έχουν μόνο ένα τεκμήριο ανά θέση (άρα όλες οι θέσεις έχουν χωρητικότητα 1) και τα βάρη σε κάθε σύνδεσμο είναι πάντα 1 (άρα και όλα τα βέλη έχουν χωρητικότητα 1).

Σε αυτά τα Petri Nets οι θέσεις αναπαριστούν **συνθήκες** και οι μεταβάσεις **γεγονότα**. Οι θέσεις από τις οποίες ξεκινούν βέλη για ένα γεγονός ονομάζονται **προ-συνθήκες** ενώ οι θέσεις στις οποίες καταλήγουν βέλη από ένα γεγονός ονομάζονται **μετα-συνθήκες**.

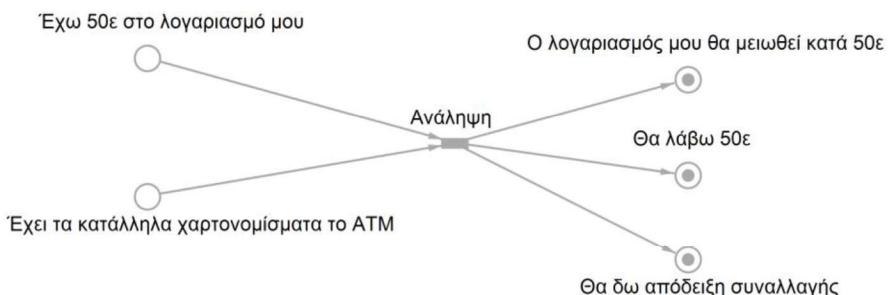
Ένα γεγονός **μπορεί να ενεργοποιηθεί** (ικανοποιηθεί) αν ικανοποιούνται όλες οι προ-συνθήκες (έχουν δηλαδή τεκμήριο όλες οι προ-συνθήκες) και μετά την ενεργοποίησή του θα ικανοποιηθούν όλες οι μετα-συνθήκες (θα αποκτήσουν δηλαδή τεκμήριο όλες οι μετα-συνθήκες).

Για παράδειγμα στο δίκτυο στο σχήμα 8.2 το γεγονός “Ανάληψη” που περιγράφει τη διαδικασία ανάληψης 50 ευρώ από ένα χρήστη έχει ως προ-συνθήκες τις “Έχω 50€ στο λογαριασμό μου” και “Έχει τα κατάλληλα χαρτονομίσματα το ATM”. Προσέξτε ότι δεν γράφουμε αν έχει τουλάχιστον 50 ευρώ το ATM γιατί θα μπορούσε να έχει μόνο 100 χαρτονομίσματα των 20 ευρώ και να μην μπορεί να ικανοποιήσει την ανάληψη που ζητά ως χρήστης.

## Ποιότητα Λογισμικού



**Σχήμα 8.2** Παράδειγμα Petri Net πριν την ενεργοποίηση του γεγονότος “Ανάληψη”



**Σχήμα 8.3** Παράδειγμα Petri Net μετά την ενεργοποίηση του γεγονότος “Ανάληψη”

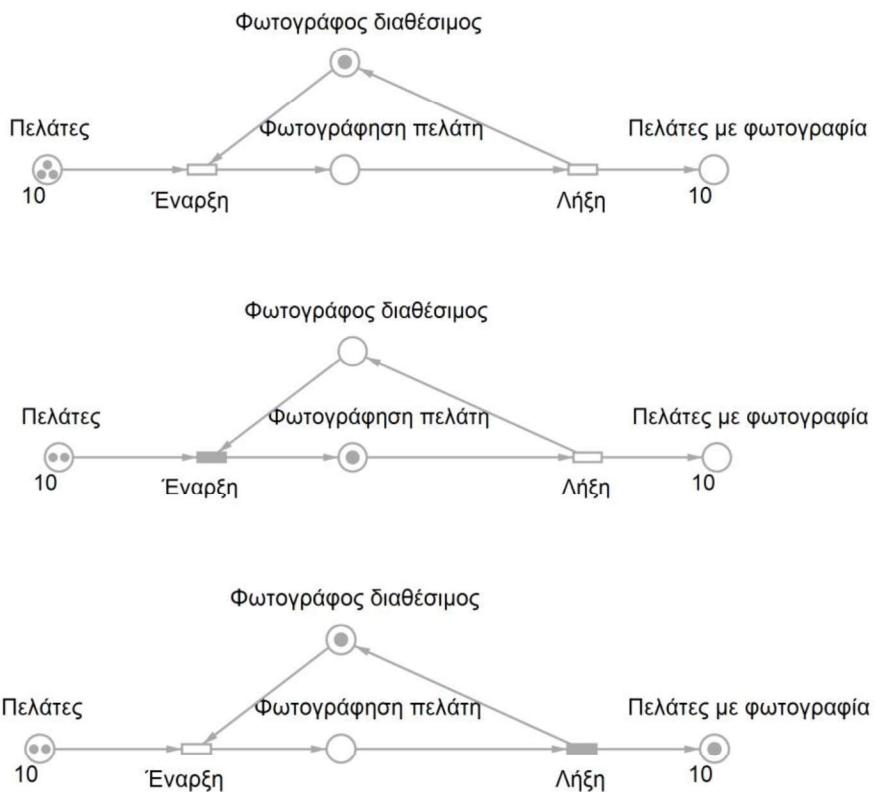
Από τη στιγμή που όλες οι προ-συνθήκες ενός γεγονότος ικανοποιούνται (όπως στο σχήμα 8.2 όπου όλες έχουν τεκμήριο) τότε αν “τρέξουμε” το δίκτυο αυτό σημαίνει ότι όλες οι μετα-συνθήκες θα ικανοποιηθούν. Άρα η επόμενη κατάσταση του δικτύου (αφού το τρέξαμε στο HPetri Sim) είναι αυτή στο σχήμα 8.3 όπου όλες οι μετα-συνθήκες έχουν ικανοποιηθεί και όλες οι προ-συνθήκες έχουν χάσει (καταναλώσει) τα τεκμήρια.

### 8.2.4 Petri Nets Θέσης-μετάβασης

Σε αυτά τα δίκτυα κάθε θέση έχει χωρητικότητα και κάθε σύνδεσμος έχει βάρος. Αυτό συνήθως σημαίνει ότι τα τεκμήρια συμβολίζουν κάτι μετρήσιμο (σε αντίθεση με τα δίκτυα συνθήκης-γεγονότος όπου συμβολίζαν την ικανοποίηση ή όχι μιας συνθήκης).

### Διαδικασίες ανάλυσης απαιτήσεων

Για παράδειγμα στα σχήματα 8.4 έως 8.5 περιγράφουμε τη διαδικασία ενός φωτογραφείου που έχει αίθουσα αναμονής για 10 πελάτες και 3 πελάτες περιμένουν ένα φωτογράφο που μπορεί να εξυπηρετήσει ένα πελάτη τη φορά. Προφανώς με τα Petri Nets μπορούμε να περιγράψουμε κάθε διαδικασία είτε αυτή αφορά λογισμικό είτε όχι!

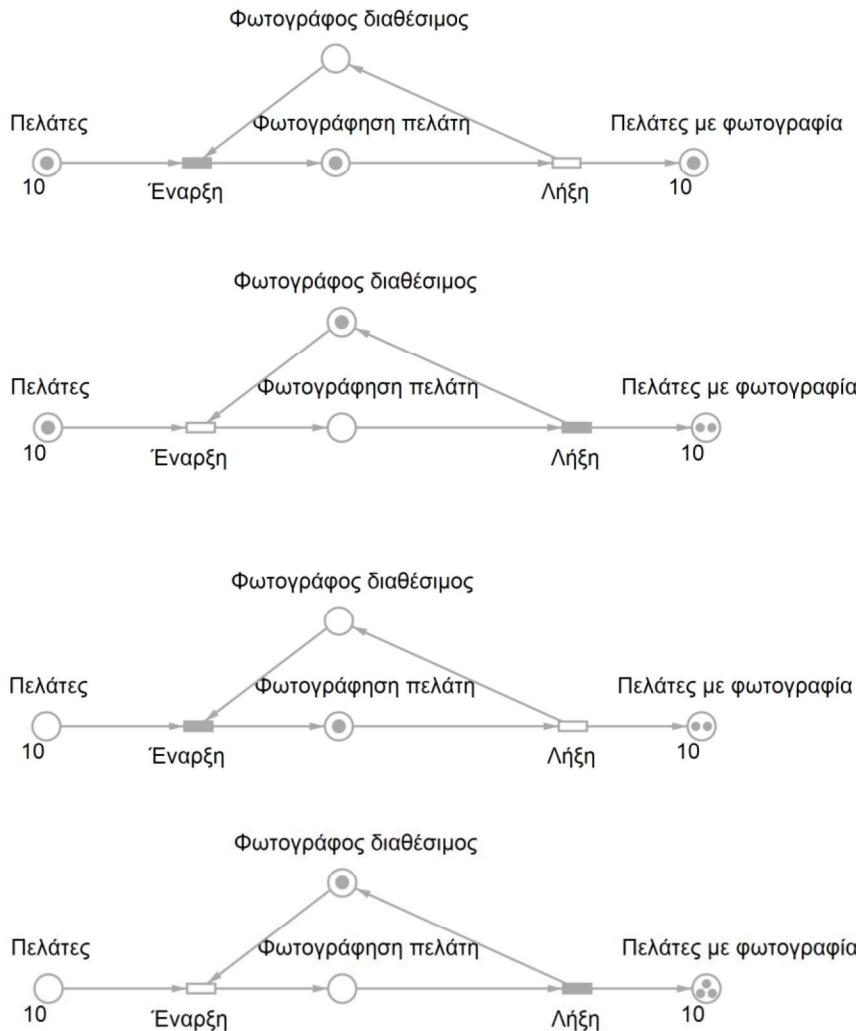


**Σχήμα 8.4** Παράδειγμα Petri Net θέσης-μετάβασης (φωτογράφηση 1<sup>ο</sup> πελάτη)

Στο σχήμα 8.4 αρχικά ο φωτογράφος είναι διαθέσιμος και 3 πελάτες είναι στην αναμονή (δείτε ότι στη θέση πελάτες αναγράφεται 10 που σημαίνει ότι η χωρητικότητα της συγκεκριμένης θέσης είναι δέκα πελάτες). Στη συνέχεια επειδή όλα τα βέλη έχουν βάρος 1 (αλλιώς θα αναγραφόταν διαφορετικά) ένας πελάτης μόνο εξυπηρετείται από τον φωτογράφο (άρα ο φωτογράφος δεν είναι πλέον διαθέσιμος) και στο επόμενο βήμα (τρίτο

## Ποιότητα Λογισμικού

στιγμιότυπο του σχήματος) ο πελάτης έχει φωτογραφηθεί και ο φωτογράφος είναι διαθέσιμος να φωτογραφήσει τον επόμενο πελάτη.



**Σχήμα 8.5** Παράδειγμα Petri Net θέσης-μετάβασης (φωτογράφηση 2<sup>ο</sup> και 3<sup>ο</sup> πελάτη)

Στο σχήμα 8.5 ο φωτογράφος φωτογραφίζει διαδοχικά τον 2<sup>ο</sup> και τον 3<sup>ο</sup> πελάτη και στο τελικό στιγμιότυπο του δικτύου και οι 3 πελάτες έχουν εξυπηρετηθεί και ο φωτογράφος αναμένει να εξυπηρετήσει νέους πελάτες.

Σε αυτό το σημείο δοκιμάστε να λύσετε τα παρακάτω:

- Πώς θα μπορούσαμε να περιγράψουμε το ίδιο φωτογραφείο αν υπήρχαν 3 χώροι φωτογράφησης και 3 διαθέσιμοι φωτογράφοι;
- Τι θα γίνει αν η χωρητικότητα της τελευταίας θέσης ήταν μόνο 1;

### 8.2.5 Βασικές δομές στα Petri Nets

Παρουσιάζουμε μερικές βασικές δομές στα Petri Nets που συνήθως χρησιμοποιούνται για την περιγραφή βασικών προδιαγραφών. Αυτές είναι:

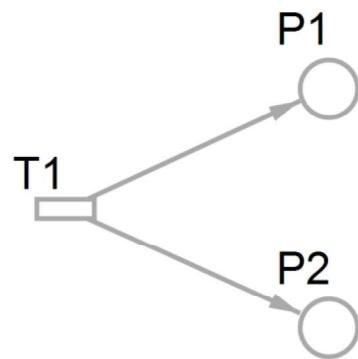
- Γεγονότα που συμβαίνουν παράλληλα
- Τυχαία επιλογή
- Δομές επανάληψης
- Μετρητές
- Πιθανότητες

#### 8.2.5.1 Γεγονότα που συμβαίνουν παράλληλα

Αν θέλουμε κάποια γεγονότα να συμβούν παράλληλα τότε αυτά απλά ξεκινούν μετά από μία μετάβαση. Όταν αυτή η μετάβαση ενεργοποιηθεί τότε εκτελούνται και τα δύο.

Στο σχήμα 8.6 η μετάβαση T1 όταν ενεργοποιηθεί θα ενεργοποιήσει τις θέσεις P1 και P2 και ότι ακολουθεί από εκεί και μετά θα εκτελεστεί παράλληλα.

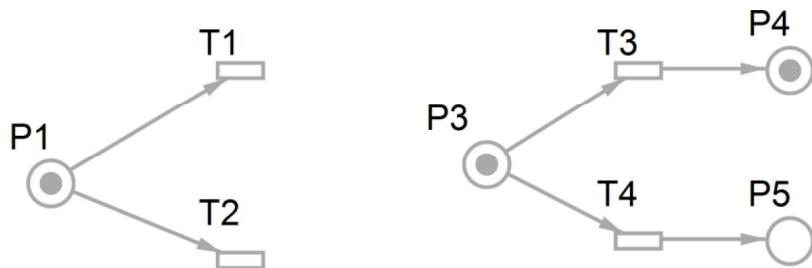
Με την ευκαιρία, το HPetri Sim ονομάζει αυτόματα τις θέσεις ως P1, P2, κτλ. (από το place) και τις μεταβάσεις ως T1, T2, κτλ. (από το transition). Για λόγους καλύτερης κατανόησης των δικτύων που σχεδιάζουμε είναι καλό να βάζουμε ονόματα που να έχουν κάποιο νόημα (όπως στα προηγούμενα παραδείγματα) και να μην τα αφήνουμε με ονόματα που δεν σημαίνουν κάτι (π.χ. P1, T1, κτλ.). Για αυτά τα παραδείγματα τα αφήσαμε έτσι γιατί δεν περιγράφουμε κάποιο πρόβλημα του πραγματικού κόσμου.



**Σχήμα 8.6** Παράλληλα γεγονότα

#### 8.2.5.2 Τυχαία επιλογή

Αν θέλουμε να περιγράψουμε ένα μη-ντετερμινιστικό δίκτυο όπου το σύστημα επιλέγει τυχαία κάτι τότε από μία θέση υπάρχουν βέλη για δύο μεταβάσεις, όπως στο σχήμα 8.7.



**Σχήμα 8.7** Τυχαία επιλογή

Προσοχή όμως, το αριστερό Petri Net είναι μη-ντετερμινιστικό και η επιλογή της T1 ή της T2 μετάβασης είναι τυχαία, αλλά για το Petri Net στο δεξί μέρος του σχήματος δεν ισχύει το ίδιο. Εκεί από τη θέση P3 θα ενεργοποιηθεί πάντα η μετάβαση T4 γιατί η T3 οδηγεί σε θέση η οποία έχει ήδη τεκμήριο και επειδή έχει χωρητικότητα 1 δεν μπορεί να ενεργοποιηθεί. Αν όμως η P4 είχε χωρητικότητα 2 τότε και το δεξιά Petri Net θα αφορούσε τυχαία επιλογή.

#### 8.2.5.3 Δομές επανάληψης

Όταν μια διαδικασία είναι επαναληπτική ή όταν τα τεκμήρια απεικονίζουν πόρους που απασχολούνται και μετά γίνονται και πάλι διαθέσιμοι τότε αυτό σχεδιάζεται με μια ακμή επιστροφής προς την προηγούμενη κατάσταση. Χαρακτηριστικό παράδειγμα είναι ο φωτογράφος στα σχήματα 8.4 και 8.5 όπου μετά τη μετάβαση “Λήξη” επιστρέφει στη θέση “Διαθέσιμος”.

#### 8.2.5.4 Μετρητές

Όταν θέλουμε να μετρήσουμε πόσες φορές συνέβη κάτι χρησιμοποιούμε μια θέση με πολύ μεγάλη<sup>4</sup> χωρητικότητα ως μετρητή. Ο μετρητής συνδέεται με ένα βέλος από μια μετάβαση που συμμετέχει στη ροή του δικτύου. Στο σχήμα 8.8 που περιγράφει ένα πιο μεγάλο παράδειγμα υπάρχουν αρκετοί τέτοιοι μετρητές.

Οι μετρητές είναι χρήσιμοι και για να ελέγξουμε Petri Nets που τρέχουν για μεγάλο αριθμό επαναλήψεων και δεν μπορούμε να παρακολουθήσουμε τη λειτουργία τους βήμα προς βήμα.

#### 8.2.5.5 Πιθανότητες

Όταν θέλουμε να περιγράφουμε κάτι που συμβαίνει με συγκεκριμένη πιθανότητα χρησιμοποιούμε τον κατάλληλο αριθμό από μεταβάσεις που να οδηγούν στη ίδια θέση.

---

<sup>4</sup> Μεγαλύτερη από όσα τεκμήρια περιμένουμε να μετρήσουμε

## Ποιότητα Λογισμικού

Αν για παράδειγμα θέλουμε να περιγράψουμε ότι από τη θέση Α μπορεί να βρεθούμε στη θέση Β με πιθανότητα  $p=1/3$  και στη θέση Γ με πιθανότητα  $p=2/3$  θα χρησιμοποιήσουμε 1 μετάβαση από την Α προς τη Β και 2 από την Α προς τη Γ. Παραδείγματα με τέτοιες πιθανότητες παρουσιάζονται στο σχήμα 8.8.

### 8.2.6 Ένα πλήρες παράδειγμα

Ακολουθεί ένα πλήρες παράδειγμα με τη λύση του (μία από τις αρκετές σωστές λύσεις). Προφανώς δοκιμάστε να λύσετε μόνοι σας το παράδειγμα πριν δείτε τη λύση του.

Έχετε κληθεί να προδιαγράψετε με χρήση Petri Nets την παρακάτω αρχική λεκτική περιγραφή για ένα σύστημα εξομοίωσης της συμπεριφοράς του μπροστινού συστήματος τροχών ενός αεροσκάφους κατά τη διαδικασία προσγείωσης.

Το σύστημα αποτελείται από ένα μηχανισμό με δύο τροχούς και μια σειρά από αισθητήρες. Αυτό που καλείστε να υλοποιήσετε έχει περιγραφεί λεκτικά ως εξής:

“Ο πιλότος δίνει εντολή να κατέβει το σύστημα τροχών. Το σύστημα του αεροσκάφους ελέγχει αν πληρούνται μια σειρά από προϋποθέσεις (π.χ. να έχει την κατάλληλη ταχύτητα, το κατάλληλο ύψος, την κατάλληλη πίεση καμπίνας, αλλά για τις ανάγκες τις εξομοίωσης θεωρήστε ότι δεν μας αφορούν) και αν όλα είναι OK το σύστημα του αεροσκάφους κατεβάζει το σύστημα των μπροστινών τροχών. Αν δεν πληρούνται όλες οι προϋποθέσεις δίνει μια ένδειξη λάθους στον πιλότο. Ο πιλότος πρέπει αμέσως να κάνει τις απαραίτητες ενέργειες (που επίσης δεν αφορούν την εξομοίωση) και να επιλέξει να προσπαθήσει ξανά. Μετά την εντολή του, το ίδιο σύστημα ελέγχει εκ νέου αν πληρούνται οι προϋποθέσεις και η διαδικασία επαναλαμβάνεται μέχρι ο πιλότος να καταφέρει να κατεβάσει το σύστημα τροχών. Για τις ανάγκες της εξομοίωσης να θεωρηθεί ότι η πιθανότητα προβλήματος (δηλαδή να μην πληρούνται όλες οι προϋποθέσεις) είναι  $1/3$  κάθε φορά που το σύστημα δέχεται την εντολή.

Στην περίπτωση που όλα είναι OK και κατέβει το σύστημα των τροχών, τότε ακολουθεί μια σειρά από ελέγχους χωριστά για κάθε τροχό (δεξί και αριστερό τροχό). Αν οι έλεγχοι και για τους δύο τροχούς είναι OK τότε ο

πιλότος μπορεί να προχωρήσει σε προσγείωση (που δεν μας αφορά σε αυτή την εργασία). Σε αυτή την περίπτωση η εξομοίωση σταματά εδώ.

Αν έστω και ένας τροχός έχει πρόβλημα τότε ο πιλότος ενημερώνεται για το ποιος τροχός είχε το πρόβλημα (δεξιά τροχός, αριστερά τροχός, και οι δύο τροχοί). Σε αυτή την περίπτωση η εξομοίωση σταματά εδώ. Για τις ανάγκες της εξομοίωσης να θεωρηθεί ότι η πιθανότητα προβλήματος σε ένα τροχό (δηλαδή να μην πληρούνται όλες οι προϋποθέσεις ελέγχων) είναι 25% για κάθε τροχό χωριστά.”

Προσπαθήστε να περιγράψετε την παραπάνω διαδικασία με Petri Nets. Στο σύστημα που θα υλοποιήσετε να βάλετε αντίστοιχους μετρητές, ώστε να καταγράφονται οι παρακάτω περιπτώσεις:

- A. Το σύστημα τροχών δεν κατέβηκε σε αυτή την προσπάθεια.
- B. Το σύστημα τροχών κατέβηκε, αλλά η διαδικασία προσγείωσης σταμάτησε λόγω προβλήματος στον δεξιά τροχό.
- C. Το σύστημα τροχών κατέβηκε, αλλά η διαδικασία προσγείωσης σταμάτησε λόγω προβλήματος στον αριστερά τροχό.
- D. Το σύστημα τροχών κατέβηκε, αλλά η διαδικασία προσγείωσης σταμάτησε λόγω προβλήματος και στους δύο τροχούς.
- E. Η διαδικασία κατεβάσματος και ελέγχου των τροχών ολοκληρώθηκε ικανοποιητικά και ο πιλότος μπόρεσε να προχωρήσει με την εξομοίωση της προσγείωσης.

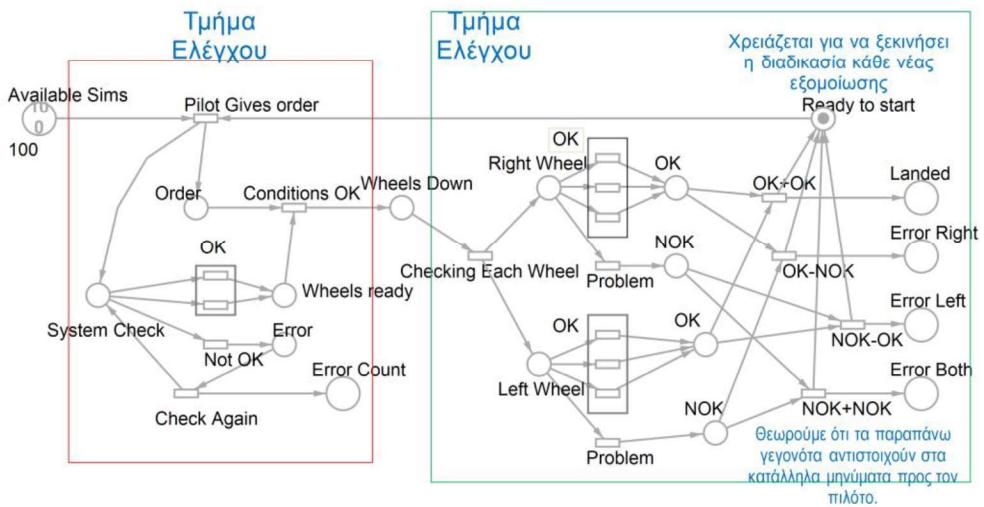
Υλοποιήστε το σύστημα στο εργαλείο HPetri Sim ορίζοντας ως αριθμό εξομοιώσεων τις 100 εξομοιώσεις (είναι ένας καλός έλεγχος για να δείτε αν τα αποτελέσματα συνάδουν με τις πιθανότητες που σας έχουν δοθεί). Το σύστημα δηλαδή θα πρέπει να “τρέξει” με επιτυχία 100 φορές και να δώσει αποτελέσματα στους παραπάνω μετρητές.

Στο σχήμα 8.8 παρουσιάζεται μια απλή λύση του ζητούμενου. Το πρώτο τμήμα που το έχουμε σχεδιάσει μέσα στο αριστερό πλαίσιο περιγράφει τη διαδικασία που ο πιλότος δίνει την εντολή, ενώ το τμήμα στο δεξιά πλαίσιο περιγράφει τη διαδικασία κάθε νέας εξομοίωσης.

Προσέξτε τη χρήση των μετρητών “Error Count”, “Landed”, “Error Right”, “Error Left”, και “Error Both” που έχουν χωρητικότητα 1000 (δεν εμφανίζεται στο σχήμα 8.8 για να μην μπερδέψει) και το πώς ορίζονται οι

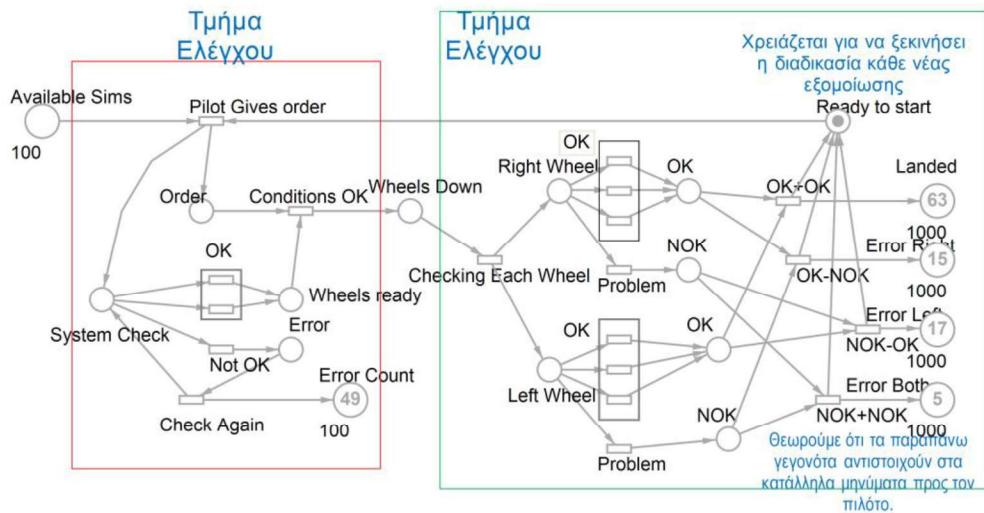
## Ποιότητα Λογισμικού

πιθανότητες όταν όλα είναι OK ή υπάρχει κάποιο πρόβλημα όπως ορίζει η περιγραφή.



**Σχήμα 8.8** Μια λύση του ζητούμενου

Στο σχήμα 8.9 φαίνεται ένα τελικό στιγμιότυπο του Petri Net μετά από την εκτέλεση 100 εξομοιώσεων.



Σχήμα 8.9 Ένα τελικό στιγμότυπο