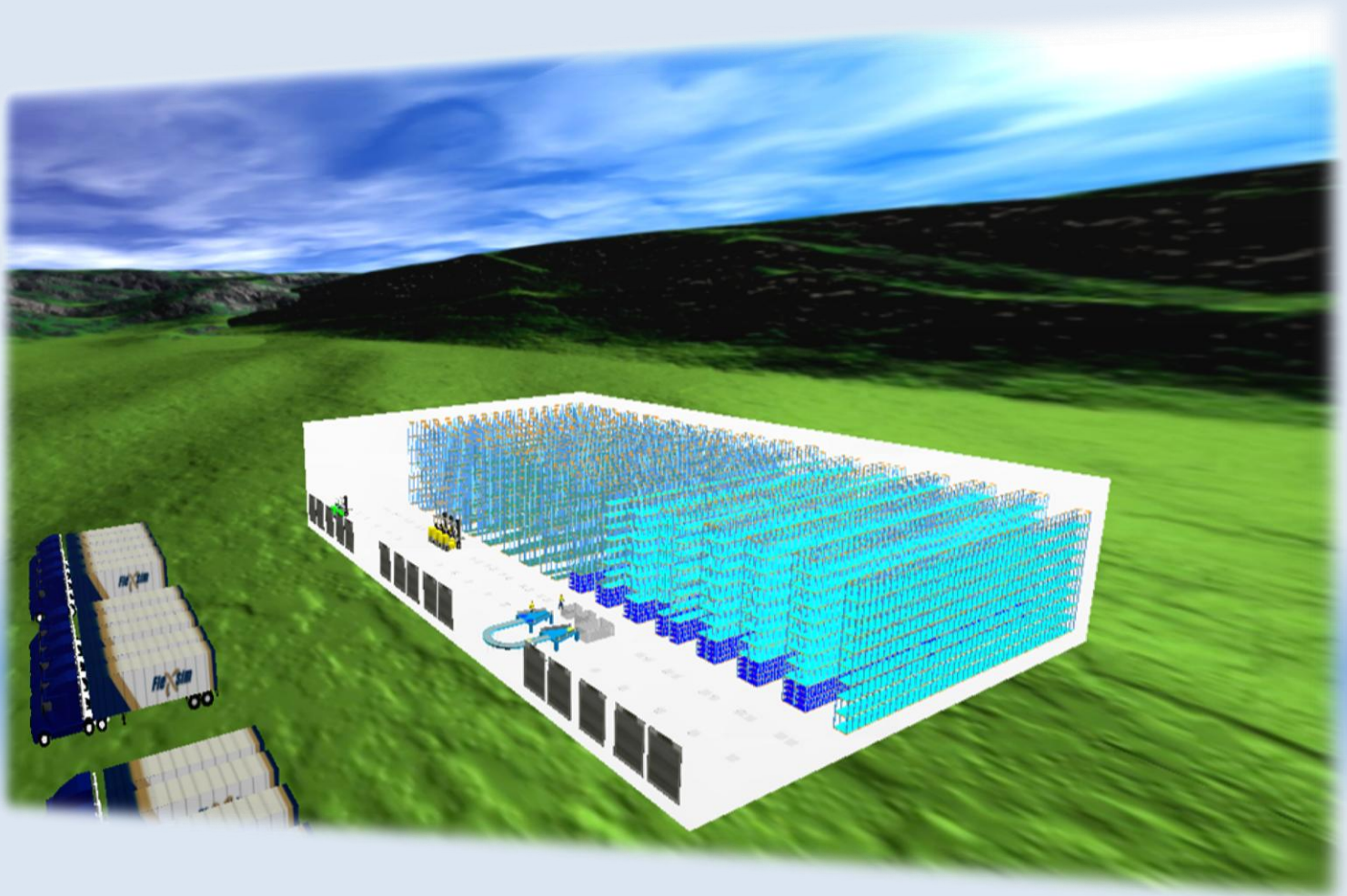


Διπλωματική Εργασία

# ***Προσομοίωση Διαδικασιών Αποθήκης με τη Χρήση του Λογισμικού FlexSim***



***Λαμπροπούλου Ασημίνα - Ελένη  
Επιβλέπων: Σ. Πόνης, Λέκτορας***

Αθήνα, Οκτώβριος 2011



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ  
Τμήμα Μηχανολόγων Μηχανικών  
Τομέας Βιομηχανικής Διοίκησης & Επιχειρησιακής Έρευνας

---

Διπλωματική Εργασία

---

***Προσομοίωση Διαδικασιών  
Αποθήκης με τη Χρήση του  
Λογισμικού FlexSim***

---

***Λαμπροπούλου Ασημίνα - Ελένη  
Επιβλέπων: Σ. Πόνης, Λέκτορας***

Αθήνα, Οκτώβριος 2011



## ***Ευχαριστίες...***

Φτάνοντας στην ολοκλήρωση της διπλωματικής μου εργασίας θα ήθελα αρχικά να ευχαριστήσω τον καθηγητή μου κ. Σ. Πόνη που μου παρείχε τη δυνατότητα να εκπονήσω την εργασία αυτή υπό την επίβλεψή του, συμβάλλοντας με τις παρατηρήσεις και τις σημαντικές συμβουλές του στην επιτυχή ολοκλήρωσή της. Ακόμα, θα ήθελα να ευχαριστήσω την οικογένεια μου και τους φίλους μου για τη σημαντική υποστήριξη που μου παρείχαν.

***Ελμίνα Λαμπροπούλου***



## **Περιεχόμενα**

<b>Εισαγωγή.....</b>	<b>9</b>
<b>Κεφάλαιο 1 : Εισαγωγικές Έννοιες.....</b>	<b>13</b>
1.1 Σύστημα .....	15
1.1.1 Χαρακτηριστικά μεγέθη συστημάτων .....	15
1.1.2 Συστατικά μέρη των συστημάτων.....	16
1.1.3 Ταξινόμηση συστημάτων.....	16
1.2 Μοντέλα συστημάτων .....	17
<b>Κεφάλαιο 2 : Προσομοίωση.....</b>	<b>19</b>
2.1 Ορισμός της προσομοίωσης.....	21
2.2 Σκοπός της προσομοίωσης .....	21
2.3 Πλεονεκτήματα, μειονεκτήματα και δυσκολίες της προσομοίωσης.....	23
2.4 Μοντέλα προσομοίωσης.....	25
2.5 Διαδικασία μελέτης προσομοίωσης.....	27
2.6 Προσομοίωση διακριτών συστημάτων (Discrete Event Simulation) .....	30
2.6.1 Συστατικά του μοντέλου προσομοίωσης .....	31
2.6.2 Μέθοδοι προσομοίωσης.....	33
2.6.3 Μεταβάσεις κατάστασης μοντέλων διακριτών συστημάτων .....	35
2.6.4 Δειγματοληψία και παραδείγματα Monte Carlo .....	38
<b>Κεφάλαιο 3 : Εισαγωγή στο Λογισμικό Προσομοίωσης FlexSim.....</b>	<b>45</b>
3.1 Λίγα λόγια για το λογισμικό FlexSim .....	47
3.2 Μοντελοποίηση .....	47
3.3 Περιπτώσεις χρήσης του λογισμικού FlexSim.....	48
3.4 Ορολογία του FlexSim .....	49
3.5 Εφαρμογές .....	50
Παράδειγμα 1 .....	50
Παράδειγμα 2 .....	52
Παράδειγμα 3 .....	55
Παράδειγμα 4 .....	57
Παράδειγμα 5 .....	61
<b>Κεφάλαιο 4 : Μελέτη Περίπτωσης - Προσομοίωση Διαδικασιών Αποθήκης ...</b>	<b>71</b>
4.1 Στοιχεία λειτουργίας της αποθήκης .....	73
4.2 Πολεοδομικοί περιορισμοί.....	76

4.3 Σχεδιασμός Αποθήκης .....	77
4.3.1 Παλέτες.....	77
4.3.2 Θυρίδες.....	78
4.3.3 Τμήματα αποθήκης.....	78
4.4 Μοντελοποίηση της αποθήκης.....	79
4.4.1 Πηγές – Sources.....	79
4.4.2 Καταβόθρες – Sinks .....	85
4.4.3 Ουρές αναμονής – Queues .....	87
4.4.4 Ράφια - Racks.....	91
4.4.5 Επεξεργαστές - Processors.....	94
4.4.6 Μεταφορείς - Transporters .....	100
4.4.7 Μεταφορικές ταινίες – Conveyors.....	105
4.4.8 Διεκπεραιωτές – Dispatchers .....	105
4.4.9 Κόμβοι δικτύου – Network Nodes.....	106
4.4.10 Πίνακες χρονικού προγραμματισμού – Time Tables .....	110
4.5 Ρύθμιση παραμέτρων του μοντέλου .....	113
4.6 Τρέξιμο μοντέλου .....	116
4.7 Αποτελέσματα προσομοίωσης.....	127
4.7.1 Διαγράμματα.....	127
4.7.2 Αναφορές.....	135
4.7.3 Μεγέθη απόδοσης.....	138
4.8 Ανάλυση κόστους.....	139
4.9 Συμπεράσματα.....	141
<b>Επίλογος.....</b>	<b>143</b>
<b>Παράρτημα .....</b>	<b>147</b>
<b>Βιβλιογραφία.....</b>	<b>153</b>



# Εισαγωγή



## **Εισαγωγή**

Στόχος του παρόντος πονήματος είναι η προσομοίωση σε Η/Υ της λειτουργίας μίας αποθήκης, με χρήση του λογισμικού FlexSim, και η εξαγωγή συμπερασμάτων ως προς το ποια είναι η οικονομικά πιο συμφέρουσα λύση για την αποθήκη.

Αρχικά, στο *Κεφάλαιο 1*, παρουσιάζονται κάποιες εισαγωγικές έννοιες οι οποίες θεωρούνται απαραίτητες για όποιον σκοπεύει να ασχοληθεί με τη μελέτη συστημάτων. Διατυπώνονται βασικοί ορισμοί, καταγράφονται κατηγοριοποιήσεις των συστημάτων και παρουσιάζονται μοντέλα ανάλυσής τους.

Στο *Κεφάλαιο 2* παρουσιάζονται οι βασικές αρχές της προσομοίωσης. Διατυπώνονται ορισμοί, αναλύεται η χρησιμότητά της, παρουσιάζονται διαφορετικά μοντέλα και αναλύεται η διαδικασία μελέτης της προσομοίωσης. Ακόμα, δίνεται ιδιαίτερη βαρύτητα στην Προσομοίωση Διακριτών Συστημάτων (Discrete Event Simulation) αναλύοντας τα συστατικά των μοντέλων και τις μεθόδους προσομοίωσης και παραθέτοντας διάφορα παραδείγματα.

Στο *Κεφάλαιο 3* γίνεται εισαγωγή στο λογισμικό προσομοίωσης FlexSim. Αναφέρονται περιπτώσεις χρήσης του λογισμικού, αναλύεται η ορολογία που χρησιμοποιείται στο υπόλοιπο της εργασίας και παρατίθενται παραδείγματα για εξοικείωση του αναγνώστη με το λογισμικό.

Το *Κεφάλαιο 4* αποτελεί και το κυρίως σώμα της εργασίας. Σε αυτό παρουσιάζεται η Μελέτη Περίπτωσης Αποθήκης. Αρχικά δίνονται στοιχεία σχετικά με τη μορφή, τη λειτουργία και τις απαιτήσεις της αποθήκης. Στη συνέχεια γίνεται μοντελοποίηση της αποθήκης με τη βοήθεια των αντικειμένων που παρέχει το λογισμικό FlexSim. Ακολουθεί πλήρης περιγραφή των αντικειμένων που χρησιμοποιήθηκαν καθώς και των ιδιοτήτων τους. Έπειτα παρουσιάζονται τα αποτελέσματα της προσομοίωσης με τη βοήθεια διαγραμμάτων, αναφορών και συγκριτικών πινάκων και ακολουθεί ανάλυση κόστους. Τέλος παρουσιάζονται τα συμπεράσματα της προσομοίωσης.

Στον *Επίλογο* αναφέρονται τα συμπεράσματα της διπλωματικής εργασίας και στο τέλος της εργασίας υπάρχει *Παράρτημα* με την κάτοψη της αποθήκης και πληροφορίες για τον εξοπλισμό της.



# Κεφάλαιο 1

## Εισαγωγικές Έννοιες



## 1.1 Σύστημα

Το σύστημα ορίζεται ως ένα σύνολο οντοτήτων, π.χ. ανθρώπους ή μηχανές, που δρουν και αλληλεπιδρούν μεταξύ τους για την επίτευξη κάποιου σκοπού. (Schmidt – Taylor, 1970)

Τα συστήματα αναπαριστούν την πραγματικότητα, επηρεάζονται από εξωτερικούς παράγοντες και χρησιμοποιούνται για την κατανόηση και τη μελέτη πραγματικών καταστάσεων. Το σύνολο των οντοτήτων που συνθέτουν ένα σύστημα για μία μελέτη μπορεί να είναι απλά ένα υποσύνολο συστήματος για άλλη μελέτη, γι' αυτό πρέπει πρώτα να καθορίζονται τα όρια του συστήματος. Κατά την ανάλυση ενός συστήματος μελετάται η έξοδος του συστήματος όταν έχει καθοριστεί η είσοδος σε αυτό, ενώ κατά το σχεδιασμό ενός συστήματος μελετώνται τα χαρακτηριστικά του συστήματος με δεδομένες τις εισόδους και τις αντίστοιχες εξόδους του.

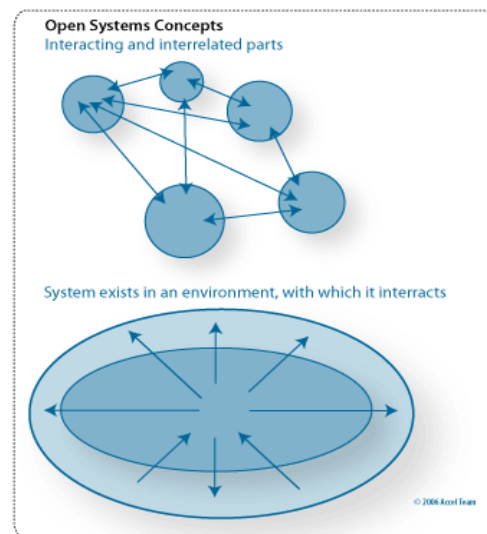
### 1.1.1 Χαρακτηριστικά μεγέθη συστημάτων

Τα μεγέθη μέτρησης και χαρακτηρισμού ενός συστήματος είναι οι παράμετροι και οι μεταβλητές. Οι παράμετροι είναι ανεξάρτητα μέτρα που διαμορφώνουν τις συνθήκες των εισόδων, ενώ οι μεταβλητές είναι μέτρα εξαρτώμενα από τις παραμέτρους και από άλλες μεταβλητές.

Η κατάσταση του συστήματος (*state*) ορίζεται ως το σύνολο των απαραίτητων μεταβλητών για την περιγραφή του συστήματος σε συγκεκριμένη χρονική στιγμή, ανάλογα με τους στόχους της μελέτης (Law-Kelton, 2000).

Οι μεταβλητές αυτές ονομάζονται **μεταβλητές κατάστασης του συστήματος (state variables)**.

Τα **γεγονότα (events)** είναι στιγμιαία συμβάντα που μπορεί να προκαλέσουν μεταβολές ή αλλαγές στην κατάσταση ενός συστήματος (Sokolowski – Banks, 2009). Διαχωρίζονται σε ενδογενή (endogenous) και εξωγενή (exogenous). Ο όρος ενδογενή χρησιμοποιείται για τα γεγονότα που συμβαίνουν μέσα στο σύστημα, ενώ ο όρος εξωγενή χρησιμοποιείται για αυτά που συμβαίνουν στο περιβάλλον του συστήματος αλλά το επηρεάζουν.



### 1.1.2 Συστατικά μέρη των συστημάτων

Πριν προχωρήσουμε στη μελέτη των συστημάτων είναι απαραίτητο να δοθούν οι ορισμοί κάποιων μεγεθών που θα χρησιμοποιηθούν στη συνέχεια.

1. **Οντότητα (entity)** είναι ένα αντικείμενο ενδιαφέροντος για το σύστημα. Μία οντότητα μπορεί να είναι δυναμική, δηλ. να «μετακινείται» μέσα στο σύστημα ή στατική, δηλ. να εξυπηρετεί άλλες οντότητες. Κάθε οντότητα δεσμεύει έναν ή περισσότερους πόρους και ο συνδυασμός τους παρέχει κάποιες υπηρεσίες στο σύστημα.
2. **Ιδιότητες (attributes)** είναι οι ιδιότητες μίας οντότητας.
3. **Πόροι ή μέσα παραγωγής (resources)** του συστήματος είναι περιορισμένα αγαθά που χρησιμοποιούνται, καταναλώνονται ή αναπληρώνονται από τις οντότητες ώστε να γίνουν κάποιες διεργασίες στο σύστημα.
4. **Δραστηριότητα (activity)** είναι μία πράξη που πραγματοποιείται σε συγκεκριμένο χρονικό εύρος. Όπως τα γεγονότα, έτσι και οι δραστηριότητες διακρίνονται σε ενδογενείς και εξωγενείς. Οι δραστηριότητες καταναλώνουν πόρους και δημιουργούν συμβάντα που αλλάζουν την κατάσταση του συστήματος.
5. **Ροή (flow)** είναι η κίνηση των οντοτήτων μέσα στο σύστημα.
6. **Ουρές αναμονής (queues)** είναι κάποια αντικείμενα στα οποία γίνεται συγκέντρωση πόρων πριν αυτοί μεταβούν σε άλλο αντικείμενο.

### 1.1.3 Ταξινόμηση συστημάτων

Τα συστήματα χωρίζονται σε δύο κύριες κατηγορίες:

- τα διακριτά συστήματα (discrete), όπου οι μεταβλητές αλλάζουν στιγμιαία σε συγκεκριμένες χρονικές στιγμές
- τα συνεχή συστήματα (continuous), όπου οι μεταβλητές κατάστασης αλλάζουν συνεχώς ως προς το χρόνο.

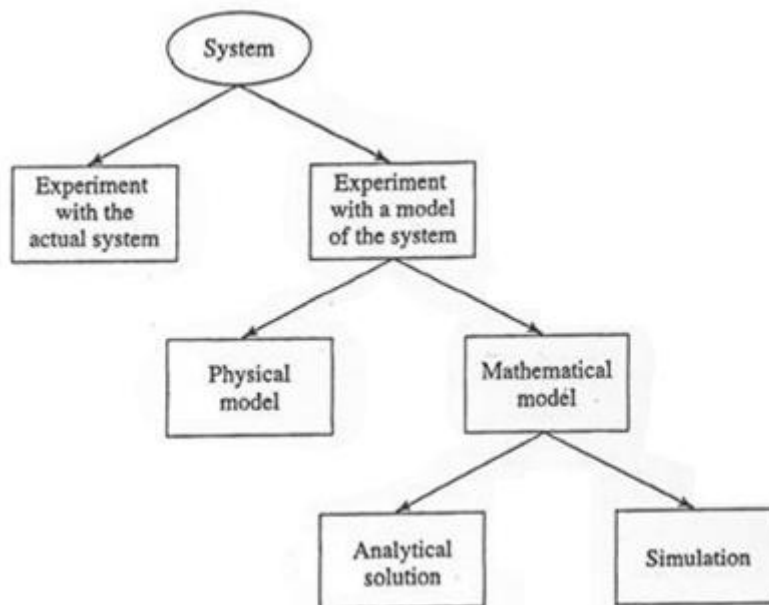
Η θεώρηση του συστήματος και των ορίων του καθώς και η κατάταξή του στις ανωτέρω κατηγορίες εξαρτάται από τον τρόπο μελέτης του.



## 1.2 Μοντέλα συστημάτων

**Μοντέλο** ενός συστήματος ονομάζεται η απλοποιημένη αναπαράστασή του ώστε αυτό να αναλυθεί ή να μελετηθεί. Στις περισσότερες μελέτες λαμβάνονται υπόψη μόνο τα χαρακτηριστικά του συστήματος που επηρεάζουν το υπό μελέτη πρόβλημα. Παρόλα αυτά το μοντέλο πρέπει να είναι αρκετά αναλυτικό ώστε η μελέτη να καταλήξει σε έγκυρα συμπεράσματα που θα αξιοποιηθούν στο πραγματικό σύστημα. Τα συστατικά μέρη των μοντέλων είναι τα ίδια με αυτά των συστημάτων.

Η χρησιμότητα των μοντέλων είναι μεγάλη ειδικά όταν δεν μπορεί να δεσμευθεί το φυσικό σύστημα για μελέτη. Τα μοντέλα, ανάλογα με τον τρόπο μελέτης του συστήματος που αναπαριστούν είναι τα ακόλουθα:



- ☞ **Φυσικό:** πρόκειται για μία φυσική αναπαράσταση του αντικειμένου που αντιπροσωπεύει και μπορεί να είναι μικρότερης, μεγαλύτερης ή ίδιας κλίμακας με το σύστημα.
- ☞ **Μαθηματικό:** Το μαθηματικό μοντέλο χρησιμοποιεί μαθηματικές έννοιες, λογικές και ποσοτικές σχέσεις για να περιγράψει τις φυσικές ιδιότητες και λειτουργίες του συστήματος.
- **Αναλυτική επίλυση:** Εάν το μαθηματικό μοντέλο είναι αρκετά απλό μπορεί να επιλυθεί με χρήση των μαθηματικών σχέσεων που το διέπουν με αναλυτική επίλυση.

- ➔ **Προσομοίωση:** Η προσομοίωση χρησιμοποιείται όταν δεν είναι δυνατή η αναλυτική επίλυση. Είναι ένας τρόπος δημιουργίας αριθμητικών μοντέλων, δηλαδή χρησιμοποιούνται αριθμητικά δεδομένα των υπό μελέτη εισόδων του συστήματος για να μελετηθεί ο τρόπος που επηρεάζουν τις εξόδους του. Σήμερα, χάρη στην τεχνολογική εξέλιξη, είναι πολύ δημοφιλής η προσομοίωση με χρήση Η/Υ.

## Κεφάλαιο 2

### Προσομοίωση



## **2.1 Ορισμός της προσομοίωσης**

*Η προσομοίωση (simulation) είναι μια εφαρμοσμένη μεθοδολογία που μπορεί να περιγράψει τη συμπεριφορά κάποιου συστήματος χρησιμοποιώντας είτε ένα μαθηματικό μοντέλο είτε ένα συμβολικό μοντέλο (Fishwick,1995). Με απλά λόγια, προσομοίωση είναι η μίμηση της λειτουργίας μιας πραγματικής διαδικασίας ή συστήματος κατά τη διάρκεια μιας χρονικής περιόδου (Banks,1998).*

Παραδοσιακά, η διαμόρφωση των συστημάτων γίνεται μέσω ενός μαθηματικού μοντέλου, το οποίο προσπαθεί να βρει τις αναλυτικές λύσεις επιτρέποντας την πρόβλεψη της συμπεριφοράς του συστήματος από ένα σύνολο παραμέτρων και αρχικών όρων. Όμως η πλειοψηφία των πραγματικών συστημάτων είναι δύσκολο ή αδύνατο να μελετηθεί μέσω μαθηματικών μοντέλων λόγω της σπανιότητας ή της έλλειψης σχεδόν υπολογίσιμης λύσης. Η προσομοίωση με χρήση Η/Υ χρησιμοποιείται συχνά ως προσθήκη, ή αντικατάσταση της διαδικασίας για τη διαμόρφωση των συστημάτων για τα οποία οι αναλυτικές λύσεις δεν είναι δυνατές. Αντίθετα, ένα μοντέλο προσομοίωσης με χρήση Η/Υ μπορεί σχεδόν πάντα να κατασκευαστεί και να εφαρμοστεί για να παράξει σενάρια που δίνουν χρήσιμες στατιστικές πληροφορίες για τη λειτουργία και την απόδοση των συστημάτων. Γι' αυτό, στην παρούσα αναφορά με τον όρο προσομοίωση θα εννοείται η προσομοίωση με χρήση Η/Υ (computer simulation). Ουσιαστικά, ένα μοντέλο προσομοίωσης είναι κάποιας μορφής πρόγραμμα Η/Υ που προσομοιώνει το υπό μελέτη σύστημα.

## **2.2 Σκοπός της προσομοίωσης**

Η προσομοίωση βοηθά τον αναλυτή να καταλάβει πόσο καλά λειτουργεί ένα σύστημα κάτω από ένα δεδομένο καθεστώς ή ένα σύνολο παραμέτρων. Μπορεί επίσης να χρησιμοποιηθεί στις μελέτες βελτιστοποίησης για να βρεθούν οι καλύτερες ή αποδεκτές τιμές των παραμέτρων.

Μέχρι τη δεκαετία του '80, η προσομοίωση ήταν αρκετά δαπανηρή και χρονοβόρα. Η εμφάνιση των προσωπικών Η/Υ, με ισχυρούς επεξεργαστές και γραφικά, προσέφερε νέες δυνατότητες που έκαναν την προσομοίωση μια ιδιαίτερα ελκυστική και οικονομικώς αποδοτική προσέγγιση για την ανάλυση πάρα πολλών συστημάτων. Πλέον ο προγραμματισμός των μοντέλων προσομοίωσης έχει απλοποιηθεί προσφέροντας τη δυνατότητα χρήσης διαγραμμάτων, κειμένων, τη

δυνατότητα διόρθωσης, εμφάνισης δυναμικών πληροφοριών και κίνηση των αντικειμένων.

Τα μοντέλα προσομοίωσης χρησιμοποιούνται για την ανάλυση πολυσύνθετων συστημάτων δημιουργώντας μια απλουστευμένη αναπαράσταση του υπό μελέτη συστήματος. Στη συνέχεια το μοντέλο δοκιμάζεται, καθοδηγούμενο από κάποιους στόχους, όπως το βελτιωμένο σχέδιο συστημάτων, την ανάλυση κόστους - οφέλους, την ευαισθησία στις σχεδιαστικές παραμέτρους, και ούτω καθεξής. Γίνονται δοκιμές με την εφαρμογή διαφόρων σεναρίων και την παρατήρηση της συμπεριφοράς του συστήματος με την πάροδο του χρόνου, καθώς επίσης και των στατιστικών του.

Είναι πολύ σημαντικό να λαμβάνονται υπόψη έγκυρα δεδομένα για την επιλογή των βασικών χαρακτηριστικών και των συμπεριφορών για τη δημιουργία του μοντέλου προσομοίωσης, να χρησιμοποιούνται απλουστευμένες προσεγγίσεις και υποθέσεις, και τα εξαγόμενα της προσομοίωσης να είναι έγκυρα.

Η προσομοίωση χρησιμοποιείται εκτενώς στη βιομηχανία ως εργαλείο απόφασης και υποστήριξης σε πολυάριθμα βιομηχανικά προβλήματα. Αυτή η τεχνική αποτελεί ένα πολύτιμο εργαλείο που χρησιμοποιείται από τους μηχανικούς κατά την αξιολόγηση της επένδυσης κεφαλαίου σε εξοπλισμό και σε φυσικές εγκαταστάσεις, όπως το κυρίως εργοστάσιο, τις αποθήκες εμπορευμάτων, και τα κέντρα διανομής. Χρησιμοποιείται για τη δοκιμαστική λειτουργία των συστημάτων, για την εκτίμηση των δυνατοτήτων και τη βελτιστοποίηση της απόδοσής τους, για δοκιμή της ασφάλειάς τους, για τη δοκιμή εναλλακτικών μεθόδων λειτουργίας, ακόμα και για κατάρτιση και εκπαίδευση. Η προσομοίωση είναι επίσης πολύ χρήσιμη όταν δεν μπορεί να δεσμευθεί το πραγματικό σύστημα σε περιπτώσεις που αυτό δεν είναι προσιτό ή είναι επικίνδυνο ή απλά δεν υπάρχει. Έτσι μπορεί να χρησιμοποιηθεί για την ανάλυση και βελτίωση υπάρχοντος συστήματος αλλά και για το σχεδιασμό νέου.

Ένας άλλος σημαντικός στόχος των κατασκευαστικών προσομοιώσεων είναι να ποσοτικοποιηθεί η απόδοση του συστήματος. Οι συνήθεις δείκτες της απόδοσης συστημάτων είναι οι εξής:

- ➔ Διεκπεραιωτική ικανότητα κάτω από μέσα και μέγιστα φορτία
- ➔ Κύκλος ζωής συστημάτων (πόσος χρόνος χρειάζεται για να παραχθεί ένα μέρος)
- ➔ Χρησιμοποίηση του πόρου, της εργασίας, και των μηχανών
- ➔ Δυσχέρειες και σημεία συμφόρησης (bottlenecks)
- ➔ Αναμονή στις θέσεις εργασίας

- ➔ Αναμονή και καθυστερήσεις που προκαλούνται από το χειρισμό των υλικών στις μηχανές και τα συστήματα
- ➔ Ανάγκες αποθήκευσης ημιετοιμών
- ➔ Ανάγκες επάνδρωσης
- ➔ Αποτελεσματικότητα του σχεδιασμού των συστημάτων
- ➔ Αποτελεσματικότητα των συστημάτων ελέγχου.

Η προσομοίωση χρησιμοποιείται επίσης σε τομείς όπως τη φυσική, τη χημεία, τη βιολογία, την ιατρική, τα οικονομικά, την εκπαίδευση, όταν είναι ακριβό ή επικίνδυνο να χρησιμοποιηθεί ο πραγματικός εξοπλισμός (διαστημικοί σταθμοί, στρατός, οχήματα, μηχανές εργοστασίων) ή σε παίγνιο επιχειρήσεων, τις κοινωνικές επιστήμες, το σχεδιασμό πόλεων, την αντιμετώπιση εκτάκτων συνθηκών, ακόμα και στη διασκέδαση.

### **2.3 Πλεονεκτήματα, μειονεκτήματα και δυσκολίες της προσομοίωσης**

Αν και η προσομοίωση είναι μια ευρέως χρησιμοποιούμενη και συνεχώς πιο δημοφιλής μέθοδος για τη μελέτη πολύπλοκων συστημάτων, εκτός από πλεονεκτήματα έχει και μειονεκτήματα, αλλά υπάρχουν και πολλές αιτίες που ορισμένες προσομοιώσεις δεν καταλήγουν στο επιθυμητό αποτέλεσμα.

Ως πλεονεκτήματα της χρήσης της προσομοίωσης μπορούν να χαρακτηριστούν τα ακόλουθα:

- ➔ Η προσομοίωση επιτρέπει την εκτίμηση της απόδοσης ενός υπάρχοντος συστήματος, κάτω από κάποιο προβλεπόμενο σύνολο συνθηκών λειτουργίας.
- ➔ Μπορούν να εξαχθούν συμπεράσματα σχετικά με τις μεταβλητές που επηρεάζουν περισσότερο το σύστημα και με τον τρόπο που αλληλεπιδρούν.
- ➔ Μπορούν να συγκριθούν μέσω της προσομοίωσης, εναλλακτικές προτεινόμενες σχεδιαστικές λύσεις ή εναλλακτικές πολιτικές λειτουργίας του συστήματος, ώστε να προσδιορισθεί η βέλτιστη λύση που ικανοποιεί τις προδιαγραφές που έχουν ορισθεί, πριν την εφαρμογή στο πραγματικό σύστημα.
- ➔ Σε ένα μοντέλο προσομοίωσης υπάρχει καλύτερος έλεγχος στις συνθήκες των πειραμάτων, σε σχέση με πιθανό πειραματισμό με το πραγματικό σύστημα.

- ➔ Η προσομοίωση επιτρέπει τη μελέτη ενός συστήματος που έχει μακρόχρονη εξέλιξη (π.χ. ένα οικονομικό σύστημα), σε πολύ μικρότερο χρόνο, ή τη μελέτη κάποιας λεπτομέρειας σε περισσότερο χρόνο.
- ➔ Είναι μια μέθοδος οικονομική, αφού είναι δυνατό να υλοποιηθεί πλέον σε μικρούς υπολογιστές με τη χρήση γλωσσών προγραμματισμού γενικού σκοπού όπως η C, η Pascal και η BASIC.
- ➔ Η προσομοίωση μπορεί να υλοποιηθεί από μηχανικούς που δεν είναι απαραίτητο να έχουν εκτεταμένες μαθηματικές γνώσεις, παρά μόνο τη δυνατότητα να κατανοούν βασικές έννοιες στατιστικής και να μπορούν να εφαρμόζουν ήδη έτοιμα μαθηματικά εργαλεία.
- ➔ Είναι πολύ σημαντική η δυνατότητα που παρέχει για οπτικοποίηση της λειτουργίας του συστήματος.

Ως μειονεκτήματα της προσομοίωσης αναφέρονται τα παρακάτω:

- ➔ Κάθε εκτέλεση ενός μοντέλου προσομοίωσης διακριτών γεγονότων παράγει μόνο εκτιμήσεις των πραγματικών χαρακτηριστικών του μοντέλου, για ένα συγκεκριμένο σύνολο παραμέτρων εισόδου. Έτσι, είναι πιθανό να χρειασθούν πολλές διαφορετικές ανεξάρτητες εκτελέσεις του μοντέλου για κάθε σύνολο παραμέτρων εισόδου που θα μελετηθεί. Για το λόγο αυτό, η προσομοίωση δεν είναι γενικά τόσο καλή μέθοδος για βελτιστοποίηση, όσο είναι για τη σύγκριση εναλλακτικών σχεδιαστικών λύσεων του συστήματος.
- ➔ Τα μοντέλα προσομοίωσης συχνά απαιτούν πολύ χρόνο και πόρους για να αναπτυχθούν.
- ➔ Ο μεγάλος όγκος αριθμών που παράγονται από μία μελέτη προσομοίωσης ή η εντύπωση που δημιουργούν οι τυχόν γραφικές αναπαραστάσεις των αποτελεσμάτων της, συχνά ενισχύουν μία τάση να δίνεται μεγαλύτερη εμπιστοσύνη στα αποτελέσματα αυτά από όσο πρέπει. Αν το μοντέλο δεν είναι μία αρκετά έγκυρη αναπαράσταση του συστήματος, τα αποτελέσματα της προσομοίωσης, ανεξάρτητα του πόσο εντυπωσιακά είναι, θα προσθέσουν λίγη χρήσιμη πληροφορία για το πραγματικό σύστημα.

Υπάρχουν όμως και πολλά προβλήματα που προκύπτουν κατά την προσομοίωση που οφείλονται είτε σε λάθη του αναλυτή, είτε σε ελλείψεις του λογισμικού:



- ➔ Όχι καλά ορισμένοι στόχοι κατά την έναρξη της μελέτης.
- ➔ Ακατάλληλο επίπεδο λεπτομέρειας του μοντέλου.
- ➔ Χειρισμός της μελέτης με προσομοίωση, σαν να ήταν βασικά μία δύσκολη άσκηση προγραμματισμού.
- ➔ Έλλειψη στοιχειωδών γνώσεων Επιχειρησιακής Έρευνας και Στατιστικής.
- ➔ Χρήση εμπορικών πακέτων προσομοιωτών που μπορεί να περιέχουν λάθη ή να μην υλοποιούν τη λογική του συστήματος.
- ➔ Αποτυχία στη σωστή καταγραφή των πηγών τυχαιότητας του συστήματος.
- ➔ Χρήση αυθαίρετων κατανομών (π.χ. κανονική ή ομοιόμορφη) για την περιγραφή των εισόδων του προσομοιωτή.
- ➔ Ανάλυση των δεδομένων εξόδου από μία εκτέλεση του προσομοιωτή, με τη χρήση στατιστικών τύπων που προϋποθέτουν ανεξαρτησία.
- ➔ Χρήση λανθασμένων μέτρων απόδοσης.

## 2.4 Μοντέλα προσομοίωσης

Δεδομένου ότι με την προσομοίωση μελετώνται μαθηματικά μοντέλα, θα χρησιμοποιείται στο εξής ο όρος **μοντέλο προσομοίωσης**. Τα μοντέλα προσομοίωσης διακρίνονται στις ακόλουθες κατηγορίες (Law-Kelton, 2000):

- ➔ **Στατικά – Steady state:** Τα στατικά μοντέλα χρησιμοποιούνται για την αναπαράσταση συστημάτων σε συγκεκριμένο χρόνο ή συστημάτων που δεν μεταβάλλονται με το χρόνο, όπως τα μοντέλα Monte Carlo.
- ➔ **Δυναμικά - Dynamic:** Τα δυναμικά μοντέλα προσομοίωσης χρησιμοποιούνται για την αναπαράσταση συστημάτων που μεταβάλλονται με το χρόνο.
- ➔ **Ντετερμινιστικά - Deterministic:** Στα ντετερμινιστικά μοντέλα προσομοίωσης δεν εμπεριέχονται τυχαίες μεταβλητές. Είσοδος είναι ένα γνωστό σύνολο δεδομένων και παράγεται ένα μοναδικό σύνολο εξόδων.
- ➔ **Στοχαστικά - Stochastic:** Τα στοχαστικά μοντέλα προσομοίωσης εμπεριέχουν τουλάχιστον μία τυχαία μεταβλητή. Οι έξοδοι που παράγονται είναι τυχαίες και αντιμετωπίζονται ως προσεγγίσεις του υπό μελέτη συστήματος και όχι σαν μία ακριβής αναπαράσταση της συμπεριφοράς του.

- ➔ **Συνεχή - Continuous:** Τα συνεχή μοντέλα προσομοίωσης ορίζονται όπως και τα συνεχή συστήματα. Αναφέρονται σε μοντέλα που η κατάσταση τους μεταβάλλεται με το χρόνο. Συνήθως αυτά τα μοντέλα περιγράφονται με διαφορικές εξισώσεις που αντιπροσωπεύουν τους ρυθμούς μεταβολής των μεταβλητών του μοντέλου.
- ➔ **Διακριτά - Discrete:** Στα διακριτά μοντέλα προσομοίωσης περιγράφεται ένα σύστημα που εξελίσσεται με το χρόνο και η κατάσταση του αλλάζει ασυνεχώς, σε συγκεκριμένα συμβάντα.

Αν και τα μοντέλα προσομοίωσης ορίζονται κατά τον ίδιο τρόπο με τα συστήματα, ένα διακριτό μοντέλο προσομοίωσης δεν μοντελοποιεί πάντα ένα διακριτό σύστημα και αντιστρόφως. Το ίδιο ισχύει και για τα συνεχή μοντέλα προσομοίωσης.

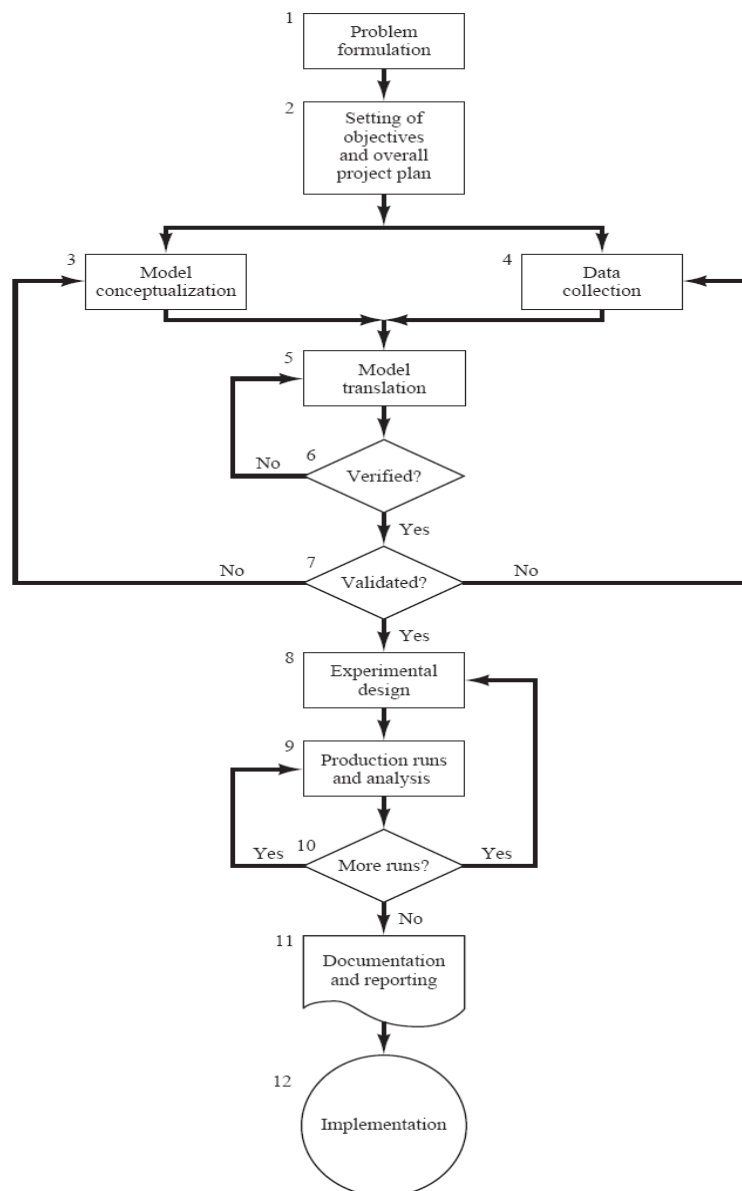
Τα μοντέλα προσομοίωσης κατηγοριοποιούνται επίσης ως προς το εύρος εφαρμογής τους, τον τρόπο μοντελοποίησης και τα χαρακτηριστικά του συστήματος Η/Υ πάνω στο οποίο εκτελούνται. Έτσι διακρίνονται στις ακόλουθες κατηγορίες:

- ➔ **Μοντέλα γενικού/ ειδικού σκοπού (General/ Special- Purpose Simulation):** Ο διαχωρισμός γίνεται ανάλογα με το σκοπό χρήσης του κάθε μοντέλου. Αν αφορούν πολλές εφαρμογές και προσομοιώνουν συστήματα που δεν ανήκουν σε ένα μόνο τομέα (π.χ. κατασκευαστικά, βιομηχανικά κτλ.) ονομάζονται μοντέλα γενικού σκοπού, ενώ αν αφορούν μία συγκεκριμένη εφαρμογή (π.χ. τηλεπικοινωνίες, παροχή υπηρεσιών υγείας, κατασκευές κτλ.) ονομάζονται μοντέλα ειδικού σκοπού. Ωστόσο υπάρχουν συστήματα που είναι γενικού σκοπού αλλά έχουν ειδικές δυνατότητες για συγκεκριμένες εφαρμογές.
- ➔ **Αντικειμενοστραφής προσομοίωση (Object- Oriented Simulation):** Στα μοντέλα προσομοίωσης αυτού του τύπου χρησιμοποιούνται αντικείμενα που αλληλεπιδρούν μεταξύ τους καθώς η προσομοίωση εξελίσσεται στο χρόνο. Τα αντικείμενα εμπεριέχουν δεδομένα που περιγράφουν την κατάσταση τους και εκτελούν διάφορες λειτουργίες μέσω κατάλληλα διαμορφωμένων μεθόδων.
- ➔ **Κατανεμημένη προσομοίωση (Distributed Simulation):** Η κατανεμημένη προσομοίωση αφορά στη δημιουργία ενός γενικευμένου μοντέλου προσομοίωσης το οποίο είναι σύνθεση ενός ή περισσοτέρων μοντέλων προσομοίωσης. Τα μοντέλα αυτά «τρέχουν» σε ένα δίκτυο υπολογιστών, επιδιώκοντας τη σημαντική μείωση του χρόνου εκτέλεσης. Για τη διασύνδεση αυτών των κατανεμημένων προσομοιωτών έχουν δημιουργηθεί ειδικά

πρωτόκολλα και πρότυπα (Distributed Interactive Simulation (DIS) standard, High Level Architecture (HLA)).

## 2.5 Διαδικασία μελέτης προσομοίωσης

Παρακάτω αναλύονται τα βασικά στάδια που πρέπει να ακολουθηθούν για την εκτέλεση μίας μελέτης προσομοίωσης, ώστε να αποφευχθούν τα προβλήματα που αναφέρθηκαν παραπάνω.



- ➔ Αρχικά γίνεται **διαμόρφωση του προβλήματος**. Ο αναλυτής πρέπει να είναι βέβαιος ότι έχει κατανοήσει ακριβώς το πρόβλημα.

- ➔ **Καθορισμός των στόχων και του γενικότερου σχεδίου του έργου**, καθορίζονται δηλαδή τα ζητούμενα και οι στόχοι της μελέτης και αποφασίζεται αν η προσομοίωση είναι η κατάλληλη μέθοδος επίλυσης του προβλήματος. Το σχέδιο μελέτης πρέπει να περιλαμβάνει στοιχεία όπως τα εναλλακτικά συστήματα που θα εξεταστούν, τους δείκτες αποδοτικότητας (benchmarks) για την αξιολόγησή τους, αλλά και στοιχεία σχετικά με το χρονικό προγραμματισμό του έργου και τις απαιτήσεις σε μέσα παραγωγής. Τέλος, πριν προχωρήσουμε στην υλοποίηση της προσομοίωσης επιλέγεται το λογισμικό που θα χρησιμοποιηθεί.
- ➔ Στη συνέχεια γίνεται **εννοιολογική σύλληψη του μοντέλου**. Σε αυτό το στάδιο επιλέγεται ο βαθμός στον οποίο το μοντέλο θα αναπαριστά το σύστημα. Αρχικά συγκεντρώνονται τα σημαντικότερα στοιχεία του προβλήματος, μοντελοποιούνται τα κυριότερα στοιχεία του συστήματος και ενσωματώνονται οι βασικές παραδοχές των λειτουργιών του. Στη συνέχεια αυτό το απλοποιημένο μοντέλο εμπλουτίζεται μέχρι να αποτελεί μία ικανοποιητική προσέγγιση της πραγματικότητας. Η καθολική αναπαράσταση του συστήματος συνήθως οδηγεί σε υπερβολικούς χρόνους ανάπτυξης και αύξηση του κόστους.
- ➔ Έπειτα γίνεται **συλλογή δεδομένων** σχετικά με τη δομή και τα λειτουργικά χαρακτηριστικά του συστήματος και προσδιορίζονται, στο βαθμό που είναι αυτό δυνατό, οι βασικές παράμετροι του μοντέλου καθώς και οι κατάλληλες κατανομές των δεδομένων εισόδου. Οι στόχοι της μελέτης υποδεικνύουν και το είδος των στοιχείων που πρέπει να συγκεντρωθούν. Όσο αλλάζει η πολυπλοκότητα του μοντέλου τόσο αλλάζουν και τα στοιχεία που πρέπει να συγκεντρωθούν.
- ➔ Στη συνέχεια **αναπτύσσεται το τυπικό μοντέλο**, δηλαδή τα χαρακτηριστικά που επιλέχθηκαν για τη μοντελοποίηση υλοποιούνται με τον καθορισμό των κατάλληλων συστατικών στοιχείων (οντότητες, πόροι κτλ.).
- ➔ Στην περίπτωση βέβαια που το μοντέλο κριθεί επαρκές, ο αναλυτής προχωρά στη **δημιουργία του περιβάλλοντος προγραμματισμού**, είτε με χρήση μίας γλώσσας προγραμματισμού είτε με χρήση ενός κατάλληλου λογισμικού προσομοίωσης.
- ➔ Αφού υλοποιηθεί το μοντέλο σε προγραμματιστικό περιβάλλον ακολουθεί **επαλήθευση (verification)** για το αν η προσομοίωση λειτουργεί σωστά, δηλαδή αν οι παράμετροι εισόδου και η λογική δομή του μοντέλου απεικονίζονται σωστά στο πρόγραμμα.

- ➔ Στη συνέχεια **πραγματοποιούνται δοκιμές.**
- ➔ Μετά τις δοκιμαστικές εκτελέσεις του προγράμματος εξετάζεται αν το μοντέλο αναπαριστά σωστά το πραγματικό σύστημα. Η **επικύρωση (validation)** πραγματοποιείται μετά από επαναληπτική διαδικασία σύγκρισης του μοντέλου με την πραγματική συμπεριφορά του συστήματος και χρησιμοποιώντας τις διαφορές μεταξύ των δύο για τη βελτίωση του μοντέλου. Αυτή η διαδικασία επαναλαμβάνεται μέχρις ότου η ακρίβεια του μοντέλου κριθεί αποδεκτή.
- ➔ Έπειτα γίνεται **σχεδιασμός των πειραμάτων**, δηλαδή προσδιορίζονται τα εναλλακτικά σενάρια, ο αριθμός επαναλήψεων για κάθε σενάριο, η χρονική διάρκεια προετοιμασίας του μοντέλου και η χρονική διάρκεια της κάθε επανάληψης.
- ➔ Αφού έχουν ολοκληρωθεί τα παραπάνω βήματα προχωράμε στην **εκτέλεση της προσομοίωσης.**
- ➔ Αφού ολοκληρωθεί ο προκαθορισμένος αριθμός επαναλήψεων της προσομοίωσης, γίνονται **πρόσθετες επαναλήψεις** αν αυτό είναι απαραίτητο και καθορίζεται ο σχεδιασμός που θα ακολουθηθεί γι' αυτές.
- ➔ Από τα εξαγόμενα της προσομοίωσης γίνεται **ανάλυση εξόδου**, με σκοπό να προσδιοριστεί η αποδοτικότητα συγκεκριμένων διατάξεων του συστήματος και να συγκριθούν οι διατάξεις μεταξύ τους.
- ➔ Έπειτα, γίνεται **τεκμηρίωση και παρουσίαση των αποτελεσμάτων.** Η τεκμηρίωση του προγράμματος είναι απαραίτητη επειδή διευκολύνει τη χρήση του προγράμματος από άλλο αναλυτή ή χρήστη βοηθώντας τον να κατανοήσει τη λειτουργία του μοντέλου και προσδίδει αξιοπιστία στη μελέτη. Ο αναλυτής οφείλει να εξηγήσει τη διαδικασία δημιουργίας και επικύρωσης του μοντέλου ώστε τα αποτελέσματα της μελέτης να χρησιμοποιηθούν στη λήψη αποφάσεων.
- ➔ Τέλος γίνεται **χρήση των αποτελεσμάτων.** Η επιτυχία αυτής της φάσης εξαρτάται από το πόσο καλά έχουν πραγματοποιηθεί τα υπόλοιπα βήματα. Επίσης, εξαρτάται από το πόσο συμμετείχε ο τελικός χρήστης σε όλη τη διαδικασία της προσομοίωσης. Εάν ο χρήστης έχει κατανοήσει τη φύση του μοντέλου και των αποτελεσμάτων του, η πιθανότητα της εφαρμογής των αποτελεσμάτων ενισχύεται. Αντίθετα, εάν το μοντέλο και οι βασικές υποθέσεις δεν έχουν κοινοποιηθεί σωστά, η χρήση των αποτελεσμάτων δεν είναι πολύ πιθανή, ανεξάρτητα από την εγκυρότητα του μοντέλου προσομοίωσης.

## 2.6 Προσομοίωση διακριτών συστημάτων (Discrete Event Simulation)

Από τα παραπάνω μοντέλα προσομοίωσης, πολύ σημαντικά είναι αυτά που αναλύουν διακριτά συστήματα λόγω της ευρείας εφαρμογής τους στη διαχείριση τεχνικών έργων. Γι' αυτό το λόγο είναι αυτά που θα αναλυθούν περαιτέρω.

Στην προσομοίωση διακριτών συστημάτων, περιγράφεται ένα σύστημα που εξελίσσεται με το χρόνο και η κατάστασή του αλλάζει ασυνεχώς με την εμφάνιση κάποιου συμβάντος. Λόγω του δυναμικού χαρακτήρα των μοντέλων αυτών πρέπει να υπάρχει δυνατότητα αποθήκευσης της τρέχουσας τιμής του προσομοιωμένου χρόνου, ενώ είναι απαραίτητος και ένας μηχανισμός αύξησής του από μία τιμή σε μία άλλη. Το τμήμα της προσομοίωσης που είναι υπεύθυνο για τον έλεγχο του χρόνου ονομάζεται **ρολόι προσομοίωσης (simulation clock)** και υπάρχουν δύο βασικές προσεγγίσεις για τον έλεγχο αυτό, η εξέλιξη με βάση το χρόνο του επομένου γεγονότος (next-event time advance) και η εξέλιξη σταθερής αύξησης του χρόνου (fixed-increment time advance).

- ➔ **Μηχανισμός επόμενου γεγονότος (next-event time advance):** Αρχικά το ρολόι προσομοίωσης αρχικοποιείται στο μηδέν και καθορίζονται οι στιγμές εμφάνισης των μελλοντικών γεγονότων. Το ρολόι της προσομοίωσης προχωρά στη χρονική στιγμή του επόμενου γεγονότος προσπερνώντας όλον τον ενδιάμεσο χρόνο, κατά τον οποίο δεν συμβαίνει τίποτα. Η εφαρμογή περιλαμβάνει την ύπαρξη ενός καταλόγου ή λίστας γεγονότων, όπου καταγράφονται τα γεγονότα που πρόκειται να συμβούν στο μέλλον. Η διαδικασία εξέλιξης του ρολογιού προσομοίωσης από το ένα γεγονός στο άλλο, συνεχίζεται μέχρι να ικανοποιηθεί κάποια προκαθορισμένη συνθήκη τερματισμού της προσομοίωσης.
- ➔ **Μηχανισμός σταθερού χρονικού διαστήματος (fixed-increment time advance):** Σύμφωνα με αυτόν τον μηχανισμό το ρολόι της προσομοίωσης αυξάνει κατά ένα μικρό και σταθερό χρονικό διάστημα. Όλα τα γεγονότα που εμφανίζονται τη συγκεκριμένη χρονική στιγμή αλλά και όλα όσα προηγήθηκαν αυτής θεωρείται ότι συμβαίνουν κατά τη συγκεκριμένη χρονική στιγμή.

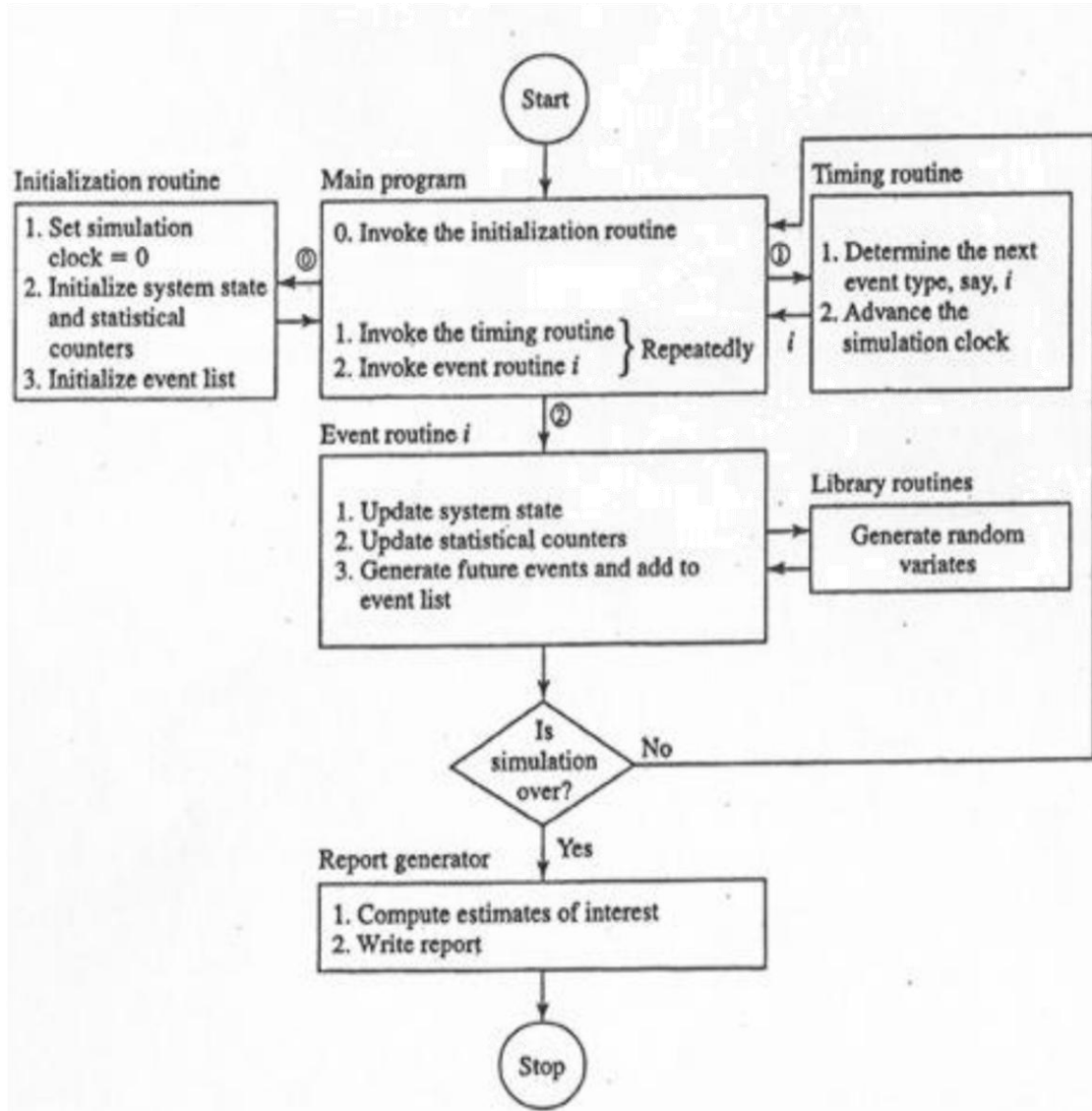
Ο μηχανισμός επόμενου γεγονότος χρησιμοποιείται κυρίως σε διακριτά συστήματα, ενώ ο μηχανισμός σταθερού διαστήματος χρησιμοποιείται για την προσομοίωση συνεχών συστημάτων.

### 2.6.1 Συστατικά του μοντέλου προσομοίωσης

Η πλειοψηφία των μοντέλων προσομοίωσης διακριτών συστημάτων, που ελέγχονται με μηχανισμούς επόμενου γεγονότος και είναι γενικού σκοπού αποτελούνται από τα παρακάτω συστατικά στοιχεία:

- ➔ **Κατάσταση συστήματος (system state)** είναι το σύνολο των μεταβλητών κατάστασης που είναι απαραίτητες για την περιγραφή του συστήματος σε μία χρονική στιγμή.
- ➔ **Ρολόι προσομοίωσης (simulation clock)** είναι η μεταβλητή που επιστρέφει τον τρέχοντα χρόνο της προσομοίωσης.
- ➔ **Λίστα γεγονότων (event list)** είναι η λίστα που καθορίζει τη χρονική αλληλουχία των γεγονότων.
- ➔ **Στατιστικοί μετρητές (statistical counters)** είναι οι μεταβλητές που χρησιμοποιούνται για την αποθήκευση στατιστικών μετρήσεων για την απόδοση του συστήματος.
- ➔ **Ρουτίνα αρχικοποίησης (initialization routine)** είναι ένα υποπρόγραμμα που αρχικοποιεί το μοντέλο προσομοίωσης τη χρονική στιγμή μηδέν.
- ➔ **Ρουτίνα χρονισμού (timing routine)** είναι ένα υποπρόγραμμα που προσδιορίζει το επόμενο γεγονός από τη λίστα γεγονότων και στη συνέχεια αυξάνει το ρολόι της προσομοίωσης στη χρονική στιγμή που το γεγονός αυτό θα εμφανισθεί.
- ➔ **Ρουτίνα γεγονότων (event routine)** είναι ένα υποπρόγραμμα που ενημερώνει την κατάσταση του συστήματος κάθε φορά που εμφανίζεται ένας συγκεκριμένος τύπος γεγονότος (υπάρχει μία τέτοια ρουτίνα για κάθε τύπο γεγονότος).
- ➔ **Βιβλιοθήκη ρουτινών (library routines)** είναι ένα σύνολο από υποπρογράμματα που χρησιμοποιούνται για την παραγωγή τυχαίων δειγμάτων από κατανομές πιθανοτήτων που έχουν καθοριστεί για το συγκεκριμένο μοντέλο προσομοίωσης.
- ➔ **Γεννήτρια αναφορών (report generator)** είναι ένα υποπρόγραμμα που πραγματοποιεί υπολογισμούς των επιθυμητών μεγεθών απόδοσης του συστήματος και τα παρουσιάζει σε μορφή αναφοράς μετά το τέλος της προσομοίωσης.

- ➔ **Κύριο πρόγραμμα (main program)** είναι το πρόγραμμα που καλεί τη ρουτίνα χρονισμού για να καθορίσει το επόμενο γεγονός και μετά μεταβιβάζει τον έλεγχο στην αντίστοιχη ρουτίνα γεγονότος που θα ενημερώσει την κατάσταση του συστήματος. Ελέγχει επίσης αν πρέπει να τερματιστεί η προσομοίωση και ενεργοποιεί τη γεννήτρια αναφορών όταν η προσομοίωση τερματιστεί.



Η λογική οργάνωση των παραπάνω εννοιών και η εκτέλεση της προσομοίωσης εξαρτάται από τη μέθοδο προσομοίωσης που θα επιλεγεί.



### 2.6.2 Μέθοδοι προσομοίωσης

Έχουν προταθεί διάφοροι μηχανισμοί για την προσομοίωση διακριτών συστημάτων γενικού σκοπού με χρήση μηχανισμών επόμενου γεγονότος, μεταξύ των οποίων είναι η προσέγγιση διεργασιών (process-based approach), η προσέγγιση συμβάντων (event-based approach), η προσέγγιση δραστηριοτήτων (activity-based approach) και η προσέγγιση τριών φάσεων (three-phase approach) (Pidd, 1998).

Η μέθοδος που θα ακολουθηθεί για την πραγματοποίηση της μελέτης επηρεάζει τόσο την ανάπτυξη του λογισμικού όσο και τον τρόπο μοντελοποίησης του συστήματος.

Παρακάτω αναλύονται οι μέθοδοι προσομοίωσης και η καταλληλότητά τους για τους τύπους συστημάτων και μοντέλων.

- ➔ **Προσέγγιση διεργασιών (process-based approach):** Αυτή η μεθοδολογία βασίζεται στην εκτέλεση διεργασιών που περιγράφουν την προσομοίωση ανεξάρτητων στοιχείων ή τμημάτων του μοντέλου. Το πρόγραμμα προσομοιώνει τη ροή μίας οντότητας μέσα στο σύστημα, η οποία, αν είναι δυναμική, προχωρά έως ότου καθυστερήσει σε κάποιον κόμβο, εισαχθεί σε μία δραστηριότητα ή βγει από το σύστημα, ή αν είναι στατική δεν κινείται. Όταν η οντότητα σταματήσει κάπου, ο χρόνος προσομοίωσης αυξάνεται στη χρονική στιγμή που αντιστοιχεί στην επόμενη κίνηση μίας άλλης οντότητας. Πιο συγκεκριμένα, οι διεργασίες επικοινωνούν μεταξύ τους με τη χρήση σημάτων που προσομοιώνουν την επίδραση του ενός στοιχείου του μοντέλου πάνω σε άλλο. Υπάρχουν δύο είδη λιστών γεγονότων τα οποία εκτελούνται σε χρονολογική σειρά. Το ένα είδος χρησιμοποιείται από τα διάφορα τμήματα, ενώ το άλλο είναι γενική χρονική λίστα, την οποία διαχειρίζεται το πρόγραμμα ελέγχου και περιέχει τα σήματα διατμηματικής επικοινωνίας. Η ροή ή κίνηση των οντοτήτων του συστήματος περιγράφει όλες τις καταστάσεις στις οποίες μπορεί να περιέλθει ένα αντικείμενο του συστήματος (δεσμεύουν, χρησιμοποιούν ή αποδεσμεύουν μέσα παραγωγής). Η προσέγγιση διεργασιών είναι κατάλληλη για συστήματα, όπου οι οντότητες μπορούν να διαφοροποιηθούν στη βάση των διαφορετικών ιδιοτήτων που κάθε μία κατέχει. Αντίστοιχα, τα μέσα παραγωγής πρέπει να έχουν λίγες ιδιότητες, περιορισμένο αριθμό καταστάσεων (π.χ. αδρανής, απασχολημένος, μη-διαθέσιμος) και να μην αλληλεπιδρούν μεταξύ τους.

- ➔ **Προσέγγιση συμβάντων (event-based approach):** Το σύστημα περιγράφεται στη βάση συγκεκριμένων συμβάντων, κάθε ένα από τα οποία αντιστοιχεί σε μία ξεχωριστή ρουτίνα γεγονότων. Η μεθοδολογία στηρίζεται στην αύξηση του χρόνου της προσομοίωσης όταν συμβαίνει κάποιο επόμενο γεγονός. Όταν συμβαίνει ένα γεγονός απελευθερώνονται κάποια μέσα παραγωγής που με τη σειρά τους προκαλούν τον προγραμματισμό και την εκτέλεση και άλλων ρουτινών. Τότε, η ρουτίνα γεγονότων επαναπροσδιορίζει τις διαθέσιμες οντότητες του συστήματος και προγραμματίζει χρονικά τις δραστηριότητες στις οποίες μπορούν να συμμετάσχουν. Θεωρείται αποδοτική μέθοδος και συχνά συνδυάζεται με την προσέγγιση δραστηριοτήτων ή και διεργασιών.
- ➔ **Προσέγγιση δραστηριοτήτων (activity-based approach):** ή μεθοδολογία των δύο φάσεων. Η φιλοσοφία της μεθόδου βασίζεται στο ότι ο αναλυτής θα πρέπει να αναγνωρίζει τα αντικείμενα (οντότητες) του συστήματος, τις δραστηριότητες που αυτά εκτελούν και τις προϋποθέσεις που πρέπει να ικανοποιηθούν για να πραγματοποιηθεί η εκτέλεσή τους. Έτσι το πρόγραμμα της προσομοίωσης αποτελείται από ανεξάρτητες λειτουργικές μονάδες οι οποίες αναμένουν την ενεργοποίησή τους. Ο έλεγχος των κριτηρίων γίνεται σε συγκεκριμένες χρονικές στιγμές και καθορίζει εάν ένα γεγονός μπορεί να πραγματοποιηθεί ή όχι. Εάν το γεγονός πραγματοποιηθεί, τότε η κατάσταση του συστήματος μεταβάλλεται. Η προσέγγιση δραστηριοτήτων μοντελοποιεί ικανοποιητικά συστήματα στα οποία τα διάφορα συστατικά μέρη (components) αλληλεπιδρούν έντονα μεταξύ τους και καθορίζονται από πολύπλοκες αρχικές συνθήκες (π.χ. πολλές οντότητες με πλήθος ιδιοτήτων και καταστάσεων που πρέπει να συνεργαστούν σε δυναμικό περιβάλλον).
- ➔ **Προσέγγιση τριών φάσεων (three-phase approach):** Η μέθοδος είναι εξέλιξη της προσέγγισης δραστηριοτήτων και βελτιώνει την αποδοτικότητά της. Διαχωρίζει δύο ειδών δραστηριότητες, αυτές που θα πραγματοποιηθούν σε συγκεκριμένη χρονική στιγμή (δραστηριότητες B) και αυτές που θα πραγματοποιηθούν όταν συμβεί κάποια άλλη δραστηριότητα (δραστηριότητες C). Ο χρόνος πραγματοποίησης των ανεξάρτητων δραστηριοτήτων δεν καθορίζεται επακριβώς από το μοντέλο ούτε πρέπει να είναι γνωστός πριν από την εκκίνηση της προσομοίωσης. Ο χρόνος πολλές φορές υπολογίζεται από την εκτέλεση ενός άλλου γεγονότος. Κάθε μία από τις τρεις φάσεις πραγματοποιεί μία συγκεκριμένη λειτουργία. Η πρώτη φάση ελέγχει τους χρόνους πραγματοποίησης των γεγονότων και αναζητά εκείνες τις

δραστηριότητες που θα συμβούν στον τρέχοντα χρόνο της προσομοίωσης. Η δεύτερη φάση εξασφαλίζει ότι όλες οι δραστηριότητες που πρέπει να εκτελεστούν την τρέχουσα χρονική στιγμή της προσομοίωσης εκτελούνται σωστά. Μετά ακολουθεί η τρίτη φάση όπου ελέγχεται εάν οι εξαρτημένες δραστηριότητες μπορούν να πραγματοποιηθούν, δηλ. εάν ικανοποιούνται τα κριτήρια της πραγματοποίησής τους. Όταν καμία δραστηριότητα δεν μπορεί να πραγματοποιηθεί πια, το ρολόι αυξάνει τον χρόνο της προσομοίωσης και άλλος ένας κύκλος τριών φάσεων ξεκινά μέχρι και την ολοκλήρωση της προσομοίωσης. Η μέθοδος θεωρείται καλύτερη σε σχέση με την προσέγγιση δραστηριοτήτων, γιατί ο έλεγχος ικανοποίησης κριτηρίων γίνεται μόνο για τις δραστηριότητες C, για τις οποίες και έχει νόημα να γίνει. Συνεπώς, ο μηχανισμός της προσομοίωσης δεν αναλώνεται σε άσκοπους ελέγχους των B δραστηριοτήτων και η απόδοση του συστήματος αυξάνεται.

### **2.6.3 Μεταβάσεις κατάστασης μοντέλων διακριτών συστημάτων**

Στο παράδειγμα προσομοίωσης διακριτών συμβάντων, το πρότυπο προσομοίωσης βρίσκεται σε μία κατάσταση S σε οποιοδήποτε στιγμή στο χρόνο. Η κατάσταση ενός συστήματος είναι ένα σύνολο πληροφοριών που περιλαμβάνουν τις εμφανείς μεταβλητές του συστήματος και επιτρέπει την περιγραφή της εξέλιξης του συστήματος στο χρόνο. Σε ένα πρόγραμμα προσομοίωσης, η κατάσταση του συστήματος αποθηκεύεται σε μια ή περισσότερες μεταβλητές που αντιπροσωπεύουν διάφορες δομές δεδομένων (π.χ., ο αριθμός πελατών σε μια ουρά αναμονής, ή η ακριβής σειρά τους στην αναμονή). Κατά συνέπεια, η κατάσταση μπορεί να καθοριστεί με διάφορους τρόπους, ανάλογα με τις ανάγκες του μοντέλου, και το απαραίτητο επίπεδο λεπτομέρειας ενσωματώνεται στο μοντέλο.

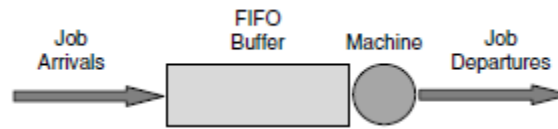
Τα παραδείγματα που ακολουθούν βρίσκονται στο βιβλίο T. Altiook, B. Melamed (2007), *Simulation Modeling and Analysis with Arena*, USA: Elsevier, Inc.

#### **Παράδειγμα 1**

##### **Μία μηχανή**

Για παράδειγμα, θεωρούμε μια μηχανή (χωρίς το ενδεχόμενο βλάβης) η οποία τροφοδοτείται από μία πηγή. Τα στοιχεία που βρίσκουν τη μηχανή απασχολημένη (να εκτελεί μια άλλη εργασία) πρέπει να περιμένουν, στη συνέχεια υποβάλλονται σε

επεξεργασία κατά τη σειρά άφιξής τους κατά το σύστημα FIFO (First In First Out) ή FCFS (First Come First Served), και το προκύπτον σύστημα καλείται ουρά αναμονής ή σύστημα αναμονής. Υποθέτουμε ότι οι χρόνοι μεταξύ των αφίξεων και οι χρόνοι επεξεργασίας δίνονται (ενδεχομένως τυχαία).



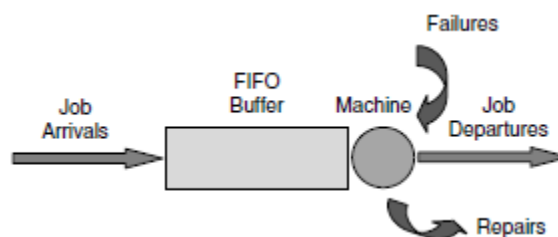
Σχήμα 2.6.3.1

Για την αναπαράσταση αυτού του συστήματος ως DES (Discrete Event Simulation), καθορίζεται η κατάσταση  $S(t)$  ως ο αριθμός εργασιών στο σύστημα στο χρόνο  $t$ . Επομένως,  $S(t)=5$  σημαίνει ότι στο χρόνο  $t$ , η μηχανή είναι απασχολημένη αφού εκτελεί την πρώτη εργασία και 4 ακόμα εργασίες είναι σε αναμονή. Υπάρχουν δύο τύποι γεγονότων: αφίξεις και ολοκληρώσεις διαδικασίας. Έστω ότι πραγματοποιήθηκε μια άφιξη στο χρόνο  $t$ , όταν υπήρχαν  $S(t)=n$  εργασίες στο σύστημα. Κατόπιν η τιμή του  $S$  στο χρόνο  $t$  γίνεται από  $n \rightarrow n+1$ . Ομοίως, η ολοκλήρωση μίας διαδικασίας περιγράφεται από τη μετάβαση  $n \rightarrow n-1$ .

## Παράδειγμα 2

### Μία μηχανή με ενδεχόμενο βλάβης

Έστω ότι η προηγούμενη μηχανή έχει τώρα το ενδεχόμενο βλάβης. Εκτός από τις διαδικασίες άφιξης και επεξεργασίας, τώρα πρέπει να περιγραφούν και οι χρόνοι βλάβης και επισκευής. Γίνεται η υπόθεση ότι η μηχανή παθαίνει βλάβη μόνο όταν εκτελεί μια εργασία, και ότι μετά την ολοκλήρωση της επισκευής, η εργασία πρέπει να εκτελεστεί από την αρχή.



Σχήμα 2.6.3.2

Η κατάσταση  $S(t)$  είναι ένα ζεύγος μεταβλητών,  $S(t) = (N(t), V(t))$ , όπου  $N(t)$  είναι ο αριθμός εργασιών στην αναμονή, και  $V(t)$  η κατάσταση της διαδικασίας, δηλαδή μη απασχολημένη - idle, απασχολημένη - busy, χαλασμένη - down, όλες στο χρόνο  $t$ . Σε ένα πρόγραμμα προσομοίωσης, το  $V(t)$  κωδικοποιείται, για παράδειγμα με ακέραιους αριθμούς, ως εξής  $0=$ idle,  $1=$ busy, and  $2=$ down. Μία εργασία παραμένει στη μηχανή όταν η κατάστασή της είναι busy ή down.

Τα γεγονότα είναι αφίξεις, ολοκληρώσεις διαδικασίας, βλάβες μηχανών, και επισκευές μηχανών.

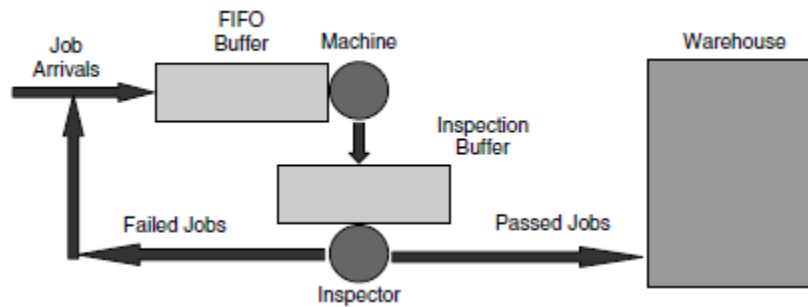
Οι μεταβάσεις της κατάστασης είναι:

- ➔ Άφιξη γεγονότος:  $(n, v) \rightarrow (n + 1, v)$
- ➔ Ολοκλήρωση διαδικασίας:  $(n, 1) \rightarrow \begin{cases} (0, 0), & \text{αν } n = 1 \\ (n - 1, 1), & \text{αν } n > 1 \end{cases}$
- ➔ Εμφάνιση βλάβης:  $(n, 1) \rightarrow (n, 2)$
- ➔ Ολοκλήρωση επισκευής:  $(n, 2) \rightarrow (n, 1)$

### Παράδειγμα 3

#### **Μία μηχανή με σταθμό επιθεώρησης και σχετικό απόθεμα**

Θεωρούμε τη μηχανή χωρίς το ενδεχόμενο βλάβης. Όταν ολοκληρώνεται η επεξεργασία των στοιχείων αυτά πηγαίνουν σε ένα σταθμό επιθεώρησης που έχει δική του ουρά αναμονής, όπου τα επεξεργασμένα στοιχεία ελέγχονται για ελαττώματα. Τα στοιχεία που περνούν τον έλεγχο αποθηκεύονται σε μια αποθήκη τελικών προϊόντων. Ωστόσο, τα στοιχεία που αποτυγχάνουν οδηγούνται πίσω στο τέλος της ουράς αναμονής της μηχανής για επανεπεξεργασία. Εκτός από τους χρόνους μεταξύ των αφίξεων και τους χρόνους επεξεργασίας, χρειαζόμαστε εδώ μια περιγραφή του χρόνου επιθεώρησης καθώς επίσης και ένα μηχανισμό για την απόφαση του ελέγχου (pass/fail) (π.χ., οι εργασίες αποτυγχάνουν με κάποια πιθανότητα, ανεξάρτητα η μια από την άλλη).



Σχήμα 2.6.3.3

Η κατάσταση  $S(t)$  είναι συνδυασμός τριών μεταβλητών,  $S(t) = (N(t), I(t), K(t))$ , όπου  $N(t)$  είναι ο αριθμός στοιχείων στη μηχανή και την ουρά αναμονής της,  $I(t)$  είναι ο αριθμός στοιχείων στο σταθμό επιθεώρησης, και  $K(t)$  είναι το περιεχόμενο προς αποθήκευση, όλα στο χρόνο  $t$ . Τα συμβάντα μπορεί να είναι αφίξεις, ολοκληρώσεις διαδικασίας, αποτυχίες στην επιθεώρηση (που ακολουθούνται με τη δρομολόγηση στο τέλος της ουράς αναμονής της μηχανής), και την επιτυχή επιθεώρηση (που ακολουθείται από την αποθήκευση στην αποθήκη ετοιμών).

Οι μεταβάσεις της κατάστασης είναι:

- ➔ Άφιξη γεγονότος:  $(n, i, k) \rightarrow (n + 1, i, k)$
- ➔ Ολοκλήρωση διαδικασίας:  $(n, i, k) \rightarrow (n - 1, i + 1, k)$
- ➔ Ολοκλήρωση ελέγχου:

$$(n, i, k) \rightarrow \left\{ \begin{array}{ll} (n + 1, i - 1, k), & \text{αν το στοιχείο απέτυχε στον έλεγχο} \\ (n, i - 1, k + 1), & \text{αν το στοιχείο πέρασε τον έλεγχο} \end{array} \right\}$$

#### 2.6.4 Δειγματοληψία και παραδείγματα Monte Carlo

Τα μοντέλα προσομοίωσης Monte Carlo χρησιμοποιούν τυχαίες αφίξεις με τη βοήθεια στατιστικών κατανομών. Οι αλγόριθμοι ή/και ο κώδικάς τους χρησιμοποιούν τυχαίες γεννήτριες αριθμών (RNG) που παράγουν ομοιόμορφα κατανομημένες τιμές («εξίσου πιθανές») μεταξύ 0 και 1. Αυτές οι τιμές μετασχηματίζονται έπειτα για να προσαρμοστούν σε κάποια κατανομή.

Τα μοντέλα DES χρησιμοποιούν μια στατιστική προσέγγιση στην αξιολόγηση της απόδοσης συστημάτων. Στην πραγματικότητα η αξιολόγηση απόδοσης που βασίζεται στην προσομοίωση μπορεί να θεωρηθεί σαν στατιστικό πείραμα. Συνεπώς,

οι απαραίτητοι δείκτες απόδοσης του υπό μελέτη μοντέλου δεν υπολογίζονται ακριβώς, αλλά υπολογίζονται από ένα σύνολο υποθέσεων. Μια τυπική στατιστική διαδικασία είναι η εξής:

1. Ο σχεδιαστής εκτελεί πολλαπλά τρέξιμα της προσομοίωσης του υπό μελέτη μοντέλου, με χρήση τυχαίων αριθμών. Κάθε τρέξιμο λέγεται επανάληψη.

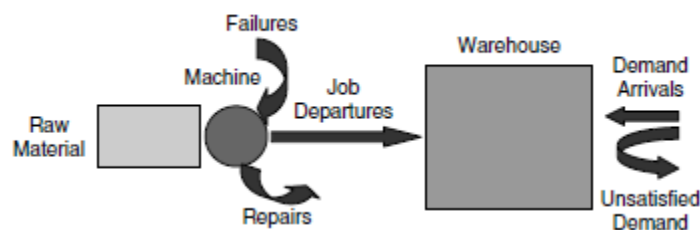
2. Από κάθε επανάληψη υπολογίζονται ένας ή περισσότεροι δείκτες απόδοσης. Τα παραδείγματα περιλαμβάνουν το μέσο χρόνο αναμονής στην ουρά, τα μέσα επίπεδα των εκτελούμενων εργασιών WIP (work in process), και τις πιθανότητες χρόνου διακοπής της μηχανής.

3. Οι τιμές της απόδοσης που έχουν δοθεί είναι τυχαίες και ανεξάρτητες, και όλες μαζί διαμορφώνουν ένα στατιστικό δείγμα. Για μια πιο αξιόπιστη εκτίμηση της πραγματικής τιμής κάθε μέτρου απόδοσης, υπολογίζονται οι αντίστοιχες τιμές κατά μέσο όρο και ορίζονται τα διαστήματα εμπιστοσύνης.

### Παράδειγμα

#### Σταθμός εργασιών που υπόκειται σε βλάβες και σε έλεγχο

Θα παρουσιαστεί ένα λεπτομερές παράδειγμα που αποτυπώνει την τυχαία φύση των μοντέλων DES και των επαναλήψεων της προσομοίωσης, καθώς και την τυχαία κατάσταση του συστήματος και τις πρότυπες διαδρομές. Σκοπός είναι να μελετηθεί η συμπεριφορά του συστήματος και να υπολογιστούν οι δείκτες της απόδοσης που μας ενδιαφέρουν. Θεωρούμε το ακόλουθο σύστημα.



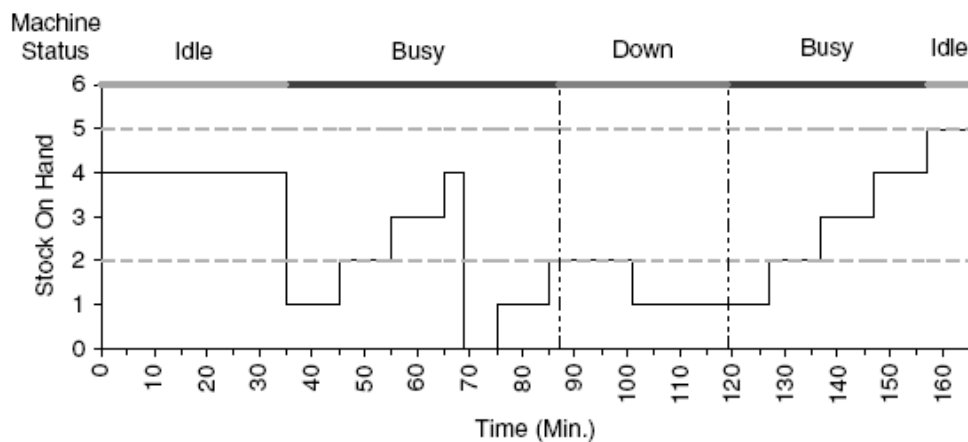
Σχήμα 2.6.4.1

Το σύστημα αποτελείται από μία μηχανή η οποία τροφοδοτείται συνεχώς με εργασίες και μία αποθήκη ετοιμών. Επιπλέον για τη μηχανή υπάρχει το ενδεχόμενο βλάβης και η κατάστασή της διατηρείται στην τυχαία μεταβλητή  $V(t)$ , που δίνεται ως εξής:

$$V(t) = \begin{cases} 0, & \text{idle} \\ 1, & \text{busy} \\ 2, & \text{down} \end{cases}$$

Η κατάσταση  $S(t)$  ορίζεται από ένα ζεύγος μεταβλητών,  $S(t)=(V(t), K(t))$ , όπου  $V(t)$  είναι η κατάσταση της μηχανής όπως περιγράφηκε παραπάνω και  $K(t)$  είναι το ύψος των έτοιμων προϊόντων στην αποθήκη, όλα σε χρόνο  $t$ . Για παράδειγμα, η κατάσταση  $S(t)= (2, 3)$  σημαίνει ότι σε χρόνο  $t$  η κατάσταση της μηχανής είναι down, δηλαδή πιθανώς επισκευάζεται, και η αποθήκη έχει τρία έτοιμα προϊόντα. Οι παραγγελίες φτάνουν στην αποθήκη και ικανοποιούνται με το ζητούμενο αριθμό ετοιμων προϊόντων. Παραγγελίες που υπερβαίνουν τον αριθμό αποθεμάτων ικανοποιούνται μερικώς και δεν υπάρχει επιλογή συμπλήρωσής τους. Ο χρόνος επεξεργασίας ενός στοιχείου είναι 10 λεπτά. Σε αυτό το παράδειγμα, η μηχανή δεν λειτουργεί ανεξάρτητα αλλά ελέγχεται από την αποθήκη ως εξής: Όταν το ύψος του αποθέματος φτάνει ή πέφτει κάτω από δύο μονάδες (σημείο αναπαραγγελίας), η αποθήκη στέλνει αίτημα στη μηχανή για εφοδιασμό μέχρι το επίπεδο  $R= 5$  μονάδων (target level ή base-stock level). Η μηχανή ξεκινά την επεξεργασία μίας σειράς διεργασιών μέχρι το ύψος του αποθέματος να φτάσει την τιμή  $R$ , και τότε σταματά τη λειτουργία της. Αυτό το καθεστώς ελέγχου είναι γνωστό ως πολιτική  $(r, R)$  συνεχούς ελέγχου αποθέματος, και η αντίστοιχη πολιτική αναφέρεται ως σύστημα pull.

Έστω ότι τα γεγονότα του DES μοντέλου του παραπάνω σταθμού εργασιών εμφανίζονται με τη σειρά που φαίνεται στο ακόλουθο σχήμα, το οποίο απεικονίζει το διαθέσιμο απόθεμα της αποθήκης σαν συνάρτηση του χρόνου, και δείχνει επίσης την κατάσταση της μηχανής με την πάροδο του χρόνου.



Σχήμα 2.6.4.2



Η εξέταση του παραπάνω σχήματος δείχνει ότι στο χρόνο  $t=0$ , η μηχανή είναι σε κατάσταση idle και η αποθήκη ετοιμών έχει τέσσερα τελειωμένα προϊόντα, δηλ.  $V(0)=0$  και  $K(0)=4$ . Ο πρώτος πελάτης φθάνει στην αποθήκη ετοιμών στο χρόνο  $t=35$  και ζητά τρεις μονάδες. Δεδομένου ότι το διαθέσιμο απόθεμα μπορεί να ικανοποιήσει αυτήν την παραγγελία, μειώνεται κατά τρεις μονάδες, επομένως  $K(35)=1$ . Το διαθέσιμο απόθεμα πέφτει κάτω από το σημείο αναπαραγγελίας  $r$  και αποστέλλεται αίτημα ξαναγεμίματος της αποθήκης στη μηχανή που προχωρά στην παραγωγή πρόσθετου προϊόντος προκειμένου να ανέβει το διαθέσιμο απόθεμα στην τιμή  $R$ . Η κατάσταση της μηχανής αλλάζει αυτόματα από idle σε busy. Για τα επόμενα 30 λεπτά δεν υπάρχει ζήτηση και το απόθεμα αυξάνεται βαθμιαία όταν τα έτοιμα προϊόντα φθάνουν από τη μηχανή, μέχρι να φθάσει τις 4 μονάδες στο χρόνο  $t=65$ .

Στο χρόνο  $t=69$ , ένας δεύτερος πελάτης φθάνει και δίνει μία παραγγελία ίση ή μεγαλύτερη από το διαθέσιμο απόθεμα, εξαντλώντας όλο το απόθεμα. Δεδομένου ότι η ανικανοποίητη παραγγελία δεν συμπληρώνεται αργότερα, είναι  $K(69)=0$ . Εάν επιτρεπόταν η συμπλήρωση της παραγγελίας (backorder), θα παρακολουθούσαμε το μέγεθος του backorder αναπαριστώντας το σαν αρνητικό απόθεμα.

Στο χρόνο  $t=75$ , η μονάδα που άρχισε να επεξεργάζεται στη μηχανή στο χρόνο  $t=65$  έχει τελειώσει και προχωρά στην αποθήκη ετοιμών, έτσι ώστε  $K(75)=1$ . Μια άλλη μονάδα τελειώνει την επεξεργασία στη μηχανή στο χρόνο  $t=85$ .

Στο χρόνο  $t=87$ , η μηχανή παθαίνει βλάβη και αρχίζει η επισκευή της (κατάσταση down). Η επισκευή ολοκληρώνεται στο χρόνο  $t=119$  και η κατάσταση της μηχανής αλλάζει σε busy. Ενώ η μηχανή επισκευάζεται, φθάνει ένας πελάτης στο χρόνο  $t=101$ , και η παραγγελία του μειώνει το διαθέσιμο απόθεμα κατά μια μονάδα, έτσι ώστε  $K(101)=1$ . Στο χρόνο  $t=119$ , η επισκευασμένη μηχανή συνεχίζει την επεξεργασία της μονάδας που διακόπηκε και τελειώνει στο χρόνο  $t=127$ .

Από το χρόνο  $t=127$  ως το  $t=157$  δεν φθάνει κανένας πελάτης στην αποθήκη ετοιμών, και συνεπώς το απόθεμα φθάνει στο target level  $R=5$ , στο χρόνο  $t=157$ . Η προσομοίωση τελικά ολοκληρώνεται στο χρόνο  $T=165$ .

### **Υπολογισμός στατιστικών στοιχείων**

Έχοντας δημιουργήσει ένα απλό παράδειγμα λειτουργίας ενός συστήματος, μπορούμε τώρα να προχωρήσουμε στον υπολογισμό των σχετικών στατιστικών (δείκτες απόδοσης).

**Πιθανοτική κατανομή της κατάστασης της μηχανής.** Εξετάζεται η κατάσταση της μηχανής στο χρονικό διάστημα  $[0, T]$ . Είναι  $T_1$  ο συνολικός χρόνος για

την κατάσταση *idle*,  $T_B$  ο χρόνος της κατάστασης *busy* και  $T_D$  ο χρόνος της κατάστασης *down* στο διάστημα  $[0, T]$ . Η πιθανοτική κατανομή της κατάστασης της μηχανής υπολογίζεται από τα κλάσματα του χρόνου της κάθε κατάστασης προς το συνολικό χρόνο προσομοίωσης:

$$\Pr\{idle\} = \frac{T_I}{T} = \frac{35 + (165 - 157)}{165} = 0,261$$

$$\Pr\{busy\} = \frac{T_B}{T} = \frac{(87 - 35) + (157 - 119)}{165} = 0,545$$

$$\Pr\{down\} = \frac{T_D}{T} = \frac{119 - 87}{165} = 0,194$$

Όλες οι παραπάνω πιθανότητες υπολογίζονται από τους μέσους χρόνους οι οποίοι εδώ δίνουν την κατάσταση της μηχανής. Η λογική που κρύβεται κάτω από αυτούς τους ορισμούς είναι απλή. Εάν ένας παρατηρητής «εξετάσει» το σύστημα σε μία τυχαία χρονική στιγμή, η πιθανότητα να βρίσκεται η μηχανή σε μία συγκεκριμένη κατάσταση είναι ανάλογη προς το συνολικό χρόνο που δαπανάται από τη μηχανή σε εκείνη την κατάσταση.

**Ρυθμοαπόδοση μηχανών (throughput).** Εξετάζεται ο αριθμός των ολοκληρωμένων εργασιών  $C_T$  στη μηχανή στο διάστημα  $[0, T]$ . Η ρυθμοαπόδοση είναι ένα μέτρο του ποσοστού ολοκληρωμένων εργασιών, δηλαδή, ο αναμενόμενος αριθμός ολοκληρωμένων εργασιών (και, επομένως, αναχωρήσεων) ανά μονάδα χρόνου, που υπολογίζεται ως εξής:

$$\bar{o} = \frac{C_T}{T} = \frac{9}{165} = 0,0545$$

**Επίπεδο εξυπηρέτησης πελατών.** Εξετάζονται οι πελάτες που φθάνουν στην αποθήκη ετοιμών με μια ζήτηση για τα προϊόντα. Ο αριθμός πελατών των οποίων η ζήτηση ικανοποιείται πλήρως στο διάστημα  $[0, T]$  είναι  $N_S$ , και  $N_T$  ο συνολικός αριθμός των πελατών που έφθασε στο  $[0, T]$ . Το επίπεδο εξυπηρέτησης πελατών,  $\xi$ , είναι η πιθανότητα να ικανοποιηθεί πλήρως η ζήτηση μιας άφιξης στην αποθήκη ετοιμών. Αυτός ο δείκτης απόδοσης υπολογίζεται ως εξής:

$$\xi = \frac{N_S}{N_T} = \frac{2}{3} = 0,6667$$

Υποθέτοντας ότι η ζήτηση του πελάτη που φθάνει στο  $t = 69$  δεν ικανοποιείται πλήρως. Ο δείκτης  $\xi$  είναι ένας μέσος όρος πελατών, ο οποίος εκφράζει εδώ τη

μορφή της σχετικής συχνότητας των ικανοποιημένων πελατών. Επιπλέον,  $J_k$  είναι η μη ικανοποιηθείσα μερίδα της ζήτησης του πελάτη  $k$  (ενδεχομένως 0), ο μέσος όρος των οποίων δίνεται από τη σχέση:

$$\bar{J} = \frac{\sum_{k=1}^M J_k}{M},$$

όπου  $M$  είναι ο συνολικός αριθμός των πελατών.



## Κεφάλαιο 3

**Εισαγωγή στο Λογισμικό**

**Προσομοίωσης FlexSim**



### **3.1 Λίγα λόγια για το λογισμικό FlexSim**

Στην παρούσα αναφορά θα χρησιμοποιηθεί σαν λογισμικό προσομοίωσης το FlexSim Simulation Software (Version 4). Πρόκειται για ένα εργαλείο ανάλυσης που βοηθά τους μηχανικούς και τους αναλυτές να λάβουν τις αποφάσεις πάνω στο σχεδιασμό και τη λειτουργία ενός συστήματος.

Το FlexSim είναι λογισμικό προσομοίωσης διακριτών συστημάτων που παρέχει τρισδιάστατη ρεαλιστική γραφική απεικόνιση και εκτενείς εκθέσεις απόδοσης, βοηθώντας στον προσδιορισμό των προβλημάτων και στην αξιολόγηση των εναλλακτικών λύσεων σε μικρό χρονικό διάστημα. Χρησιμοποιώντας το FlexSim για τη διαμόρφωση ενός συστήματος πριν αυτό υλοποιηθεί, ή για τη δοκιμή κάποιας πολιτικής πριν αυτή εφαρμοστεί, αποφεύγονται πολλά από τα λάθη που αντιμετωπίζονται συχνά στο ξεκίνημα ενός νέου συστήματος.

### **3.2 Μοντελοποίηση**

Το FlexSim είναι ένα λογισμικό προσομοίωσης διακριτών συμβάντων. Αυτό σημαίνει ότι χρησιμοποιείται για τη μοντελοποίηση συστημάτων που αλλάζουν κατάσταση σε συγκεκριμένες χρονικές στιγμές ως αποτέλεσμα συγκεκριμένων γεγονότων. Οι συνήθεις καταστάσεις μπορεί να είναι μη απασχολημένο, απασχολημένο, φραγμένο ή χαλασμένο, και κάποια παραδείγματα των γεγονότων θα ήταν η άφιξη των παραγγελιών, η μετακίνηση προϊόντων, και η βλάβη των μηχανών. Τα στοιχεία που επεξεργάζονται με τη βοήθεια της διακριτής προσομοίωσης είναι συχνά φυσικά προϊόντα, αλλά μπορεί και να είναι πελάτες, σχέδια, εργασίες, τηλεφωνήματα, ηλεκτρονικά μηνύματα, κ.λπ. Αυτά τα στοιχεία προχωρούν μέσω μιας σειράς διεργασιών, αναμονής και μεταφορών σε αυτό που λέγεται ροή διαδικασίας. Κάθε βήμα της διαδικασίας μπορεί να απαιτεί έναν ή περισσότερους πόρους όπως μια μηχανή, ένα μεταφορέα, ένα χειριστή, ένα όχημα ή κάποιο εργαλείο. Μερικοί από αυτούς τους πόρους είναι στάσιμοι και μερικοί είναι κινητοί, μερικοί πόροι αφιερώνονται σε μία συγκεκριμένη εργασία και άλλοι πρέπει να μοιραστούν σε πολλαπλές εργασίες.

Υπάρχουν τρία βασικά προβλήματα που μπορούν να επιλυθούν με το FlexSim:

1. **Προβλήματα υπηρεσιών:** η ανάγκη να υποβληθούν σε επεξεργασία οι πελάτες και τα αιτήματά τους στο υψηλότερο επίπεδο ικανοποίησης με το χαμηλότερο δυνατό κόστος.

2. **Προβλήματα παραγωγής:** η ανάγκη να γίνει το σωστό προϊόν στο σωστό χρόνο με το χαμηλότερο δυνατό κόστος.
3. **Προβλήματα εφοδιαστικής:** η ανάγκη να φτάσει το σωστό προϊόν στη σωστή θέση στο σωστό χρόνο με το χαμηλότερο δυνατό κόστος.

### **3.3 Περιπτώσεις χρήσης του λογισμικού FlexSim**

Το FlexSim έχει χρησιμοποιηθεί επιτυχώς:

- ➔ για να βελτιώσει τη χρησιμοποίηση εξοπλισμού
- ➔ για να μειώσει το χρόνο αναμονής και το μέγεθος της ουράς αναμονής
- ➔ για να διαθέσει αποτελεσματικά τους πόρους
- ➔ για να περιορίσει προβλήματα έλλειψης αποθεμάτων
- ➔ για να περιορίσει τα αρνητικά αποτελέσματα των βλαβών
- ➔ για να περιορίσει τα αρνητικά αποτελέσματα των απορριμμάτων και των αποβλήτων
- ➔ για να μελετήσει εναλλακτικές ιδέες επένδυσης
- ➔ για να καθορίσει τους χρόνους ρυθμοαπόδοσης μερών
- ➔ για να μελετήσει σχέδια μείωσης δαπανών
- ➔ για να καθορίσει το βέλτιστο μέγεθος παρτίδας και την αλληλουχία των μερών
- ➔ για να επιλύσει ζητήματα χειρισμού υλικών
- ➔ για να μελετήσει την επίδραση των χρόνων προετοιμασίας (setup time) και αλλαγής εργαλείων (tool changeover)
- ➔ για να βελτιστοποιήσει τη λογική καθορισμού προτεραιοτήτων και αποστολής για τα αγαθά και τις υπηρεσίες
- ➔ για να εκπαιδεύσει τους χειριστές στη γενική συμπεριφορά του συστήματος και τη σχετική με την εργασία απόδοση
- ➔ για να δείξει το σχέδιο και τις δυνατότητες καινούριων εργαλείων
- ➔ να διαχειριστεί την καθημερινή λήψη αποφάσεων

Το FlexSim έχει χρησιμοποιηθεί επιτυχώς στις μελέτες σχεδίου συστημάτων και στη διαχείριση των συστημάτων σε καθημερινή λειτουργική βάση. Έχει



χρησιμοποιηθεί επίσης για λόγους κατάρτισης και εκπαίδευσης. Ένα μοντέλο κατάρτισης FlexSim μπορεί να παρέχει γνώση στις σύνθετες εξαρτήσεις και τη δυναμική ενός πραγματικού συστήματος.

Το FlexSim παρέχει διάφορα εργαλεία για τη δημιουργία μοντέλων που επεξεργάζονται μετρήσιμα στοιχεία αλλά και για τη δημιουργία μοντέλων που επεξεργάζονται “ρευστά στοιχεία”. Παρακάτω αναφέρονται ενδεικτικά κάποια παραδείγματα προσομοίωσης μοντέλων μετρήσιμων στοιχείων.

### 3.4 Ορολογία του FlexSim

Πριν περάσουμε σε παραδείγματα και στο κυρίως σώμα της προσομοίωσης πρέπει πρώτα να γνωρίσουμε την ορολογία που χρησιμοποιεί το FlexSim.

- **Αντικείμενα του FlexSim – FlexSim objects:** τα αντικείμενα του FlexSim προσομοιώνουν διαφορετικές μορφές πόρων στην προσομοίωση. Ένα παράδειγμα αντικειμένου είναι η ουρά αναμονής – queue που συμπεριφέρεται σαν σημείο αποθήκευσης. Μία ουρά αναμονής μπορεί να αντιπροσωπεύει μία σειρά ανθρώπων, μία περιοχή αποθήκευσης σε ένα εργοστάσιο, ή μία ουρά αναμονής κλήσεων σε τηλεφωνικό κέντρο εξυπηρέτησης πελατών. Ένα άλλο παράδειγμα αντικειμένου είναι ο επεξεργαστής – processor, που προσομοιώνει μία καθυστέρηση ή ένα χρόνο επεξεργασίας. Αυτό το αντικείμενο μπορεί να αντιπροσωπεύει μία μηχανή σε εργοστάσιο, ένα τραπεζικό υπάλληλο που εξυπηρετεί πελάτες, κλπ.
- **Στοιχεία ροής – Flowitems/ Items:** είναι δυναμικά αντικείμενα τα οποία κινούνται στο μοντέλο. Μπορούν να αντιπροσωπεύουν παλέτες, χαρτιά, ανθρώπους, τηλεφωνικές κλήσεις, παραγγελίες ή οτιδήποτε κινείται στη διεργασία που προσομοιώνεται. Αυτά τα στοιχεία μπορούν να υποστούν επεξεργασία και να μεταφερθούν στο μοντέλο από πόρους χειρισμού υλικού. Τα flowitems παράγονται από τις πηγές – sources. Όταν έχουν περάσει από το μοντέλο και δεν χρησιμεύουν πια, αποστέλλονται σε καταβόθρες – sinks.
- **Τύπος αντικειμένου – Itemtype:** είναι μία ετικέτα που τοποθετείται στα στοιχεία ροής – flowitems. Μπορεί να αντιπροσωπεύει ένα γραμμωτό κώδικα, τον τύπο του προϊόντος ή τον αριθμό του είδους. Το FlexSim έχει ρυθμιστεί έτσι ώστε να χρησιμοποιεί τον τύπο – itemtype σαν σημείο αναφοράς για τη δρομολόγηση των στοιχείων – flowitems.

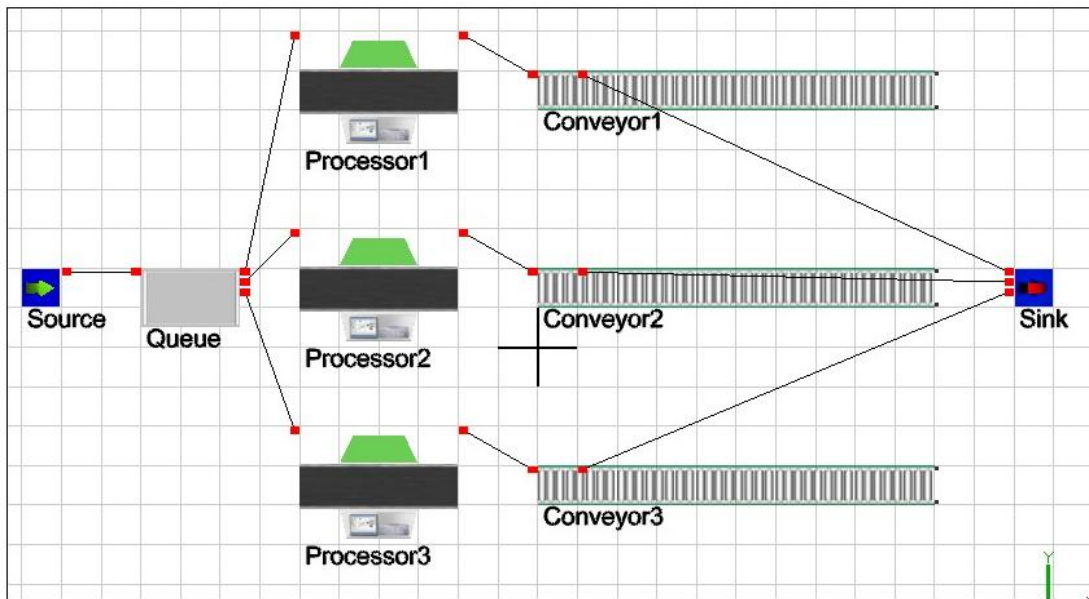
- **Ροή διαδικασίας – Process flow:** είναι ένα σύνολο από στάδια επεξεργασίας, αναμονής και μεταφοράς στο μοντέλο. Κάθε στάδιο αντιπροσωπεύει και μία εργασία και απαιτεί έναν ή περισσότερους πόρους.
- **Θύρες – Ports:** Κάθε αντικείμενο στο FlexSim έχει άπειρο αριθμό θυρών μέσω των οποίων επικοινωνεί με άλλα αντικείμενα. υπάρχουν τρία είδη θυρών, **εισόδου – input**, **εξόδου – output ports** και **κεντρικές– central**. Οι θύρες εισόδου και εξόδου χρησιμοποιούνται για τη δρομολόγηση των αντικειμένων, ενώ οι κεντρικές χρησιμοποιούνται για τη δημιουργία αναφορών από ένα αντικείμενο σε ένα άλλο.

Στα παραδείγματα που ακολουθούν θα παρουσιαστούν σταδιακά τα υπόλοιπα αντικείμενα του FlexSim αλλά και οι δυνατότητες που παρέχει το λογισμικό για την ανάλυση του μοντέλου.

### 3.5 Εφαρμογές

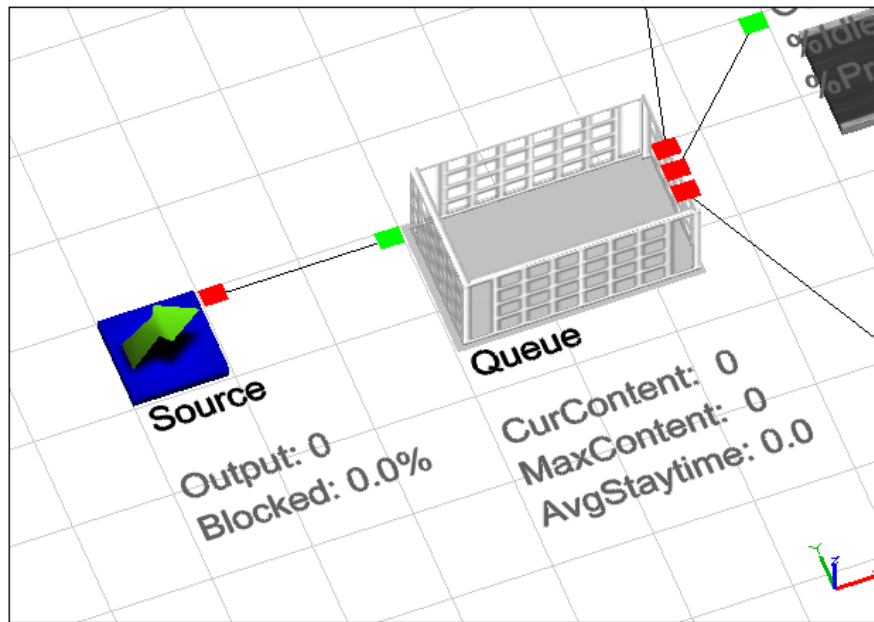
#### Παράδειγμα 1

- Δημιουργείται μία γραμμή παραγωγής που αποτελείται από μία πηγή (source), μία ουρά αναμονής (queue), τρεις επεξεργαστές (processors), τρεις μεταφορικές ταινίες (conveyors) και μία καταβόθρα (sink).
- Δημιουργούνται οι κατάλληλες συνδέσεις μεταξύ των αντικειμένων.
- Στις ιδιότητες της πηγής επιλέγεται ο τρόπο άφιξης των στοιχείων, δηλαδή κανονική κατανομή με διάστημα 20'' μεταξύ των αφίξεων και τυπική απόκλιση 2''.
- Ορίζεται ο τύπος και το χρώμα του κάθε στοιχείου. Τα στοιχεία που θα εξέρχονται από την πηγή θα παίρνουν τυχαίες ακέραιες τιμές από το 1 ως το 3. Το διαφορετικό χρώμα του κάθε στοιχείου διευκολύνει στην καλύτερη εποπτεία του συστήματος.
- Καθορίζεται η χωρητικότητα της ουράς αναμονής βάζοντας σαν ανώτερο όριο τα 25 στοιχεία.



Σχήμα 3.5.1

- Εφόσον έχει ορισθεί ότι τα στοιχεία που θα εξέρχονται από την πηγή θα παίρνουν τυχαίες ακέραιες τιμές από το 1 ως το 3, τώρα ορίζεται ότι θα επεξεργάζονται από τον αντίστοιχο τύπο μηχανής, συνδέοντας τις μηχανές με τις κατάλληλες θύρες.
- Καθορίζεται ο χρόνος επεξεργασίας των στοιχείων, για κάθε μηχανή ξεχωριστά, ορίζοντας ότι ικανοποιεί εκθετική κατανομή με χρόνο 30'' επεξεργασίας. Η ταχύτητα των μεταφορικών ταινιών είναι 1 m/sec.
- Πλέον το μοντέλο είναι έτοιμο για δοκιμή. Πατώντας το reset επαναφέρεται το σύστημα και οι παραμέτροι του μοντέλου στην αρχική τους κατάσταση.
- Πατώντας το κουμπί start ξεκινά η προσομοίωση.
- Τα στοιχεία εισέρχονται στην ουρά αναμονής και στη συνέχεια πηγαίνουν στους επεξεργαστές. Από εκεί τα στοιχεία συνεχίζουν στις μεταφορικές ταινίες και έπειτα στην καταβόθρα. Η ταχύτητα της προσομοίωσης μπορεί να προσαρμοστεί από τη μπάρα Run Speed.
- Κατά τη διάρκεια της προσομοίωσης καταγράφονται κάποια στατιστικά στοιχεία για τη λειτουργία κάθε αντικειμένου του συστήματος που αναγράφονται πάνω από αυτό.

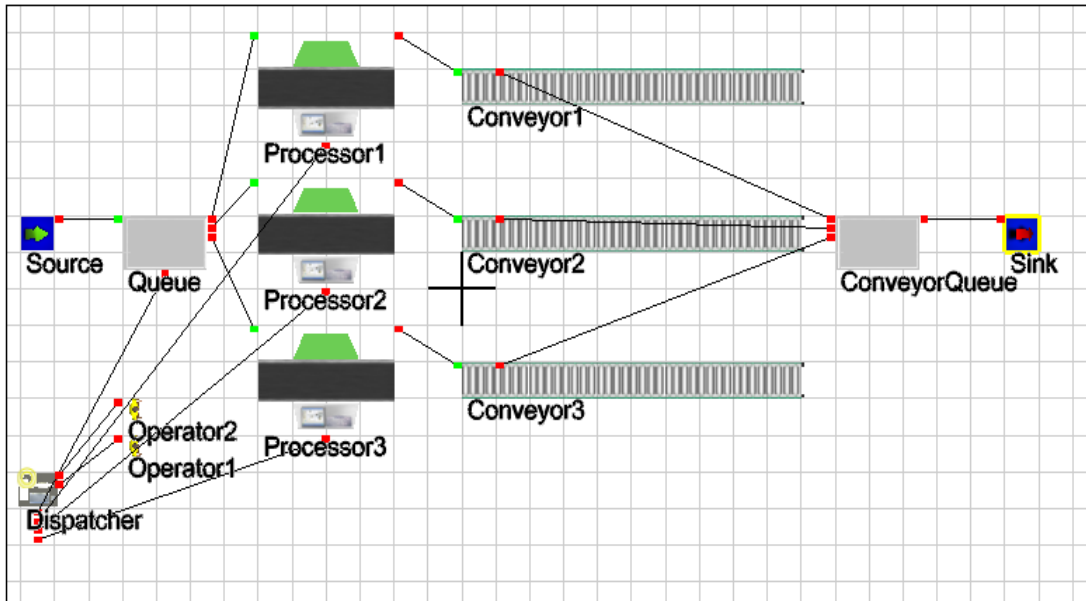


Σχήμα 3.5.2

## Παράδειγμα 2

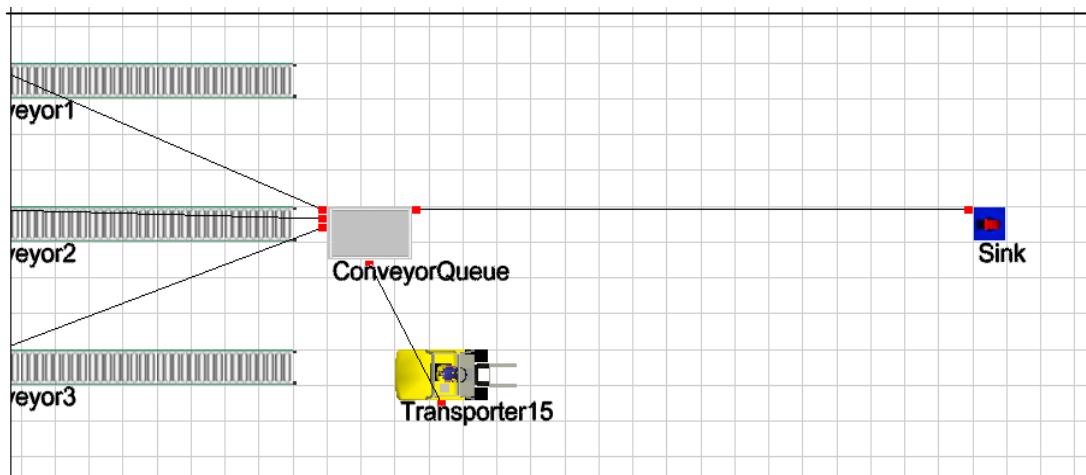
Εξελίσσοντας το παραπάνω μοντέλο ακολουθούνται τα παρακάτω βήματα:

- Εισάγεται ένας διεκπεραιωτής - dispatcher ο οποίος μοιράζει τις εργασίες σε μία ομάδα χειριστών ή μεταφορέων. Εδώ εισάγονται δύο χειριστές οι οποίοι θα μεταφέρουν τα στοιχεία από την ουρά αναμονής στους επεξεργαστές και δημιουργούνται οι κατάλληλες συνδέσεις με τα μέρη του συστήματος. Ο διεκπεραιωτής καλεί το χειριστή που είναι ελεύθερος εκείνη τη στιγμή για τη μεταφορά του στοιχείου.
- Επίσης συνδέεται ο διεκπεραιωτής με τους επεξεργαστές και δημιουργούνται οι κατάλληλες συνδέσεις με τους χειριστές έτσι ώστε αυτοί να είναι υπεύθυνοι και για την προετοιμασία των επεξεργασιών.
- Ορίζεται ο χρόνος προετοιμασίας των επεξεργασιών.
- Αποσυνδέονται οι μεταφορικές ταινίες από την καταβόθρα, προστίθεται μία ουρά αναμονής δεξιά από τις μεταφορικές ταινίες και στη συνέχεια δημιουργούνται ξανά οι κατάλληλες συνδέσεις.



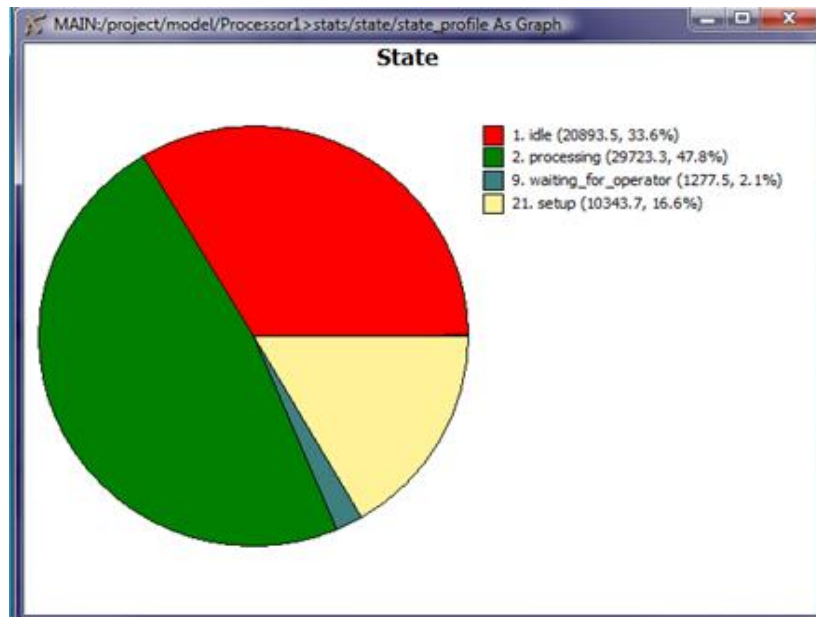
Σχήμα 3.5.3

- Προστίθεται ένας μεταφορέας (περονοφόρο) ο οποίος θα μεταφέρει στοιχεία από την ουρά αναμονής των μεταφορικών ταινιών στην καταβόθρα και δημιουργούνται οι κατάλληλες συνδέσεις. Εφόσον ο μεταφορέας θα είναι μόνο ένας δεν χρειάζεται διεκπεραιωτής.

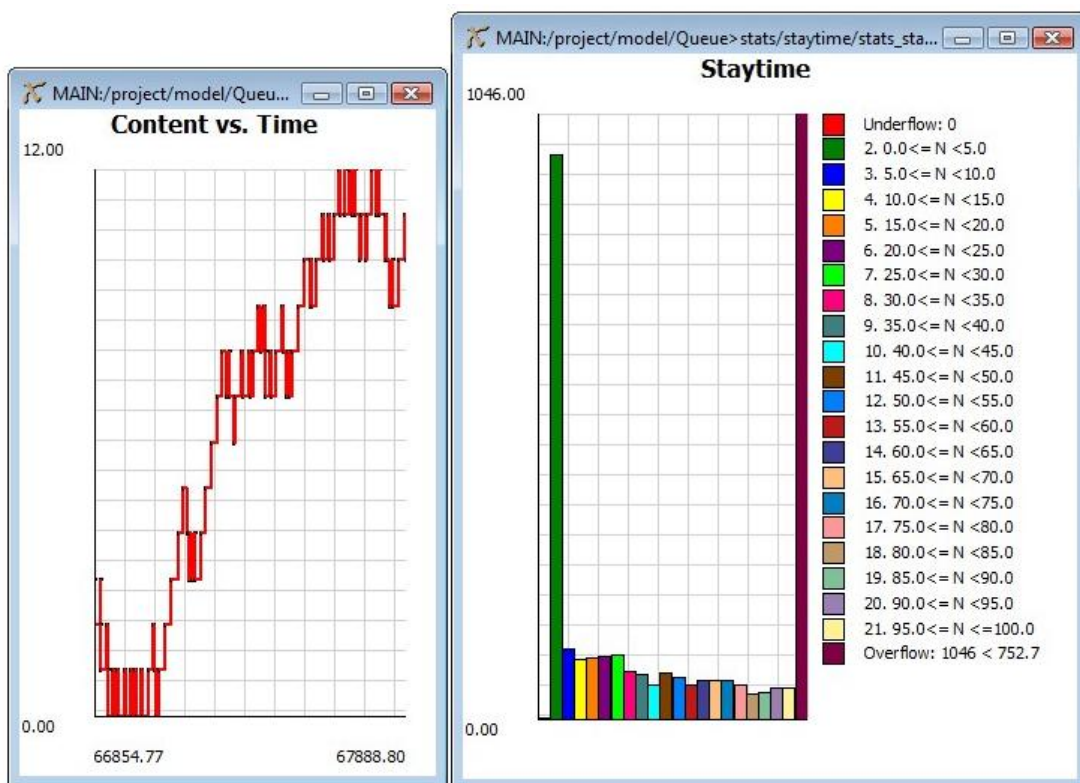


Σχήμα 3.5.4

- Ενεργοποιείται η καταγραφή στατιστικών και τρέχει το μοντέλο προσομοίωσης. Εν τω μεταξύ το μοντέλο ελέγχεται για το κατά πόσο λειτουργούν όλα σωστά.



Σχήμα 3.5.5



Σχήμα 3.5.6

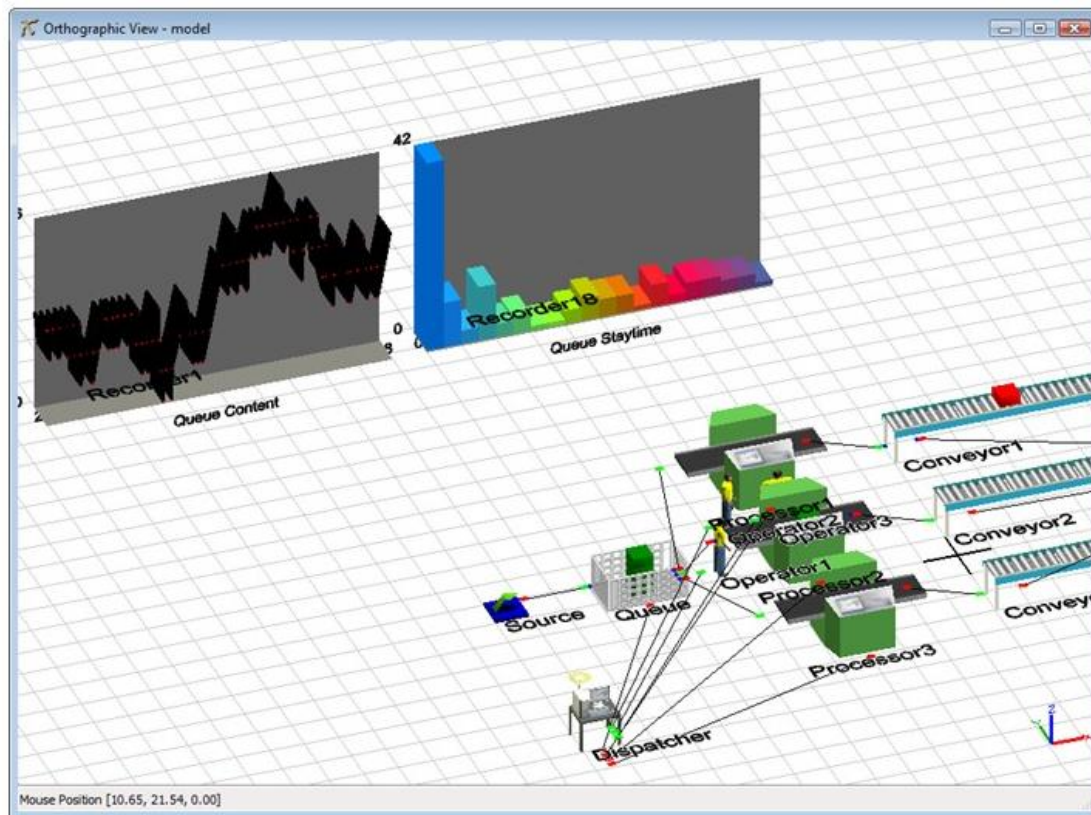
Σχήμα 3.5.7

➔ Η λειτουργία του συστήματος θα βελτιωθεί αν προστεθεί ένας ακόμα χειριστής.

### Παράδειγμα 3

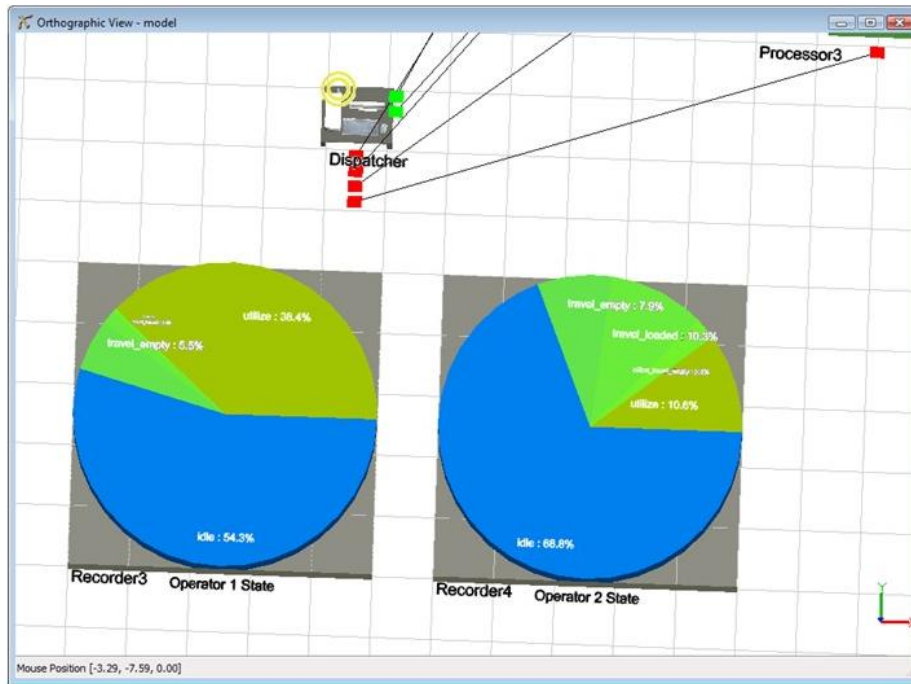
Εξελίσσοντας το παραπάνω μοντέλο ακολουθούνται τα παρακάτω βήματα:

- Ενεργοποιείται η καταγραφή στατιστικών για όλα τα αντικείμενα.
- Εισάγεται ένας καταγραφέας (Recorder) για να καταγράψει το περιεχόμενο της ουράς αναμονής σε συνάρτηση με το χρόνο.
- Εισάγεται ένας καταγραφέας (Recorder) για να απεικονίζει με τη βοήθεια ιστογράμματος το χρόνο παραμονής των στοιχείων στην ουρά αναμονής.



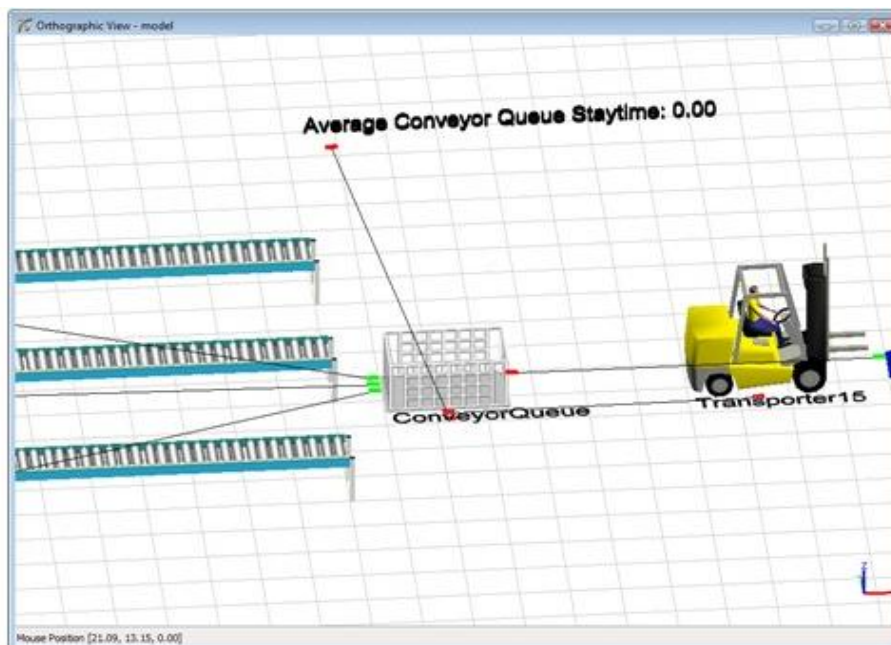
Σχήμα 3.5.8

- Εισάγεται ένας καταγραφέας (Recorder) για κάθε επεξεργαστή για να απεικονίζει, με τη βοήθεια διαγράμματος σε μορφή πίτας, στατιστικά για την κατάσταση του κάθε επεξεργαστή.



Σχήμα 3.5.9

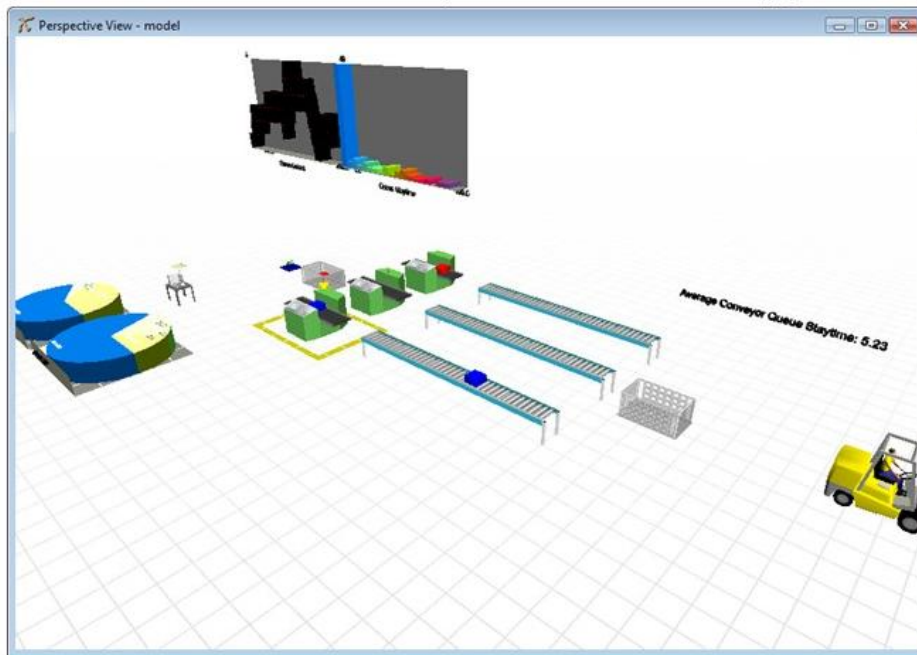
- Εισάγεται ένα τρισδιάστατο μήνυμα όπου φαίνεται ο μέσος χρόνος παραμονής των στοιχείων στην ουρά αναμονής των μεταφορικών ταινιών (Conveyor Queue).



Σχήμα 3.5.10

- Γίνεται επαναφορά του συστήματος στην αρχική κατάσταση και τρέχει το μοντέλο.



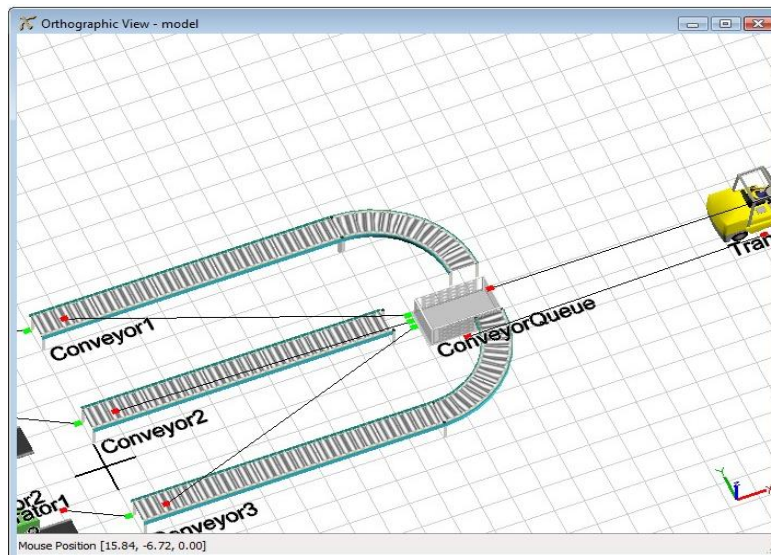


Σχήμα 3.5.11

#### Παράδειγμα 4

Εξελίσσοντας περισσότερο το μοντέλο ακολουθούνται τα παρακάτω βήματα:

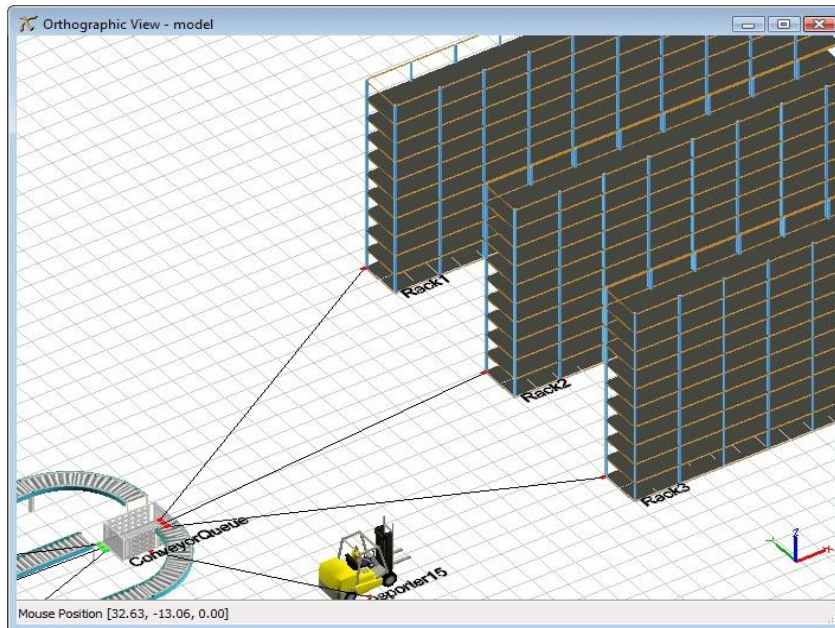
- Αλλάζει η μορφή των μεταφορικών ταινιών Conveyor 1 και Conveyor 3, προσθέτοντας ένα καμπύλο τμήμα στο τέλος τους.



Σχήμα 3.5.12

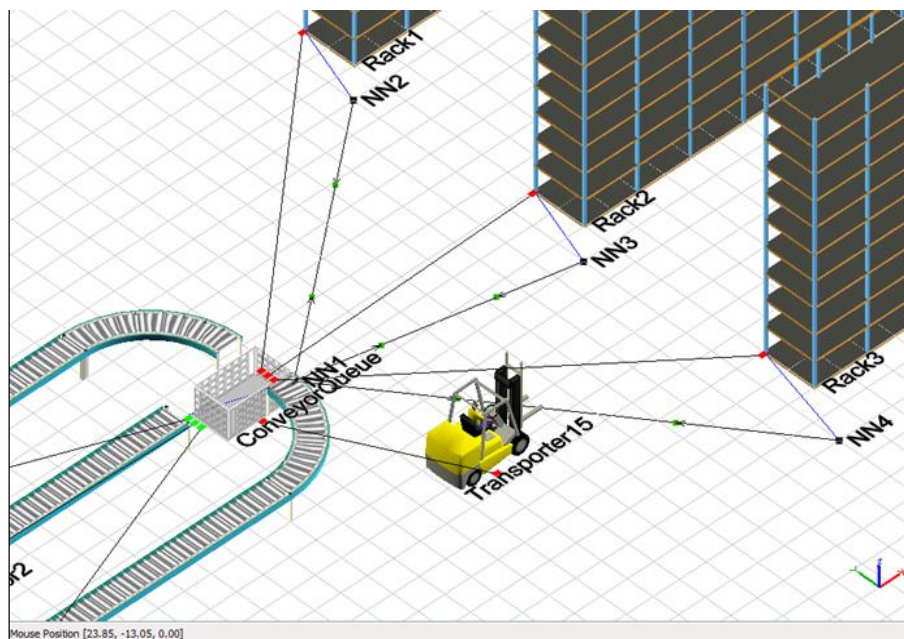
- Διαγράφεται η καταβόθρα και αντικαθίσταται με τρεις στήλες ραφιών αποθήκευσης τα οποία συνδέονται με την ουρά αναμονής των μεταφορικών ταινιών.

- Δημιουργείται ένας πίνακας όπου ορίζεται σε ποια στήλη ραφιών θα αποστέλλεται κάθε επεξεργασμένο στοιχείο.

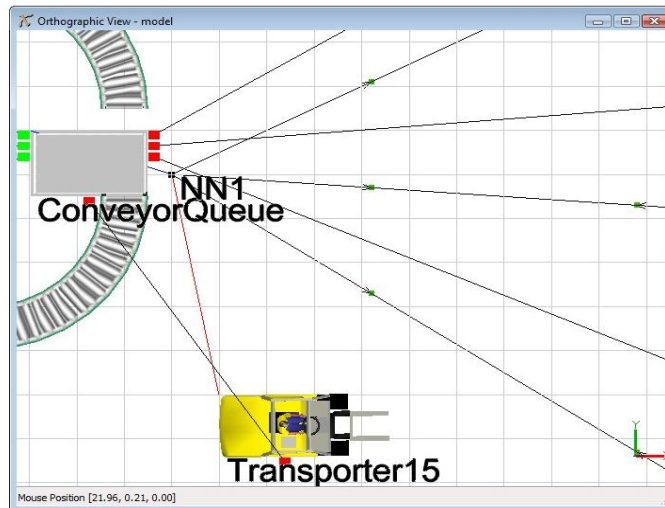


Σχήμα 3.5.13

- Προστίθενται κόμβοι (NetworkNodes) κοντά στην ουρά αναμονής των μεταφορικών ταινιών, και σε καθεμία από τις στήλες ραφιών για να βελτιωθεί η διαδρομή που ακολουθεί ο μεταφορέας – περνοφόρο και συνδέονται με τα αντίστοιχα μέρη του συστήματος καθώς και με το μεταφορέα. Τα σημεία αυτά θα είναι τα σημεία pick-up και drop-off αντίστοιχα.

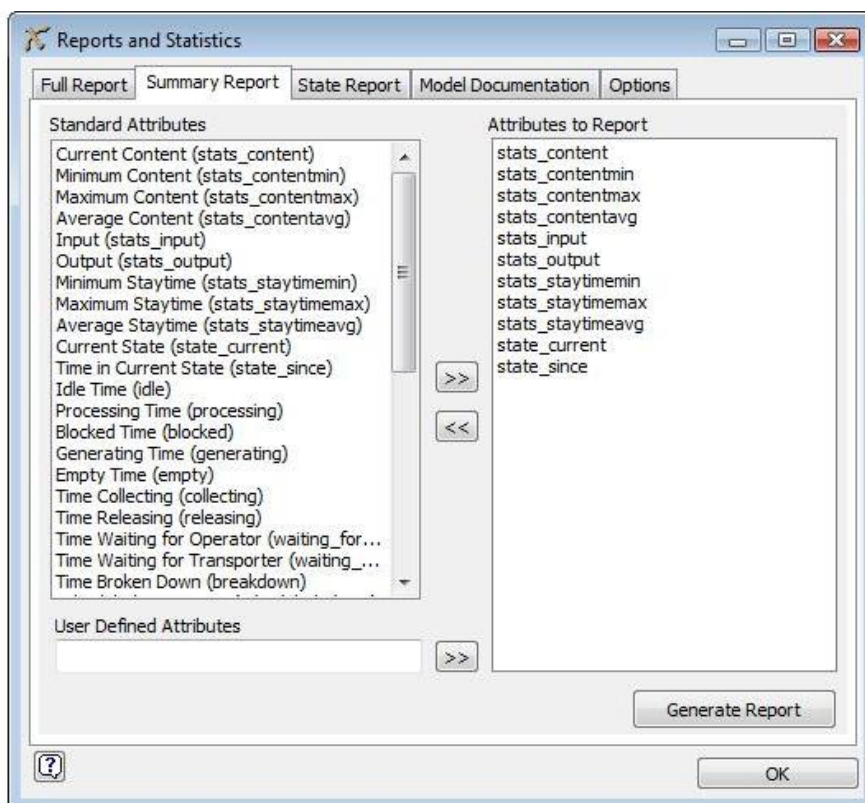


Σχήμα 3.5.14



Σχήμα 3.5.15

- Γίνεται επαναφορά του συστήματος στην αρχική κατάσταση και τρέχει το μοντέλο.
- Μετά το τέλος της προσομοίωσης μπορούμε να δούμε τα στατιστικά στοιχεία που έχουν καταγραφεί με τη βοήθεια αναφορών, όπως η γενική αναφορά και η αναφορά κατάστασης. Οι αναφορές εξάγονται σε αρχείο csv και εμφανίζονται αυτόματα στο Excel.



Σχήμα 3.5.16

Object	Class	stats_cont	stats_cont	stats_cont	stats_cont	stats_inpu	stats_outu	stats_stay	stats_stay	stats_stay	state_curr	state_since
Source	Source	0	0	0	0	1	0	1339	0	326.6003	7.882783	5
Queue	Queue	24	0	25	22.33462	1339	1315	1.777517	3185.945	618.1947	8	37376.36
Processor	Processor	0	0	1	0.46037	461	461	10.0347	162.5524	37.29556	1	37376.36
Processor	Processor	1	0	1	0.498806	466	465	10.037	183.433	40.07985	2	37376.36
Processor	Processor	1	0	1	0.923999	388	387	10.31463	2307.401	89.17754	2	37376.36
Conveyor	Conveyor	0	0	3	0.405734	461	461	14.71239	68.82785	32.89554	6	37376.36
Conveyor	Conveyor	1	0	13	1.42399	465	464	10	723.6134	114.6433	4	37376.36
Conveyor	Conveyor	25	0	25	23.34033	387	362	14.71239	5121.235	2343.239	4	37376.36
Dispatche	Dispatche	0	0	0	0	0	0	0	0	0	2	37376.36
Operator1	Operator	0	0	1	0.016472	262	262	1.844796	2.949462	2.346952	1	37376.36
Operator2	Operator	0	0	1	0.068573	1053	1053	1.844763	2.958871	2.433187	1	37376.36
Conveyor	Queue	10	0	10	9.881959	1287	1277	0.729	339.848	288.1934	10	37376.36
Transport	Transport	1	0	1	0.516961	1277	1276	8.288835	19.38042	15.14272	15	37376.36
Rack1	Rack	359	1	359	157.3819	359	0	0	0	0	2	37376.36
Rack2	Rack	457	1	457	236.5265	457	0	0	0	0	2	37376.36
Rack3	Rack	460	1	460	239.7007	460	0	0	0	0	2	37376.36

Σχήμα 3.5.17

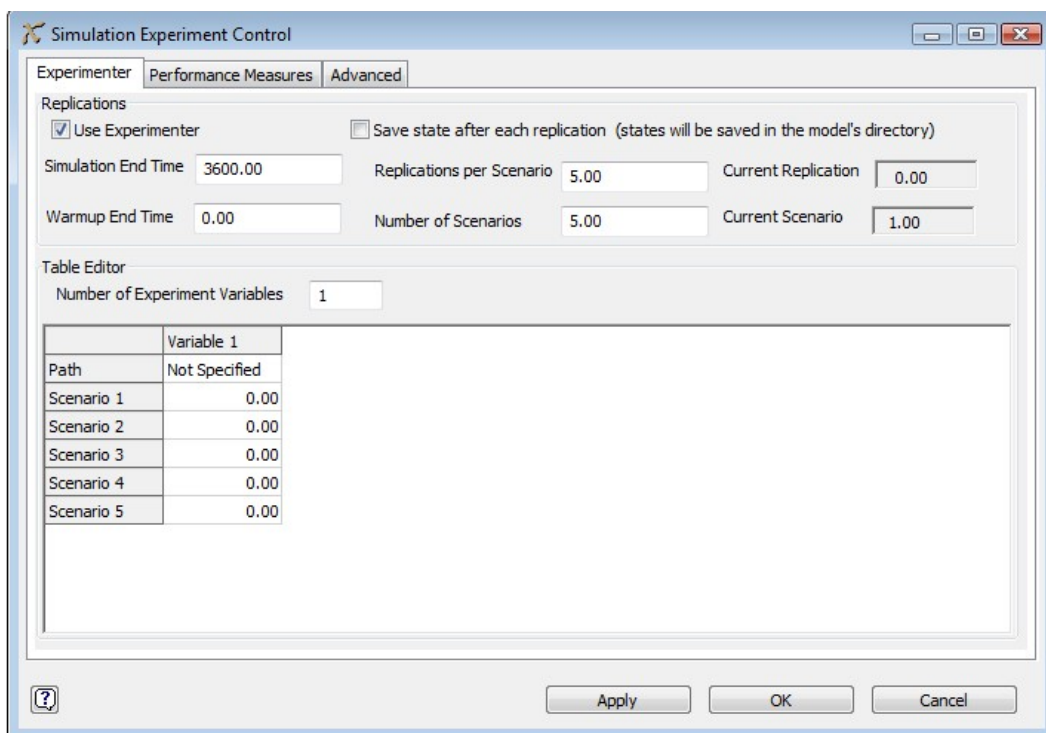
Object	Class	idle	processing	busy	blocked	generating	empty	collecting	releasing	waiting_for	waiting_for	breakdown
Source	Source	0.00%	0.00%	0.00%	28.24%	71.76%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
Queue	Queue	0.00%	0.00%	0.00%	0.00%	0.00%	0.22%	0.00%	99.76%	0.00%	0.03%	0.00%
Processor	Processor	54.00%	32.60%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	1.07%	0.00%	0.00%
Processor	Processor	50.10%	37.10%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.33%	0.00%	0.00%
Processor	Processor	7.59%	30.23%	0.00%	50.74%	0.00%	0.00%	0.00%	0.00%	1.05%	0.00%	0.00%
Conveyor	Conveyor	0.00%	0.00%	0.00%	21.45%	0.00%	63.70%	0.00%	0.00%	0.00%	0.00%	0.00%
Conveyor	Conveyor	0.00%	0.00%	0.00%	50.57%	0.00%	42.68%	0.00%	0.00%	0.00%	0.00%	0.00%
Conveyor	Conveyor	0.00%	0.00%	0.00%	96.68%	0.00%	2.09%	0.00%	0.00%	0.00%	0.00%	0.00%
Dispatche	Dispatche	0.00%	100.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
Operator1	Operator	69.46%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
Operator2	Operator	76.13%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
Conveyor	Queue	0.00%	0.00%	0.00%	0.00%	0.00%	0.49%	0.00%	0.00%	99.51%	0.00%	0.00%
Transport	Transport	0.38%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
Rack1	Rack											
Rack2	Rack											
Rack3	Rack											

Σχήμα 3.5.18

Ακόμα το FlexSim μπορεί να αποθηκεύσει όλα τα στατιστικά στοιχεία των αντικειμένων και των οντοτήτων σε βάση δεδομένων. Στη συνέχεια, με τη χρήση αυτής της βάσης μέσω του FlexSim Chart παρέχονται τα εξής:

- γενική αναφορά και αναφορά κατάστασης του μοντέλου

- ☛ συγκριτικά διαγράμματα για κάποια αντικείμενα
  - ☛ διαγράμματα διαφόρων μεταβλητών των αντικειμένων ως προς το χρόνο
  - ☛ διαγράμματα κατάστασης των αντικειμένων
  - ☛ διαγράμματα Gant για τις αλλαγές κατάστασης των αντικειμένων
  - ☛ διαγράμματα Gant που μας πληροφορούν για την πορεία των οντοτήτων στο σύστημα
  - ☛ οικονομική έκθεση
- Μπορούμε να τρέξουμε πολλές φορές το μοντέλο προσομοίωσης, για διαφορετικό σενάριο κάθε φορά, χρησιμοποιώντας το εργαλείο “Experimenter”. Σε αυτό δηλώνεται ένας αριθμός μεταβλητών και να ορίζεται η τιμή που αυτές θα παίρνουν σε κάθε σενάριο.

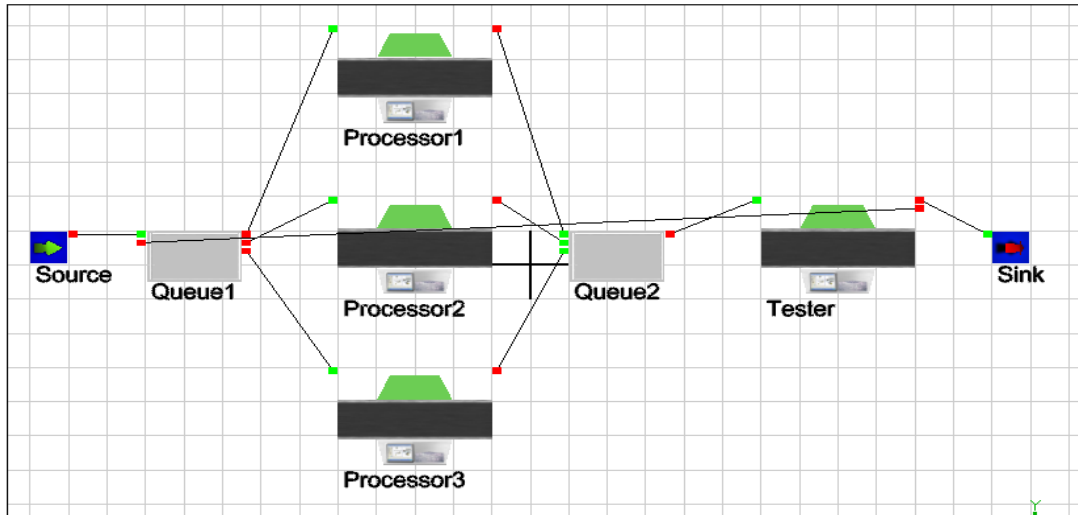


Σχήμα 3.5.19

### Παράδειγμα 5

Σε αυτό το παράδειγμα θα δείξουμε και την εμφάνιση σημείου συμφόρησης (bottleneck).

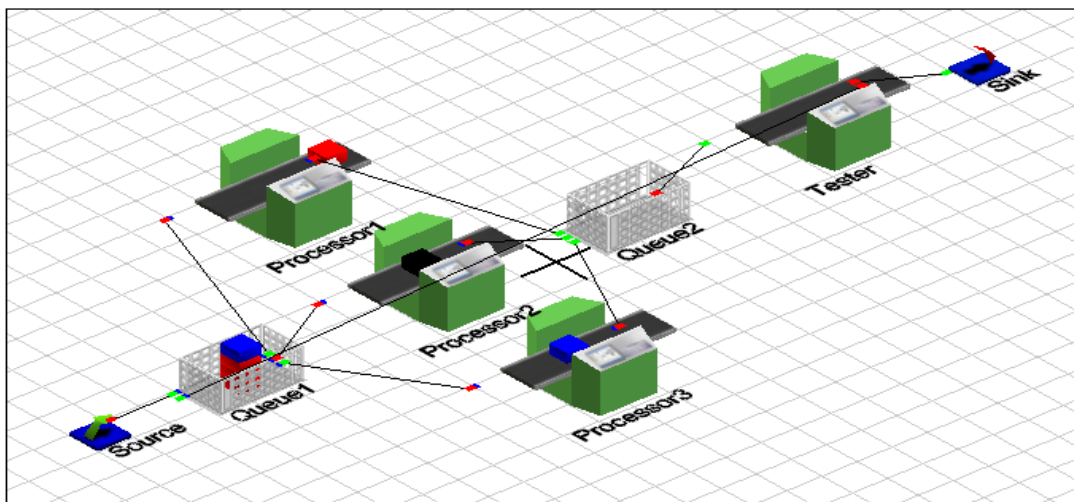
- Δημιουργούνται τα εξής αντικείμενα: μία πηγή (Source), δύο ουρές αναμονής (Queues), τέσσερις επεξεργαστές (Processors) και μία καταβόθρα (Sink), τα οποία τοποθετούνται και ονομάζονται όπως φαίνεται στο σχήμα.
- Δημιουργούνται οι κατάλληλες συνδέσεις μεταξύ των αντικειμένων.



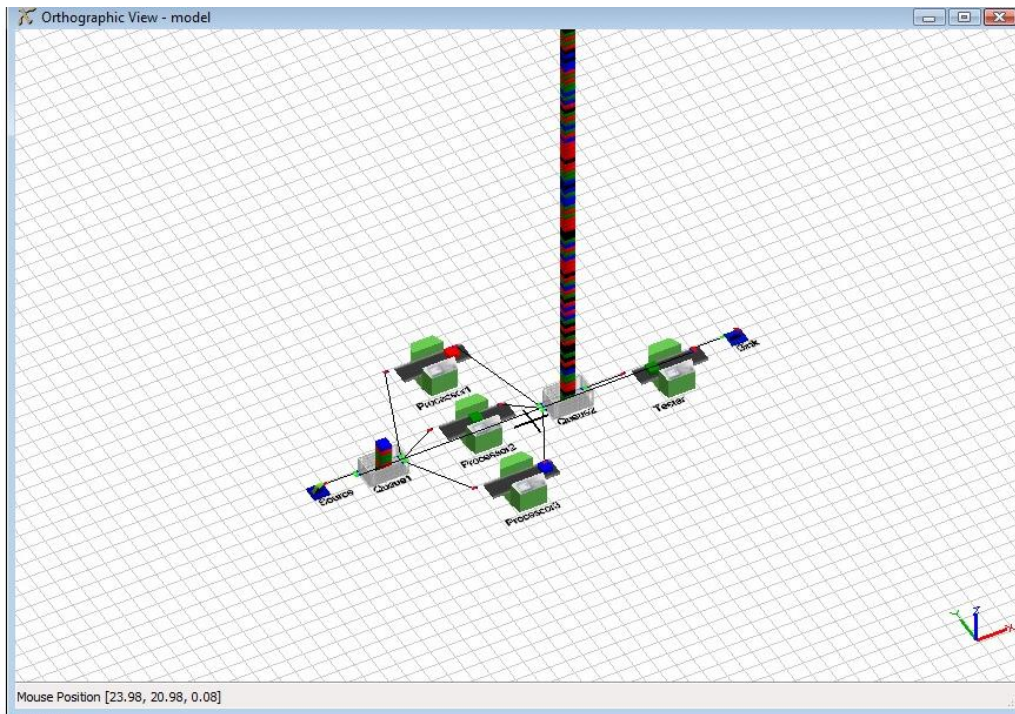
Σχήμα 3.5.20

- Στις ιδιότητες της πηγής επιλέγεται ο τρόπος άφιξης των στοιχείων, δηλαδή εκθετική κατανομή με διάστημα 5" μεταξύ των αφίξεων.
- Ορίζεται ο τύπος και το χρώμα του κάθε στοιχείου. Τα στοιχεία που θα εξέρχονται από την πηγή θα παίρνουν τυχαίες ακέραιες τιμές από το 1 ως το 3. Το διαφορετικό χρώμα του κάθε στοιχείου διευκολύνει στην καλύτερη εποπτεία του συστήματος.
- Καθορίζεται η χωρητικότητα της ουράς αναμονής Queue1 βάζοντας σαν ανώτερο όριο τα 10000 στοιχεία.
- Εφόσον έχει ορισθεί ότι τα στοιχεία που θα εξέρχονται από την πηγή θα παίρνουν τυχαίες ακέραιες τιμές από το 1 ως το 3, τώρα ορίζεται ότι θα επεξεργάζονται από τον αντίστοιχο τύπο μηχανής, συνδέοντας τις μηχανές με τις κατάλληλες θύρες.
- Καθορίζεται ο χρόνος επεξεργασίας των στοιχείων, για κάθε μηχανή ξεχωριστά, ορίζοντας ότι ικανοποιεί εκθετική κατανομή με χρόνο επεξεργασίας 10".
- Καθορίζεται η χωρητικότητα της ουράς αναμονής Queue2 βάζοντας σαν ανώτερο όριο τα 10000 στοιχεία.

- Καθορίζεται ο χρόνος επεξεργασίας του ελεγκτή σε 4”.
- Ορίζεται ότι το 80% των στοιχείων που ελέγχει ο ελεγκτής Tester είναι αποδεκτά και το 20% των στοιχείων είναι σκάρτα. Πρέπει να καθοριστεί η διαδρομή που θα ακολουθούν τα στοιχεία που υπόκεινται σε έλεγχο. Τα αποδεκτά στοιχεία θα πηγαίνουν στην καταβόθρα, ενώ τα σκάρτα θα επιστρέφουν στην πρώτη ουρά αναμονής Queue1 για επανεπεξεργασία. Επιλέγεται διαφορετικό χρώμα για τα σκάρτα προϊόντα για καλύτερη εποπτεία της διαδικασίας.
- Πλέον το μοντέλο είναι έτοιμο για δοκιμή. Πατώντας το reset επαναφέρονται το σύστημα και οι παράμετροι του μοντέλου στην αρχική τους κατάσταση.
- Πατώντας το κουμπί start ξεκινά η προσομοίωση.
- Τα στοιχεία εισέρχονται στην ουρά αναμονής και στη συνέχεια πηγαίνουν στους επεξεργαστές. Από εκεί τα στοιχεία πηγαίνουν σε δεύτερη ουρά αναμονής, μετά στον ελεγκτή και έπειτα στην καταβόθρα ή στην πρώτη ουρά αναμονής για επανεπεξεργασία. Η ταχύτητα της προσομοίωσης αλλάζει από τη μπάρα Run Speed.



Σχήμα 3.5.21



Σχήμα 3.5.22

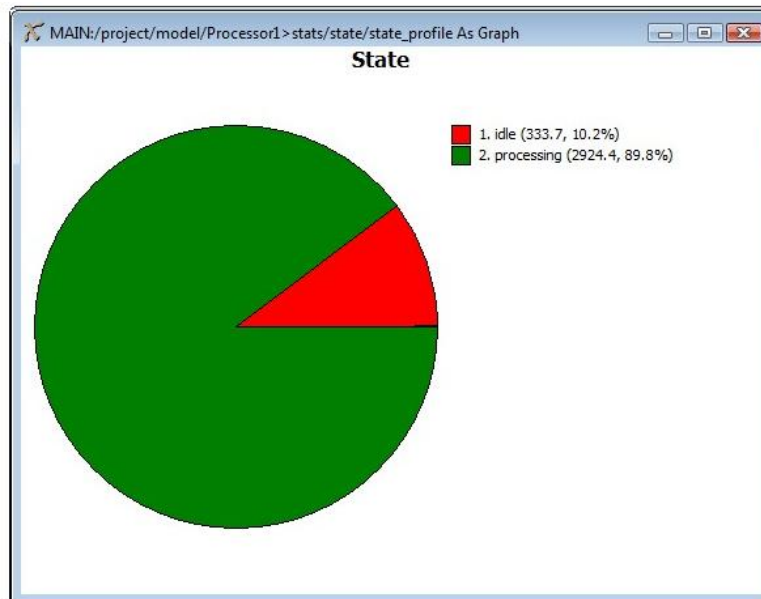
### Βρίσκοντας το σημείο συμφόρησης – bottleneck

Υπάρχουν πολλοί τρόποι για να προσδιοριστεί το σημείο που εμφανίζεται συμφόρηση - bottleneck στο μοντέλο.

- Ο πιο απλός είναι να εξεταστεί, από το τρέξιμο του μοντέλου προσομοίωσης, το μέγεθος της κάθε ουράς αναμονής. Εάν κάποια ουρά αναμονής του μοντέλου έχει συνεχώς πολλά στοιχεία, είναι μία ένδειξη ότι ο επόμενος σταθμός επεξεργασίας δημιουργεί συμφόρηση - bottleneck στο σύστημα.
- Ένας άλλος τρόπος για να βρεθεί το σημείο bottleneck είναι εξετάζοντας τα στατιστικά της κατάστασης κάθε επεξεργαστή. Αν οι τρεις επεξεργαστές είναι συνεχώς απασχολημένοι ενώ ο ελεγκτής δεν είναι απασχολημένος τότε το bottleneck μάλλον εμφανίζεται στους επεξεργαστές. Από την άλλη πλευρά, αν ο ελεγκτής είναι συνεχώς απασχολημένος τότε το bottleneck μάλλον βρίσκεται στον ελεγκτή.

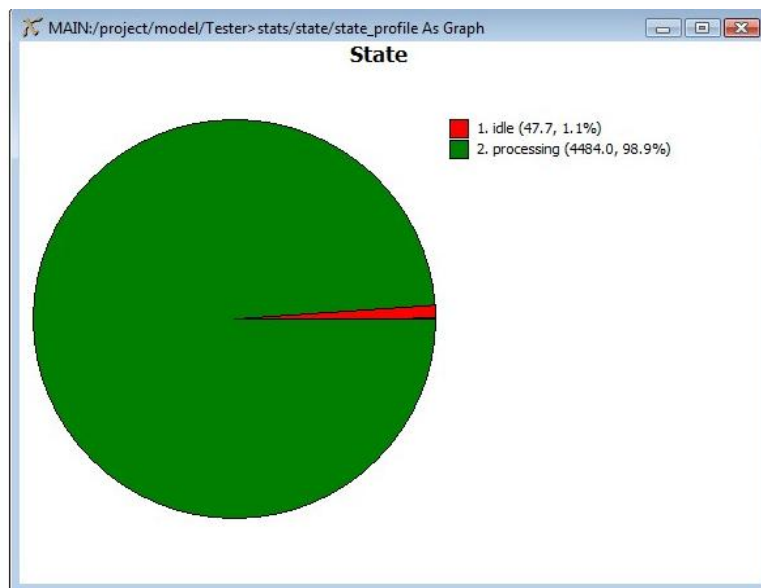
Τρέχουμε το μοντέλο τουλάχιστον μέχρι τις 50000 μονάδες χρόνου. Ελέγχουμε τα στατιστικά της κατάστασης των σταθμών επεξεργασίας.





Σχήμα 3.5.23

Ελέγχουμε τα στατιστικά της κατάστασης του ελεγκτή.



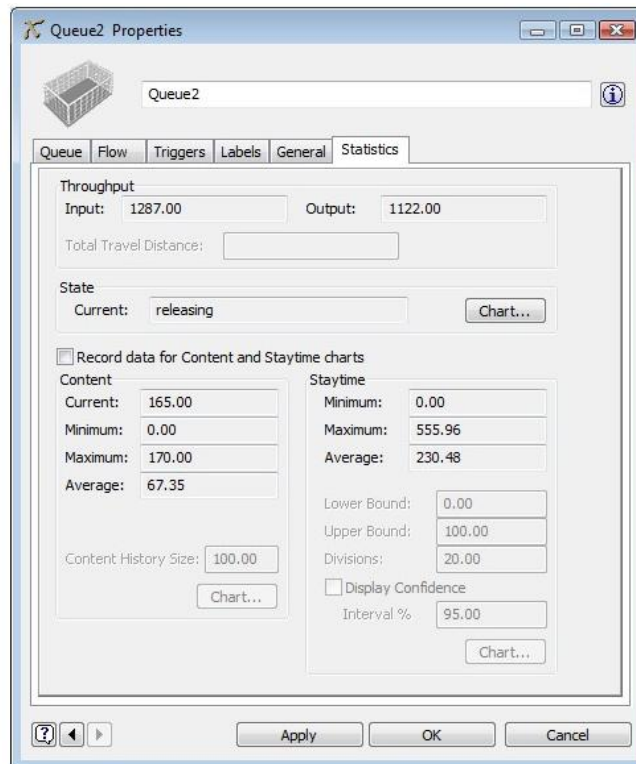
Σχήμα 3.5.24

Από τα παραπάνω διαγράμματα είναι εμφανές ότι το bottleneck προκαλείται από τον ελεγκτή.

Η αντιμετώπιση του bottleneck εξαρτάται από διάφορους παράγοντες κόστους-κέρδους, καθώς και από τους μελλοντικούς στόχους της διάταξης.

Έτσι μπορεί να χρειαστεί να προστεθεί δεύτερος ελεγκτής αν αυξηθεί ο ρυθμός αφίξεως των στοιχείων ή αν μελλοντικά μας ενδιαφέρει το μέγεθος της ουράς αναμονής του ελεγκτή.

Παρακάτω φαίνονται τα στατιστικά για την ουρά αναμονής του ελεγκτή.



Σχήμα 3.5.25

Αν αυτά δεν είναι αποδεκτά για το σύστημα θα πρέπει να προστεθεί δεύτερος ελεγκτής.

### Τυχειότητα αφίξεων

Πριν αποφασιστεί ότι όντως θα προστεθεί ένας ακόμα ελεγκτής εξετάζεται η τυχειότητα των αφίξεων. Εφόσον υπάρχουν αφίξεις κάθε 5'' και κάθε στοιχείο πηγαίνει στην καταβόθρα κάθε 5'' κατά μέσο όρο, δεν θα έπρεπε να συσσωρεύονται καθόλου στοιχεία στην ουρά αναμονής.

Ο λόγος που συσσωρεύονται στοιχεία είναι λόγω της στατιστικής κατανομής των αφίξεων. Ο ρυθμός άφιξης είναι 5'' σύμφωνα με εκθετική κατανομή. Αυτό σημαίνει ότι κατά διαστήματα δεν θα υπάρχει καμία άφιξη και άλλοτε θα φτάνουν γρηγορότερα δημιουργώντας bottleneck.

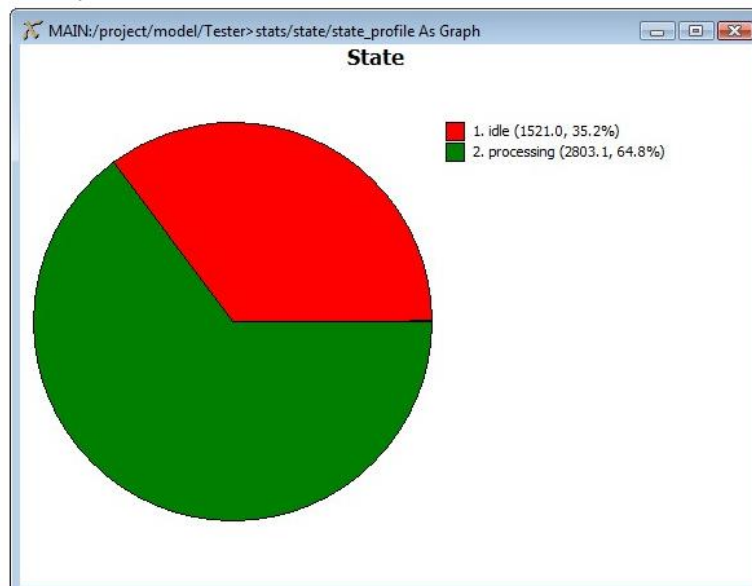
Αλλάζουμε τη στατιστική κατανομή σε κανονική με μέση τιμή 5'' και τυπική απόκλιση 5''. Τρέχοντας ξανά το μοντέλο παρατηρείται ότι το μέσο μέγεθος της ουράς αναμονής και ο μέσος χρόνος παραμονής σε αυτή δεν αυξάνονται τόσο. Αυτή ήταν μία σημαντική βελτίωση και έγινε μόνο αλλάζοντας τη στατιστική κατανομή των αφίξεων.

### Αύξηση του ρυθμού αφίξεων

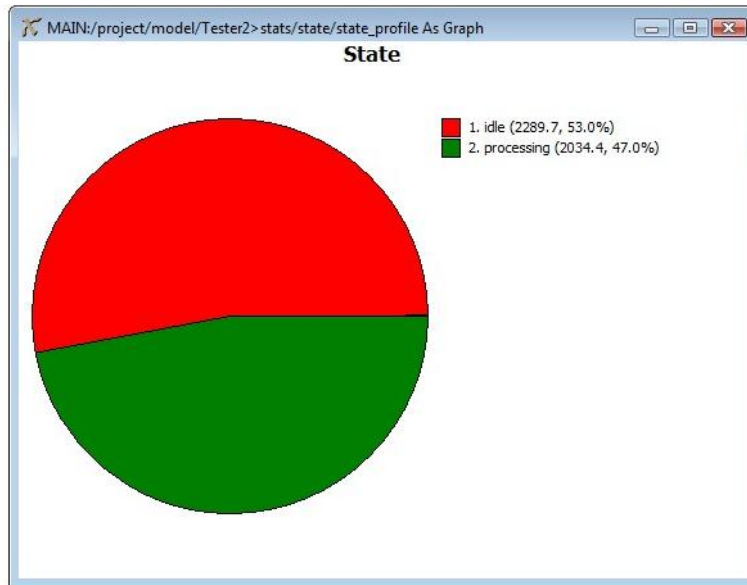
Αν αυξηθεί κατά 15% ο ρυθμός άφιξης των στοιχείων, οι αφίξεις θα γίνονται κάθε 4,25". Εφόσον ο ελεγκτής χρησιμοποιούνταν ήδη στο 100%, είναι προφανές ότι χρειαζόμαστε δεύτερο ελεγκτή.

- Αρχικά αλλάζει στις ιδιότητες της πηγής ο χρόνος μεταξύ των αφίξεων των στοιχείων σε 4,25". Οι αφίξεις θα ακολουθούν κανονική κατανομή με τυπική απόκλιση 5".
- Προσθέτεται ένας δεύτερος επεξεργαστής ο οποίος θα λειτουργεί σαν ελεγκτής και ορίζεται ο χρόνος επεξεργασίας σε 4".
- Όπως και στον πρώτο ελεγκτή ορίζεται το ποσοστό των προϊόντων που θα θεωρούνται αποδεκτά και το αντίστοιχο ποσοστό των σκάρτων.
- Γίνεται επαναφορά του συστήματος στην αρχική κατάσταση με reset και τρέχει το μοντέλο για το λιγότερο 50000 μονάδες χρόνου.
- Τώρα η δεύτερη ουρά αναμονής είναι σχεδόν άδεια, ενώ η ουρά αναμονής για τους επεξεργαστές γεμίζει αρκετά συχνά.

Παρακάτω φαίνονται τα διαγράμματα της κατάστασης του αρχικού ελεγκτή και του δεύτερου ελεγκτή αντίστοιχα.



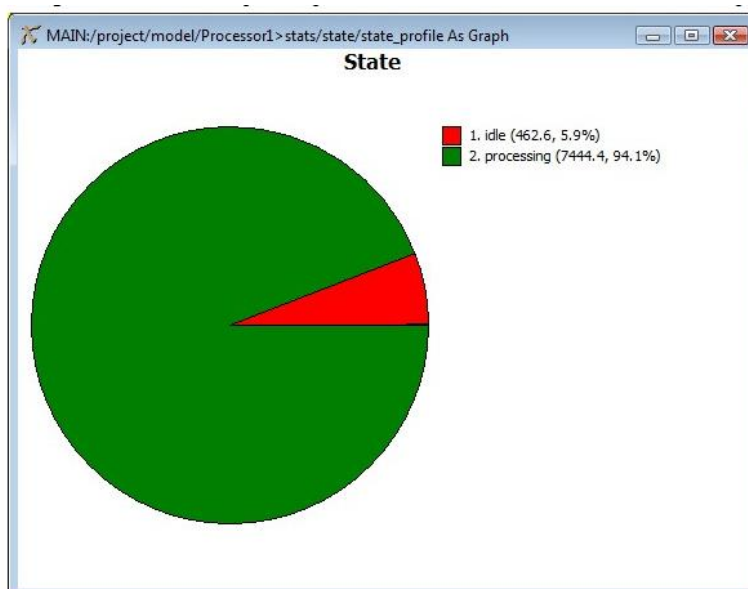
Σχήμα 3.5.26



Σχήμα 3.5.27

Ο λόγος που τα δύο διαγράμματα είναι διαφορετικά είναι επειδή η ουρά αναμονής των ελεγκτών στέλνει στοιχεία στον πρώτο διαθέσιμο ελεγκτή. Όταν είναι ελεύθεροι και οι δύο ελεγκτές το αντικείμενο θα σταλεί στον αρχικό. Τα στοιχεία πηγαίνουν στο δεύτερο ελεγκτή μόνο αν ο αρχικός είναι ήδη απασχολημένος.

Αντίστοιχα, το διάγραμμα κατάστασης των τριών επεξεργαστών είναι το ακόλουθο:



Σχήμα 3.5.28

Τώρα η χρησιμοποίηση των επεξεργαστών είναι σχεδόν στο 100%. Το bottleneck τώρα βρίσκεται πριν τους τρεις επεξεργαστές. Τώρα σε περίπτωση που θέλουμε περεταίρω αύξηση του ρυθμού άφιξης των στοιχείων και αύξηση της χρησιμοποίησης των ελεγκτών, θα πρέπει να προστεθούν περισσότεροι επεξεργαστές. Και για αυτή την απόφαση πρέπει να προηγηθεί ανάλυση κόστους – οφέλους.



## Κεφάλαιο 4

### Μελέτη Περίπτωσης

### Προσομοίωση Διαδικασιών Αποθήκης





#### 4.1 Στοιχεία λειτουργίας της αποθήκης

Η αποθήκη χρησιμοποιείται για την αποθήκευση παλετών και στη συνέχεια για σύνθεση παραγγελιών και την αποστολή τους σε πελάτες. Οι αποθηκευόμενες παλέτες είναι όλες ευρωπαϊκές διαστάσεων 80x120 cm και ύψους: 195cm το 20%, 170cm το 40% και 105cm το 20%. Επίσης, αποθηκεύονται λίγα τεμάχια σε θυρίδες, όταν αυτά δεν σχηματίζουν παλέτες. Οι διαστάσεις των θυρίδων είναι 45x60x45 cm. Η σύνθεση της αποθήκης σε παλετοθέσεις και θέσεις για θυρίδες είναι η εξής:

Είδος θέσης	Διαστάσεις	Αριθμός θέσεων	Σύνολο
Παλέτες τύπου Α	80x120x195 cm	1428	7192
Παλέτες τύπου Β	80x120x170 cm	2738	
Παλέτες τύπου C	80x120x105 cm	3026	
Θυρίδες	80x120x195 cm	3716	3716

Πίνακας 4.1.1

Η διακίνηση των παλετών στην αποθήκη γίνεται με περονοφόρα. Οι θέσεις των θυρίδων είναι ισόγειες για να μπορεί να γίνεται η διακίνηση των προϊόντων χειροκίνητα από μεταφορείς και όχι από περονοφόρα.

➔ **Πλήθος κωδικών που χειρίζεται η αποθήκη**

Η αποθήκη χειρίζεται 310 κωδικούς.

➔ **Τύπος συσκευασίας ανά κωδικό**

Κάθε κωδικός αποτελείται από ένα γράμμα, Α, Β ή C και από έναν αριθμό. Το γράμμα του κάθε κωδικού υποδηλώνει το είδος της συσκευασίας του. Οι τύποι συσκευασίας ανά κωδικό είναι οι εξής:

Τύπος συσκευασίας	Κωδικοί
Παλέτες ύψους 195 cm	A1, A2, ... , A85
Παλέτες ύψους 170 cm	B1, B2, ... , B180
Παλέτες ύψους 105 cm	C1, C2, ... , C45

Πίνακας 4.1.2

### ➔ Αρχικό απόθεμα

Η λειτουργία της αποθήκης ξεκινά χωρίς αρχικό απόθεμα. Έχει οριστεί το διάστημα των 2400 λεπτών, δηλαδή 5 ημερών, ως χρόνος προθέρμανσης του μοντέλου – *warmup time* στον οποίο η αποθήκη παραλαμβάνει προϊόντα και τα τοποθετεί στα ράφια, χωρίς να εκτελείται κάποια άλλη λειτουργία της. Έτσι, η σύνθεση και η αποστολή παραγγελιών ξεκινούν μετά από αυτό το διάστημα.

### ➔ Αριθμός αφίξεων ανά ημέρα

Η αποθήκη έχει σχεδιασθεί έτσι ώστε να πραγματοποιεί 20 παραλαβές την ημέρα.

### ➔ Χρόνοι άφιξης φορτηγών

Η άφιξη των φορτηγών παραλαβών ορίζεται από ένα πίνακα που λέγεται πρόγραμμα αφίξεων – *arrival schedule*, σύμφωνα με τον οποίο φθάνουν φορτηγά περίπου κάθε 23 λεπτά.

### ➔ Μέγεθος και σύνθεση παραλαβών

Τα φορτηγά που φέρνουν προϊόντα στην αποθήκη έχουν χωρητικότητα κατά μέσο όρο  $24 \text{ m}^3$ . Υπολογίστηκε ότι η σύνθεση των παραλαβών και των αποστολών είναι περίπου ίδια με τη σύνθεση της αποθήκης. Έτσι βρέθηκε ο μέσος όγκος της κάθε παλέτας ως εξής:

$$V_{\text{παλέτας}} = 0,8 \times 1,2 \times (0,2 \times 1,95 + 0,4 \times 1,7 + 0,4 \times 1,05) \Rightarrow$$

$$V_{\text{παλέτας}} = 1,43 \text{ m}^3$$

Άρα το φορτηγό των παραλαβών μεταφέρει περίπου 16 παλέτες σε πλήρες φορτίο (*full truck load*). Στο μοντέλο έχει υποτεθεί ότι η κάθε παραλαβή είναι 16 παλέτες και η σύνθεσή της είναι τυχαία, ικανοποιεί όμως την υπόθεση ότι αποτελείται από 20% παλέτες τύπου A, 40% τύπου B και 40% τύπου C.

Το φορτηγό των αποστολών μεταφέρει 11 παλέτες σε πλήρες φορτίο. Έχει υποτεθεί ότι η κάθε παραγγελία αποτελείται από 2 παλέτες ύψους 195 cm, 5 παλέτες ύψους 170 cm και 4 παλέτες ύψους 105 cm.

### ➔ Αριθμός αποστολών ανά ημέρα

Η αποθήκη πραγματοποιεί 6 αποστολές την ημέρα.

### ➔ Χρόνοι άφιξης φορτηγών

Η άφιξη των φορτηγών αποστολών ορίζεται από ένα πίνακα που λέγεται πρόγραμμα αφίξεων – *arrival schedule*, σύμφωνα με τον οποίο φθάνουν φορτηγά περίπου κάθε 73 λεπτά, με πρώτη άφιξη στα 2499 λεπτά.

### ➔ Μέγεθος και σύνθεση αποστολών

Τα φορτηγά που παίρνουν παραγγελίες για πελάτες έχουν χωρητικότητα κατά μέσο όρο  $16 \text{ m}^3$ . Εφόσον ο μέσος όγκος της παλέτας βρέθηκε  $V_{\text{παλέτας}} = 1,43 \text{ m}^3$ , το φορτηγό των αποστολών μεταφέρει περίπου 11 παλέτες σε πλήρες φορτίο. Έχει υποθεθεί ότι η κάθε παραγγελία αποτελείται από 2 παλέτες ύψους 195 cm, 5 παλέτες ύψους 170 cm και 4 παλέτες ύψους 105 cm.

Στη συνέχεια του κεφαλαίου θα αναζητηθεί η λύση που ικανοποιεί τις απαιτήσεις της αποθήκης. Έτσι, θα παρουσιαστούν τρεις προτάσεις. Και στις τρεις περιπτώσεις γίνεται χρήση των ακόλουθων:

- ☞ ενός περονοφόρου για την εκφόρτωση και τη φόρτωση των φορτηγών
- ☞ δύο εργατών/μεταφορέων που είναι υπεύθυνοι για το χειρισμό των επεξεργαστών του τμήματος σύνθεσης παραγγελιών και για τη μεταφορά των κιβωτίων σε θυρίδες

Αυτό που αλλάζει σε κάθε περίπτωση είναι ο αριθμός των περονοφόρων που χρησιμοποιούνται για τη διακίνηση των παλετών στην αποθήκη. Έτσι, στην πρώτη περίπτωση, Scenario 1, γίνεται χρήση ενός περονοφόρου, στη δεύτερη περίπτωση, Scenario 2, γίνεται χρήση δύο περονοφόρων, ενώ στην τρίτη περίπτωση, Scenario 3, γίνεται χρήση τριών περονοφόρων.

Έπειτα θα γίνει αξιολόγηση και σύγκριση των σεναρίων με τη βοήθεια δεικτών απόδοσης, διαγραμμάτων και αναφορών. Ακόμα θα γίνει εκτίμηση για το κόστος εφαρμογής της κάθε πρότασης. Τέλος, θα ληφθούν υπόψη όλες οι παράμετροι ώστε να καταλήξουμε στη λύση που ικανοποιεί τις απαιτήσεις της αποθήκης και ταυτόχρονα είναι οικονομικά συμφέρουσα.

#### 4.2 Πολεοδομικοί περιορισμοί

Η αποθήκη θα εγκατασταθεί σε οικοπέδο διαστάσεων  $210 \times 180$  m, δηλαδή συνολικού εμβαδού  $37800 \text{ m}^2$ . Οι συντελεστές του ΓΟΚ στην περιοχή είναι:

- ➔ Συντελεστής Κάλυψης: 30%

$$\text{Συντελεστής Κάλυψης} = \frac{\text{Επιτρεπόμενη Κάλυψη}}{\text{Συνολικό Εμβαδό Οικοπέδου}} \Rightarrow$$

$$\text{Επιτρεπόμενη Κάλυψη} = 11340 \text{ m}^2$$

- ➔ Συντελεστής Δόμησης: 60%

$$\text{Συντελεστής Δόμησης} = \frac{\text{Επιτρεπόμενη Δόμηση}}{\text{Συνολικό Εμβαδό Οικοπέδου}} \Rightarrow$$

$$\text{Επιτρεπόμενη Δόμηση} = 22680 \text{ m}^2$$

- ➔ Συντελεστής Όγκου: 3,6

$$\text{Συντελεστής Όγκου} = \frac{\text{Επιτρεπόμενος Όγκος}}{\text{Συνολικό Εμβαδό Οικοπέδου}} \Rightarrow$$

$$\text{Επιτρεπόμενος Όγκος} = 136080 \text{ m}^3$$

- ➔ Μέγιστο επιτρεπόμενο ύψος: 16 m
- ➔ Πρασιά: 15 m

Σύμφωνα με τις απαιτήσεις για μελλοντική επέκταση, η παρούσα αποθήκη θα καταλάβει τη μισή έκταση του οικοπέδου. Άρα:

- ➔ *Επιτρεπόμενη κάλυψη : **5670m<sup>2</sup>***
- ➔ *Επιτρεπόμενη Δόμηση : 22680m<sup>2</sup>*
- ➔ *Επιτρεπόμενος Όγκος : 136080 m<sup>3</sup>*
- ➔ *Μέγιστο επιτρεπόμενο ύψος: 16 m*
- ➔ *Πρασιά: **15 m***

### 4.3 Σχεδιασμός Αποθήκης

Για καλύτερη ροή των προϊόντων η αποθήκη έχει λόγο διαστάσεων 2: 1. Για το σχεδιασμό της αποθήκης επιλέγουμε σύστημα FIFO (First In First Out). Τα υλικά δεν είναι ευαίσθητα στο χρόνο, αλλά η χρησιμοποίηση του FIFO επιβάλλεται σταδιακά από τους διεθνείς κανονισμούς και από τα συστήματα διασφάλισης ποιότητας. Χρησιμοποιούμε άναρχο σύστημα αποθήκευσης για καλύτερη εκμετάλλευση του χώρου, στο οποίο είναι καθορισμένες οι θέσεις για τους διαφορετικούς τύπους παλετών.

#### 4.3.1 Παλέτες

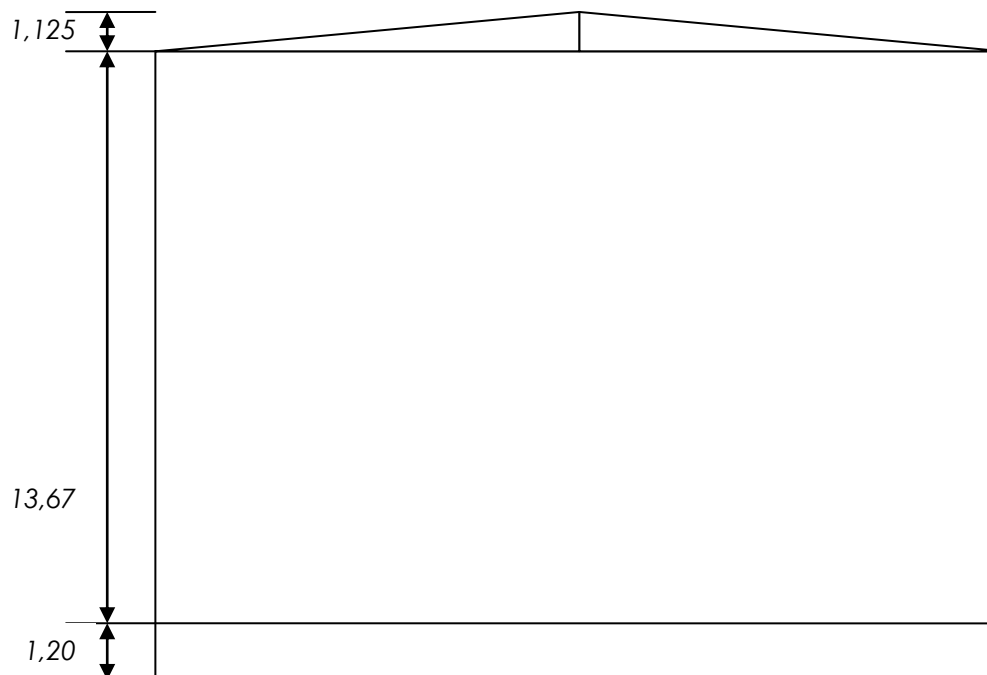
Η διάταξη των ραφιών είναι τύπου back to back και θα χρησιμοποιηθεί το περνοφόρο στενών διαδρόμων ETX 515 της εταιρίας Jungheinrich το οποίο απαιτεί διαδρόμους πλάτους 2,8 m.

Πριν προχωρήσουμε στον υπολογισμό του αριθμού των παλετών κατά ύψος, πρέπει πρώτα να υπολογίσουμε το καθαρό ύψος της αποθήκης. Το κτίριο της αποθήκης έχει κολώνες και οροφές κάθε 22,5m. Η οροφή έχει κλίση 10% και το κτίριο είναι όλο υπερυψωμένο σε ύψος 1,20m για λόγους κατασκευής ραμπών φορτοεκφόρτωσης για τα φορτηγά.

Επομένως, το καθαρό ύψος της αποθήκης θα είναι 13,65 m. Μεταξύ των παλετών θα χρησιμοποιηθούν δοκίδες πάχους 0,1 m και πάνω από κάθε παλέτα θα υπάρχει ελεύθερος χώρος 0,2 – 0,25 m ώστε να μπορεί το περνοφόρο να τις προσεγγίσει εύκολα. Έτσι, βρίσκουμε πόσες παλέτες κατά ύψος χωράνε από κάθε τύπο.

- $n_1 \cdot 1,95 + (n_1 - 1) \cdot 0,1 + (n_1 - 4) \cdot 0,2 + (n_1 - 2) \cdot 0,25 \leq 13,675 \rightarrow n_1 = 6$  παλέτες
- $n_2 \cdot 1,70 + (n_2 - 1) \cdot 0,1 + n_2 \cdot 0,25 \leq 13,675 \rightarrow n_2 = 6$  παλέτες
- $n_3 \cdot 1,05 + (n_3 - 1) \cdot 0,1 + n_3 \cdot 0,25 \leq 13,675 \rightarrow n_3 = 9$  παλέτες

Παρακάτω φαίνεται η μορφή της αποθήκης.



Σχήμα 4.3.1

#### 4.3.2 Θυρίδες

Όσον αφορά τις θυρίδες, σε αυτές πάνω από κάθε στρώση υπάρχει δοκίδα πάχους 0,05 m. Σε ύψος χωράνε 4 θυρίδες σε παλετοθέση ύψους 1,95 m. Πάνω από τις θυρίδες μπορούν να τοποθετηθούν ράφια παλετών.

#### 4.3.3 Τμήματα αποθήκης

Η αποθήκη έχει διαστάσεις 90,9m x 61,9m. Το εμβαδό της είναι 5626,71 m<sup>2</sup>. Μπροστά από τον αποθηκευτικό χώρο βρίσκεται το τμήμα παραλαβοαποστολών, μήκους 90,9m και πλάτους 17m. Μέρος αυτού του χώρου χρησιμοποιείται για τη σύνθεση των παραγγελιών. Στο τμήμα σύνθεσης παραγγελιών οδηγούνται παλέτες ύψους 170 cm και 105 cm οι οποίες “σπάνε” σε κιβώτια. Στη συνέχεια, ανάλογα με τη σύνθεση της κάθε παραγγελίας δημιουργούνται νέες παλέτες ύψους 170 cm οι οποίες περιέχουν 12 κιβώτια διαφορετικών κωδικών. Έτσι σε αυτό το τμήμα οδηγούνται παλέτες κωδικών που η ζήτησή τους περιορίζεται σε λίγα μόνο κιβώτια και όχι σε ολόκληρη την παλέτα. Τα κιβώτια που δεν αποστέλλονται σε κάποια παραγγελία αποθηκεύονται στις θυρίδες. Σε αυτό το τμήμα δεν οδηγούνται παλέτες ύψους 195 cm επειδή αυτές περιέχουν μόνο ένα μεγάλο τεμάχιο.

Μπροστά από το τμήμα των παραλαβοαποστολών υπάρχουν ράμπες για την εκφόρτωση και τη φόρτωση των φορητών πλάτους 4 m η κάθε μία. Υπάρχουν 10 ράμπες εκφόρτωσης και 5 ράμπες φόρτωσης.

Στο Παράρτημα υπάρχει και η κάτοψη της αποθήκης σχεδιασμένη με τη βοήθεια του Autocad.

#### **4.4 Μοντελοποίηση της αποθήκης**

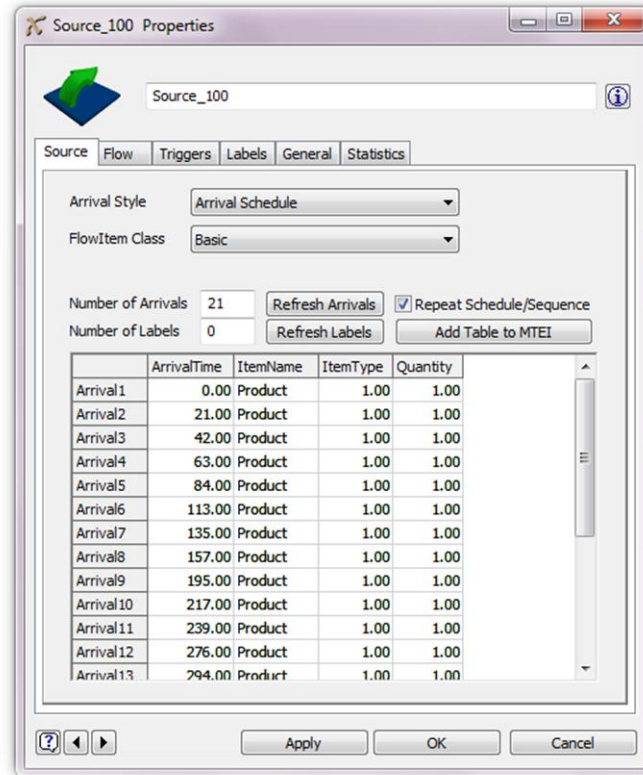
Τα αντικείμενα που έχουν χρησιμοποιηθεί για τη μοντελοποίηση της αποθήκης είναι πηγές – sources, καταβόθρες – sinks, ουρές αναμονής – queues, ράφια – racks, επεξεργαστές – processors, μεταφορείς – transporters, μεταφορικές ταινίες – conveyors, διεκπεραιωτές – dispatchers και κόμβοι δικτύου – network nodes. Παρακάτω ακολουθεί περιγραφή του κάθε αντικειμένου καθώς και σχήματα, μετά την ενσωμάτωση των αντικειμένων στο μοντέλο. Στα σχήματα έχει γίνει απόκρυψη των συνδέσεων μεταξύ των αντικειμένων για λόγους ευκρίνειας.

##### **4.4.1 Πηγές – Sources**

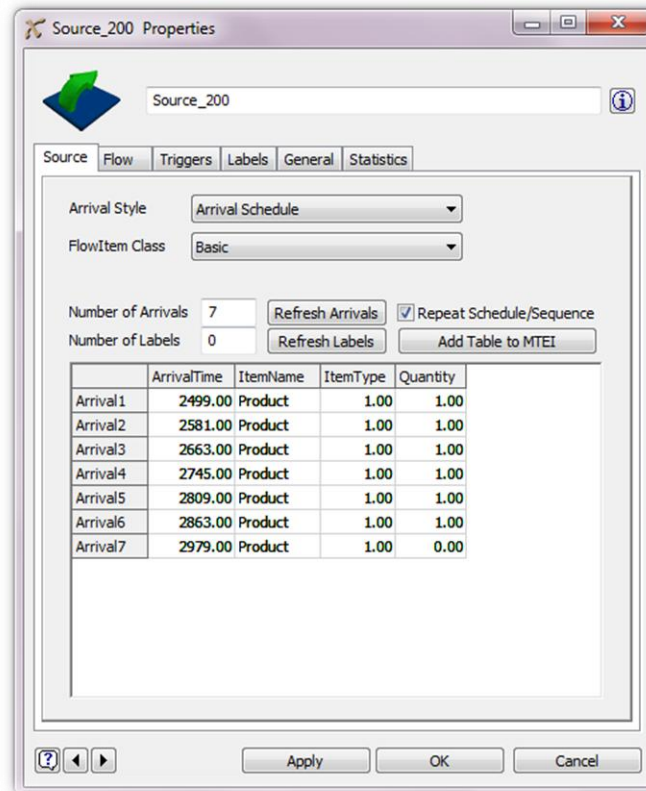
Στο μοντέλο, οι αφίξεις προϊόντων μοντελοποιούνται από το στοιχείο της πηγής. Έτσι στην περίπτωση μας έχουμε τα ακόλουθα είδη πηγών:

##### **Πηγές που τροφοδοτούν τα φορητά παραλαβών/ αποστολών**

Πρόκειται για δύο πηγές, μία για τις παραλαβές και μία για τις αποστολές, οι οποίες δημιουργούν ένα στοιχείο, σύμφωνα με το πρόγραμμα αφίξεων που έχει ορισθεί. Είναι συνδεδεμένες με 10 φορητά παραλαβών και 5 φορητά αποστολών και από τις επιλογές ροής στοιχείων έχει ορισθεί ότι τα φορητά χρησιμοποιούνται κυκλικά (για τις παραλαβές και τις αποστολές αντίστοιχα). Οι πηγές είναι συνδεδεμένες με 10 και 5 επεξεργαστές αντίστοιχα στους οποίους μεταφέρονται τα στοιχεία. Αυτές οι πηγές δεν έχουν πρακτική σημασία για το μοντέλο, έχουν χρησιμοποιηθεί μόνο για να προκληθεί κίνηση των φορητών προς τις ράμπες.

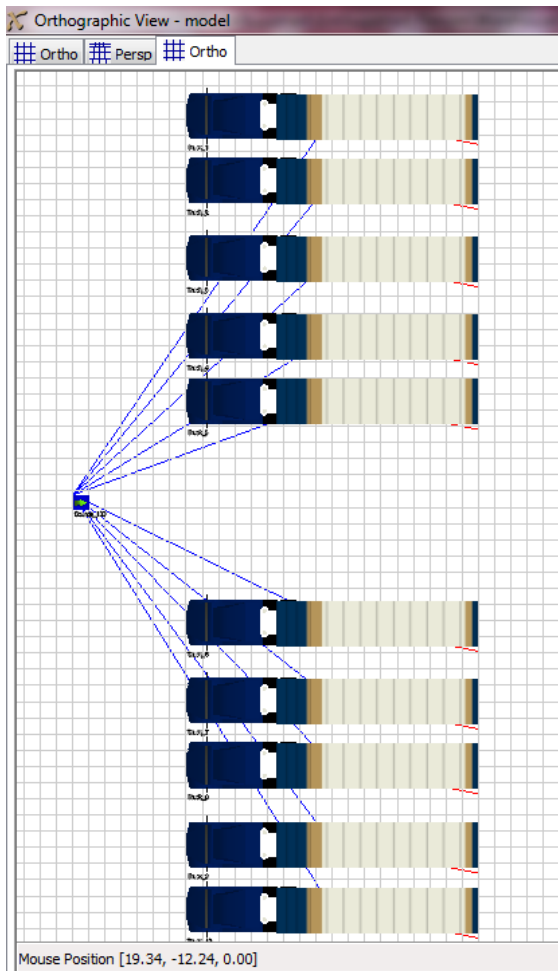


Σχήμα 4.4.1.1

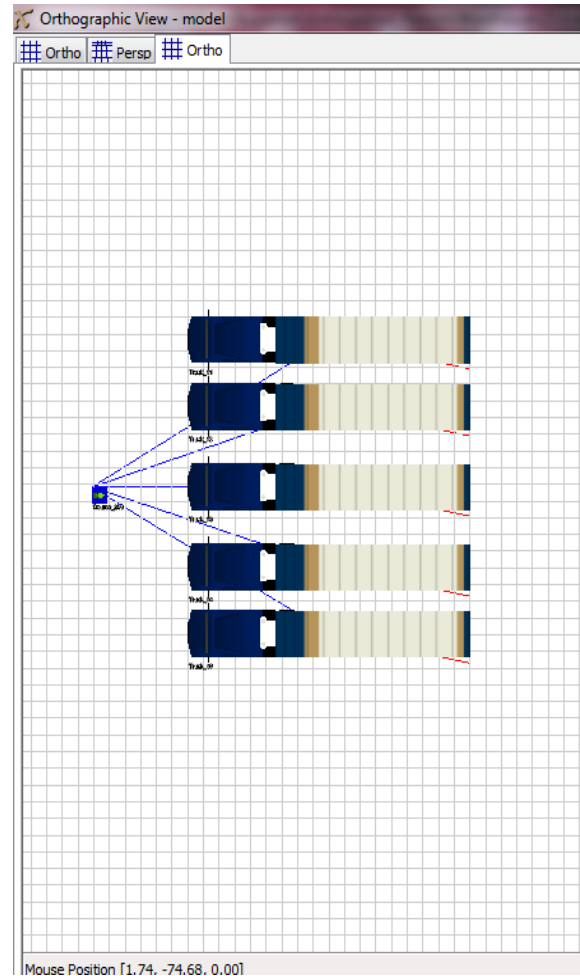


Σχήμα 4.4.1.2





Σχήμα 4.4.1.3

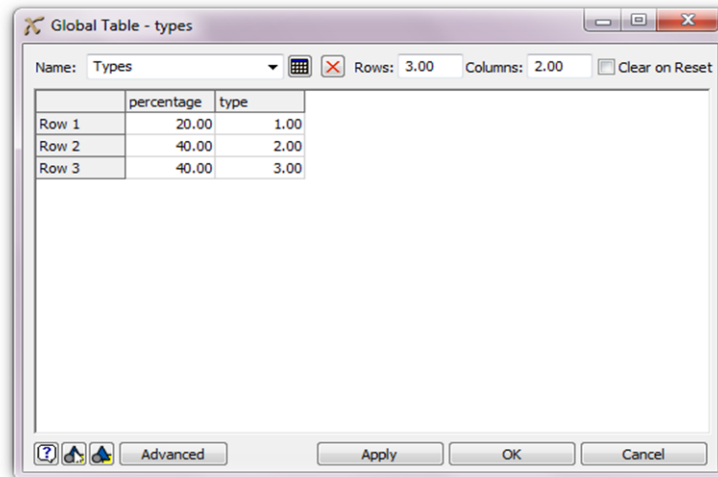


Σχήμα 4.4.1.4

### Πηγές που παράγουν στοιχεία στο τμήμα παραλαβών

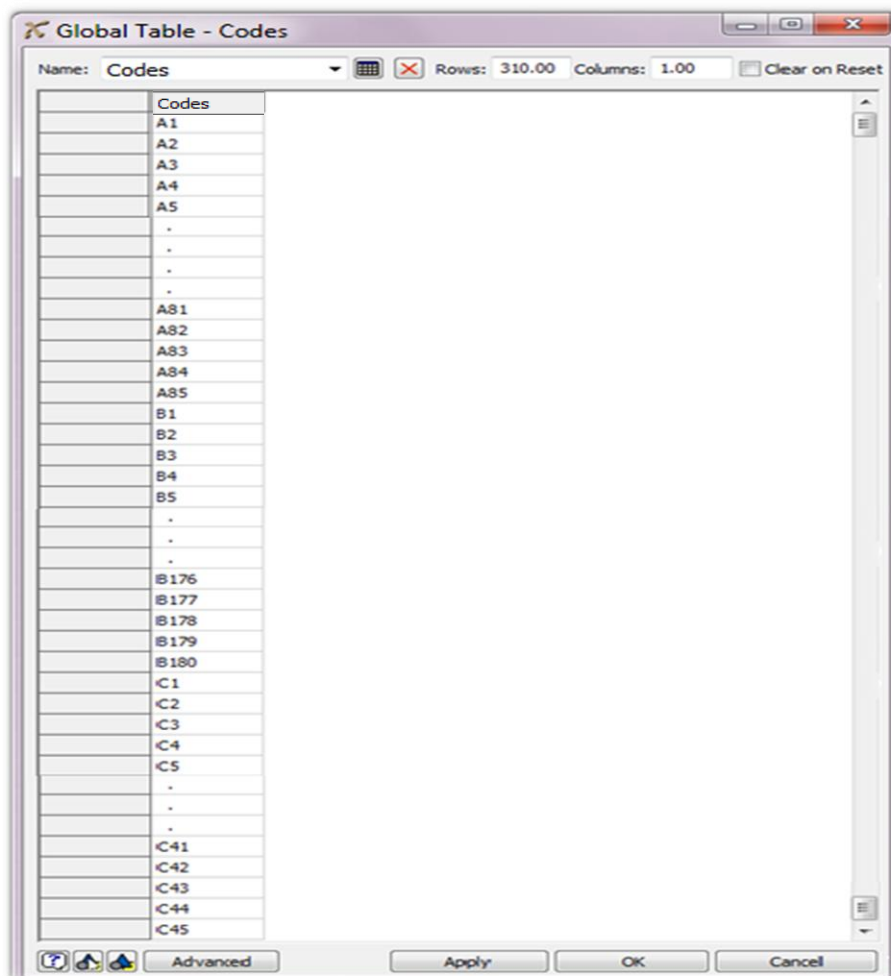
Πρόκειται για 10 πηγές, όσες και οι ράμπες εκφόρτωσης, οι οποίες αρχικά είναι ανενεργές και ενεργοποιούνται κάθε φορά που φτάνει στη ράμπα κάποιο φορτηγό. Τα στοιχεία που δημιουργούν αυτές οι πηγές είναι φορτωμένες παλέτες (loaded pallets).

- Κατά τη δημιουργία του κάθε στοιχείου ορίζεται ο τύπος – itemtype του καθενός. Για τη διαδικασία αυτή χρησιμοποιείται ο ακόλουθος πίνακας “Types”, ο οποίος δίνει πληροφορίες για το ποσοστό των στοιχείων που πρέπει να δημιουργηθούν από κάθε τύπο.



Σχήμα 4.4.1.5

- Έπειτα, ανάλογα με τον τύπο του στοιχείου ορίζονται οι διαστάσεις του και επιλέγεται με τυχαίο τρόπο ο κωδικός προϊόντος στον οποίο θα αντιστοιχεί. Για τη διαδικασία αυτή χρησιμοποιείται ο ακόλουθος πίνακας “Codes”.



Σχήμα 4.4.1.6

- ➔ Τέλος, προστίθεται στο αντικείμενο ετικέτα. Σε ένα πεδίο αναγράφεται ο κωδικός και σε ένα άλλο αναγράφεται ο χρόνος άφιξης του στοιχείου.

Παρακάτω φαίνεται ο κώδικας που εκτελείται για τις διαδικασίες αυτές.

```
treenode item = parnode(1);
treenode current = ownerobject(c);
int rownumber = parval(2);

setitemtype(item, dempirical("Types", 0));

int type = getitemtype(item);

switch(type)
{
    case 1:
    {
        setsize(item, 0.8, 1.2, 1.95);
        addlabel
        (item, "code", gettablestr("Codes", uniform(1, 85, 1), 1));
        break;
    }
    case 2:
    {
        setsize(item, 0.8, 1.2, 1.70);
        addlabel
        (item, "code", gettablestr("Codes", uniform(86, 265, 1), 1));
        break;
    }
    case 3:
    {
        setsize(item, 0.8, 1.2, 1.05);
        addlabel
        (item, "code", gettablestr("Codes", uniform(266, 310, 1), 1));
        break;
    }
}

int creation_time=getcreationtime(item);

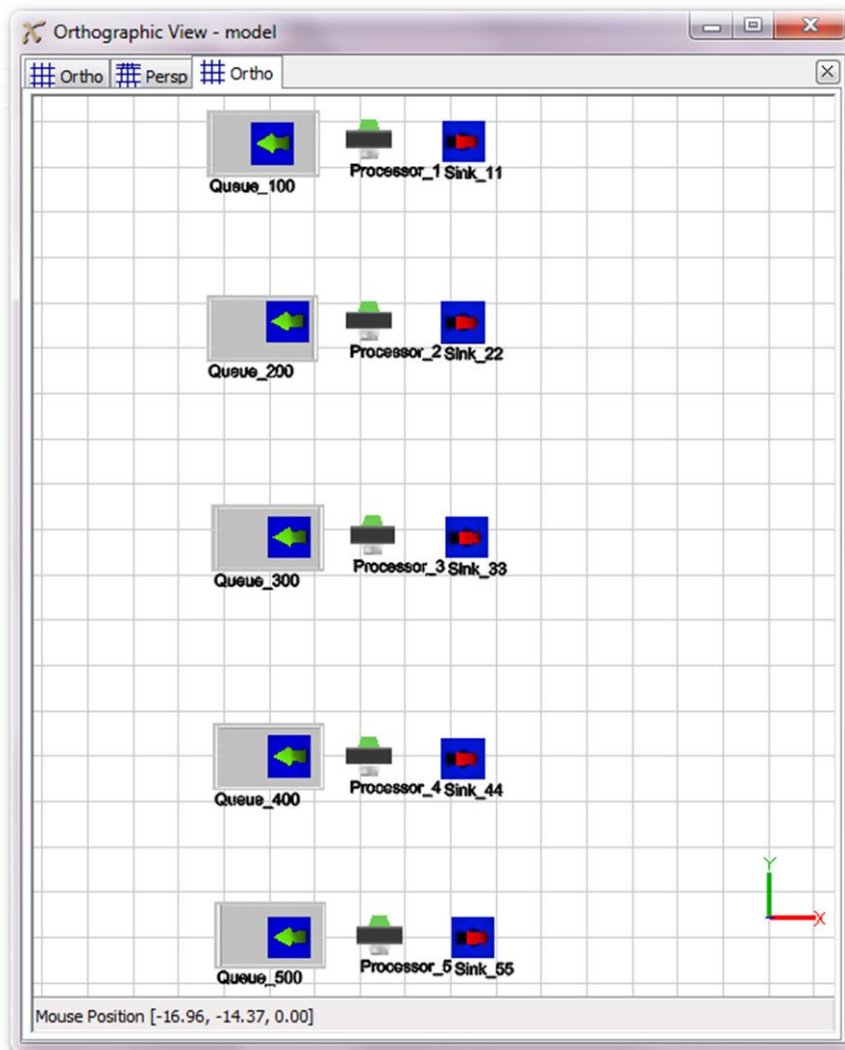
addlabel(item, "creation time", creation_time);
```

- ➔ Με την άφιξη του φορτηγού πρέπει η πηγή των παλετών να δημιουργήσει ταυτόχρονα 16 στοιχεία. Επίσης έχει ορισθεί ότι μετά την άφιξη 16 παλετών η πηγή σταματά την παραγωγή οντοτήτων και ενεργοποιείται ξανά με την εκ νέου άφιξη φορτηγού. Αυτά γίνονται με τη βοήθεια του ακόλουθου κώδικα:

```
treenode current = ownerobject(c);

int outval=getoutput(current);
switch(outval)
{
    case 15:
    {
        closeoutput(current);
        break;
    }
    case 31:
    {
        closeoutput(current);
        break;
    }
    case 47:
    {
        closeoutput(current);
        break;
    }
    case 63:
    {
        closeoutput(current);
        break;
    }
    case 79:
    {
        closeoutput(current);
        break;
    }
        .
        .
        .
        .
        .
        .
    case 751:
    {
        closeoutput(current);
        break;
    }
    case 767:
    {
        closeoutput(current);
        break;
    }
    case 783:
    {
        closeoutput(current);
        break;
    }
    case 799:
    {
        closeoutput(current);
        break;
    }
    case 815:
    {
        closeoutput(current);
        break;
    }
}
```

- ➔ Στη συνέχεια συνδέεται η κάθε πηγή με μία ουρά αναμονής, όπου παραμένουν τα στοιχεία μέχρι την εκφόρτωσή τους.



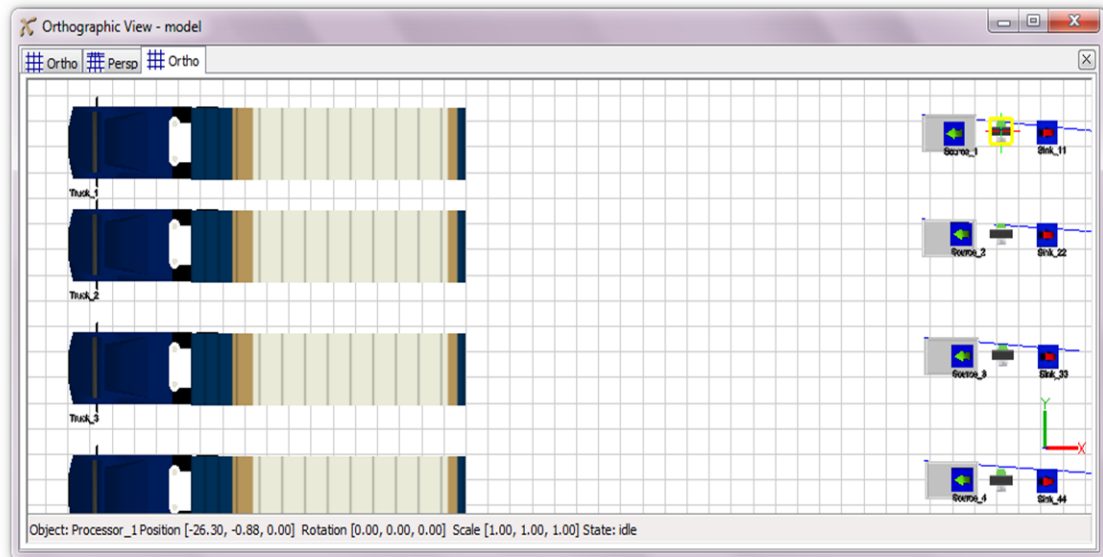
Σχήμα 4.4.1.7

#### 4.4.2 Καταβόθρες – Sinks

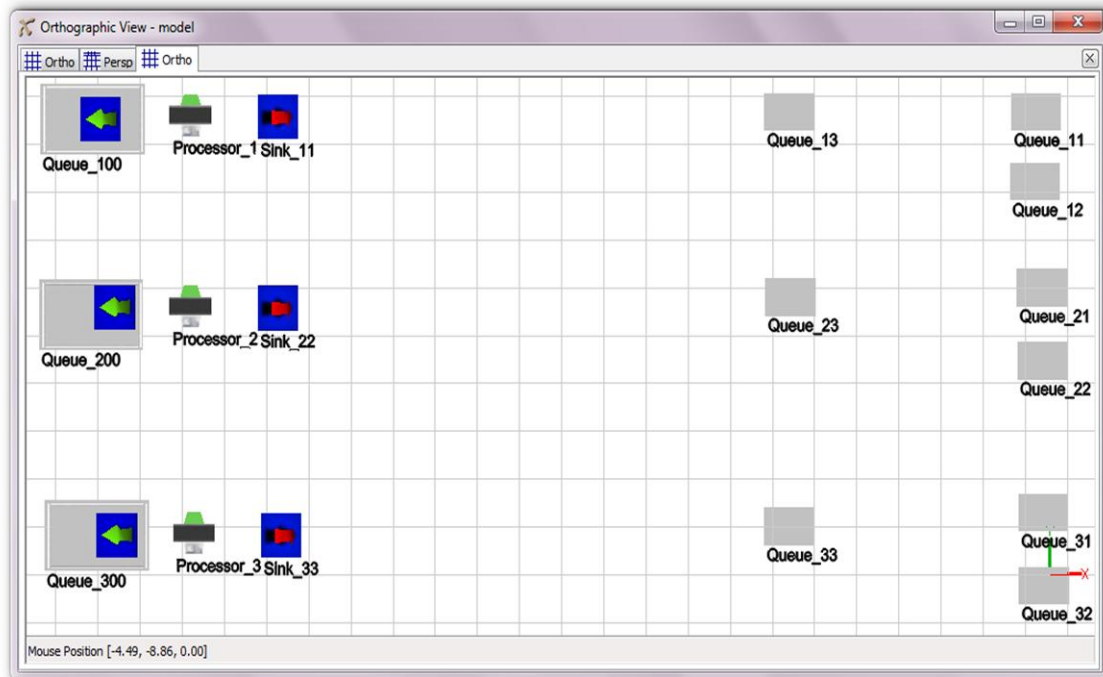
Πρόκειται για αντικείμενα που “καταστρέφουν” τα στοιχεία όταν γίνει μεταφορά τους σε αυτά. Όταν δηλαδή ένα στοιχείο μεταβεί σε μία καταβόθρα, καταστρέφεται και παύει να υπάρχει στο μοντέλο. Στο μοντέλο έχουν χρησιμοποιηθεί τα ακόλουθα είδη καταβόθρων:

### Καταβόθρες στο τμήμα παραλαβών/ αποστολών

Αυτές οι καταβόθρες βρίσκονται μετά τους επεξεργαστές όπου μεταφέρονται τα αντικείμενα της πηγής *Source\_100*. Έτσι, μετά την τυπική τους επεξεργασία, τα αντικείμενα καταστρέφονται στην καταβόθρα. Αντίστοιχες καταβόθρες υπάρχουν και για την πηγή *Source\_200* του τμήματος αποστολών με ακριβώς την ίδια λειτουργία.



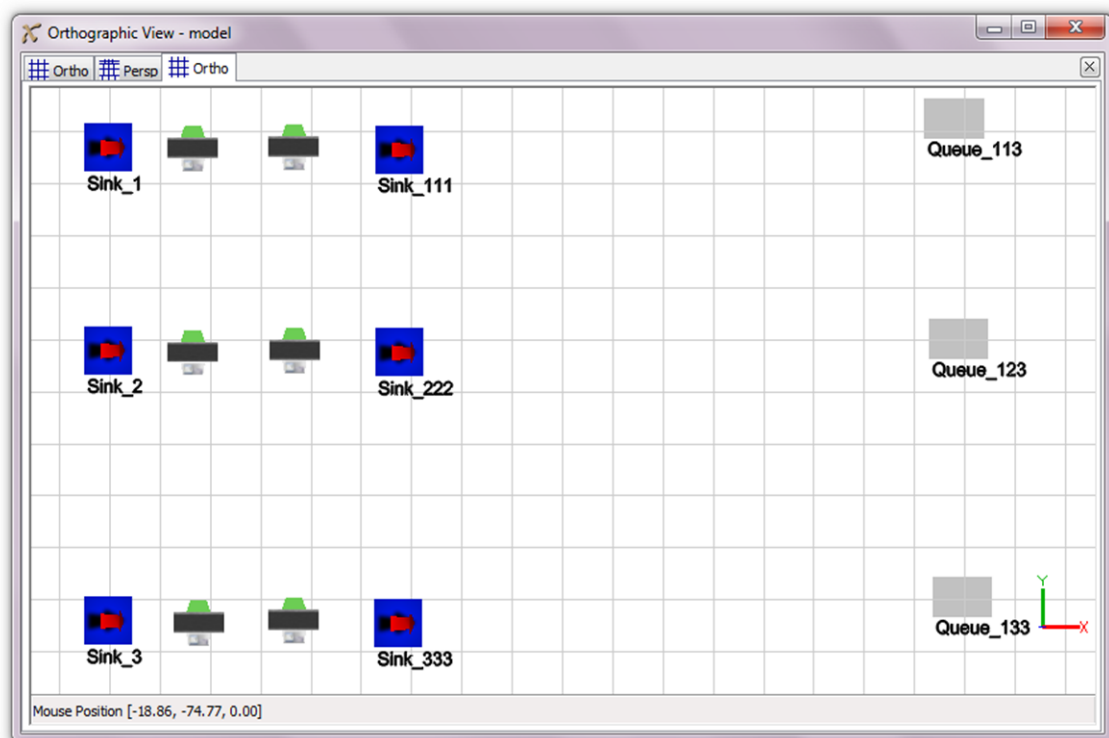
Σχήμα 4.4.2.1



Σχήμα 4.4.2.2

### Καταβόθρες στο τμήμα αποστολών

Η διαδικασία αποστολής παραγγελιών μοντελοποιείται από αυτές τις καταβόθρες. Οι παραγγελόμενες παλέτες συγκεντρώνονται στις ουρές αναμονής του τμήματος αποστολών. Όταν φτάσει το φορηγό στη ράμπα ξεκινά η διαδικασία φόρτωσής του, δηλαδή στο μοντέλο αρχίζει η τροφοδοσία των επεξεργαστών που βρίσκονται μετά τις ουρές αναμονής. Τελικά, τα προϊόντα αυτά καταλήγουν στις καταβόθρες και το φορηγό επιστρέφει στην αρχική του θέση.



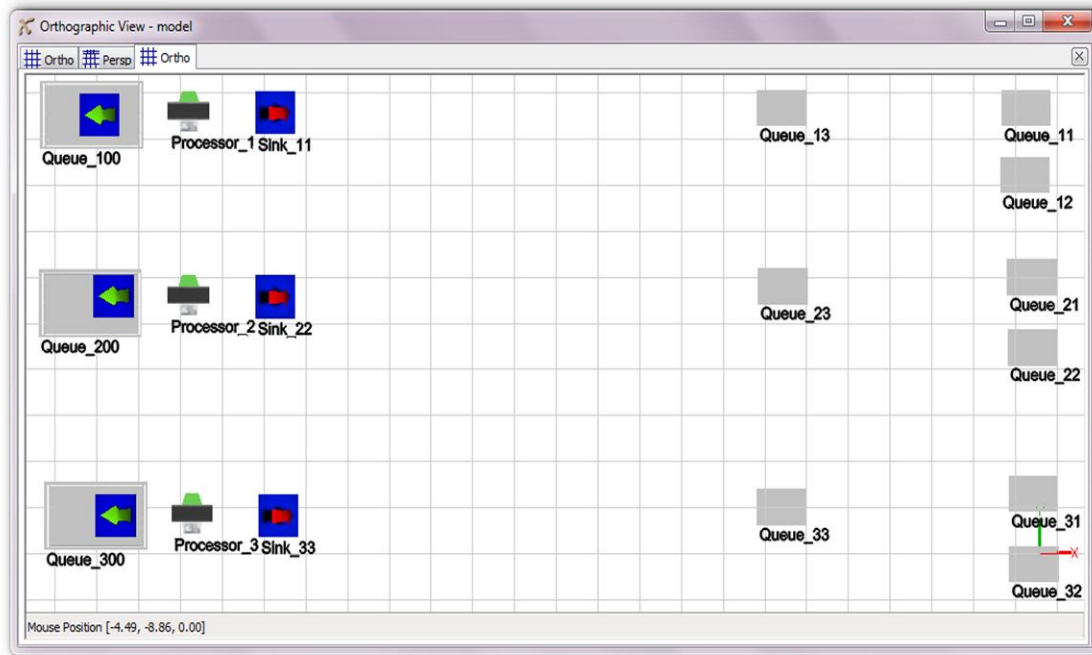
Σχήμα 4.4.2.3

### 4.4.3 Ουρές αναμονής – Queues

Σε διάφορα σημεία στο μοντέλο υπάρχουν ουρές αναμονής πριν από κάποια διαδικασία. Σε αυτές συσσωρεύονται στοιχεία και περιμένουν τη μεταφορά τους σε κάποιο αντικείμενο του μοντέλου ή την επεξεργασία τους. Έτσι έχουμε τα ακόλουθα είδη ουρών αναμονής:

## Ουρές αναμονής στο τμήμα παραλαβών

Όπως αναφέρθηκε προηγουμένως, στο χώρο των ραμπών, μετά από κάθε πηγή υπάρχει μία ουρά αναμονής. Εκεί συσσωρεύονται τα στοιχεία μέχρι την εκφόρτωσή τους. Αυτές οι ουρές αναμονής συνδέονται με άλλες τρεις που βρίσκονται στο χώρο παραλαβών και αντιστοιχούν στους τύπους των παλετών Α, Β και C. Το περνοφόρο που πραγματοποιεί την εκφόρτωση των φορτηγών μεταφέρει την κάθε παλέτα στην αντίστοιχη ουρά αναμονής, ανάλογα με τον τύπο της.



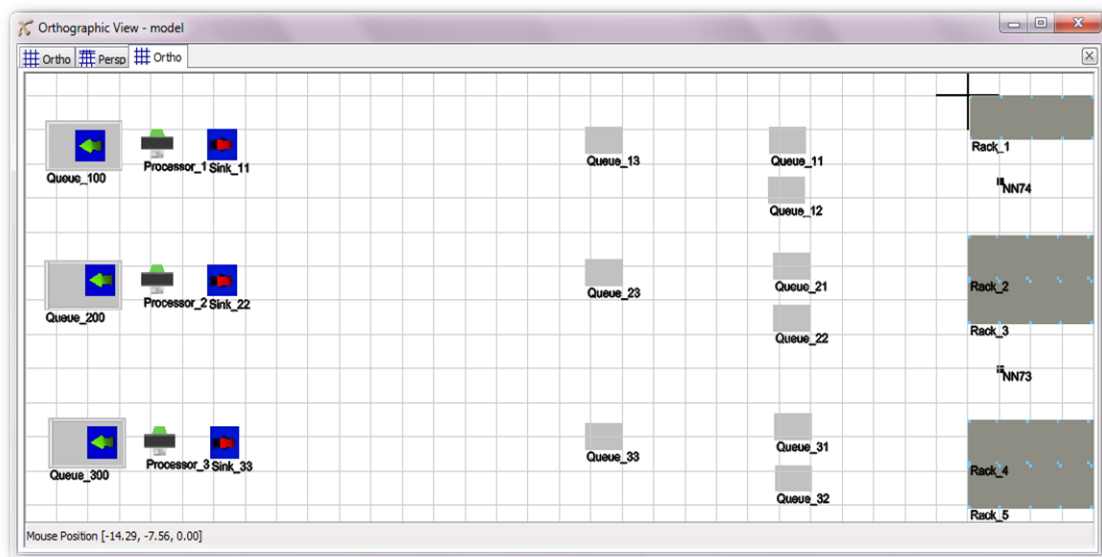
Σχήμα 4.4.3.1

Κατά την έξοδο των στοιχείων από την πρώτη ουρά αναμονής, δίνεται σε αυτά ένα όνομα της μορφής "Item\_", ακολουθούμενο από τον αύξοντα αριθμό του στοιχείου, σύμφωνα με τη σειρά άφιξής του. Σε αυτό βοηθά ένας πίνακας ο οποίος αποθηκεύει τον αύξοντα αριθμό του τελευταίου στοιχείου που εισήχθη στο σύστημα καθώς και ο ακόλουθος κώδικας:

```
treenode item = parnode(1);
treenode current = ownerobject(c);
int port = parval(2);
int rows=gettablelrows("Items");
string tablename = "Items";
setname(item,concat("Item_", numtostring(inc(gettablecell("Items",1,1),1),
0,0)));
for(int x=1; x<=rows; x++)
{
if(gettablenum("Items",x,1)==0)
{
settablestr(tablename,x,1,getname(item));
return(x);
}
}
}
```



Οι ουρές αναμονής που βρίσκονται στο χώρο παραλαβών είναι με τη σειρά τους συνδεδεμένες με τον αντίστοιχο τύπο ραφιών, όπου δηλαδή αποθηκεύονται οι παλέτες A, B ή C αντίστοιχα. Έχει ορισθεί το μέγιστο περιεχόμενό αυτών των ουρών αναμονής να είναι 16 στοιχεία. Στις επιλογές ροής αντικειμένων έχει ορισθεί να γίνεται μεταφορά των παλετών στο πρώτο διαθέσιμο ράφι του αντίστοιχου τύπου μέσω περονοφόρου.



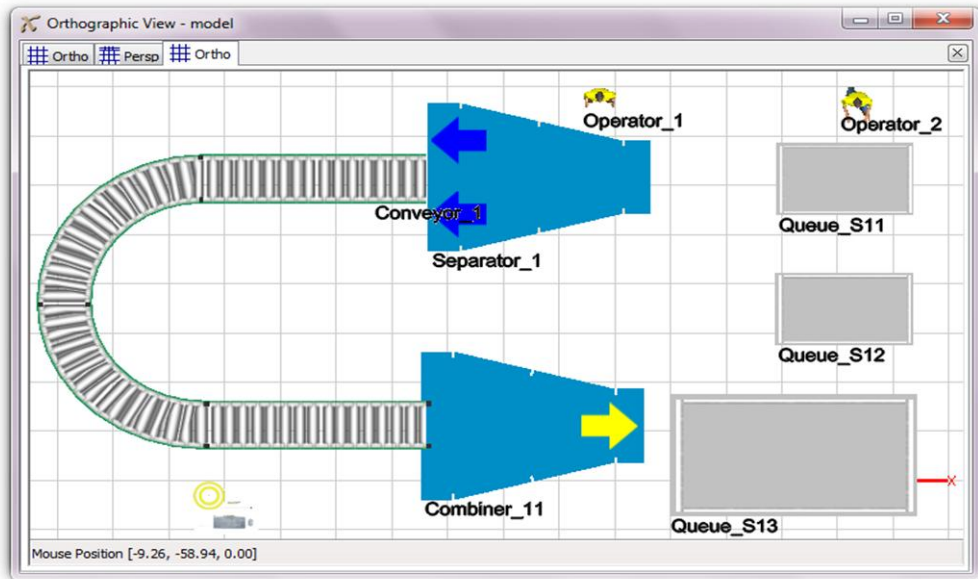
Σχήμα 4.4.3.2

### Ουρές αναμονής στο τμήμα σύνθεσης παραγγελιών

Υπάρχει μία ουρά αναμονής μπροστά από τον επεξεργαστή διαχωρισμού κιβωτίων - separator, η οποία είναι συνδεδεμένη με τα ράφια τύπου B και C και με τον επεξεργαστή. Σε αυτή συσσωρεύονται οι παλέτες τύπου B και C πριν την επεξεργασία τους για το διαχωρισμό των κιβωτίων. Το μέγιστο περιεχόμενό της έχει ορισθεί να είναι ένα στοιχείο.

Μετά τον επεξεργαστή διαχωρισμού, όσα κιβώτια δεν έχουν χρησιμοποιηθεί για τη σύνθεση κάποιας παραγγελίας συσσωρεύονται σε ουρά αναμονής απ' όπου τα παραλαμβάνει μεταφορέας και τα μεταφέρει στην πρώτη διαθέσιμη για αποθήκευση θυρίδα. Κατά την έξοδο των στοιχείων από αυτή την ουρά αναμονής, δίνεται σε αυτά ένα όνομα της μορφής "Item\_", ακολουθούμενο από τον αύξοντα αριθμό του στοιχείου, με τρόπο όμοιο με αυτόν που αναλύθηκε και νωρίτερα.

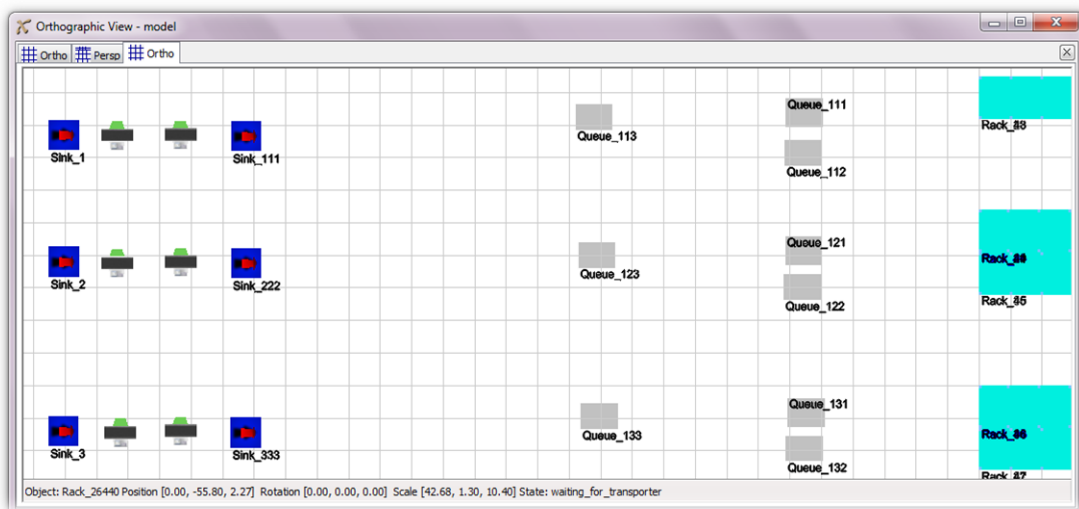
Τέλος, μετά τη σύνθεση νέων παλετών αυτές συσσωρεύονται σε μία ουρά αναμονής απ' όπου τις μεταφέρει περονοφόρο στο τμήμα αποστολών. Το μέγιστο περιεχόμενο αυτής της ουράς αναμονής έχει ορισθεί σε 10 στοιχεία.



Σχήμα 4.4.3.3

### Ουρές αναμονής στο τμήμα αποστολών

Σε αυτό το τμήμα υπάρχουν 3 ουρές αναμονής μπροστά από κάθε ράμπα που αντιστοιχούν στους τύπους των παλετών Α, Β και C. Έχει ορισθεί το μέγιστο περιεχόμενό τους να είναι 2, 5 και 4 στοιχεία αντίστοιχα, δίνουν δηλαδή άθροισμα 11 παλετών που είναι το μέγεθος των αποστελλόμενων παραγγελιών. Κάθε ουρά αναμονής είναι συνδεδεμένη με τον αντίστοιχο τύπο ραφιών, όπου δηλαδή αποθηκεύονται οι παλέτες Α ή C αντίστοιχα και από τα οποία τροφοδοτούνται, καθώς και με τις ουρές αναμονής όπου συσσωρεύονται οι ανασυντιθέμενες παλέτες. Επίσης είναι συνδεδεμένες με τις αντίστοιχες καταβόθρες – sinks όπου αποστέλλονται τα στοιχεία μετά την άφιξη των φορτηγών.



Σχήμα 4.4.3.4

#### 4.4.4 Ράφια - Racks

- ➔ Αρχικά στήνεται ο χώρος της αποθήκης με τη χρήση ραφιών – *racks*. Επιλέγεται ο αριθμός των στηλών και των επιπέδων που θα έχει το καθένα και ορίζονται οι διαστάσεις 0,97x1,3 cm και ύψος αντίστοιχο με τον τύπο των στοιχείων που αποθηκεύονται σε αυτά. Οι διαστάσεις των θυρίδων (των ραφιών όπου θα αποθηκεύονται κιβώτια) είναι 0,478x1,3x0,54cm. Στις επιλογές του rack ορίζεται ο μέγιστος αριθμός στοιχείων που θα αποθηκεύονται σε αυτά. Επίσης γίνεται η επιλογή το στοιχείο να αποθηκεύεται στο πρώτο διαθέσιμο κελί κατά στήλη και κατά επίπεδο, και ο μέγιστος αριθμός αντικειμένου ανά κελί να είναι 1.
- ➔ Στις παραμέτρους - triggers του ραφιού επιλέγουμε τα εξής:

A. *On Entry*: Κατά την είσοδο στο ράφι κάθε στοιχείο παίρνει μία ετικέτα με τα εξής στοιχεία:

- ☞ *Rack*: σε ποιο ράφι τοποθετήθηκε το στοιχείο
- ☞ *Bay*: σε ποια στήλη ανήκει το κελί του στοιχείου
- ☞ *Level*: σε ποιο επίπεδο ανήκει το κελί του στοιχείου
- ☞ *Entry time*: ο χρόνος προσομοίωσης στον οποίο τοποθετήθηκε το στοιχείο στο ράφι

Επίσης, κατά την είσοδο μίας παλέτας στο ράφι, γίνεται εγγραφή στον αντίστοιχο πίνακα (RacksA, RacksB, RacksC, Racks4) με τα εξής στοιχεία:

- ☞ *Κωδικός θέσης του στοιχείου*
- ☞ *Όνομα στοιχείου (της μορφής "Item\_")*
- ☞ *Κωδικός προϊόντος*
- ☞ *Ράφι*
- ☞ *Στήλη*
- ☞ *Επίπεδο*
- ☞ *Χρόνος εισαγωγής στο ράφι*

Έτσι είναι δυνατό να γνωρίζουμε σε ποιο κελί βρίσκεται η κάθε παλέτα σε κάθε χρονική στιγμή. Με τη βοήθεια των κατάλληλων εντολών πρώτα ορίζονται οι μεταβλητές και στη συνέχεια αποδίδονται στην ετικέτα. Εκτελείται ο ακόλουθος κώδικας:

```

treenode item = parnode(1);
treenode current = ownerobject(c);
int port = parval(2);

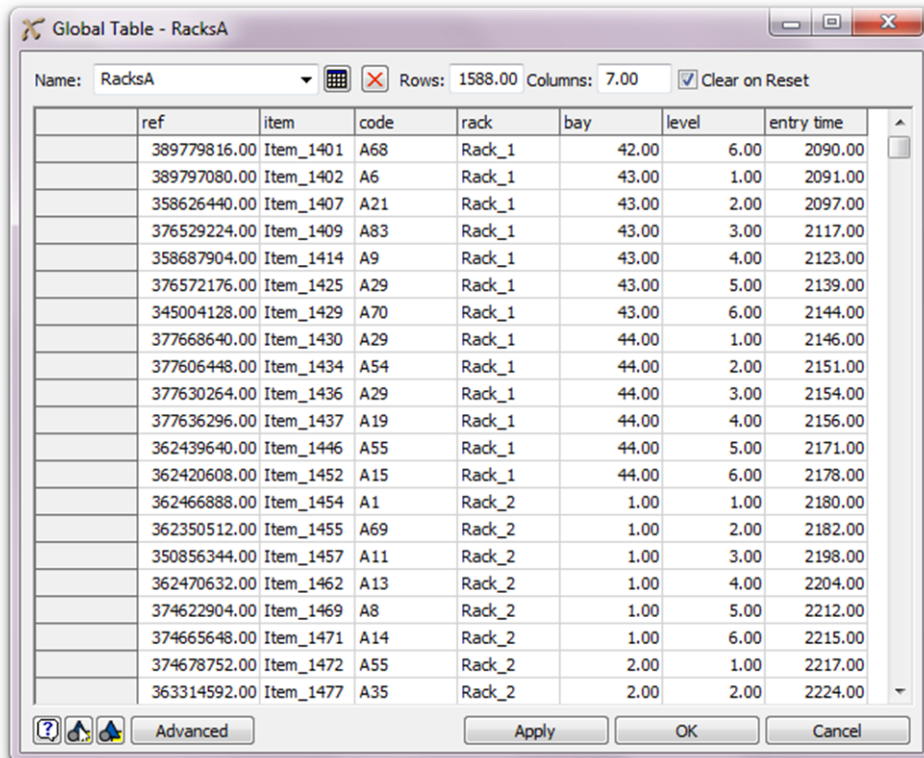
int entry_time=getentrytime(item);
double ref=tonum(item);
int bay=rackgetbayofitem(current,item);
int level=rackgetlevelofitem(current,item);
int rows=gettablerows("RacksA");

for(int x=1; x<=rows; x++)
{
if(gettablenum("RacksA",x,1)==0)
{
settablenum("RacksA",x,1,-1);
addlabel(item,"rack",getnodename(current));
addlabel(item,"bay",bay);
addlabel(item,"level",level);
addlabel(item,"entry time",entry_time);

settablenum("RacksA",x,1,ref);
settablestr("RacksA",x,2,getlabelstr(item,"code"));
settablestr("RacksA",x,3,getlabelstr(item,"rack"));
settablenum("RacksA",x,4,bay);
settablenum("RacksA",x,5,level);
settablenum("RacksA",x,6,entry_time);

return(x);
}
}
return(-1);

```



Global Table - RacksA

Name: RacksA Rows: 1588.00 Columns: 7.00  Clear on Reset

ref	item	code	rack	bay	level	entry time
389779816.00	Item_1401	A68	Rack_1	42.00	6.00	2090.00
389797080.00	Item_1402	A6	Rack_1	43.00	1.00	2091.00
358626440.00	Item_1407	A21	Rack_1	43.00	2.00	2097.00
376529224.00	Item_1409	A83	Rack_1	43.00	3.00	2117.00
358687904.00	Item_1414	A9	Rack_1	43.00	4.00	2123.00
376572176.00	Item_1425	A29	Rack_1	43.00	5.00	2139.00
345004128.00	Item_1429	A70	Rack_1	43.00	6.00	2144.00
377668640.00	Item_1430	A29	Rack_1	44.00	1.00	2146.00
377606448.00	Item_1434	A54	Rack_1	44.00	2.00	2151.00
377630264.00	Item_1436	A29	Rack_1	44.00	3.00	2154.00
377636296.00	Item_1437	A19	Rack_1	44.00	4.00	2156.00
362439640.00	Item_1446	A55	Rack_1	44.00	5.00	2171.00
362420608.00	Item_1452	A15	Rack_1	44.00	6.00	2178.00
362466888.00	Item_1454	A1	Rack_2	1.00	1.00	2180.00
362350512.00	Item_1455	A69	Rack_2	1.00	2.00	2182.00
350856344.00	Item_1457	A11	Rack_2	1.00	3.00	2198.00
362470632.00	Item_1462	A13	Rack_2	1.00	4.00	2204.00
374622904.00	Item_1469	A8	Rack_2	1.00	5.00	2212.00
374665648.00	Item_1471	A14	Rack_2	1.00	6.00	2215.00
374678752.00	Item_1472	A55	Rack_2	2.00	1.00	2217.00
363314592.00	Item_1477	A35	Rack_2	2.00	2.00	2224.00

Advanced Apply OK Cancel

Σχήμα 4.4.1

B. *On Exit*: Κατά την έξοδο του από το ράφι κάθε στοιχείο συμπληρώνει στην ετικέτα του τον χρόνο εξόδου – Exit time από το ράφι. Επίσης, διαγράφεται η εγγραφή από τον αντίστοιχο πίνακα (RacksA, RacksB, RacksC, Racks4). Εκτελείται ο ακόλουθος κώδικας:

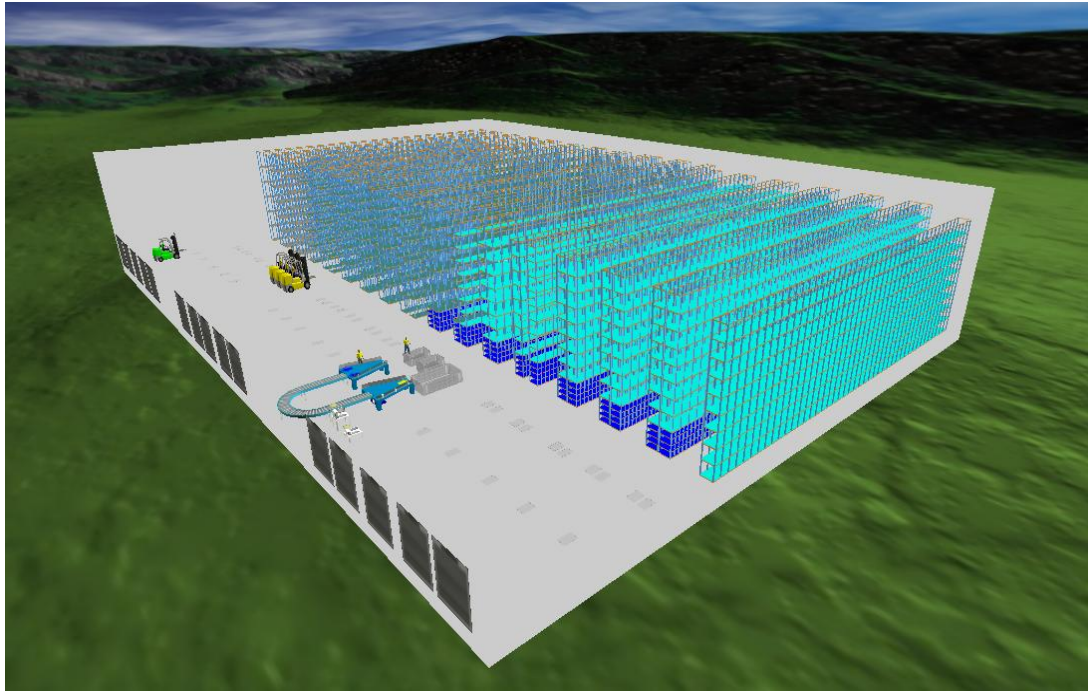
```
treenode item = parnode(1);
treenode current = ownerobject(c);
int port = parval(2);
double ref=tonum(item);

int exit_time=time();

addlabel(item,"exit time",exit_time);
deletetablerow("RacksA",searchdata_n(reftable("RacksA"),ref));
addtablerow("RacksA");
```

- ➔ Συνδέεται το κάθε ράφι με την αντίστοιχη ουρά αναμονής όπου θα αποστέλλονται τα στοιχεία, δηλαδή είτε την ουρά αναμονής όπου θα αποστέλλονται παλέτες για να διασπαστούν σε κιβώτια είτε την ουρά αναμονής (που αντιστοιχεί σε αυτό τον τύπο στοιχείου) στο τμήμα αποστολών.
- ➔ Συνδέονται τα ράφια με το διεκπεραιωτή – dispatcher, ώστε αυτός να ρυθμίζει ποιο περονοφόρο πρέπει να σταλεί για τη μεταφορά κάποιου στοιχείου από το ράφι.
- ➔ Στις επιλογές ροής αντικειμένων επιλέγεται τα προϊόντα να αποστέλλονται στις θύρες εξόδου – output ports με κυκλικό τρόπο. Επίσης ενεργοποιείται η επιλογή να καλείται περονοφόρο για την έξοδο των στοιχείων από το ράφι.
- ➔ Αντίστοιχα, συνδέονται οι θυρίδες με την ταινία μεταφοράς πριν τον επεξεργαστή σύνθεσης παλετών. Ενεργοποιείται η επιλογή να καλούνται μεταφορείς για τη μεταφορά των στοιχείων στην ταινία μεταφοράς και οι θυρίδες συνδέονται με το διεκπεραιωτή που συντονίζει τους μεταφορείς.

Στο σχήμα που ακολουθεί φαίνεται η διάταξη της αποθήκης. Με γκρι χρώμα είναι τα ράφια που αποθηκεύονται οι παλέτες ύψους 195 cm, με πράσινο οι θέσεις για τις παλέτες ύψους 170 cm, με γαλάζιο οι θέσεις για παλέτες ύψους 105 cm και με μπλε οι θέσεις των θυρίδων. Μπροστά από τα ράφια φαίνεται ο χώρος παραλαβοαποστολών και οι ράμπες για την εκφόρτωση και τη φόρτωση των φορητών. Στο αριστερό μέρος της αποθήκης φαίνεται ο χώρος παραλαβών που διαθέτει 10 ράμπες φόρτωσης, και δεξιά φαίνεται ο χώρος αποστολών που διαθέτει 5 ράμπες φόρτωσης.



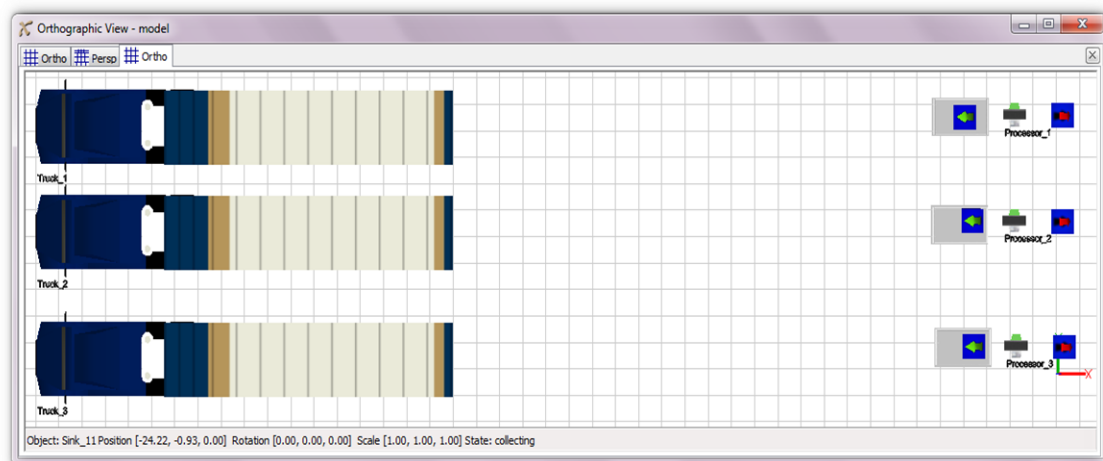
Σχήμα 4.4.4.2

#### 4.4.5 Επεξεργαστές - Processors

Στο μοντέλο έχουν τοποθετηθεί επεξεργαστές στο τμήμα παραλαβών, αποστολών και στο τμήμα σύνθεσης παραγγελιών.

##### Επεξεργαστές στο τμήμα παραλαβών

Όπως έχει αναφερθεί παραπάνω, η χρήση αυτών των επεξεργαστών δεν έχει πρακτική σημασία για το μοντέλο. Επεξεργάζονται ένα αρχικό αντικείμενο που δημιουργεί η πηγή Source\_100 για να προκαλέσει την κίνηση των φορτηγών.



Σχήμα 4.4.5.1

Όταν το αντικείμενο εισαχθεί στους επεξεργαστές, αυτοί στέλνουν κάποιο μήνυμα στην πηγή που δημιουργεί παλέτες σύμφωνα με τον ακόλουθο κώδικα:

```

treenode item = parnode(1);
treenode current = ownerobject(c);
int port = parval(2);
int NoDelay = -1;
double delaytime = 0;
treenode toobject = outobject(current,2);
treenode fromobject = current;
double param1 = 0;
double param2 = 0;
double param3 = 0;
int condition = true;

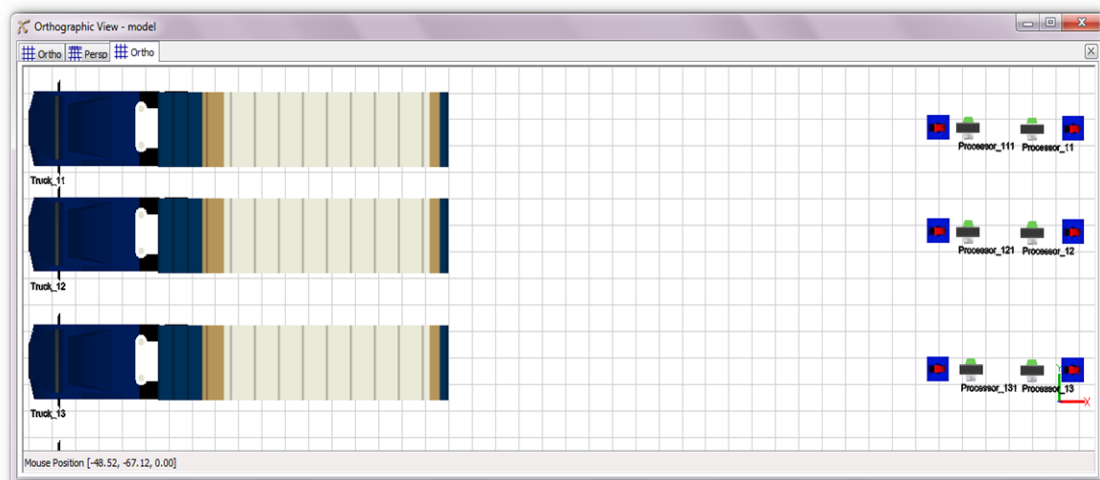
if(condition)
{
    if(delaytime == NoDelay)
        sendmessage(toobject,fromobject,param1,param2,param3);
    else senddelayedmessage(toobject, max(0,delaytime),
        fromobject,param1,param2,param3);
}

```

### Επεξεργαστές στο τμήμα αποστολών

Σε αυτό το τμήμα υπάρχουν δύο είδη επεξεργαστών.

- Οι πρώτοι εκτελούν ακριβώς τις ίδιες λειτουργίες με τους επεξεργαστές του τμήματος παραλαβών. Η μόνη διαφορά τους είναι ότι όταν εισαχθεί κάποιο αντικείμενο σε αυτούς, στέλνουν μήνυμα στους δεύτερους επεξεργαστές, οι οποίοι βρίσκονται μετά τις ουρές αναμονής παλετών, για την ενεργοποίησή τους.



Σχήμα 4.4.5.2

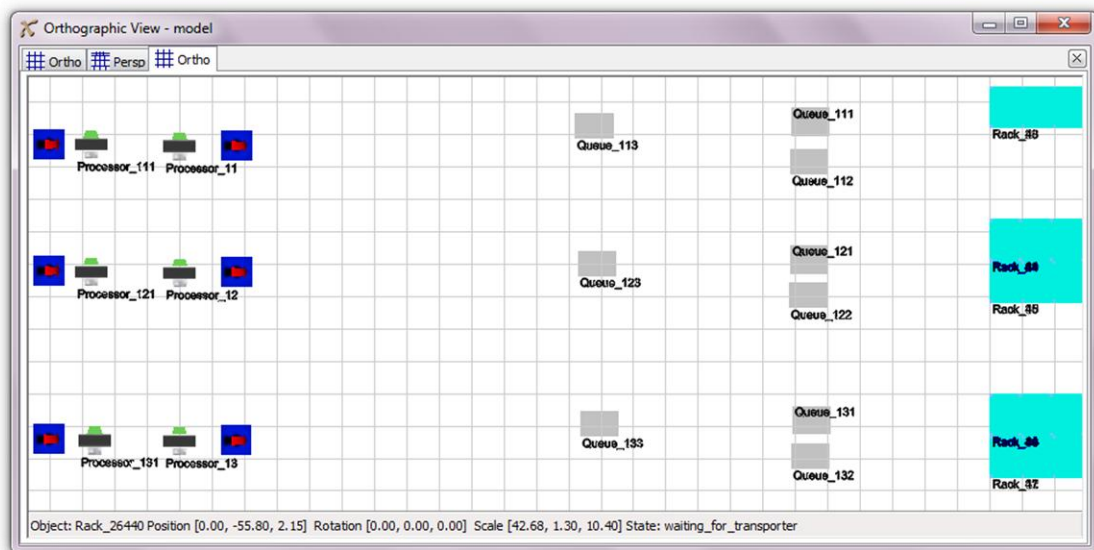
```

treenode item = parnode(1);
treenode current = ownerobject(c);
int port = parval(2);
int NoDelay = -1;
double delaytime = 0;
treenode toobject = centerobject(current,1);
treenode fromobject = current;
double param1 = 0;
double param2 = 0;
double param3 = 0;
int condition = true;

if(condition)
{
    if(delaytime == NoDelay)
        sendmessage(toobject,fromobject,param1,param2,param3);
    else senddelayedmessage(toobject, max(0,delaytime),
        fromobject,param1,param2,param3);
}

```

- ➔ Οι δεύτεροι επεξεργαστές αρχικά είναι ανενεργοί. Ενεργοποιούνται κάθε φορά που φτάνει στη ράμπα κάποιο φορτηγό και λαμβάνουν μήνυμα από τους πρώτους επεξεργαστές. Έτσι αρχίζουν να λαμβάνουν παλέτες από τις ουρές αναμονής και να τις επεξεργάζονται. Αυτή η διαδικασία μοντελοποιεί τη διαδικασία φόρτωσης των φορτηγών. Έχει οριστεί ότι μετά την επεξεργασία 11 παλετών ο επεξεργαστής σταματά την επεξεργασία στοιχείων και ενεργοποιείται ξανά με την εκ νέου άφιξη φορτηγού. Αυτά γίνονται με τη βοήθεια του ακόλουθου κώδικα:



Σχήμα 4.4.5.3



```
treenode item = parnode(1);
treenode current = ownerobject(c);
int port = parval(2);

int inval=getinput(current);
switch(inval)
{
    case 11:
    {
        closeinput(current);
        break;
    }
    case 22:
    {
        closeinput(current);
        break;
    }
    .
    .
    .
    case 209:
    {
        closeinput(current);
        break;
    }
    case 220:
    {
        closeinput(current);
        break;
    }
    default:
    {
        openinput(current);
        break;
    }
}
```

### Επεξεργαστές στο τμήμα σύνθεσης παραγγελιών

Σε αυτό το τμήμα υπάρχουν ο επεξεργαστής διαχωρισμού παλετών και ο επεξεργαστής σύνθεσης παλετών.

### Επεξεργαστής διαχωρισμού παλετών – Separator

Αυτός ο επεξεργαστής βρίσκεται στο τμήμα σύνθεσης παραγγελιών. Πριν από αυτόν υπάρχει μία ουρά αναμονής στην οποία μεταφέρονται παλέτες τύπου Β και C. Η λειτουργία του επεξεργαστή αρχίζει μετά τα πρώτα 2400 λεπτά που είναι ο χρόνος προθέρμανσης του μοντέλου. Ο χρόνος επεξεργασίας των παλετών είναι 1 λεπτό.

- ➔ Όταν μία παλέτα τύπου Β εισαχθεί στο διαχωριστή διασπάται σε 12 κιβώτια, ενώ η παλέτα τύπου C διασπάται σε 8 κιβώτια.

- ➔ Κατά την είσοδό τους στην ουρά αναμονής δίνεται στα κιβώτια ο τύπος – itemtype 4, ο οποίος αποθηκεύεται και στην ετικέτα των στοιχείων, σύμφωνα με τον ακόλουθο κώδικα:

```
treenode item = parnode(1);
treenode current = ownerobject(c);
int port = parval(2);

setitemtype(item,4);
setsize(item,0.4,0.6,0.45);
```

- ➔ Στη συνέχεια συμπληρώνεται στην ετικέτα τους ένα πεδίο με τον κωδικό τους – code, και άλλο ένα με το χρόνο δημιουργίας τους.

Η διαδικασία εκτελείται σύμφωνα με τον ακόλουθο κώδικα που έχει εισαχθεί στις επιλογές διαχωρισμού των αντικειμένων:

```
treenode item = parnode(1);
treenode current = ownerobject(c);
int rownumber = parval(2);
setname(item,concat("Item_", numtostring(getoutput(current)+ 1,0,0)));
int type = getitemtype(item);

switch(type)
{
    case 2:
    {
        double value=12;
        return value;

        addlabel(item,"code",gettablestr("Codes",uniform(86,265,1),1));
        break;
    }

    case 3:
    {
        double value=8;
        return value;

        addlabel(item,"code",gettablestr("Codes",uniform(266,310,1),1));
        break;
    }
}
int creation_time=getcreationtime(item);
addlabel(item,"creation time",creation_time);
```

- ➔ Στη συνέχεια συνδέεται ο διαχωριστής με τον επεξεργαστή σύνθεσης παλετών καθώς και με την ουρά αναμονής κιβωτίων. Γίνεται τυχαία επιλογή των κιβωτίων που θα συνθέσουν μία παραγγελία ενώ τα υπόλοιπα αποθηκεύονται

στις θυρίδες. Τα κιβώτια που δεν χρησιμοποιούνται για κάποια παραγγελία μεταφέρονται από μεταφορέα στην ουρά αναμονής.

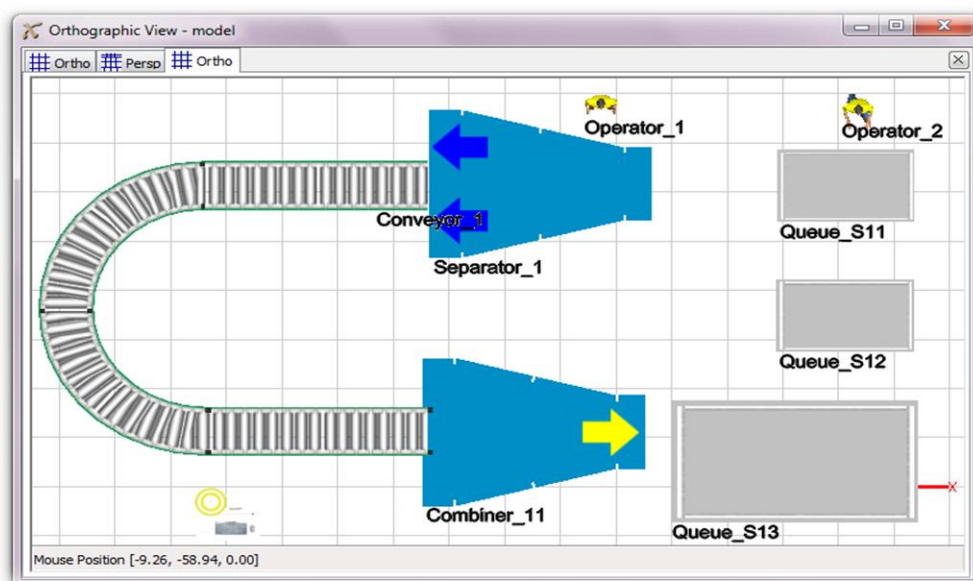
- ➔ Κατά την έξοδο των κιβωτίων από το διαχωριστή ορίζονται οι διαστάσεις τους και δίνεται σαν τύπος - itemtype το 4, σύμφωνα με τον ακόλουθο κώδικα:

```
treenode item = parnode(1);  
treenode current = ownerobject(c);  
int port = parval(2);  
  
setitemtype(item,4);  
setsize(item,0.4,0.6,0.45);
```

### Επεξεργαστής σύνθεσης παλετών – Combiner

Αυτός ο επεξεργαστής βρίσκεται μετά το διαχωριστή και η εργασία που εκτελεί είναι η σύνθεση παλετών.

- ➔ Τα κιβώτια που έχουν ζητηθεί για τη σύνθεση κάποιας παραγγελίας μεταφέρονται μέσω μεταφορικής ταινίας σε αυτό τον επεξεργαστή.
- ➔ Όταν συγκεντρωθούν 12 κιβώτια συντίθεται μία παλέτα διαστάσεων 80x120x170.
- ➔ Τέλος, συνδέεται ο επεξεργαστής με την ουρά αναμονής όπου μεταφέρονται οι ανασυντιθέμενες παλέτες.



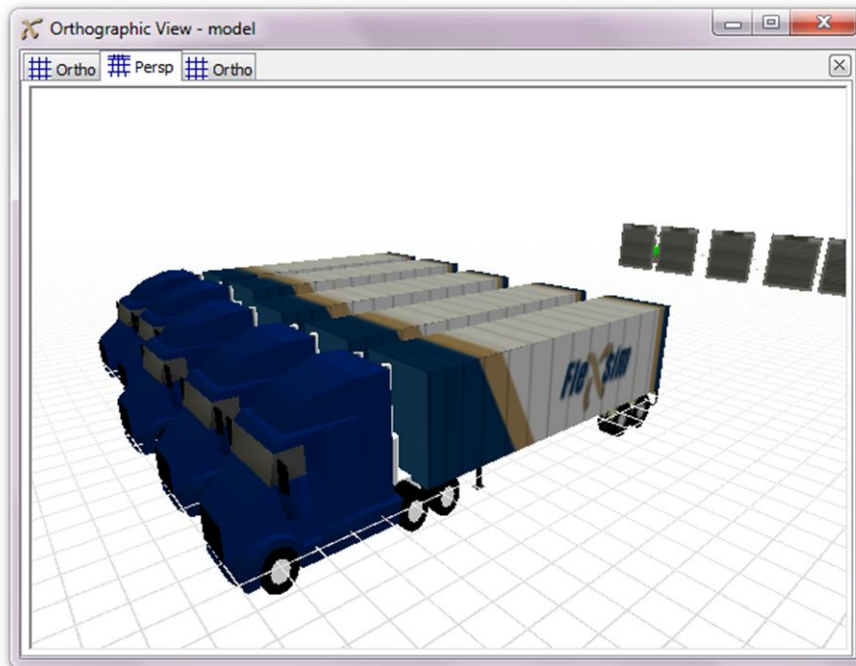
Σχήμα 4.4.5.4

#### 4.4.6 Μεταφορείς - Transporters

Στο μοντέλο υπάρχουν μεταφορείς υπεύθυνοι για τη διακίνηση των στοιχείων – items σε διάφορα σημεία. Τα είδη των μεταφορέων είναι τα ακόλουθα:

##### Φορηγά

Υπάρχουν 10 φορηγά που τροφοδοτούν το τμήμα παραλαβών και 5 που εξυπηρετούν το τμήμα αποστολών. Όπως αναφέρθηκε παραπάνω, τα φορηγά αυτά είναι συνδεδεμένα με πηγές (10 φορηγά στη μία πηγή και 5 στην άλλη). Όταν οι πηγές δημιουργήσουν ένα στοιχείο τα φορηγά το μεταφέρουν σε ένα επεξεργαστή. Αυτή η διαδικασία δεν έχει πρακτική σημασία για το μοντέλο, απλά είναι μία λύση για να επιτευχθεί η κίνηση των φορηγών προς τις ράμπες.



Σχήμα 4.4.6.1

Έχει υπολογιστεί ότι η εκφόρτωση των φορηγών ολοκληρώνεται σε 14,8 λεπτά. Το φορηγό παραμένει λοιπόν στη ράμπα για διάστημα 20 λεπτών σύμφωνα με τον ακόλουθο κώδικα:

```
treenode item = parnode(1);
treenode current = ownerobject(c);
treenode station = parnode(2);
treenode ts = createemptytasksequence(current, 0, 0);

inserttask(ts, TASKTYPE_DELAY, NULL, NULL, 20);
```

Στο τέλος της εκφόρτωσης, όταν το φορηγό είναι ξανά διαθέσιμο, επιστρέφει στην αρχική του θέση σύμφωνα με τον ακόλουθο κώδικα:

```
treenode current = ownerobject(c);
int port = parval(1);
treenode resource = parnode(2);
treenode nextts = parnode(3);
treenode lastts = parnode(4);
treenode dest = centerobject(current, 1);
int condition = content(gettasksequencequeue(resource))==0;
if(condition)
{
    if(objectexists(getnavigator(resource)))
    {
        if(distancetotravel(resource, dest)>
            xsize(resource)+max(xsize(dest),ysize(dest)))
        {
            treenode ts = createemptytasksequence(resource, 0,0);
            inserttask(ts, TASKTYPE_TRAVEL, dest);
            dispatchtasksequence(ts);
            return 1;
        }
    }
}
return 0;
```

Αντίστοιχα, έχει υπολογιστεί ότι η φόρτωση των φορηγών ολοκληρώνεται σε 10,2 λεπτά. Το φορηγό παραμένει λοιπόν στη ράμπα για διάστημα 15 λεπτών σύμφωνα με τον ακόλουθο κώδικα:

```
treenode item = parnode(1);
treenode current = ownerobject(c);
treenode station = parnode(2);
treenode ts = createemptytasksequence(current,0, 0);

inserttask(ts, TASKTYPE_DELAY, NULL, NULL, 15);
```

Στο τέλος της φόρτωσης, όταν το φορηγό είναι ξανά διαθέσιμο, επιστρέφει στην αρχική του θέση με τον ίδιο κώδικα που χρησιμοποιείται από τα φορηγά παραλαβών.

## Περονοφόρα

Για τη διακίνηση των στοιχείων στην αποθήκη χρησιμοποιούνται κυρίως περονοφόρα. Ο αριθμός τους είναι και το ζητούμενο για να πετύχουμε την ομαλή λειτουργία του μοντέλου. Τα περονοφόρα είναι υπεύθυνα για τη μετακίνηση των παλετών από και προς τις ουρές αναμονής της αποθήκης καθώς και για την τοποθέτηση των παλετών στα ράφια. Σαν τύπος περονοφόρου έχει επιλεγθεί το ETX 515 της εταιρίας Jungheinrich.



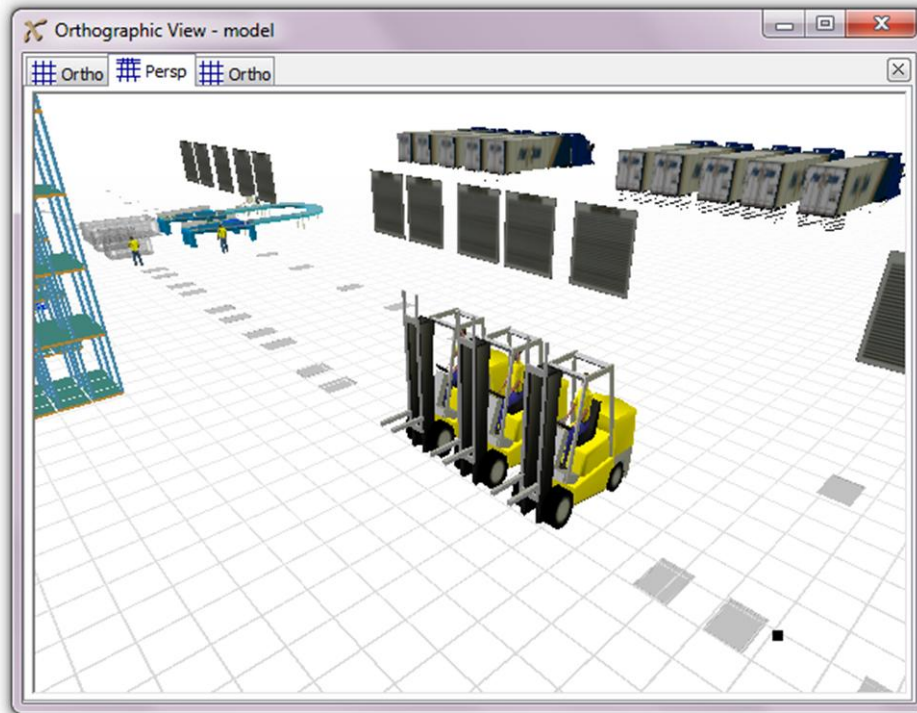
Σχήμα 4.4.6.1

Στις επιλογές του περονοφόρου καταχωρούμε τα εξής στοιχεία:

Δεδομένα απόδοσης	
Ταχύτητα ανύψωσης	27,30 m/min
Χωρητικότητα	1 τεμ.
Μέγιστη ταχύτητα	175 m/min
Χρόνος φόρτωσης	0,25 min
Χρόνος εκφόρτωσης	0,25 min

Πίνακας 4.4.6.1

Επίσης έχει επιλεγθεί να επιτρέπεται η περιστροφή του περονοφόρου κατά την κίνησή του. Περισσότερες τεχνικές λεπτομέρειες του περονοφόρου (διαστάσεις, ταχύτητα κίνησης, ταχύτητα φόρτωσης/εκφόρτωσης κτλ.) φαίνονται στο Παράρτημα.



Σχήμα 4.4.6.2

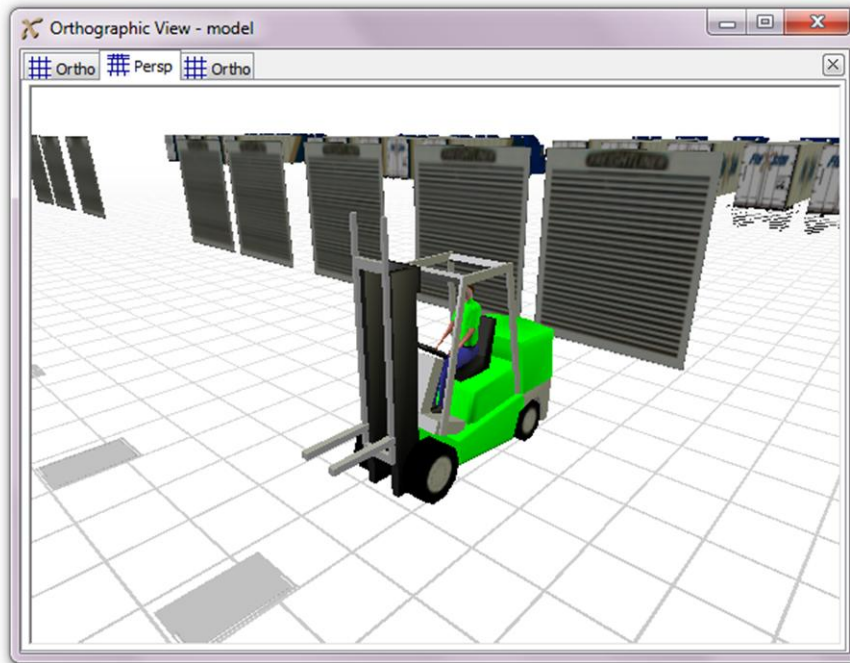
Μετά το τέλος της κάθε διαδρομής τους τα περνοφόρα επιστρέφουν στο χώρο φύλαξης τους μέχρι την επόμενη εργασία σύμφωνα με τον ακόλουθο κώδικα:

```

treenode current = ownerobject(c);
int port = parval(1);
treenode resource = parnode(2);
treenode nextts = parnode(3);
treenode lastts = parnode(4);
treenode dest = node("/NN168",model());
int condition = content(gettasksequencequeue(resource))==0;
if(condition)
{
    if(objectexists(getnavigator(resource)))
    {
        if(distancetotravel(resource, dest) >
            xsize(resource)+max(xsize(dest),ysize(dest)))
        {
            treenode ts = createemptytasksequence(resource, 0,0);
            inserttask(ts, TASKTYPE_TRAVEL, dest);
            dispatchtasksequence(ts);
            return 1;
        }
    }
}
return 0;

```

Εκτός από τα περονοφόρα που μετακινούν παλέτες από και προς τα ράφια, υπάρχει άλλο ένα περονοφόρο το οποίο χρησιμοποιείται αποκλειστικά για την εκφόρτωση και την φόρτωση των φορτηγών.



Σχήμα 4.4.6.3

### Μεταφορείς

Στο μοντέλο υπάρχουν και εργάτες που είναι μεταφορείς. Αυτοί μεταφέρουν χειροκίνητα κιβώτια από τις θυρίδες προς τη μεταφορική ταινία και από την ουρά αναμονής μετά το διαχωριστή - seperator προς τις θυρίδες. Επίσης είναι υπεύθυνοι για το χειρισμό του διαχωριστή - seperator.

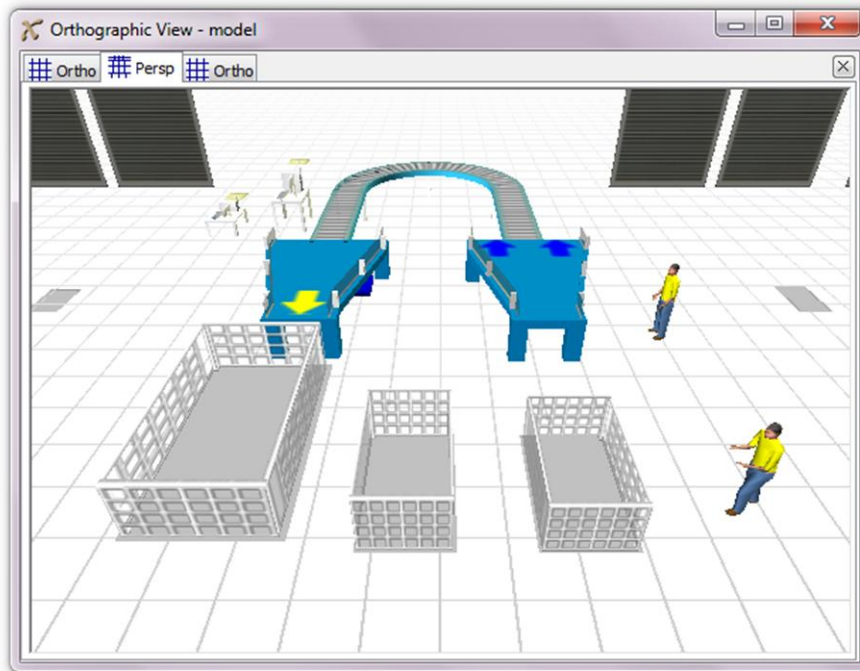
Στις επιλογές των μεταφορέων καταχωρούμε τα εξής στοιχεία:

Δεδομένα απόδοσης	
Χωρητικότητα	1 τεμ.
Μέγιστη ταχύτητα	50 m/min
Χρόνος φόρτωσης	0 min
Χρόνος εκφόρτωσης	0 min

Πίνακας 4.4.6.2



Επίσης έχει επιλεχθεί να επιτρέπεται η περιστροφή των μεταφορέων κατά την κίνησή τους.



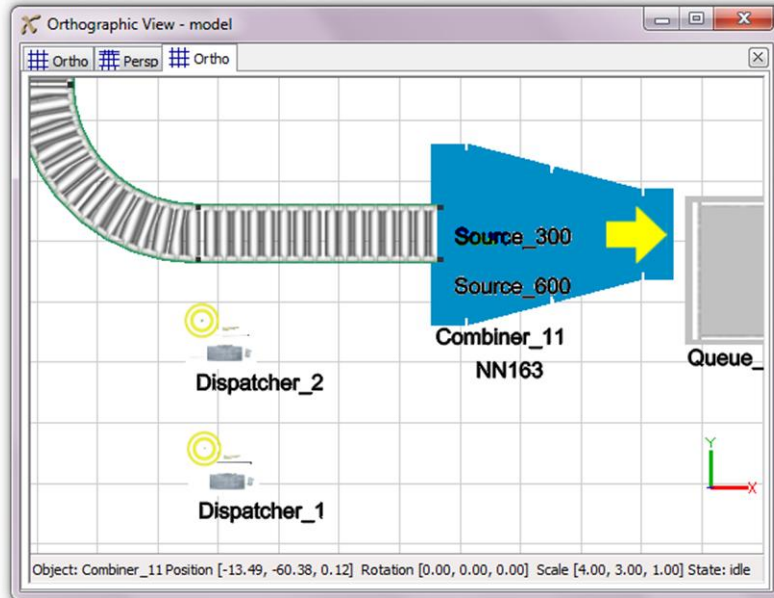
Σχήμα 4.4.6.4

#### **4.4.7 Μεταφορικές ταινίες – Conveyors**

Όπως έχει αναφερθεί και παραπάνω, στο τμήμα σύνθεσης παραγγελιών υπάρχει μεταφορική ταινία. Αυτή συνδέει το διαχωριστή – separator με τον επεξεργαστή σύνθεσης παλετών – combiner. Έχει επιλεχθεί η δυνατότητα να συσσωρεύονται στοιχεία πάνω στη μεταφορική ταινία όταν ο combiner είναι απασχολημένος. (Η μορφή της μεταφορικής ταινίας φαίνεται στο Σχήμα 4.4.6.3)

#### **4.4.8 Διεκπαιρωτές – Dispatchers**

Αυτά τα αντικείμενα είναι υπεύθυνα για τον καταμερισμό εργασιών στους μεταφορείς. Έτσι, στο μοντέλο έχει συνδεθεί ένας διεκπαιρωτής με τα περονοφόρα και ένας με τους εργάτες μεταφορείς. Στις ιδιότητες και των δύο διεκπαιρωτών έχει επιλεχθεί να καλείται το περονοφόρο, ή ο μεταφορέας αντίστοιχα, που είναι ανενεργός και βρίσκεται πιο κοντά στο στοιχείο που χρήζει μεταφοράς και οι εργασίες να ταξινομούνται βάσει του συστήματος FIFO (First In First Out).



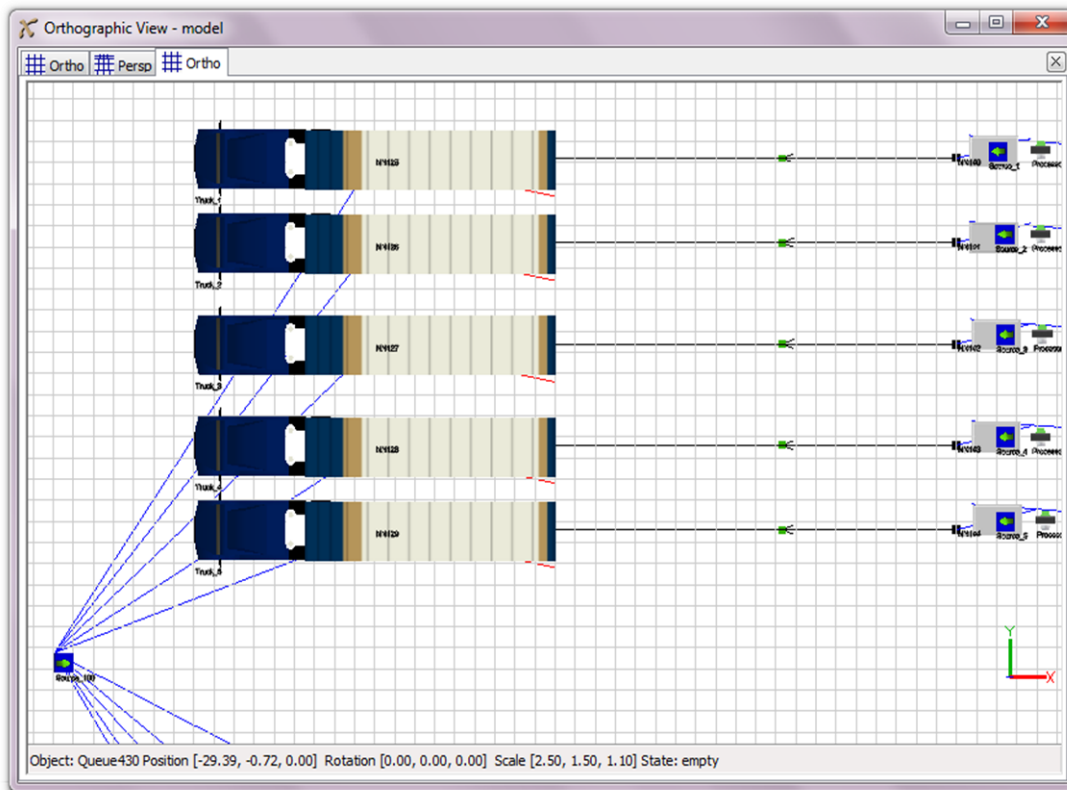
Σχήμα 4.4.8.1

#### 4.4.9 Κόμβοι δικτύου – Network Nodes

Στο μοντέλο έχει γίνει χρήση πολλών τέτοιων κόμβων με σκοπό τη δημιουργία δικτύου στην αποθήκη. Το δίκτυο αυτό ορίζει ποιες είναι οι επιτρεπόμενες διαδρομές και αποτρέπει την άναρχη κίνηση των μεταφορέων στην αποθήκη. Η μέγιστη ταχύτητα που έχει ορισθεί σε όλο το δίκτυο των οχημάτων είναι 175m/min ή 10,5km/h, ενώ στο δίκτυο που χρησιμοποιούν οι εργάτες μεταφορείς είναι 50m/min ή 3km/h. Στις επιλογές όλων των κόμβων έχει ορισθεί ότι επιτρέπεται να κινείται ένας μεταφορέας σε κάθε κόμβο. Αυτό σημαίνει ότι αν θέλει να εισέλθει δεύτερος μεταφορέας πρέπει να περιμένει την έξοδο του πρώτου. Μόνο στους κόμβους που είναι συνδεδεμένοι με τα ράφια επιτρέπεται η είσοδος δύο μεταφορέων τη φορά.

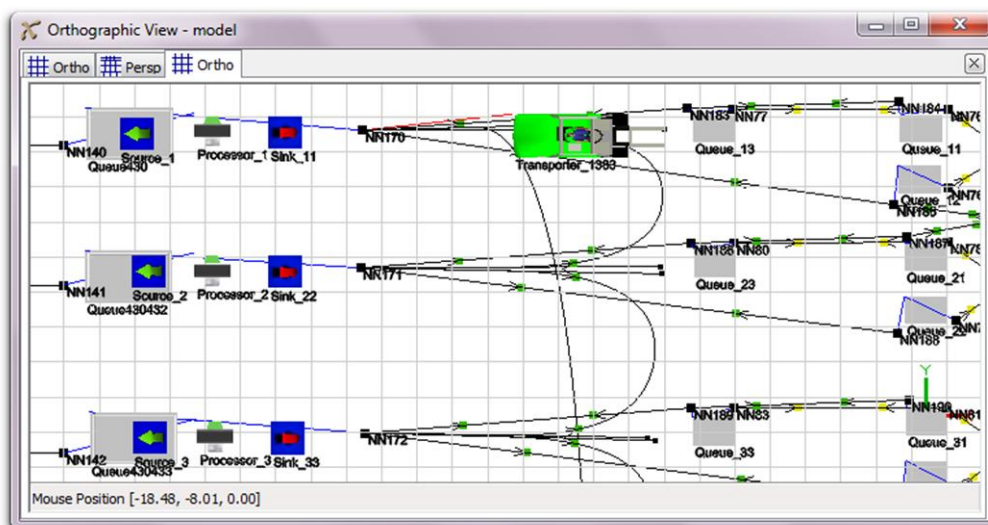
Έχουν χρησιμοποιηθεί κόμβοι οι οποίοι ορίζουν τις εξής μετακινήσεις:

- Κόμβοι που ορίζουν τις μετακινήσεις των φορτηγών από και προς την αποθήκη. Αυτοί είναι συνδεδεμένοι με τις πηγές Source\_100 και Source\_200 που παράγουν τα αρχικά στοιχεία, συνδεδεμένοι με τους επεξεργαστές στους οποίους προορίζονται τα στοιχεία, και με τα φορτηγά. Έτσι ορίζεται η διαδρομή που μπορούν να ακολουθήσουν.



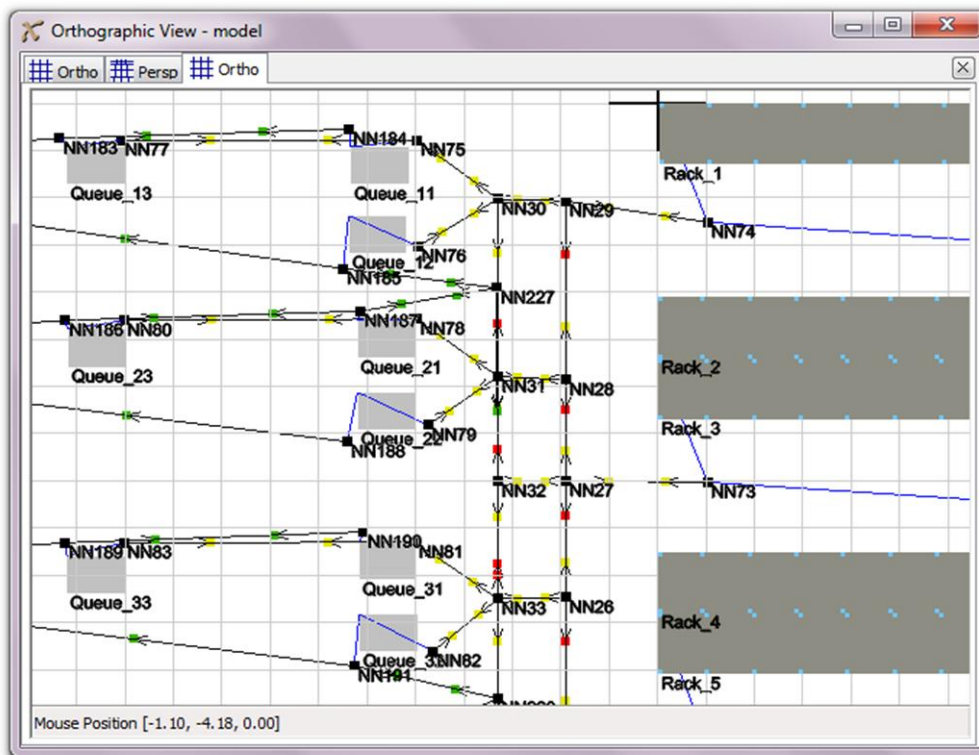
Σχήμα 4.4.9.1

- Κόμβοι που ορίζουν τις μετακινήσεις του περνοφόρου που πραγματοποιεί την εκφόρτωση και τη φόρτωση των φορτηγών. Αυτοί είναι συνδεδεμένοι με τις ουρές αναμονής που βρίσκονται στις ράμπες και τις ουρές αναμονής του χώρου παραλαβών. Αντίστοιχα, στο τμήμα αποστολών είναι συνδεδεμένοι με τις ουρές αναμονής και τους επεξεργαστές όπου καταλήγουν τα στοιχεία. Έτσι ορίζονται οι διαδρομές που μπορεί να ακολουθήσει το περνοφόρο.



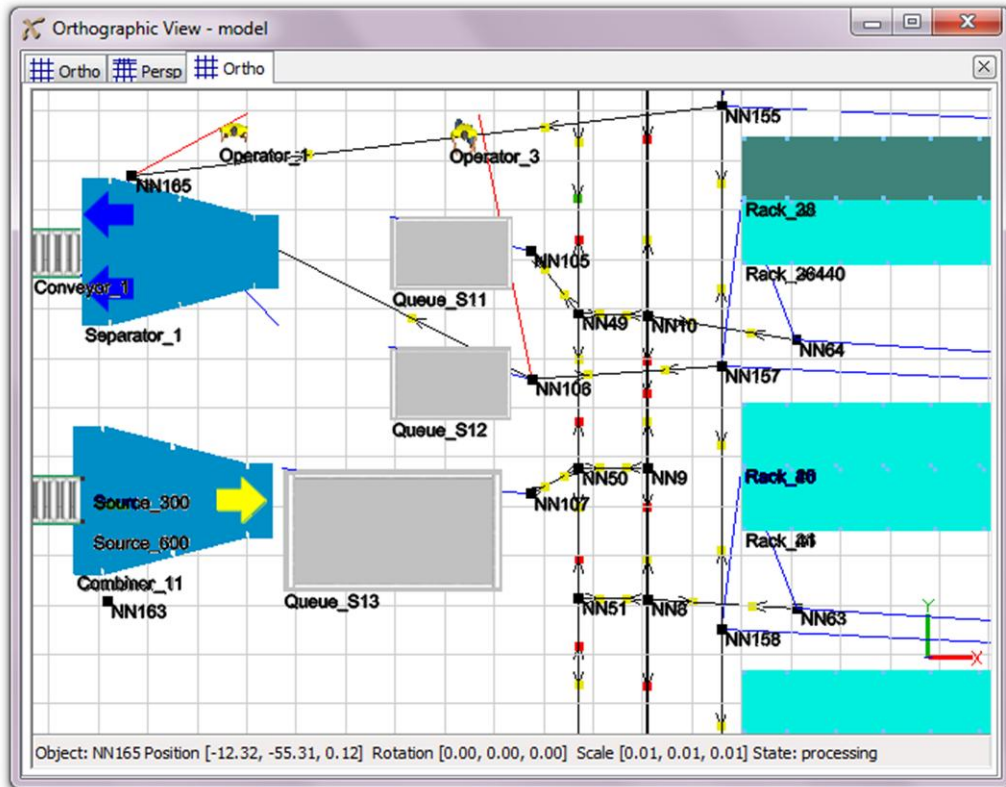
Σχήμα 4.4.9.2

- Κόμβοι που ορίζουν τις μετακινήσεις των περονοφόρων ανάμεσα στις ουρές αναμονής και στα ράφια αποθήκευσης παλετών. Αυτοί είναι συνδεδεμένοι με τις ουρές αναμονής παλετών, τα ράφια αποθήκευσης παλετών και με τα περονοφόρα. Έτσι ορίζονται οι διαδρομές που μπορούν να ακολουθήσουν τα περονοφόρα. Όπως φαίνεται και στο σχήμα, η δεξιά πλευρά του διαδρόμου χρησιμοποιείται μόνο για άνοδο των περονοφόρων, ενώ η αριστερή μόνο για κάθοδο (το κόκκινο χρώμα υποδηλώνει ότι δεν επιτρέπεται η κίνηση προς τη συγκεκριμένη κατεύθυνση, ενώ το κίτρινο ότι δεν επιτρέπονται οι προσπεράσεις). Στα κάθετα τμήματα επιτρέπεται η κίνηση και προς τις δύο κατευθύνσεις.

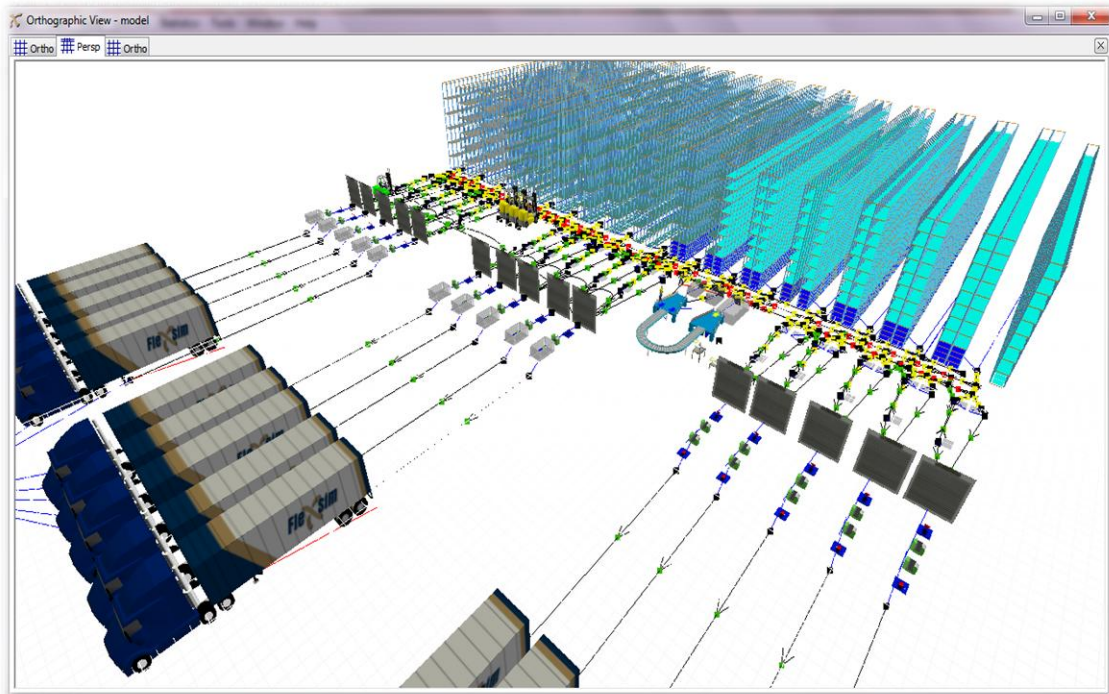


Σχήμα 4.4.9.3

- Κόμβοι που ορίζουν τις μετακινήσεις των μεταφορέων ανάμεσα στις ουρές αναμονής και στις θυρίδες. Αυτοί είναι συνδεδεμένοι με τις ουρές αναμονής κιβωτίων στο τμήμα σύνθεσης παραγγελιών. Έτσι ορίζονται οι διαδρομές που μπορούν να ακολουθήσουν οι μεταφορείς.



Σχήμα 4.4.9.4

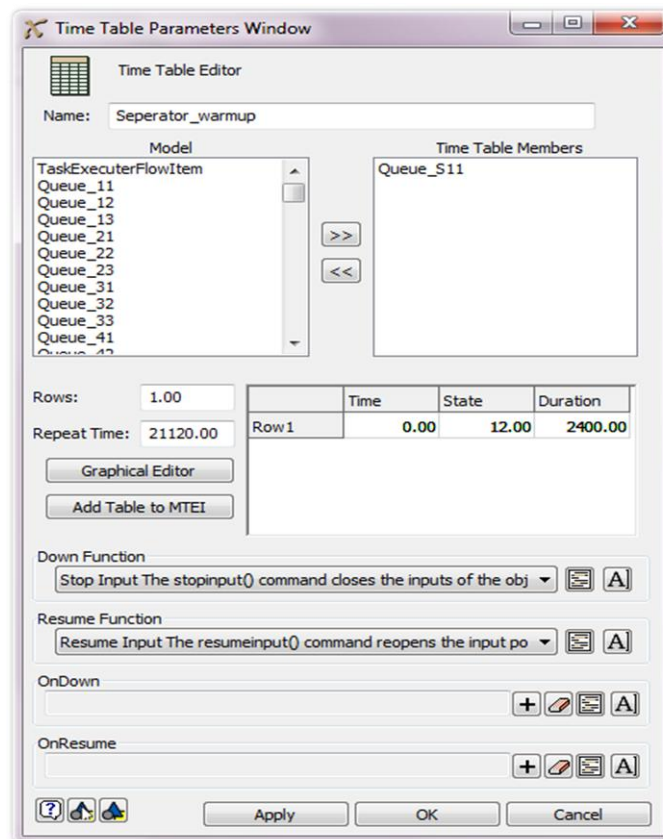


Σχήμα 4.4.9.5

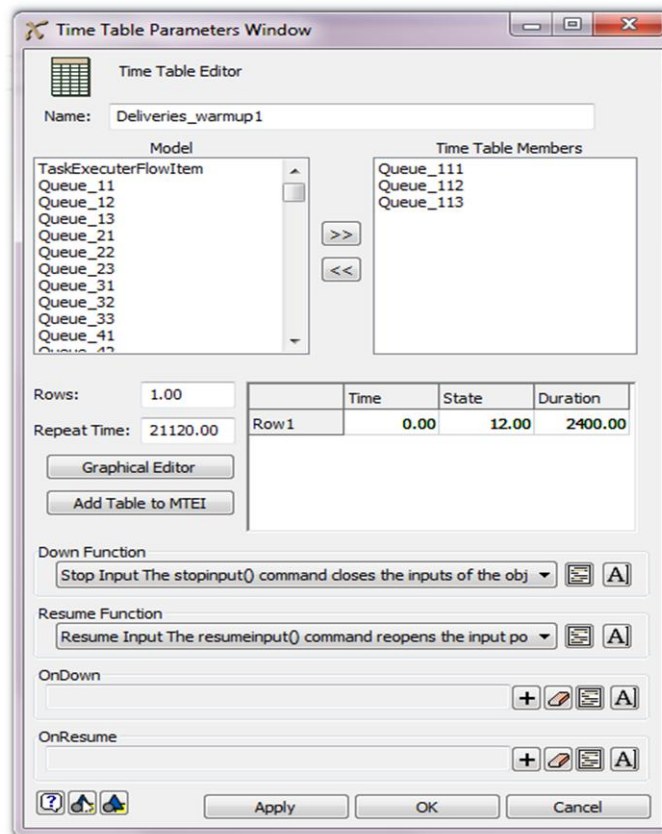
#### 4.4.10 Πίνακες χρονικού προγραμματισμού – Time Tables

Εφόσον έχουν τοποθετηθεί τα αντικείμενα που συνθέτουν το μοντέλο, πρέπει να προγραμματιστούν χρονικά.

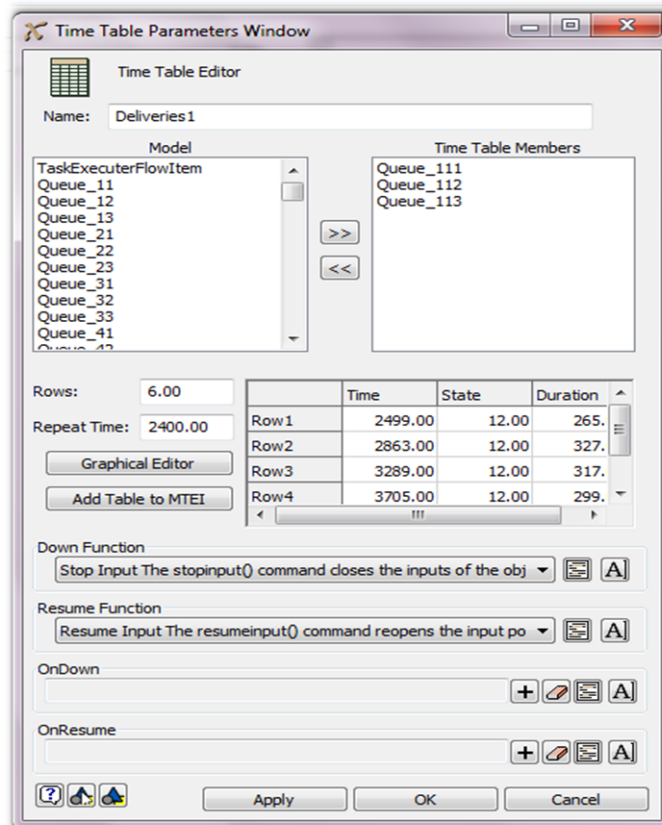
- ➔ Αρχικά πρέπει να προγραμματιστεί η λειτουργία της ουράς αναμονής πριν το διαχωριστή παλετών – separator (Σχήμα 4.4.10.1) και των ουρών αναμονής του τμήματος αποστολών ώστε να μην δέχονται στοιχεία τα πρώτα 2400 λεπτά που είναι ο χρόνος προθέρμανσης (warmup time) του μοντέλου και να αρχίζουν τη λειτουργία τους μετά από αυτό το διάστημα (Σχήμα 4.4.10.2 για την πρώτη ράμπα ενδεικτικά).
- ➔ Οι ουρές αναμονής του τμήματος αποστολών σταματούν να δέχονται παλέτες τη στιγμή που φτάνει το αντίστοιχο φορτηγό για να παραλάβει την παραγγελία. Παρακάτω φαίνεται ο πίνακας χρονικού προγραμματισμού για την ουρά αναμονής της πρώτης ράμπας ενδεικτικά (Σχήμα 4.4.10.3, Σχήμα 4.4.10.4).



Σχήμα 4.4.10.1

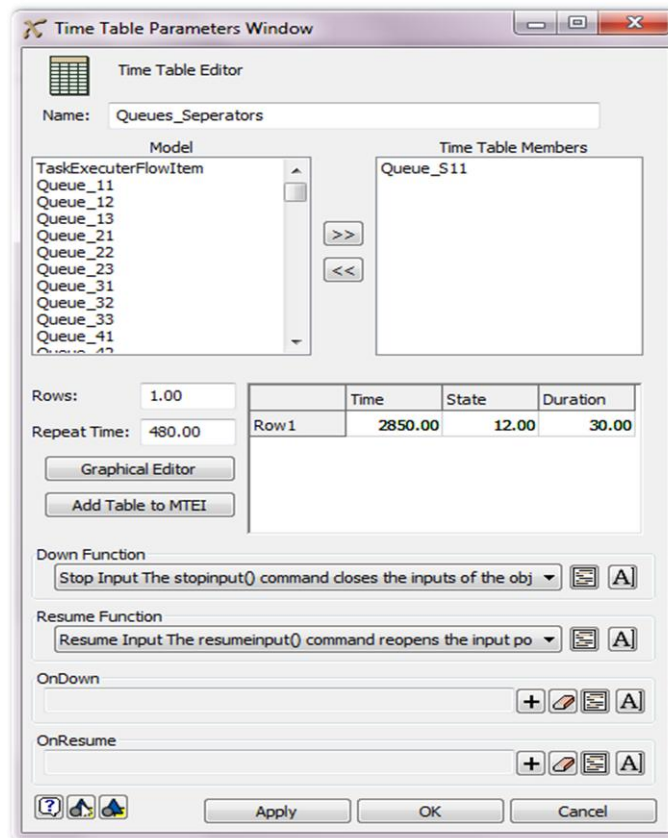


Σχήμα 4.4.10.2



Σχήμα 4.4.10.3

- Η ουρά αναμονής του διαχωριστή σταματά να δέχεται στοιχεία 30 λεπτά πριν το τέλος της ημέρας, εφόσον έχουν συντεθεί όλες οι παραγγελιόμενες παλέτες.



Σχήμα 4.4.10.4

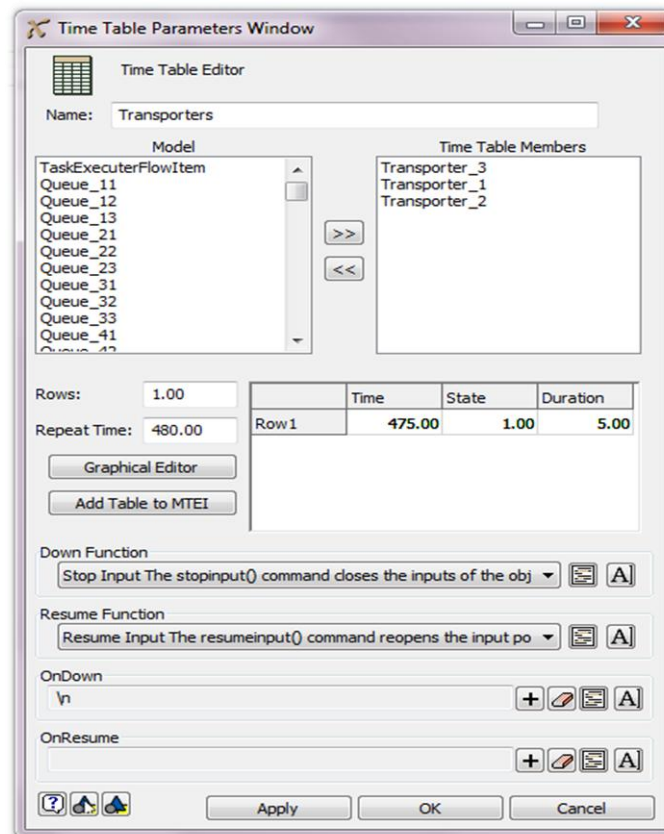
- Τα περνοφόρα, έχει προγραμματιστεί να σταματούν τη λειτουργία τους 5 λεπτά πριν το τέλος της ημέρας. Εκείνη τη χρονική στιγμή αποστέλλεται μήνυμα στα περνοφόρα να μεταβούν στο χώρο φύλαξης τους, εάν δεν βρίσκονται ήδη εκεί, σύμφωνα με τον κώδικα:

```

treenode item = parnode(1);
treenode current = ownerobject(c);
treenode station = parnode(2);
treenode dest = node("/NN168",model());
double priority = 0;
int condition = content(gettasksequencequeue(current)) == 0;
if(condition)
{
    if(objectexists(getnavigator(current)))
    {
        createtraveltask(current,dest,priority,0);
    }
}
setrot(current,0,0,0);

```





Σχήμα 4.4.10.5

#### 4.5 Ρύθμιση παραμέτρων του μοντέλου

Εφόσον έχει στηθεί το μοντέλο της αποθήκης, το επόμενο βήμα είναι να ρυθμιστούν κάποιες παραμέτρους για να ξεκινήσει η προσομοίωση. Για τη ρύθμιση αυτών των παραμέτρων, το FlexSim παρέχει ένα εργαλείο, το Experimenter. Έτσι:

- Ορίζεται η χρονική διάρκεια της προσομοίωσης – Simulation End Time σε 12960 λεπτά και ο χρόνος προθέρμανσης του μοντέλου – Warmup End Time σε 2400 λεπτά. Το μοντέλο δηλαδή μετά το χρόνο προθέρμανσης θα τρέχει για 10560 λεπτά, δηλαδή για ένα μήνα. Μετά από το χρόνο προθέρμανσης τα στατιστικά στοιχεία που έχουν συγκεντρωθεί διαγράφονται ενώ το μοντέλο συνεχίζει να τρέχει. Έτσι, συγκεντρώνονται στατιστικά στοιχεία εκ νέου και είναι αυτά που ουσιαστικά μας ενδιαφέρουν.
- Ακόμα ρυθμίζεται ο αριθμός των σεναρίων – Number of Scenarios σε 3 και ο αριθμός των επαναλήψεων ανά σενάριο – Replications per Scenario σε 20. Επίσης μπορούμε να ρυθμίσουμε πόσες και ποιες παράμετροι θα αλλάζουν σε

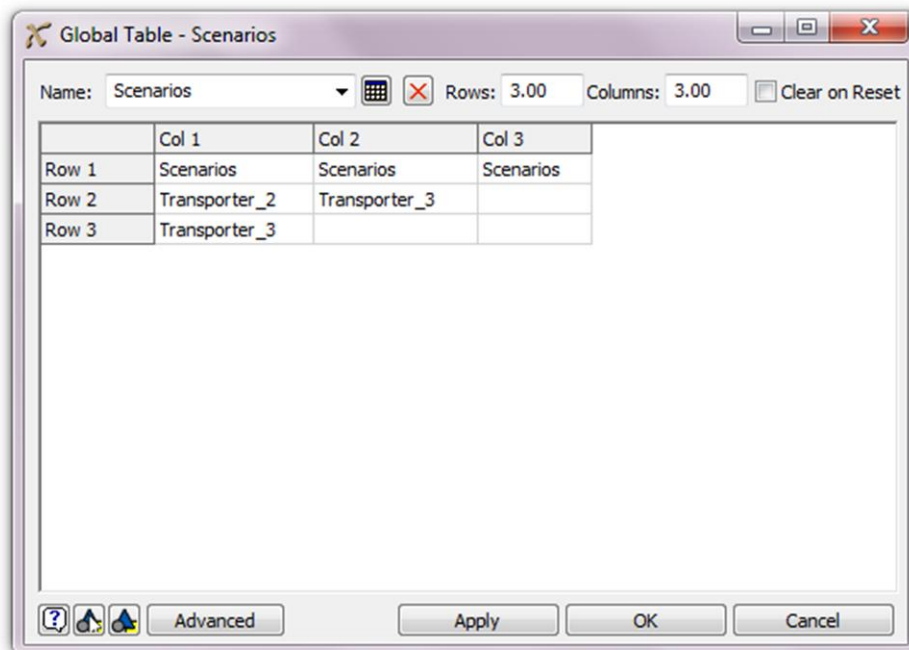
κάθε σενάριο. Στο μοντέλο δεν έχει ορισθεί κάποια παράμετρος. Ο αριθμός των περονοφόρων που χρησιμοποιείται σε κάθε σενάριο ορίζεται με τη βοήθεια ενός πίνακα, ενός πίνακα χρονικού προγραμματισμού και του ακόλουθου κώδικα που εκτελείται στην αρχή κάθε σεναρίου:

```
double replication = parval(1);
double scenario = parval(2);

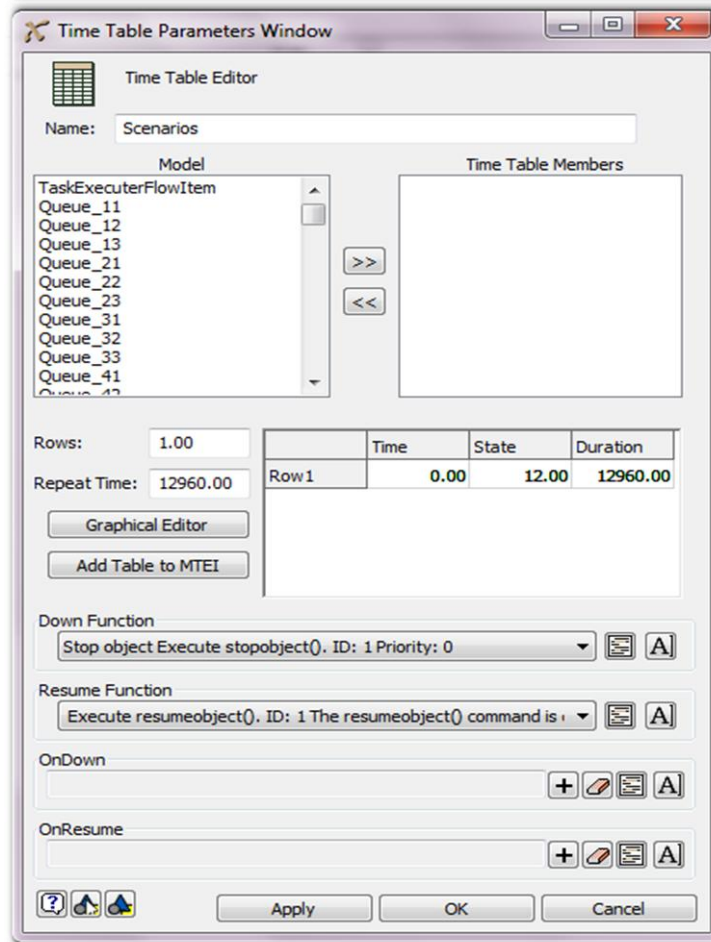
string timetablename = gettablestr("Scenarios",1,scenario);
treenode timetable =
node(apchar(concat("/Tools/TimeTables/",timetablename)),model());

memberremoveall(timetablename,1);
int rownum = 2;

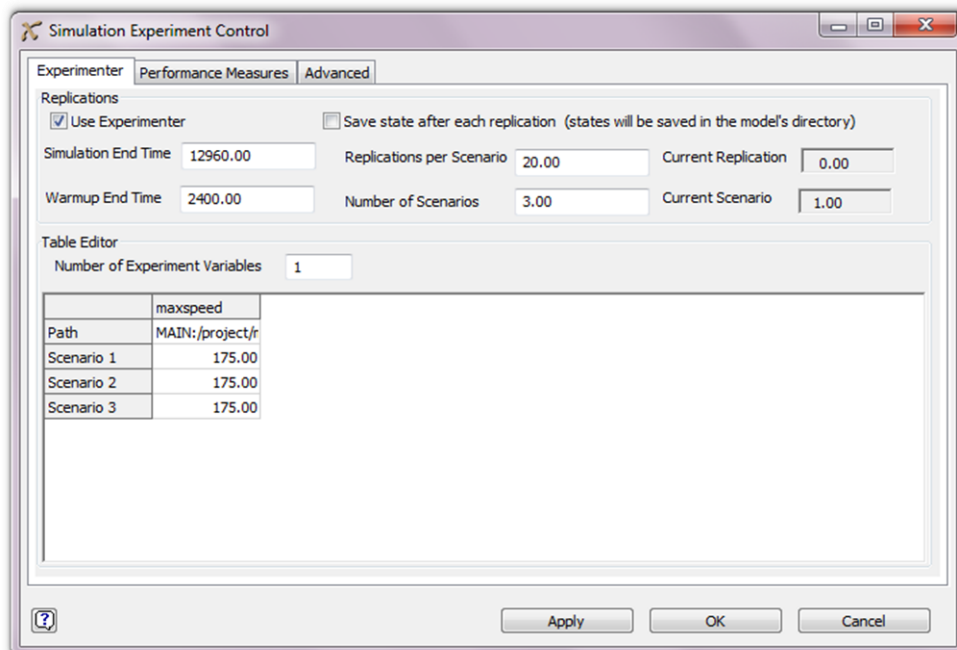
while(gettablestr("Scenarios",rownum,scenario) != NULL && rownum<10000)
{
    treenode member = node
(apchar(concat("/",gettablestr("Scenarios",rownum,scenario))),model());
    if(objectexists(member))
    {
        memberadd(member,timetablename,1);
    }
    rownum++;
}
```



Σχήμα 4.5.1

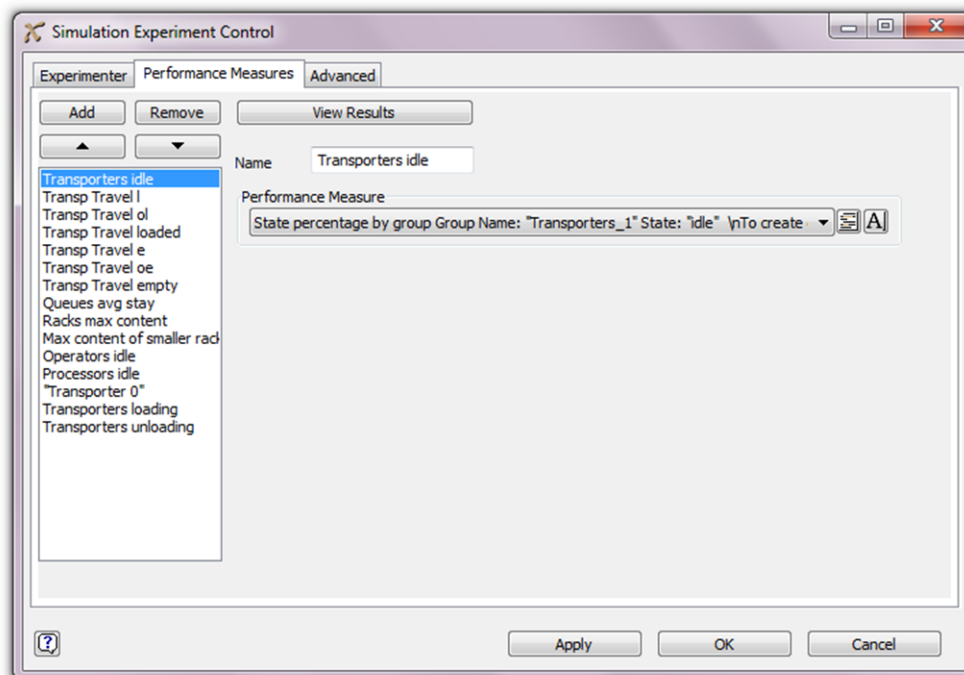


Σχήμα 4.5.2



Σχήμα 4.5.3

- Έπειτα, γίνεται επιλογή των αντικειμένων για τα οποία θα καταγράφονται στατιστικά στοιχεία και των παραμέτρων που πρέπει να εξεταστούν. Στο τέλος της προσομοίωσης προκύπτουν κάποιοι δείκτες απόδοσης του μοντέλου σε μορφή διαγραμμάτων. Έτσι, θα καταγράφονται στατιστικά στοιχεία για τη χρήση των περονοφόρων, την αξιοποίηση των εργατών, το μέσο χρόνο παραμονής στις ουρές αναμονής του τμήματος παραλαβών, τη χρήση των ραφιών και των θυρίδων.



Σχήμα 4.5.4

#### 4.6 Τρέξιμο μοντέλου

Εφόσον έχει στηθεί το μοντέλο της αποθήκης, το επόμενο βήμα είναι να γίνει προσομοίωση της λειτουργίας της τρέχοντας το μοντέλο.

- Αρχικά ενεργοποιείται η καταγραφή στατιστικών στοιχείων σε όλα τα αντικείμενα και η καταγραφή ιστορικών στοιχείων για το μοντέλο. Τα στοιχεία αυτά συλλέγονται κατά τη διάρκεια της προσομοίωσης και χρησιμοποιούνται για τη δημιουργία διαγραμμάτων και αναφορών που βοηθούν στην εξαγωγή συμπερασμάτων για το ποσοστό αξιοποίησης του κάθε πόρου.
- Γίνεται επαναφορά του συστήματος στην αρχική κατάσταση – Reset, και στη συνέχεια τρέχει το μοντέλο – Run. Αν χρειάζεται να σταματήσει το μοντέλο σε

χρόνο διαφορετικό από αυτόν που ορίστηκε στο Experimenter υπάρχει η επιλογή Stop. Η ταχύτητα της προσομοίωσης μπορεί να αλλάξει από τη μπάρα Run Speed.

- ➔ Επίσης, για ευκολότερο χειρισμό έχει δημιουργηθεί μία επιφάνεια ελέγχου - Model Control GUI. Από εκεί μπορεί να τρέξει ή να σταματήσει το μοντέλο και να αλλάξει η ταχύτητα της προσομοίωσης. Επίσης, μπορεί να αλλάξει η οπτική γωνία για καλύτερη εμποπτεία της αποθήκης.



Σχήμα 4.6.1

Παρακάτω φαίνονται κάποιες εικόνες από διαφορετικές οπτικές γωνίες της αποθήκης. Δείχνουν όλες την εικόνα της αποθήκης στα 10.000 λεπτά για κάθε περίπτωση (Scenario 1, Scenario 2 & Scenario 3).

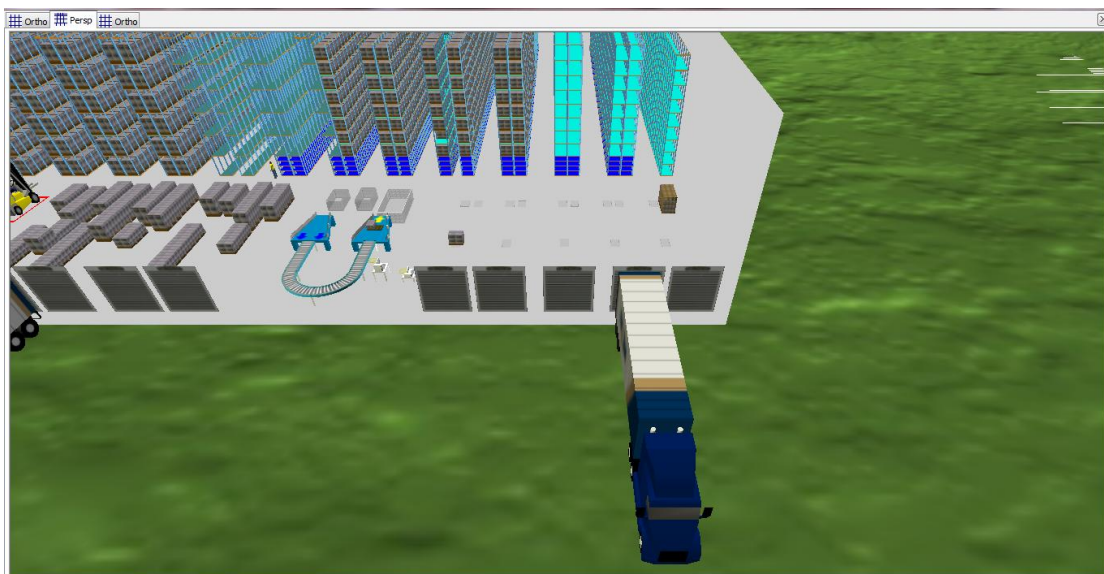
### Scenario 1

Παρακάτω φαίνεται η εικόνα του τμήματος παραλαβών. Είναι εμφανές ότι η συμφόρηση είναι πολύ μεγάλη. Το ένα περνοφόρο είναι ανεπαρκές, δεν προλαβαίνει να τοποθετήσει τις παλέτες στα ράφια και επομένως οι ουρές αναμονής του τμήματος παραλαβών δεν αδειάζουν. Έτσι, το επόμενο φορτηγό που έρχεται στην ίδια ράμπα δεν μπορεί να ξεφορτώσει παλέτες.



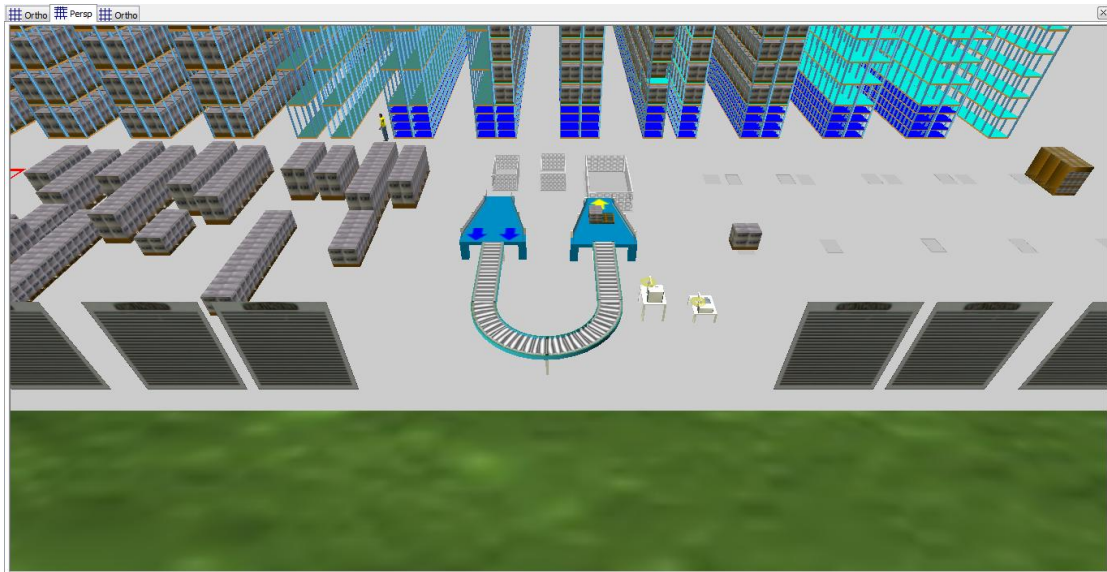
Σχήμα 4.6.2 - Receiving

Το τμήμα αποστολών δεν έχει τις παραγγελίες έτοιμες κατά την άφιξη των φορτηγών εφόσον δίνεται προτεραιότητα στην τακτοποίηση των παραλαμβανόμενων παλετών στα ράφια.

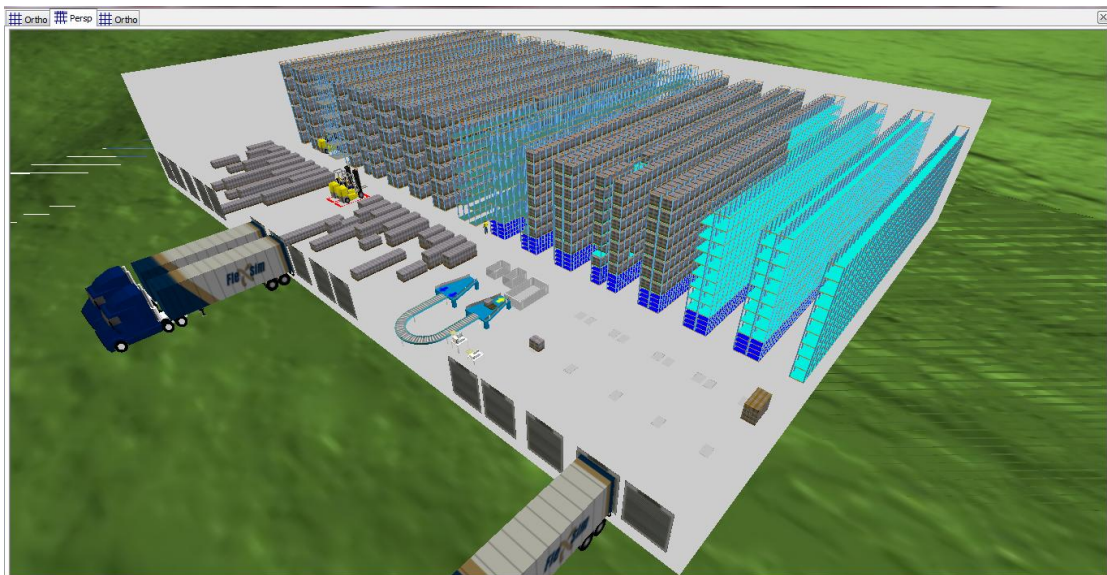


Σχήμα 4.6.3 – Delivery

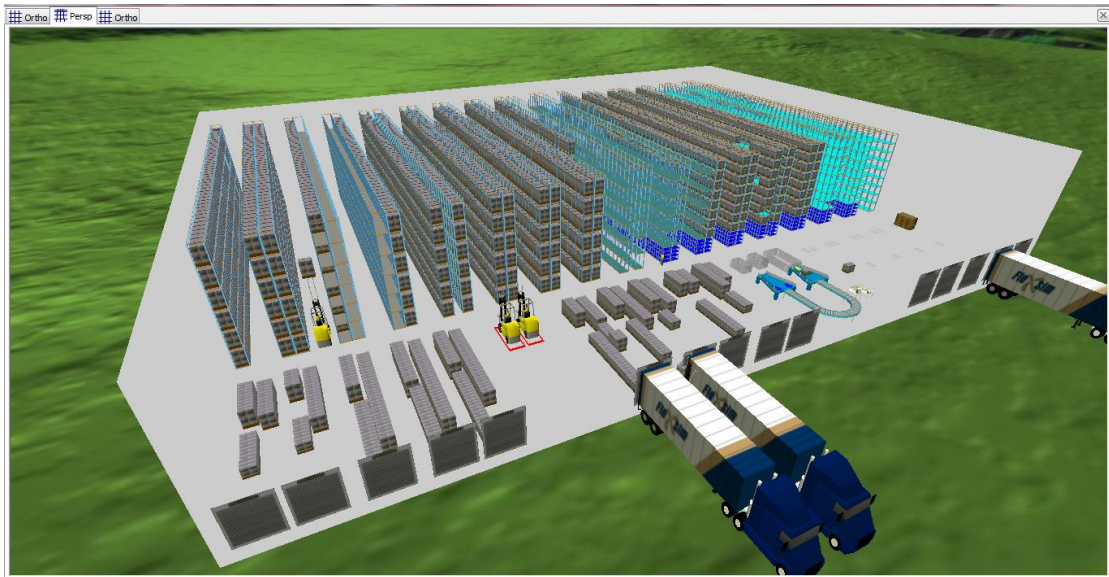
Το τμήμα σύνθεσης παραγγελιών είναι κυρίως ανενεργό αφού το περνοφόρο δεν προλαβαίνει να μεταφέρει παλέτες σε αυτό.



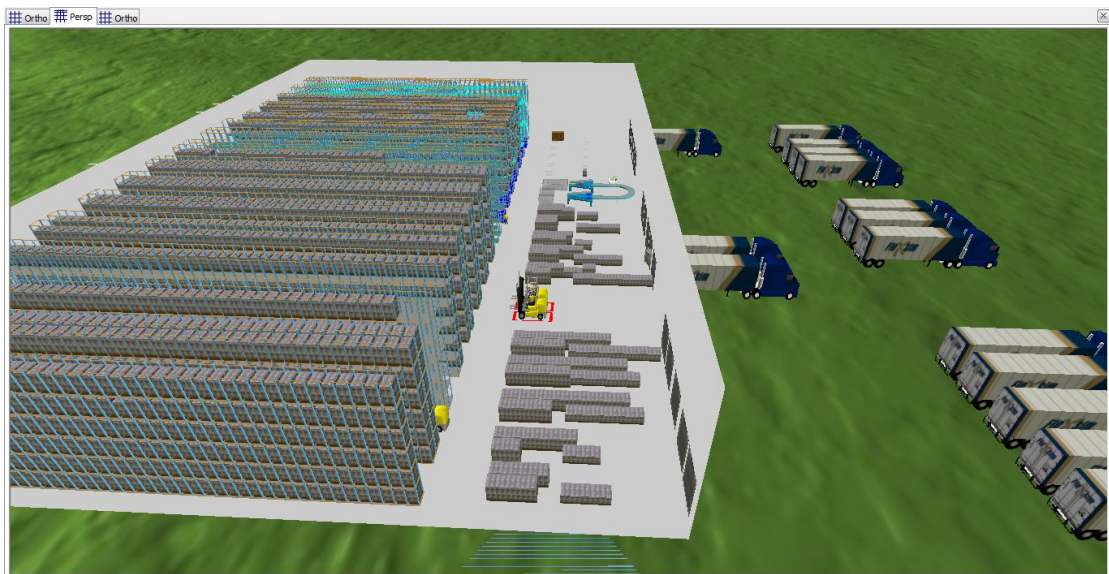
Σχήμα 4.6.4 - Consolidation



Σχήμα 4.6.5 – Overview 1



Σχήμα 4.6.6 – Overview 2

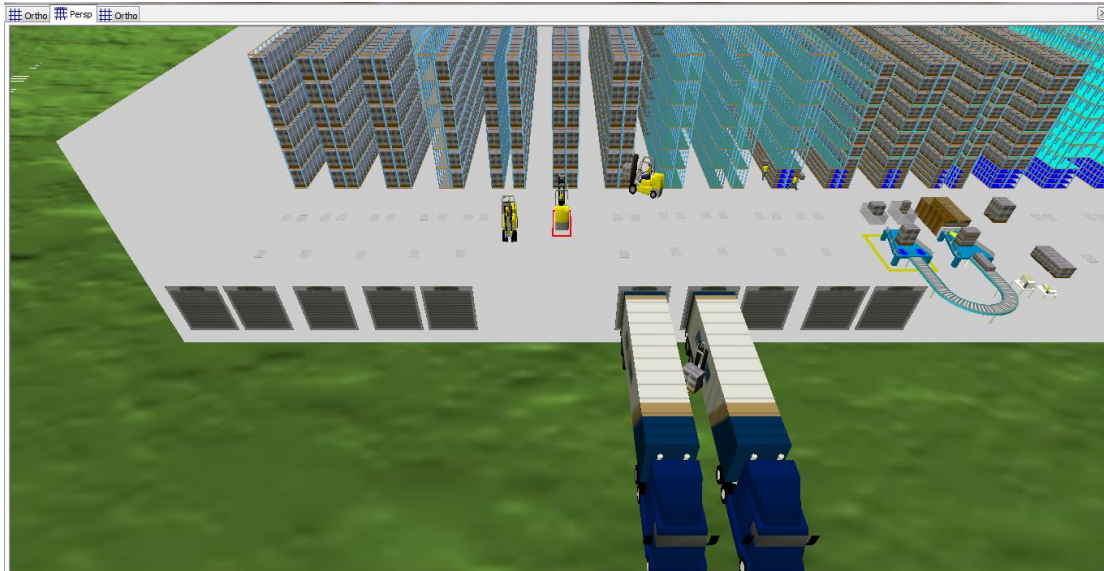


Σχήμα 4.6.7 – Overview 3



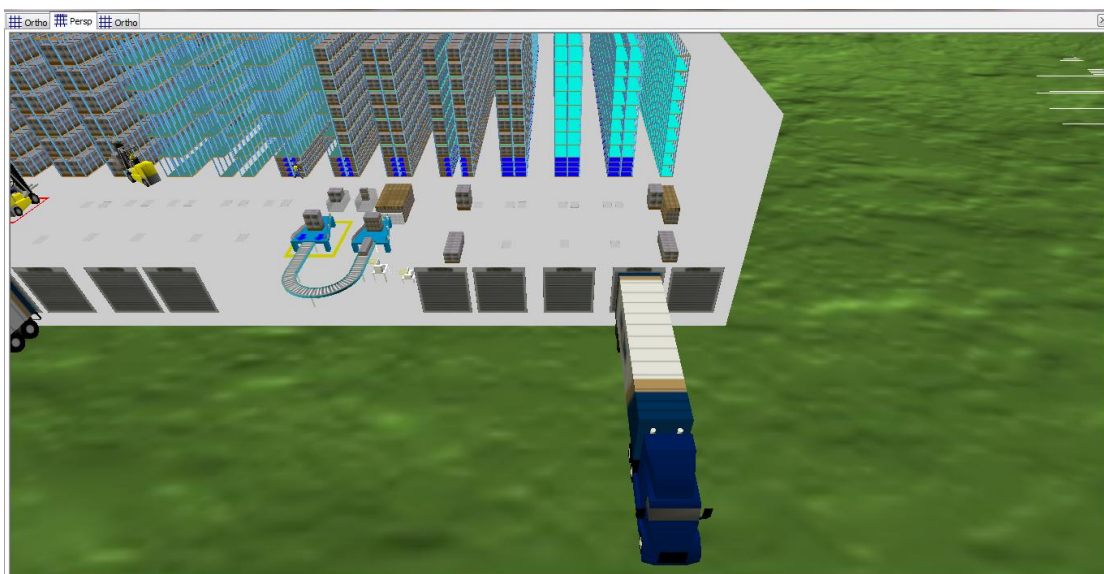
## Scenario 2

Παρακάτω φαίνεται η εικόνα του τμήματος παραλαβών για τη δεύτερη περίπτωση. Τα δύο περνοφόρα αδειάζουν πολύ γρήγορα τις ουρές αναμονής του τμήματος και αποθηκεύουν τις παλέτες στα ράφια. Έτσι δεν υπάρχει καθόλου συμφόρηση.



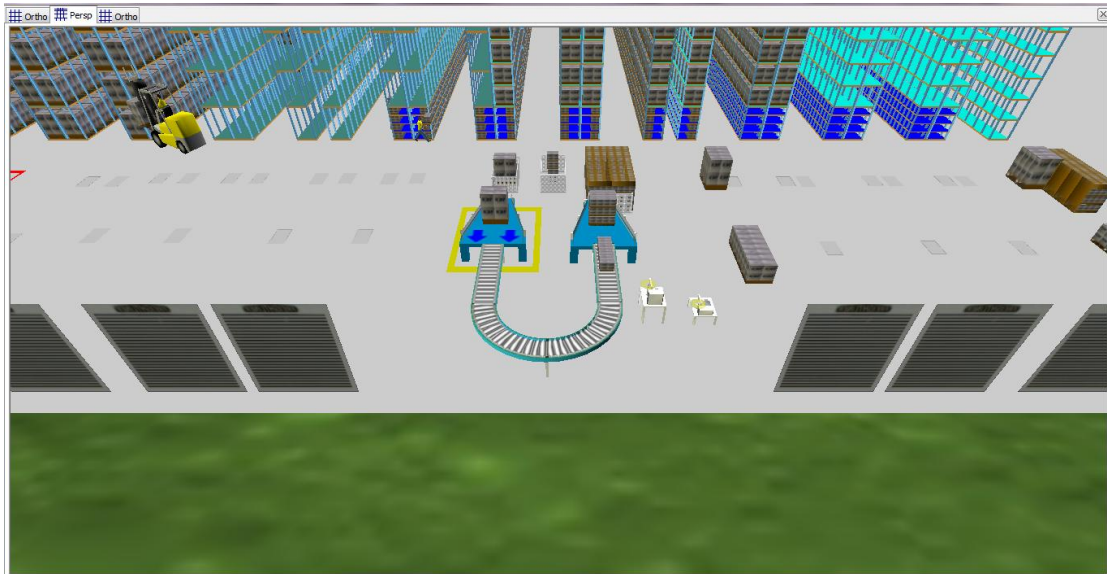
Σχήμα 4.6.8 – Receiving

Το τμήμα αποστολών έχει τις παραγγελίες έτοιμες κατά την άφιξη των φορτηγών και ταυτόχρονα έχουν αρχίσει να συγκεντρώνονται οι παλέτες και για τις επόμενες παραγγελίες.

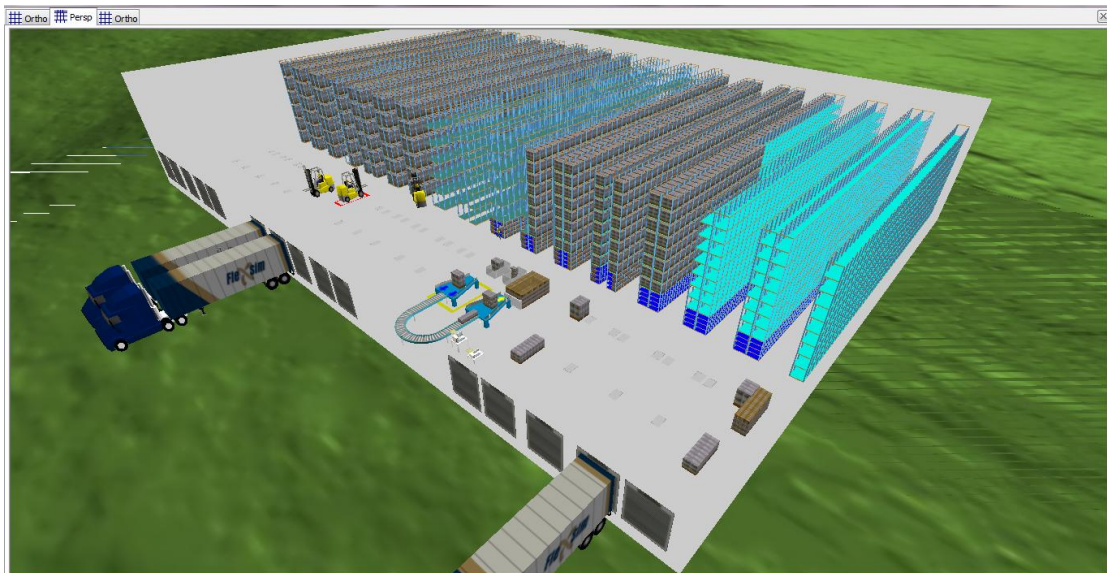


Σχήμα 4.6.9 - Delivery

Το τμήμα σύνθεσης παραγγελιών λειτουργεί συνεχώς και προετοιμάζει τις επόμενες παραγγελίες.



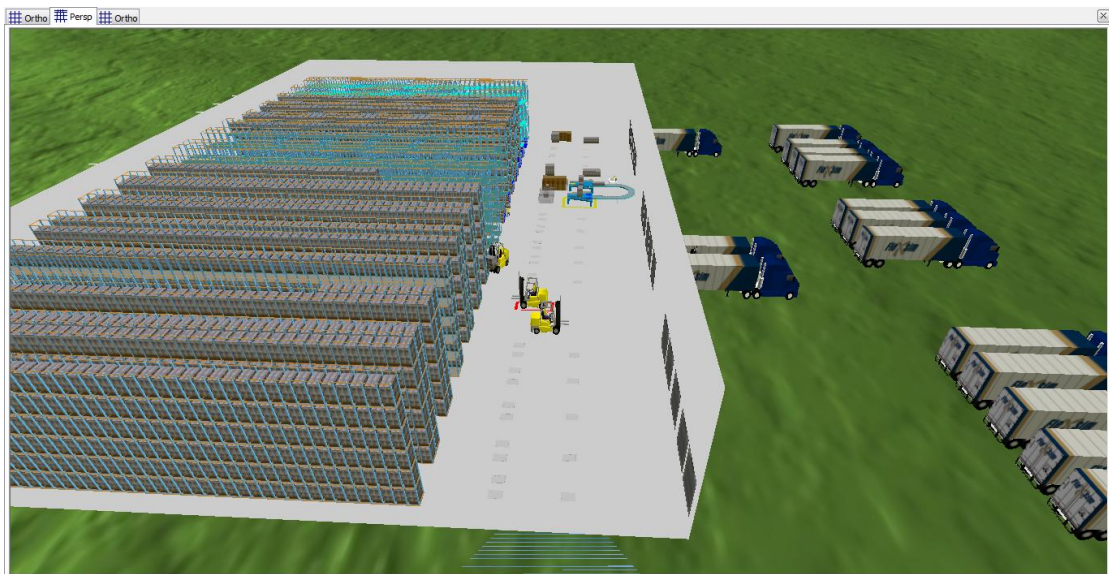
Σχήμα 4.6.10 - Consolidation



Σχήμα 4.6.11 – Overview 1



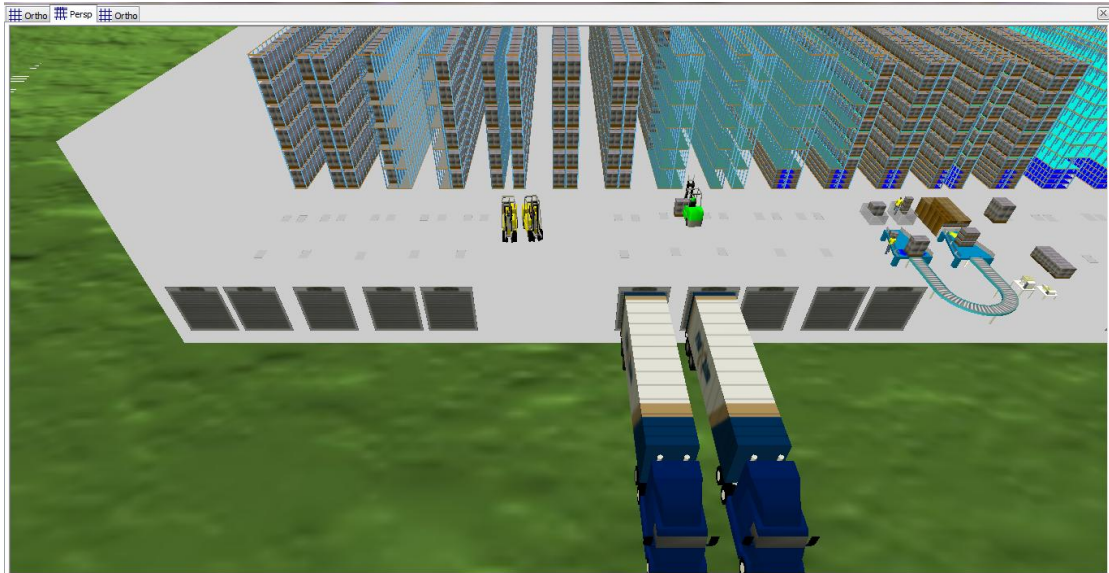
Σχήμα 4.6.12 – Overview 2



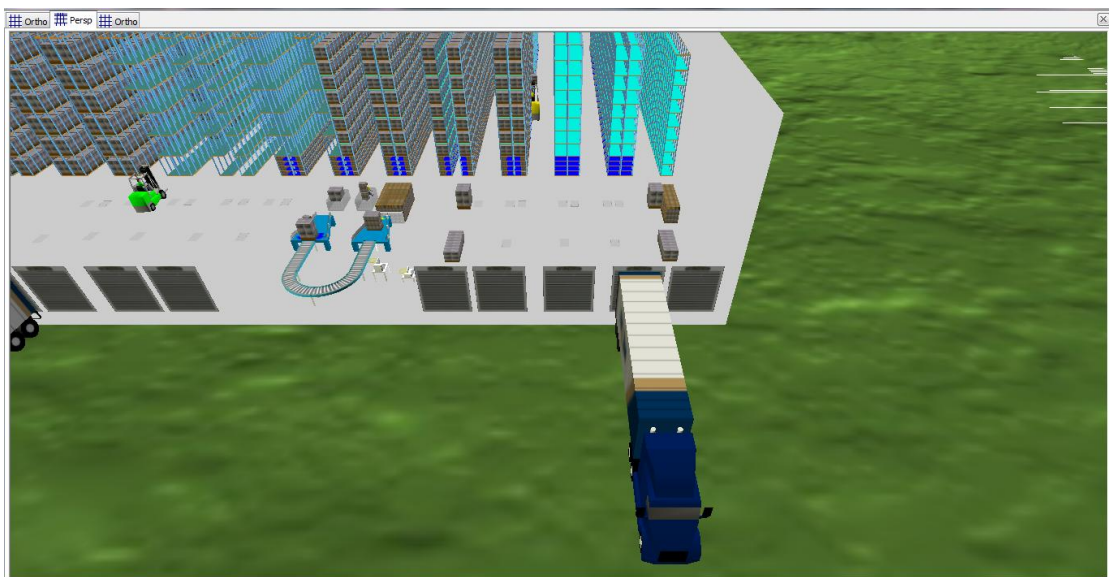
Σχήμα 4.6.13 – Overview 3

### **Scenario 3**

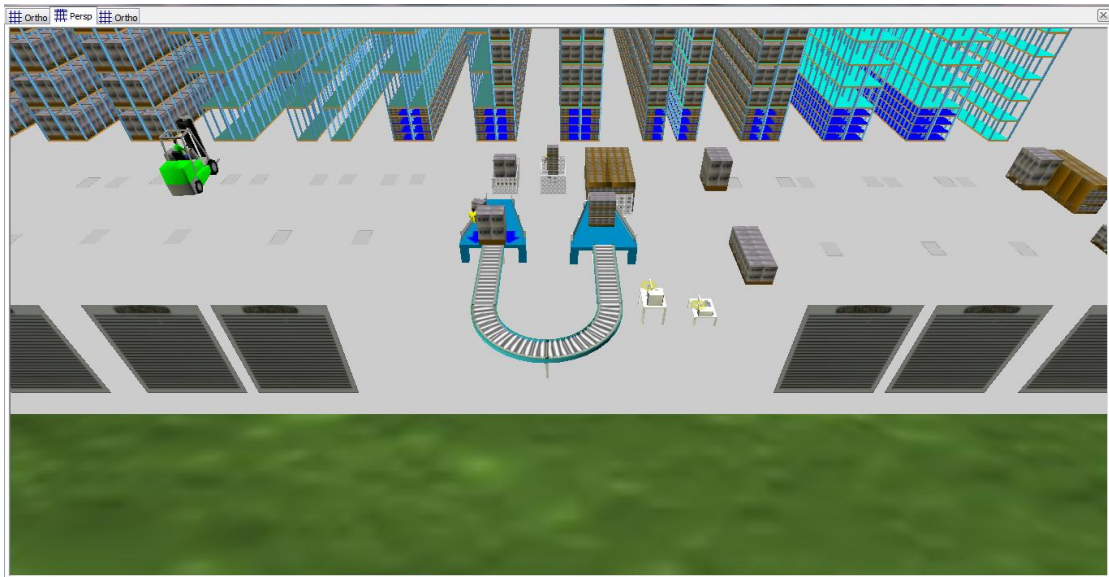
Παρακάτω φαίνεται η εικόνα του τμήματος παραλαβών για την τρίτη περίπτωση. Η εικόνα της αποθήκης είναι παρόμοια με τη δεύτερη περίπτωση, δεν δημιουργείται συμφόρηση αλλά ο αριθμός των περνοφόρων είναι υπερβολικός με αποτέλεσμα τα περνοφόρα να είναι κυρίως ανενεργά.



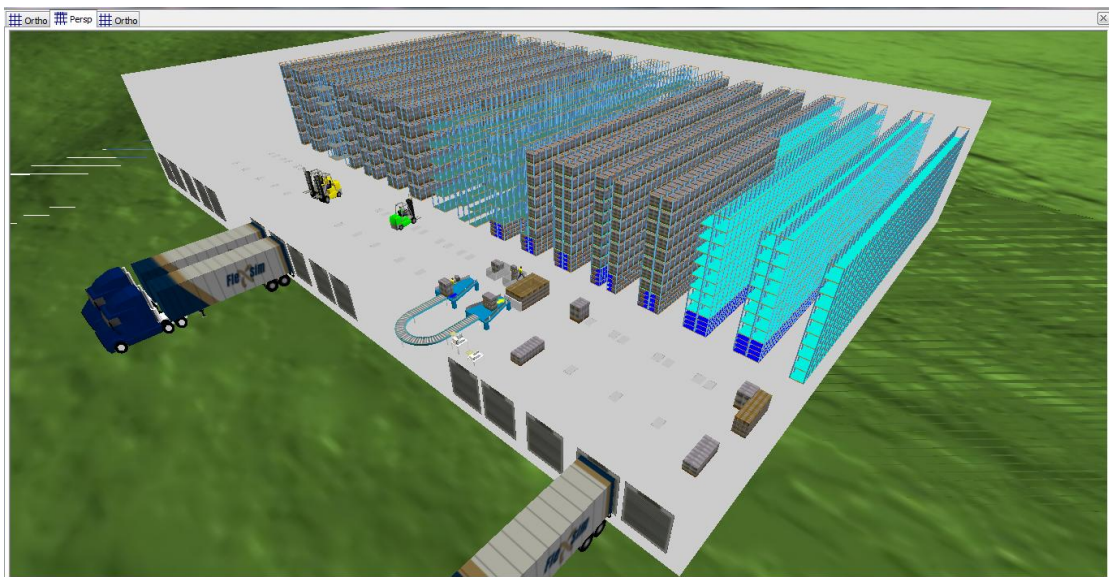
**Σχήμα 4.6.14 - Receiving**



**Σχήμα 4.6.15 - Delivery**



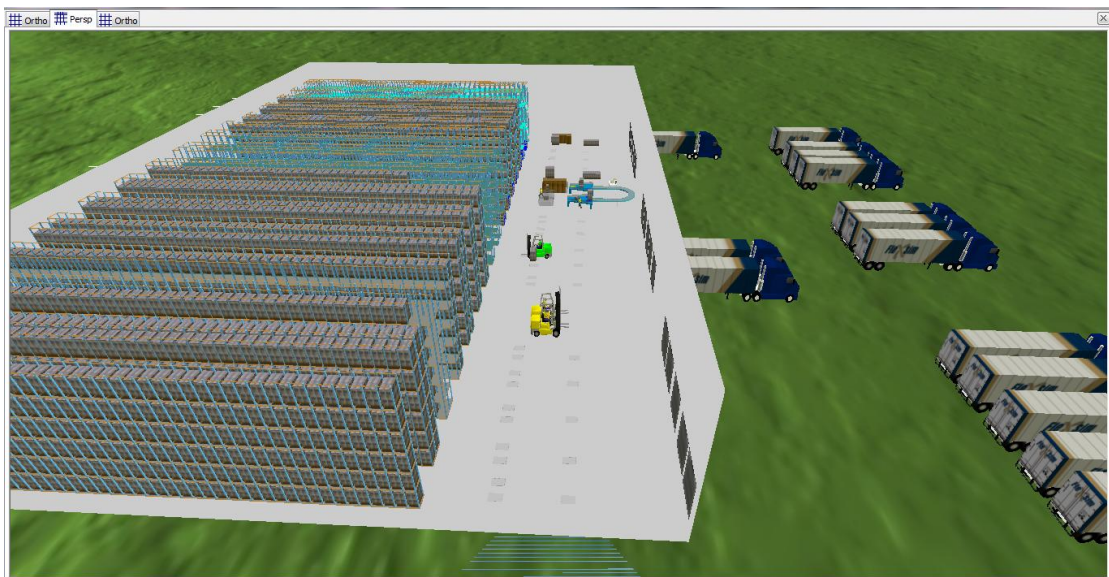
Σχήμα 4.6.16 - Consolidation



Σχήμα 4.6.17 – Overview 1



Σχήμα 4.6.18 – Overview 2



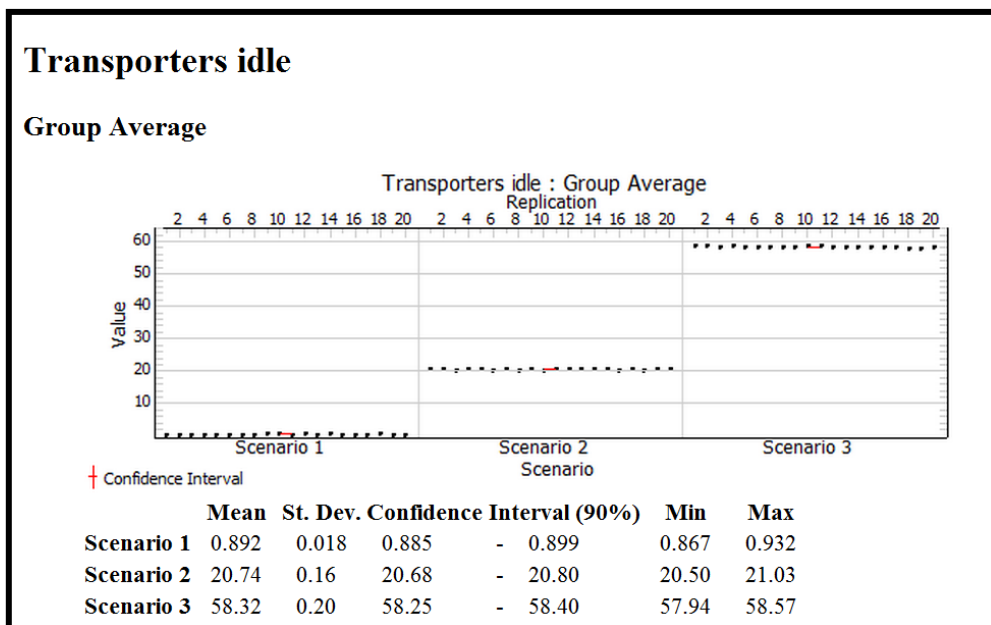
Σχήμα 4.6.19 – Overview 3

## 4.7 Αποτελέσματα προσομοίωσης

### 4.7.1 Διαγράμματα

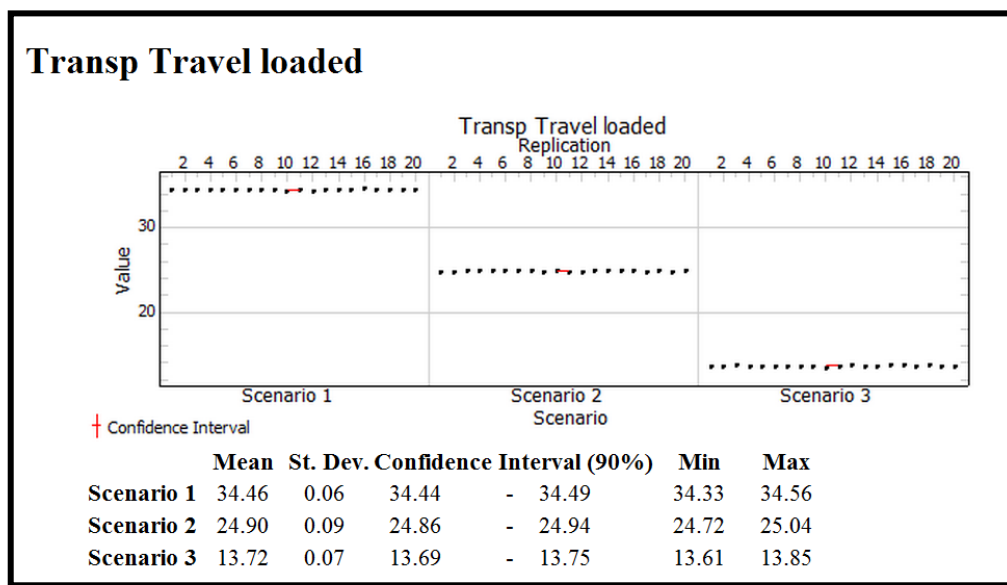
Τα ακόλουθα διαγράμματα έχουν προκύψει με τη βοήθεια του Experimenter και μας βοηθούν να καταλάβουμε κατά πόσο αξιοποιήθηκαν τα περονοφόρα σε κάθε σενάριο.

- Στο σχήμα 4.7.1.1 φαίνεται κατά μέσο όρο το ποσοστό του χρόνου που τα περονοφόρα ήταν στην κατάσταση idle, δηλαδή ανενεργά. Είναι εμφανές ότι στην πρώτη περίπτωση - scenario 1, όπου γίνεται χρήση ενός περονοφόρου για τη διακίνηση παλετών στην αποθήκη, αυτό δεν είναι καθόλου ανενεργό. Στη δεύτερη περίπτωση - scenario 2, που χρησιμοποιούνται δύο περονοφόρα, αυτά είναι ανενεργά το 20% του χρόνου. Στην τρίτη περίπτωση - scenario 3, όπου χρησιμοποιούνται τρία περονοφόρα, αυτά είναι κυρίως ανενεργά, δηλαδή το 58% του χρόνου. Από τον πρώτο κιάλας δείκτη γίνεται αντιληπτό ότι η χρήση τριών περονοφόρων για τη διακίνηση παλετών στην αποθήκη είναι υπερβολική.



Σχήμα 4.7.1.1

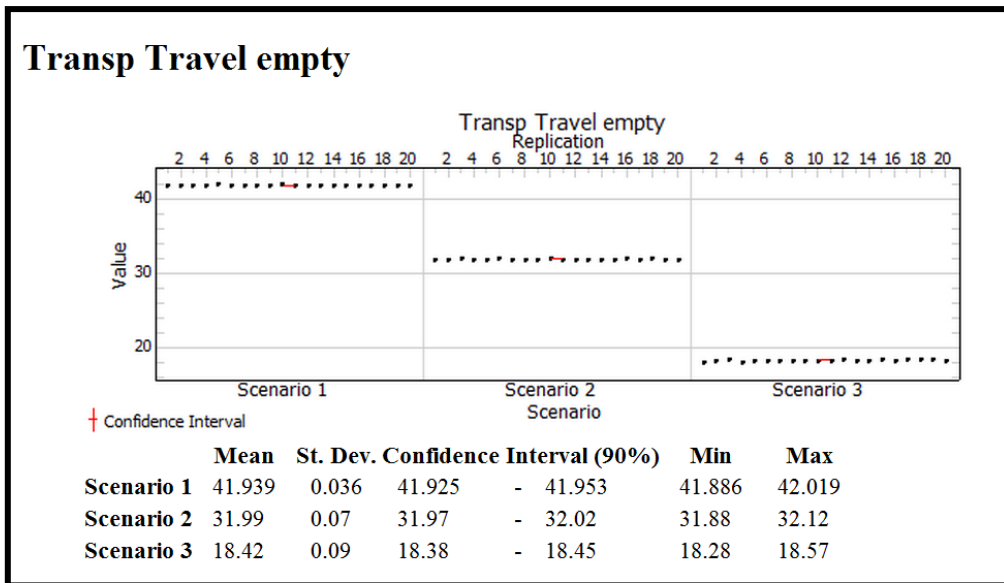
- Στο ακόλουθο διάγραμμα φαίνεται κατά μέσο όρο το ποσοστό του χρόνου που τα περονοφόρα μετέφεραν στοιχεία, δηλαδή ήταν στην κατάσταση travel loaded. Βλέπουμε ότι το ποσοστό είναι μεγαλύτερο στην πρώτη περίπτωση και μικρότερο στην τρίτη. Αυτό συμβαίνει γιατί οι εργασίες στην πρώτη περίπτωση εκτελούνται αποκλειστικά από ένα περονοφόρο, ενώ στις άλλες περιπτώσεις μοιράζονται.



Σχήμα 4.7.1.2

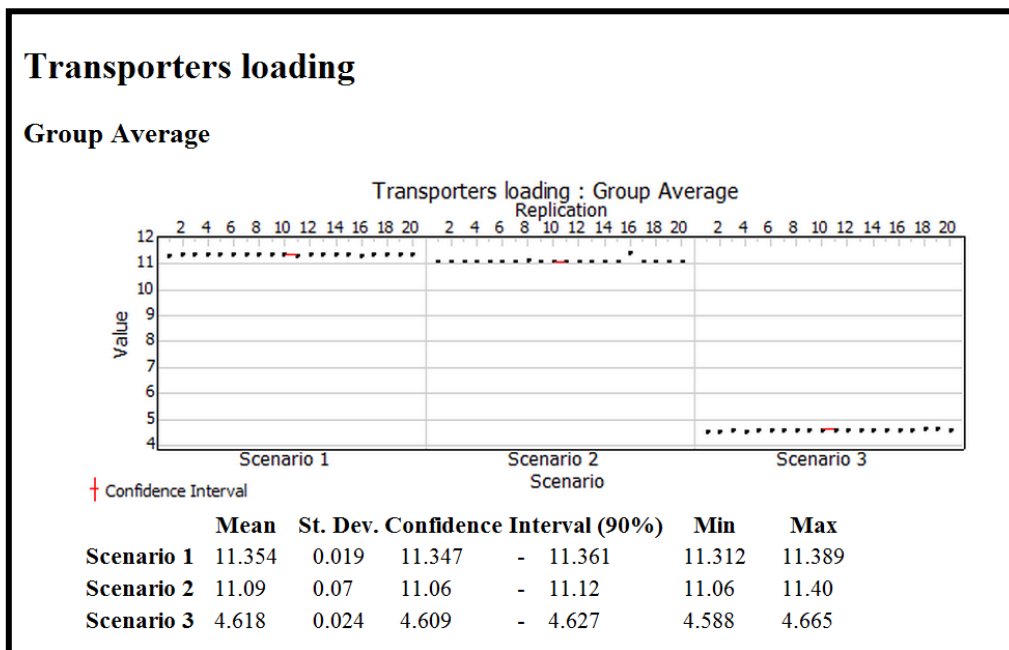
- Παρακάτω φαίνεται κατά μέσο όρο το ποσοστό του χρόνου που τα περονοφόρα κινούνταν άδεια, χωρίς να μεταφέρουν στοιχεία, ήταν δηλαδή στην κατάσταση travel empty. Τα ποσοστά ακολουθούν ίδια πορεία με αυτά της κατάστασης travel loaded, αφού όταν ένα περονοφόρο περατώσει τη μεταφορά ενός στοιχείου χρειάζεται περίπου το ίδιο χρονικό διάστημα για τη μεταφορά του σε κάποιο άλλο στοιχείο που ζητά μεταφορά. Περίπου το 6% της κατάστασης travel empty οφείλεται στο ότι τα περονοφόρα έχουν ρυθμιστεί να επιστρέφουν στην αρχική τους θέση – home location κάθε φορά που τελειώνουν μία εργασία. Έχει γίνει προσπάθεια μείωσής του ποσοστού της κατάστασης travel empty με την επιλογή να καλείται το περονοφόρο που είναι πιο κοντά στο στοιχείο που καλεί μεταφορά. Παρόλα αυτά η μείωση του ποσοστού δεν ήταν σημαντική.



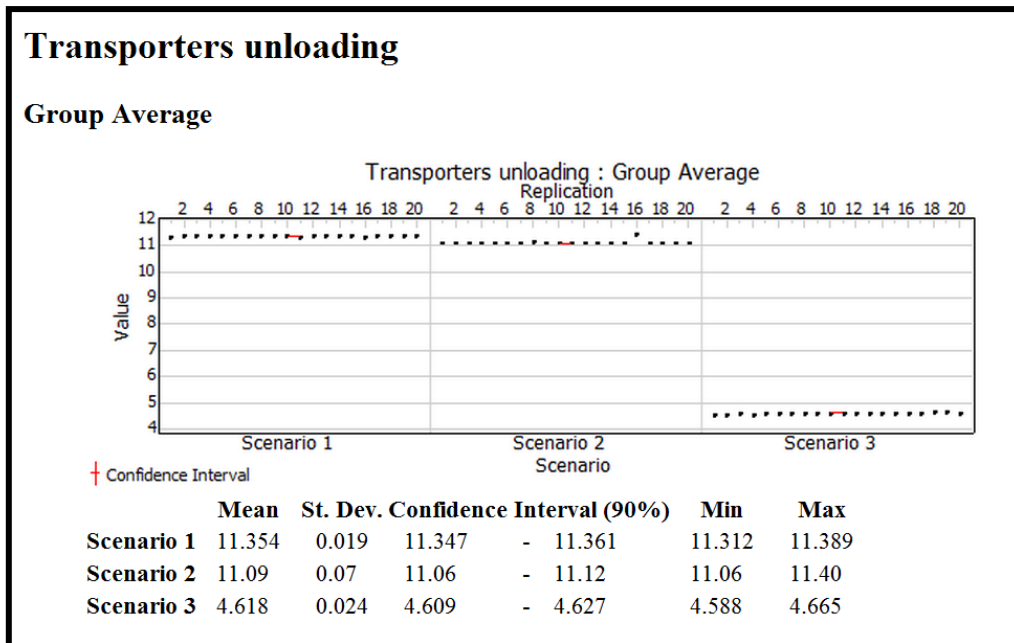


Σχήμα 4.7.1.3

- Ακόμα, στην πρώτη και δεύτερη περίπτωση, 22% του χρόνου καταναλώνεται συνολικά για τη φόρτωση και την εκφόρτωση των στοιχείων. Για την τρίτη περίπτωση το ποσοστό αυτό είναι εντελώς διαφορετικό αφού τα περονοφόρα είναι κατά κύριο λόγο ανενεργά.

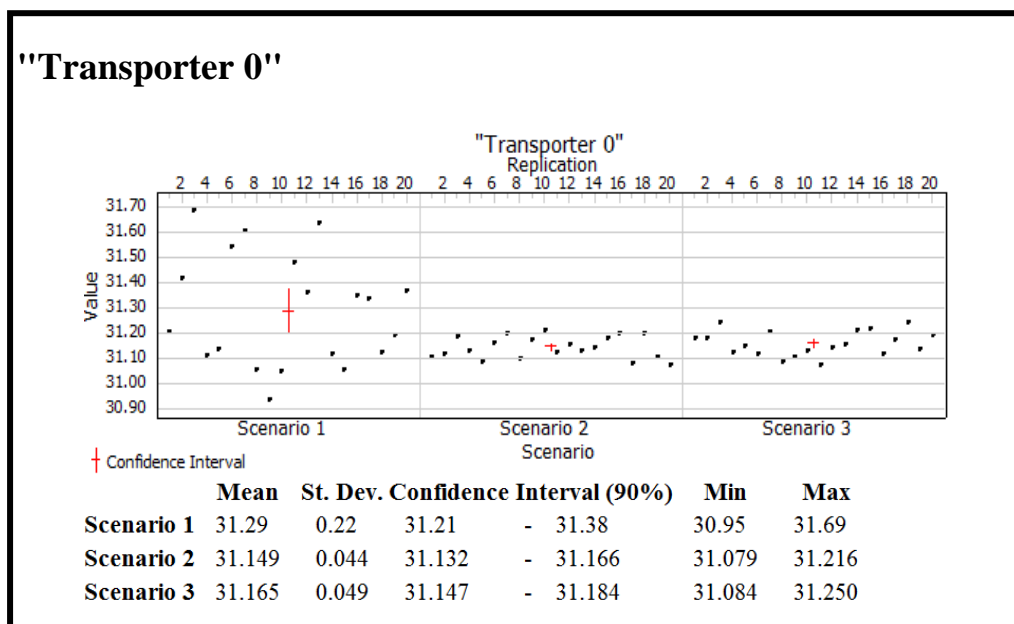


Σχήμα 4.7.1.4



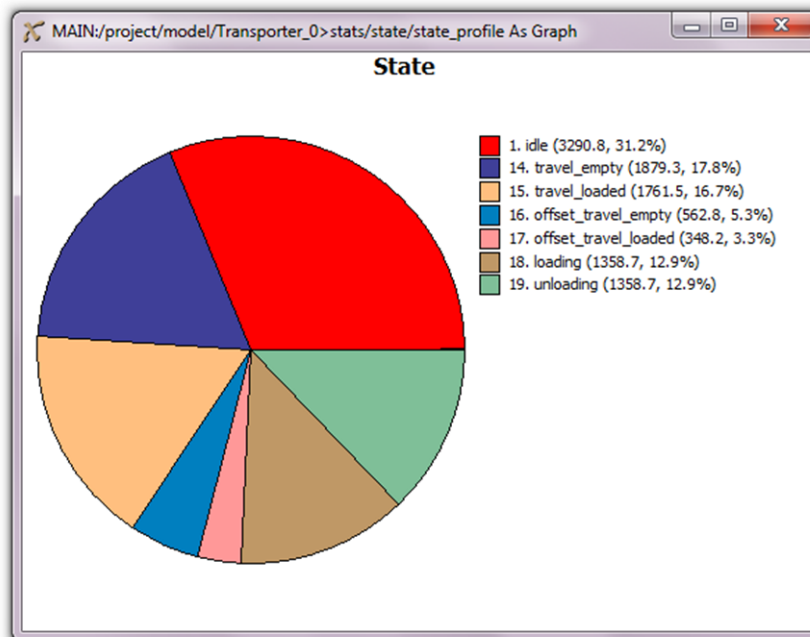
Σχήμα 4.7.1.5

- ➔ Στο επόμενο διάγραμμα φαίνεται το ποσοστό του χρόνου που είναι ανενεργό το περονοφόρο που πραγματοποιεί τη φόρτωση και την εκφόρτωση των φορητών. Τα ποσοστά είναι περίπου ίδια στις τρεις περιπτώσεις. Παρότι το ποσοστό είναι σχετικά μεγάλο έχει χρησιμοποιηθεί περονοφόρο που αναλαμβάνει αποκλειστικά τη φόρτωση και την εκφόρτωση παλετών για καλύτερο διαμερισμό των εργασιών και ευκολότερες μετακινήσεις των περονοφόρων στο δίκτυο.



Σχήμα 4.7.1.6

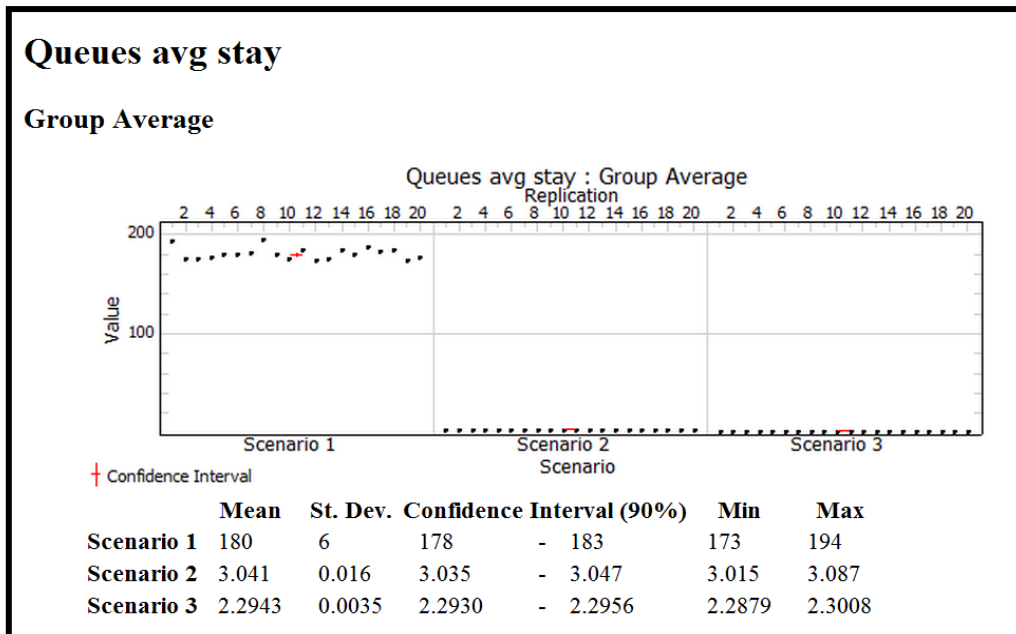
- ➔ Παρακάτω παρουσιάζεται ενδεικτικά η χρήση του περονοφόρου φόρτωσης/εκφόρτωσης παλετών σε διάγραμμα πίτας.



Σχήμα 4.7.1.7

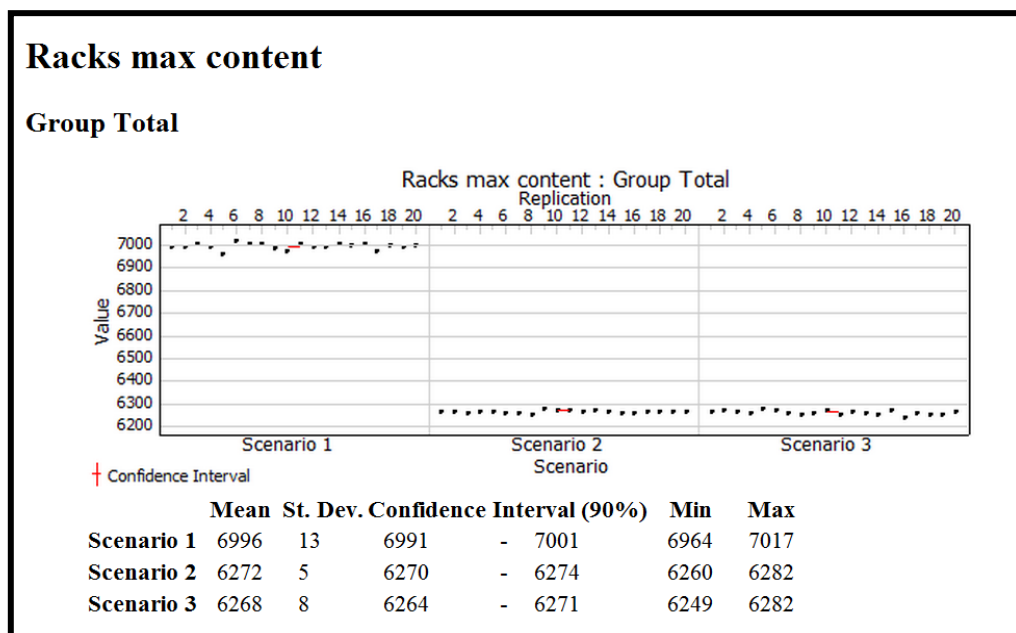
Τα ακόλουθα διαγράμματα μας βοηθούν να καταλάβουμε τη χρήση που έγινε στις ουρές αναμονής και στα ράφια αποθήκευσης παλετών σε κάθε περίπτωση.

- ➔ Παρακάτω φαίνεται ο μέσος όρος του χρόνου παραμονής των παλετών στις ουρές αναμονής του τμήματος παραλαβών. Είναι εμφανές ότι με τη χρήση ενός περονοφόρου - scenario 1 οι παλέτες παραμένουν στις ουρές αναμονής για 180 λεπτά κατά μέσο όρο, δηλαδή για 3 ώρες. Αυτό έχει σαν αποτέλεσμα να δημιουργείται συμφόρηση αφού με το πέρασμα του χρόνου οι ουρές αναμονής φτάνουν στο σημείο να μην μπορούν να δεχτούν άλλες παλέτες και τα φορητά να μην μπορούν να προχωρήσουν σε εκφόρτωση. Όταν γίνεται χρήση δύο περονοφόρων - scenario 2 η κατάσταση είναι σαφώς βελτιωμένη αφού μία παλέτα παραμένει σε ουρά αναμονής του τμήματος παραλαβών μόλις για τρία λεπτά. Η διαφορά με τη χρήση τριών περονοφόρων - scenario 3 είναι μικρή, αφού ο μέσος χρόνος παραμονής μειώνεται περίπου κατά ένα λεπτό.



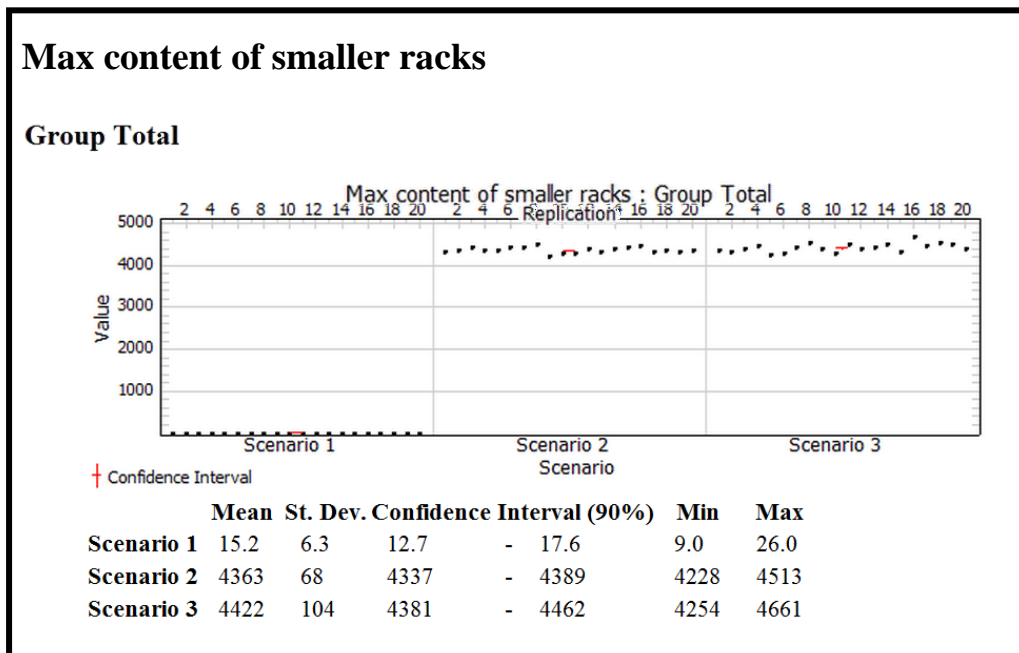
Σχήμα 4.7.1.8

- Στο επόμενο διάγραμμα φαίνεται ο συνολικός αριθμός των παλετοθέσεων που χρησιμοποιούνται σε κάθε περίπτωση. Παρατηρήθηκε ότι η χρήση ενός περονοφόρου - scenario 1 είχε σαν αποτέλεσμα να δίνεται προτεραιότητα στην τοποθέτηση των παλετών στα ράφια αλλά όχι στη σύνθεση παραγγελιών στο τμήμα αποστολών. Έτσι, οι παλέτες έμεναν αποθηκευμένες στα ράφια. Με τη χρήση δύο και τριών περονοφόρων – scenario 2 και scenario 3 αντίστοιχα, η χρήση των ραφιών είναι ίδια.



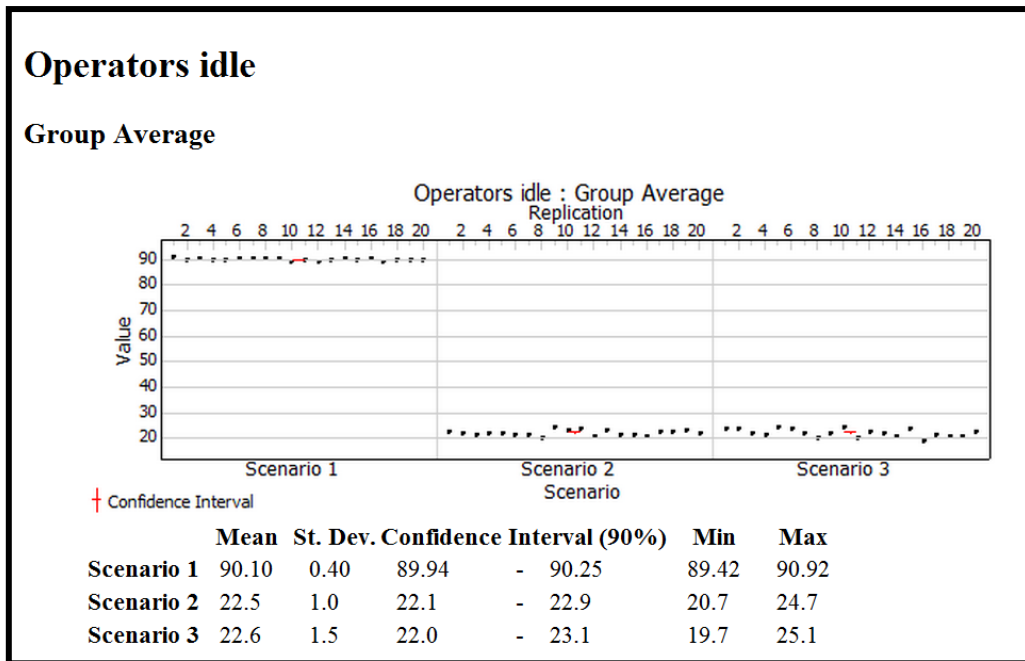
Σχήμα 4.7.1.9

- Στο επόμενο διάγραμμα φαίνεται ο συνολικός αριθμός των θυρίδων που έχουν χρησιμοποιηθεί. Στην πρώτη περίπτωση - scenario 1 χρησιμοποιούνται πολύ λίγες θυρίδες εφόσον δεν δίνεται προτεραιότητα στη σύνθεση παραγγελιών (και επομένως δεν δημιουργούνται κιβώτια από τη διάσπαση παλετών που προορίζονται για παραγγελίες). Στις άλλες δύο περιπτώσεις η χρήση των θυρίδων είναι αρκετά μεγάλη.

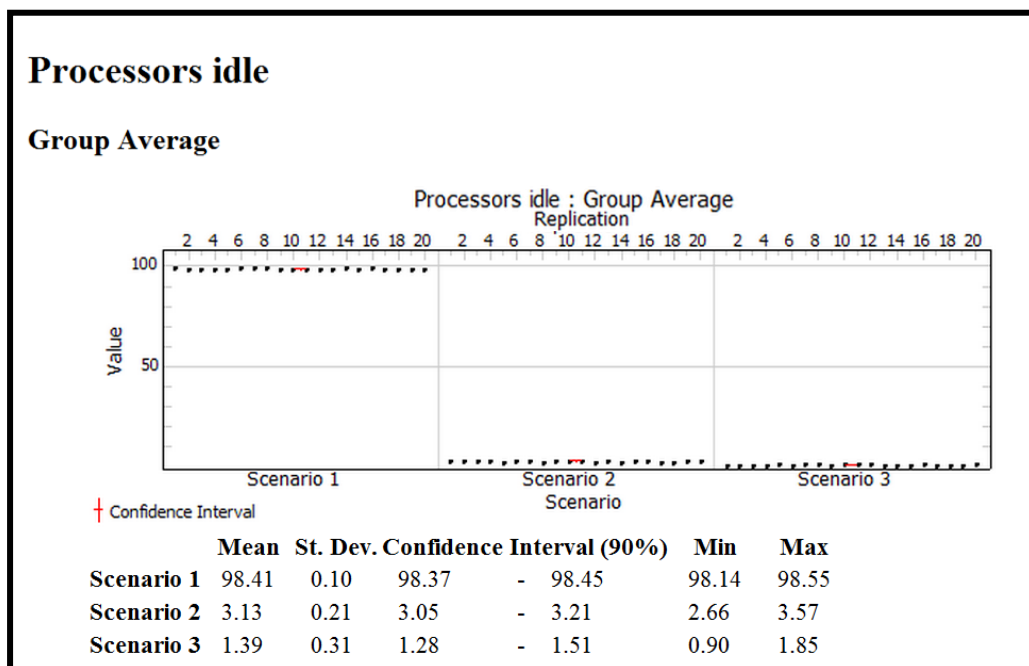


Σχήμα 4.7.1.10

- Στο επόμενα διαγράμματα φαίνεται κατά μέσο όρο το ποσοστό του χρόνου που οι εργάτες/μεταφορείς και οι επεξεργαστές του τμήματος σύνθεσης παραγγελιών αντίστοιχα ήταν ανενεργοί, δηλαδή στην κατάσταση idle. Όπως αναφέρθηκε ήδη, με τη χρήση ενός περονοφόρου - scenario 1 δεν δίνεται προτεραιότητα στη σύνθεση παραγγελιών και γι αυτό το λόγο οι μεταφορείς και οι επεξεργαστές σε αυτό το τμήμα μένουν κατά κύριο λόγο ανενεργοί. Αντίθετα, με τη χρήση δύο και τριών περονοφόρων - scenario 2 και scenario 3, οι μεταφορείς μένουν ανενεργοί για ένα λογικό διάστημα, ενώ οι επεξεργαστές σχεδόν καθόλου.



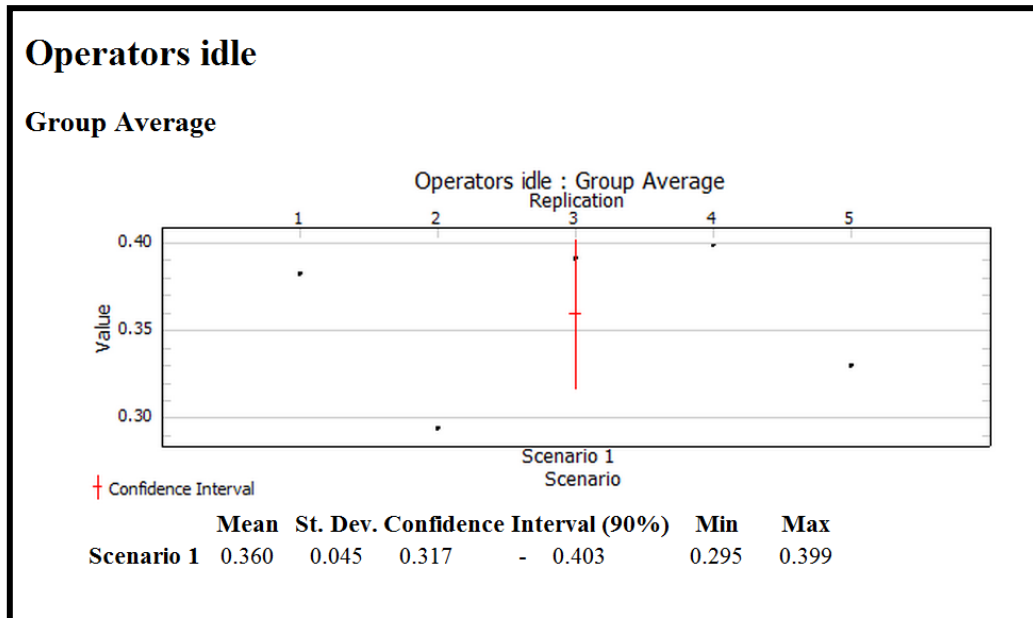
Σχήμα 4.7.1.11



Σχήμα 4.7.1.12

- Επιπλέον, έγινε ακόμα μία δοκιμή σχετικά με το αν μπορεί να χρησιμοποιηθεί μόνο ένας εργάτης/μεταφορέας στο τμήμα σύνθεσης παραγγελιών. Η δοκιμή αυτή έγινε με την υπόθεση ότι χρησιμοποιούνται δύο περονοφόρα, δηλαδή έστω ότι επιλέγεται το scenario 2. Για τη δοκιμή αυτή πραγματοποιήθηκαν 5 επαναλήψεις. Στο επόμενο διάγραμμα φαίνεται κατά μέσο όρο το ποσοστό του

χρόνου που ο μεταφορέας ήταν ανενεργός. Φαίνεται ότι ο αυτός δεν είναι καθόλου ανενεργός με αποτέλεσμα να μην μπορεί να λειτουργήσει επαρκώς το τμήμα σύνθεσης παραγγελιών. Επομένως για την καλύτερη λειτουργία του τμήματος κρίνεται απαραίτητη η ύπαρξη δύο μεταφορέων.

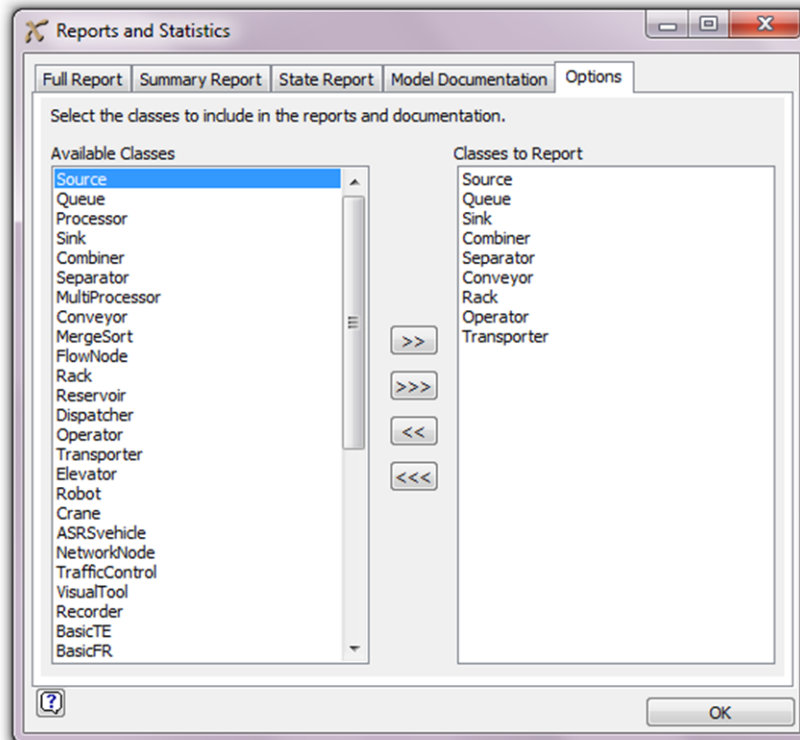


Σχήμα 4.7.1.13

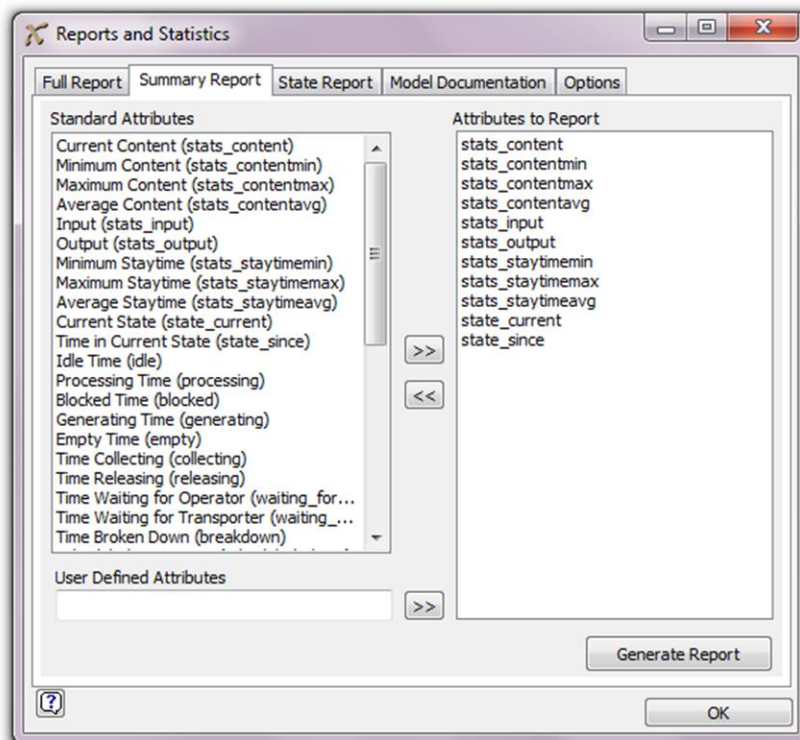
#### 4.7.2 Αναφορές

Το FlexSim παρέχει ακόμα τη δυνατότητα να μελετηθούν περισσότερα στατιστικά στοιχεία που καταγράφηκαν κατά την τελευταία επανάληψη της προσομοίωσης με τη βοήθεια αναφορών, όπως η γενική αναφορά και η αναφορά κατάστασης.

Πρώτα επιλέγουμε τα αντικείμενα για τα οποία μας ενδιαφέρει να δούμε τα στατιστικά στοιχεία. Έπειτα επιλέγουμε τα στοιχεία που θέλουμε να δούμε για κάθε αντικείμενο, όπως το μέγιστο, ελάχιστο ή μέσο περιεχόμενό του, πληροφορίες για τον αριθμό των στοιχείων που εισήλθαν και εξήλθαν από το αντικείμενο και το χρόνο παραμονής του σε αυτό. Στη συνέχεια, για την αναφορά κατάστασης, επιλέγουμε τις πιθανές καταστάσεις που μας ενδιαφέρουν. Έτσι παίρνουμε για κάθε αντικείμενο στοιχεία για το ποσοστό του χρόνου που αντιστοιχεί σε κάθε κατάσταση, π.χ. ανενεργό, σε λειτουργία, απασχολημένο κτλ. Οι αναφορές εξάγονται σε αρχείο csv και εμφανίζονται αυτόματα στο Excel. Παρακάτω παρουσιάζονται ενδεικτικά οι αναφορές για τη δεύτερη περίπτωση – scenario 2.

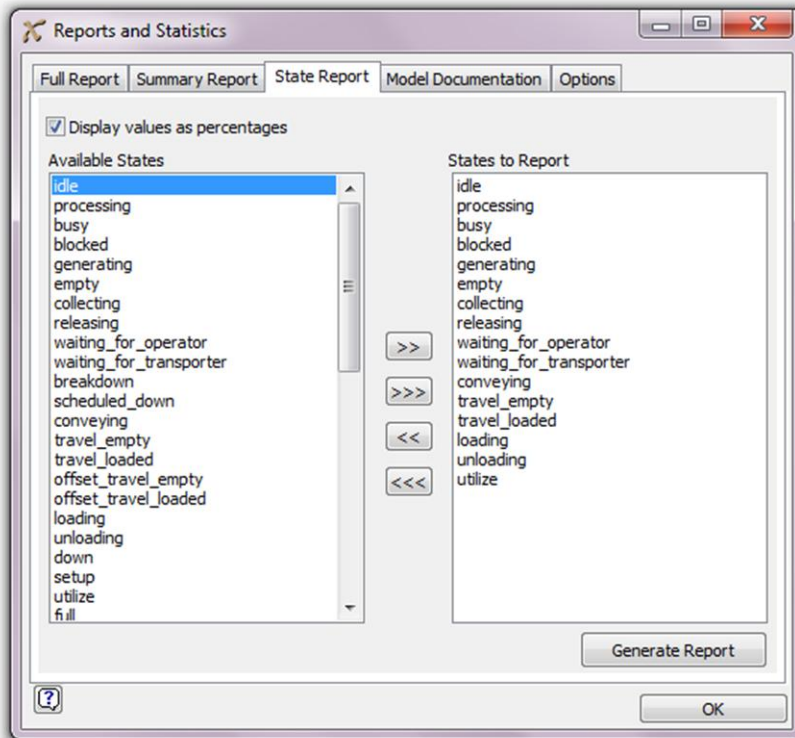


Σχήμα 4.7.2.1



Σχήμα 4.7.2.2





Σχήμα 4.7.2.3

Object	Class	stats_content	stats_contentmin	stats_contentmax	stats_contentavg	stats_input	stats_output	stats_staytimemin	stats_staytimemax	stats_staytimeavg	state_current	stat	
2	Time:	12.960.000.000											
4	Object	Class	stats_content	stats_contentmin	stats_contentmax	stats_contentavg	stats_input	stats_output	stats_staytimemin	stats_staytimemax	stats_staytimeavg	state_current	stat
5	Transporter_1	Transporter	0.000000	0.000000	1.000.000	0.232579	4.800.000.000	4.800.000.000	0.281542	1.102.886	0.626682	1.000.000	12.960
6	Transporter_2	Transporter	0.000000	0.000000	1.000.000	0.222496	4.620.000.000	4.620.000.000	0.279308	1.201.977	0.622900	1.000.000	12.960
7	Transporter_3	Transporter	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	12.000.000	12.960
8	Transporter_0	Transporter	0.000000	0.000000	1.000.000	0.267797	8.492.000.000	8.492.000.000	0.343063	0.448422	0.408450	1.000.000	12.960
9	Operator_1	Operator	1.000.000	0.000000	1.000.000	0.432985	6.849.000.000	6.848.000.000	0.195278	76.261.925	0.819235	4.000.000	12.960
10	Operator_2	Operator	0.000000	0.000000	1.000.000	0.436550	6.931.000.000	6.931.000.000	0.195278	54.675.148	0.816104	1.000.000	12.960
11	Queue_11	Queue	0.000000	0.000000	2.000.000	0.009043	149.000.000	149.000.000	0.387669	2.432.749	0.775260	6.000.000	12.960
12	Queue_12	Queue	0.000000	0.000000	3.000.000	0.016929	272.000.000	272.000.000	0.389929	2.683.506	0.795113	6.000.000	12.960
13	Queue_13	Queue	0.000000	0.000000	4.000.000	0.019668	299.000.000	299.000.000	0.421682	3.042.642	0.840126	6.000.000	12.960
14	Queue_21	Queue	0.000000	0.000000	3.000.000	0.015358	131.000.000	131.000.000	0.377940	4.386.623	1.498.962	6.000.000	12.960
15	Queue_22	Queue	0.000000	0.000000	5.000.000	0.029971	301.000.000	301.000.000	0.358825	4.575.216	1.273.609	6.000.000	12.960
16	Queue_23	Queue	0.000000	0.000000	4.000.000	0.028952	272.000.000	272.000.000	0.401046	4.399.959	1.361.396	6.000.000	12.960
17	Queue_31	Queue	0.000000	0.000000	4.000.000	0.022530	127.000.000	127.000.000	0.352484	5.150.634	2.271.015	6.000.000	12.960
18	Queue_32	Queue	0.000000	0.000000	4.000.000	0.052204	289.000.000	289.000.000	0.341275	5.526.204	2.313.689	6.000.000	12.960
19	Queue_33	Queue	0.000000	0.000000	6.000.000	0.047327	288.000.000	288.000.000	0.375123	5.350.848	2.104.634	6.000.000	12.960
20	Queue_41	Queue	0.000000	0.000000	2.000.000	0.009868	134.000.000	134.000.000	0.330308	3.660.577	0.944095	6.000.000	12.960
21	Queue_42	Queue	0.000000	0.000000	4.000.000	0.025212	305.000.000	305.000.000	0.312129	3.523.835	1.060.060	6.000.000	12.960
22	Queue_43	Queue	0.000000	0.000000	4.000.000	0.022338	265.000.000	265.000.000	0.351400	3.454.642	1.080.796	6.000.000	12.960
23	Queue_51	Queue	0.000000	0.000000	4.000.000	0.022922	145.000.000	145.000.000	0.300882	6.773.924	2.031.524	6.000.000	12.960
24	Queue_52	Queue	0.000000	0.000000	6.000.000	0.041254	257.000.000	257.000.000	0.288125	6.835.021	2.063.527	6.000.000	12.960
25	Queue_53	Queue	0.000000	0.000000	6.000.000	0.051417	302.000.000	302.000.000	0.318511	6.558.814	2.189.195	6.000.000	12.960

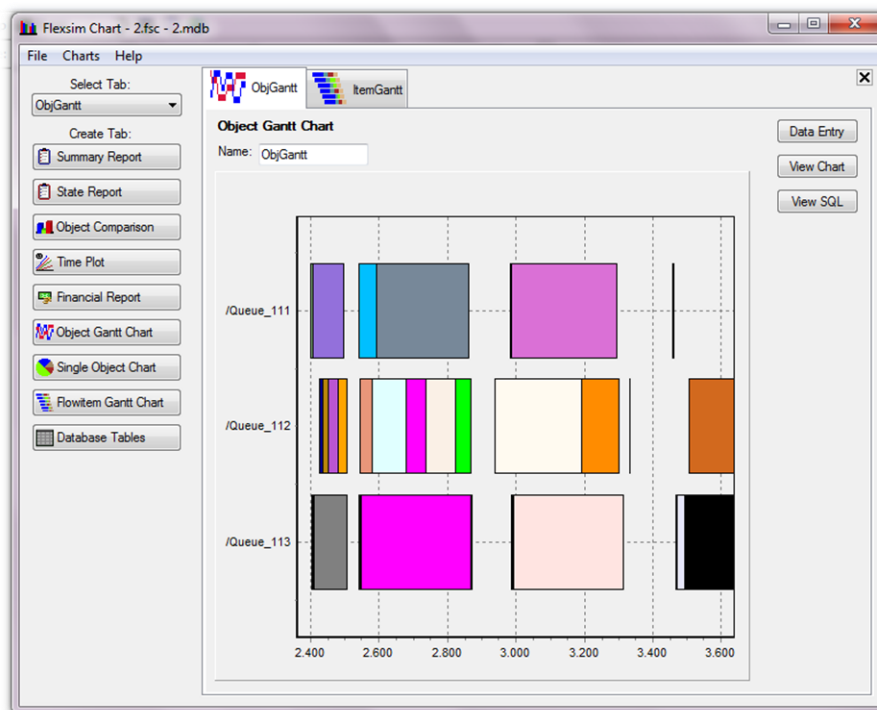
Σχήμα 4.7.2.4

Object	Class	idle	processing	busy	blocked	generating	empty	collecting	releasing	waiting_for	waiting_for	conveying	travel_empty	travel_loaded	loading	unloading	utilize
Transporter_1	Transporter	20.00%	0.00%	0.00%	0.22%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	32.48%	25.60%	11.10%	11.10%	0.00%
Transporter_2	Transporter	21.48%	0.00%	0.00%	0.16%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	31.50%	24.20%	11.08%	11.08%	0.00%
Transporter_3	Transporter	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
Transporter_0	Transporter	30.92%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	18.04%	16.69%	12.87%	12.87%	0.00%
Operator_1	Operator	4.31%	0.00%	0.00%	12.72%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	31.86%	25.98%	0.00%	0.00%	4.50%
Operator_2	Operator	3.93%	0.00%	0.00%	12.59%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	31.83%	26.61%	0.00%	0.00%	4.29%
Queue_11	Queue	0.00%	0.00%	0.00%	0.00%	0.00%	98.97%	0.00%	0.00%	0.00%	1.03%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
Queue_12	Queue	0.00%	0.00%	0.00%	0.00%	0.00%	98.16%	0.00%	0.00%	0.00%	1.84%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
Queue_13	Queue	0.00%	0.00%	0.00%	0.00%	0.00%	97.93%	0.00%	0.00%	0.00%	2.07%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
Queue_21	Queue	0.00%	0.00%	0.00%	0.00%	0.00%	98.46%	0.00%	0.00%	0.00%	1.54%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
Queue_22	Queue	0.00%	0.00%	0.00%	0.00%	0.00%	97.32%	0.00%	0.00%	0.00%	2.68%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
Queue_23	Queue	0.00%	0.00%	0.00%	0.00%	0.00%	97.43%	0.00%	0.00%	0.00%	2.57%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
Queue_31	Queue	0.00%	0.00%	0.00%	0.00%	0.00%	97.82%	0.00%	0.00%	0.00%	2.18%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
Queue_32	Queue	0.00%	0.00%	0.00%	0.00%	0.00%	96.21%	0.00%	0.00%	0.00%	3.79%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
Queue_33	Queue	0.00%	0.00%	0.00%	0.00%	0.00%	96.35%	0.00%	0.00%	0.00%	3.65%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
Queue_41	Queue	0.00%	0.00%	0.00%	0.00%	0.00%	98.88%	0.00%	0.00%	0.00%	1.12%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
Queue_42	Queue	0.00%	0.00%	0.00%	0.00%	0.00%	97.60%	0.00%	0.00%	0.00%	2.40%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
Queue_43	Queue	0.00%	0.00%	0.00%	0.00%	0.00%	97.80%	0.00%	0.00%	0.00%	2.20%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
Queue_51	Queue	0.00%	0.00%	0.00%	0.00%	0.00%	97.93%	0.00%	0.00%	0.00%	2.07%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
Queue_52	Queue	0.00%	0.00%	0.00%	0.00%	0.00%	97.19%	0.00%	0.00%	0.00%	2.81%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
Queue_53	Queue	0.00%	0.00%	0.00%	0.00%	0.00%	96.82%	0.00%	0.00%	0.00%	3.18%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%

Σχήμα 4.7.2.5

### 4.7.3 Μεγέθη απόδοσης

Ο ακόλουθος πίνακας παρουσιάζει το μέσο χρόνο αποθήκευσης, σύνθεσης παραγγελίας και φόρτωσης συγκριτικά για κάθε περίπτωση. Τα μεγέθη αυτά υπολογίστηκαν με τη βοήθεια των διαγραμμάτων Gantt αντικειμένων και στοιχείων τα οποία μπορούν να κατασκευαστούν με χρήση του FlexSim Chart.



Σχήμα 4.7.3.1

Μέγεθος	Scenario 1	Scenario 2	Scenario 3
Μέσος χρόνος αποθήκευσης-storaging	182 min	5 min	4,3 min
Μέσος χρόνος σύνθεσης παραγγελίας	1231,8 min	41,4 min	39,8 min
Μέσος χρόνος φόρτωσης	10,2 min	10,2 min	10,2 min

Πίνακας 4.7.3.1

#### 4.8 Ανάλυση κόστους

Στον παρακάτω πίνακα φαίνονται οι απαιτούμενες δαπάνες εξοπλισμού της εταιρίας για την κάθε περίπτωση. Στα έξοδα αγοράς περονοφόρων έχει υπολογιστεί η αγορά ενός επιπλέον περονοφόρου. Αυτό θα χρησιμοποιείται σε περίπτωση βλάβης κάποιου περονοφόρου.

Είδος	Συντελεστής Απόσβεσης	Scenario 1	Scenario 2	Scenario 3
Αγορά περονοφόρου φόρτωσης/εκφόρτωσης φορητών	15%	8.000,00€	8.000,00€	8.000,00€
Αγορά περονοφόρων διακίνησης παλετών	15%	16.000,00€	24.000,00€	32.000,00€
<b>Σύνολο</b>		<b>24.000,00€</b>	<b>32.000,00€</b>	<b>40.000,00€</b>

Πίνακας 4.8.1

Στον παρακάτω πίνακα παρουσιάζονται τα ετήσια λειτουργικά έξοδα της επιχείρησης όσον αφορά στον εξοπλισμό και τους εργαζόμενους. Τα λειτουργικά έξοδα των περονοφόρων αφορούν το κόστος συντήρησης αλλά και το ρεύμα που καταναλώνουν τα περονοφόρα για την κίνησή τους. Σύμφωνα με πληροφορίες που παρείχε η εταιρία που διαθέτει το περονοφόρο ETX 515 της εταιρίας Jungheinrich στην Ελλάδα, τα έξοδα αυτά ανέρχονται ετησίως στο 15% της αξίας του περονοφόρου. Ακόμα, ανάλογα με τον αριθμό των περονοφόρων θα χρειαστεί και αντίστοιχος αριθμός χειριστών. Ο μισθός των χειριστών περονοφόρων έχει προκύψει από το ημερομίσθιο που προβλέπει η συλλογική σύμβαση εργασίας των Χειριστών

Βιομηχανικών Επιχειρήσεων που είναι σε ισχύ από 1/7/2011. Αντίστοιχα, ο μισθός των εργατών/μεταφορέων έχει προκύψει από το ημερομίσθιο που προβλέπει η Εθνική Γενική Συλλογική Σύμβαση που είναι σε ισχύ από 1/7/2011.

<b>Δαπάνες</b>	<b>Scenario 1</b>	<b>Scenario 2</b>	<b>Scenario 3</b>
<b>Λειτουργικά έξοδα περονοφόρων</b>	2.400,00€	3.600,00€	4.800,00€
<b>Μισθοί χειριστών περονοφόρων</b>	25.789,34€	38.684,01€	51.578,68€
<b>Μισθοί εργατών/ μεταφορέων</b>	21.283,38€	21.283,38€	21.283,38€
<b>Σύνολο</b>	<b>49.472,72€</b>	<b>63.567,39€</b>	<b>77.662,06€</b>

Πίνακας 4.8.2

Στον ακόλουθο πίνακα φαίνεται το σύνολο των ετησίων δαπανών για κάθε περίπτωση, όπως αναλύθηκαν προηγουμένως.

<b>Συνολικές Δαπάνες</b>	<b>Scenario 1</b>	<b>Scenario 2</b>	<b>Scenario 3</b>
<b>Αποσβέσεις</b>	3.600,00€	4.800,00€	6.000,00€
<b>Λειτουργικά έξοδα</b>	49.472,72€	63.567,39€	77.662,06€
<b>Σύνολο</b>	<b>53.072,72€</b>	<b>68.367,39€</b>	<b>83.662,06€</b>

Πίνακας 4.8.3

#### 4.9 Συμπεράσματα

Σε αυτή την παράγραφο θα γίνει σύγκριση των τριών σεναρίων που περιγράφηκαν παραπάνω και επιλογή του σεναρίου που ικανοποιεί τις απαιτήσεις της αποθήκης και είναι συγχρόνως οικονομικά συμφέρον.

Ανακεφαλαιώνοντας, έχουν παρουσιαστεί τρεις προτάσεις για τη λειτουργία της αποθήκης. Και στις τρεις περιπτώσεις γίνεται χρήση των ακολούθων:

- ενός περονοφόρου για την εκφόρτωση και τη φόρτωση των φορτηγών
- δύο εργατών/μεταφορέων που είναι υπεύθυνοι για το χειρισμό των επεξεργαστών του τμήματος σύνθεσης παραγγελιών και για τη μεταφορά των κιβωτίων σε θυρίδες

Αυτό που αλλάζει σε κάθε περίπτωση είναι ο αριθμός των περονοφόρων που χρησιμοποιούνται για τη διακίνηση των παλετών στην αποθήκη. Έτσι, στην πρώτη περίπτωση, Scenario 1, γίνεται χρήση ενός περονοφόρου, στη δεύτερη περίπτωση, Scenario 2, γίνεται χρήση δύο περονοφόρων, ενώ στην τρίτη περίπτωση, Scenario 3, γίνεται χρήση τριών περονοφόρων.

Ακόμα έγινε δοκιμή πάνω στη δεύτερη περίπτωση – Scenario 2, που φαινόταν επικρατέστερη από τη διαγραμματική απεικόνιση των αποτελεσμάτων, σχετικά με το κατά πόσο είναι εφικτή η χρήση ενός εργάτη/μεταφορέα αντί για δύο. Το διάγραμμα που προέκυψε έδειξε ότι ο μεταφορέας δεν ήταν καθόλου ανενεργός, ενώ στη διάρκεια αυτής της δοκιμής παρατηρήθηκε ότι το τμήμα σύνθεσης παραγγελιών υπολειπορούσε επειδή δεν μπορούσε να λειτουργήσει επαρκώς με έναν μόνο εργάτη. Έτσι δεν μελετήθηκε περαιτέρω αυτή η περίπτωση αφού κρίθηκε απαραίτητη η χρήση δύο εργατών.

Στον πίνακα που ακολουθεί φαίνονται συγκεντρωτικά κάποια στοιχεία αναφορικά με τις τρεις προτάσεις, που μας βοηθούν στη σύγκρισή τους, όπως το ποσοστό χρησιμοποίησης των περονοφόρων, το ποσοστό χρησιμοποίησης των ραφιών και των θυρίδων, ο μέσος χρόνος σύνθεσης ανά παραγγελία, παρατηρήσεις για την κάθε πρόταση και τέλος το ετήσιο κόστος για την εφαρμογή της.

Έτσι, από τον παρακάτω πίνακα είναι φανερό ότι παρά το μικρό του κόστος το Scenario 1 απορρίπτεται επειδή δεν μπορεί να ανταπεξέλθει στις απαιτήσεις της αποθήκης (ανεπαρκής αριθμός περονοφόρων που έχει σαν αποτέλεσμα να υπολειπορούσε το τμήμα σύνθεσης παραγγελιών και το τμήμα αποστολών της αποθήκης). Ακόμα, η διαφορά στην απόδοση του Scenario 3 σε σύγκριση με το Scenario 2 είναι πολύ μικρή, ενώ αφενός δεν γίνεται σωστή αξιοποίηση των πόρων (μικρό ποσοστό χρησιμοποίησης των περονοφόρων) και αφετέρου το ετήσιο κόστος διαφέρει σημαντικά. Συμπερασματικά προκύπτει ότι για τη σωστή λειτουργία της

αποθήκης πρέπει να υιοθετηθεί το Scenario 2, δηλαδή να γίνει χρήση ενός περονοφόρου φόρτωσης/εκφόρτωσης, δύο περονοφόρων για τις υπόλοιπες μετακινήσεις των παλετών και δύο εργατών/μεταφορέων .

	Scenario 1	Scenario 2	Scenario 3
Ποσοστό χρησιμοποίησης των περονοφόρων	99,11%	79,26%	41,68%
Ποσοστό χρησιμοποίησης των ραφιών	74,30%	66,61%	66,57%
Ποσοστό χρησιμοποίησης των θυρίδων	0,31%	88,53%	89,73%
Μέσος χρόνος σύνθεσης παραγγελίας	1231,8 min	41,4 min	39,8 min
<b>Παρατηρήσεις</b>	<p>Το ποσοστό χρησιμοποίησης του περονοφόρου δηλώνει ότι αυτό αποδίδει το μέγιστο και δεν υπάρχει δυνατότητα για μεγαλύτερη απόδοση. Σε αυτή την περίπτωση η παραμονή των παλετών στις ουρές αναμονής είναι τόσο μεγάλη που τελικά για να αντιμετωπιστεί η συμφόρηση δίνεται προτεραιότητα στην αποθήκευση των παλετών και όχι στη σύνθεση των παραγγελιών. Εκεί οφείλεται και το μεγάλο ποσοστό αξιοποίησης των ραφιών σε αντίθεση με τη μικρή αξιοποίηση των θυρίδων. Αυτό το σενάριο δεν ικανοποιεί τις παρούσες απαιτήσεις της αποθήκης, πόσο μάλλον μία μελλοντική αύξηση των εργασιών της.</p>	<p>Το ποσοστό χρησιμοποίησης των περονοφόρων είναι πολύ ικανοποιητικό. Σε αυτή την περίπτωση οι παλέτες μένουν μόλις 3 λεπτά στις ουρές αναμονής και εκτελείται το ίδιο ικανοποιητικά και η αποθήκευση των παλετών και η σύνθεση παραγγελιών. Το ποσοστό χρησιμοποίησης των ραφιών και των θυρίδων είναι ικανοποιητικό. Αυτό το σενάριο ικανοποιεί τόσο τις παρούσες απαιτήσεις όσο και πιθανή μελλοντική αύξηση των δραστηριοτήτων της αποθήκης.</p>	<p>Το ποσοστό χρησιμοποίησης των περονοφόρων είναι μικρό. Σε αυτή την περίπτωση οι παλέτες μένουν μόλις 2 λεπτά στις ουρές αναμονής και εκτελείται το ίδιο ικανοποιητικά και η αποθήκευση των παλετών και η σύνθεση παραγγελιών. Το ποσοστό χρησιμοποίησης των ραφιών και των θυρίδων είναι ικανοποιητικό. Αυτό το σενάριο ικανοποιεί τις παρούσες απαιτήσεις αλλά και πιθανή μελλοντική αύξηση των δραστηριοτήτων της αποθήκης. Το ετήσιο κόστος όμως σε αυτή την περίπτωση είναι αποτρεπτικό, δεδομένου ότι τα περονοφόρα θα υπολειμθούν.</p>
<b>Κόστος</b>	<b>53.072,72€</b>	<b>68.367,39€</b>	<b>83.662,06€</b>

Πίνακας 4.9.1

## Επίλογος





## **Επίλογος**

Η αλματώδης ανάπτυξη της τεχνολογίας έχει σαν αποτέλεσμα την όλο και μεγαλύτερη διείσδυση των ηλεκτρονικών υπολογιστών σε θέματα που μέχρι πρότινος επιλύονταν αποκλειστικά με τη χρήση μαθηματικών μοντέλων. Έτσι, η παρούσα μελέτη πραγματεύθηκε την προσομοίωση των διαδικασιών μίας αποθήκης με χρήση του λογισμικού FlexSim.

Αρχικά παρατέθηκαν κάποιες εισαγωγικές έννοιες για τα συστήματα, τα μοντέλα και την προσομοίωση και στη συνέχεια παρουσιάστηκαν παραδείγματα χρήσης του λογισμικού. Στη συνέχεια αναλύθηκε το κυρίως μέρος της εργασίας που αφορούσε μία Μελέτη Περίπτωσης, όπου παρουσιάστηκε από την αρχή η δημιουργία του μοντέλου της αποθήκης, το τρέξιμό του και η εξαγωγή αποτελεσμάτων και συμπερασμάτων.

Σχολιάζοντας περαιτέρω την εμπειρία χρήσης της προσομοίωσης με τη βοήθεια ηλεκτρονικού υπολογιστή σαν μέθοδο επίλυσης ενός προβλήματος εφοδιαστικής αλυσίδας, πρόκειται για μία πολύ ενδιαφέρουσα μέθοδο ανάλυσης και πολύ διαφορετική από τις παραδοσιακές μεθόδους. Στα πλεονεκτήματα της μεθόδου περιλαμβάνονται η δυνατότητα οπτικοποίησης της λειτουργίας του συστήματος, η δυνατότητα άμεσης εφαρμογής αλλαγών στο μοντέλο και άμεσης παρατήρησης των αποτελεσμάτων που επιφέρουν, και τέλος η δυνατότητα σύγκρισης εναλλακτικών προτάσεων και εύρεσης της βέλτιστης πριν την εφαρμογή στο πραγματικό σύστημα.

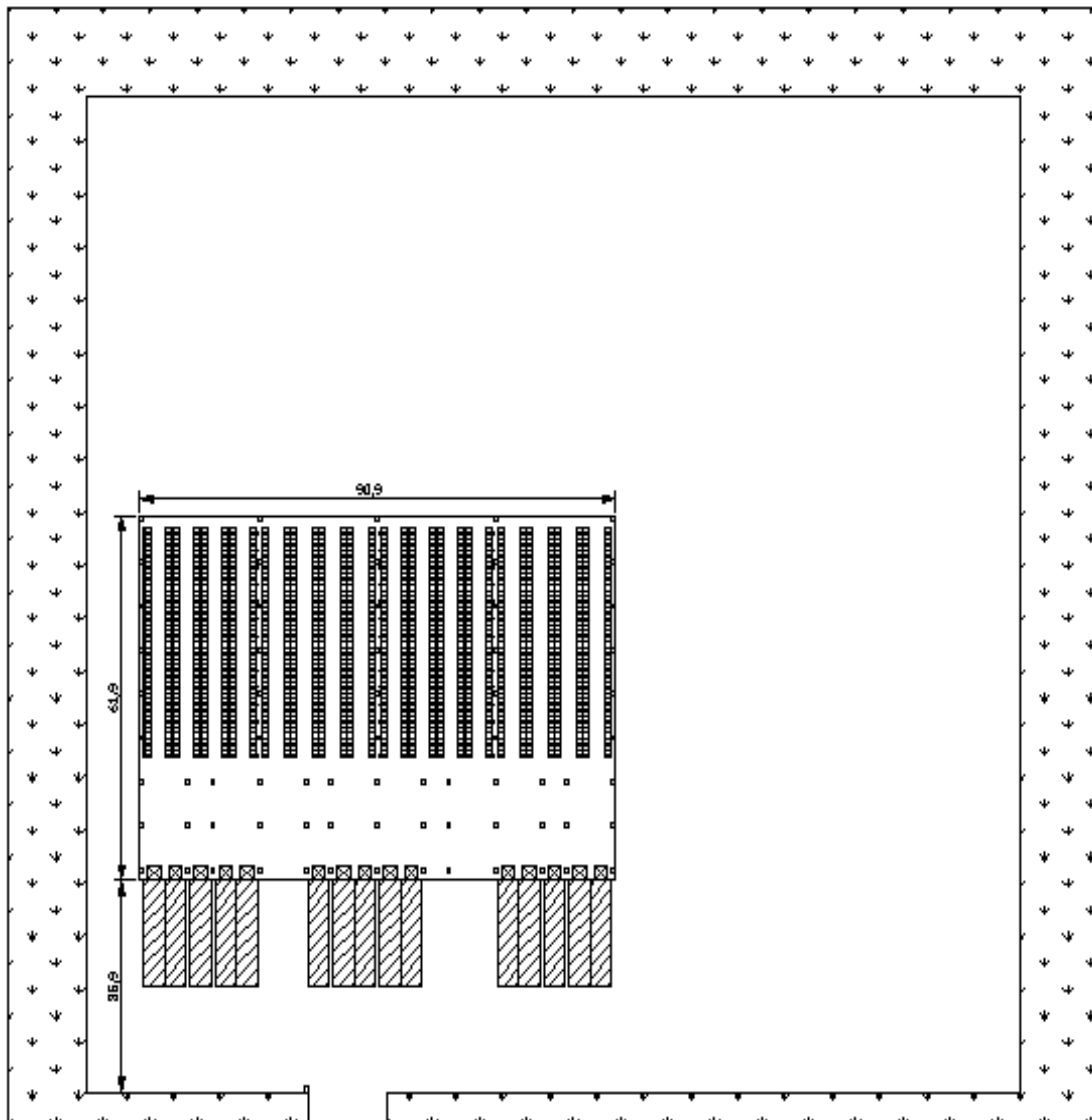
Ειδικότερα, στα πλεονεκτήματα χρήσης του λογισμικού FlexSim περιλαμβάνονται το ιδιαίτερα εξελιγμένο γραφικό περιβάλλον που παρέχει και το γεγονός ότι διευκολύνει το χρήστη παραθέτοντας σε κάθε αντικείμενο λίστες με τις πιο συχνές και χρήσιμες εντολές ενώ παράλληλα αφήνει το περιθώριο να εισάγει ο χρήστης το δικό του κώδικα που εξυπηρετεί τις ανάγκες του επακριβώς. Ωστόσο, υπήρχαν κάποια μειονεκτήματα που δυσχέραιναν τη χρήση του, όπως η έλλειψη επιλογής αναίρεσης, και κάποια εργαλεία που ενώ παρέχονται από το λογισμικό δεν λειτουργούν. Παρόλα αυτά το λογισμικό αναπτύσσεται συνεχώς και επιλύονται προβλήματα προηγούμενων εκδόσεων.

Τέλος, είναι προφανές ότι το μοντέλο που παρουσιάστηκε αντιπροσωπεύει την πραγματικότητα στο βαθμό που αυτό είναι εφικτό. Το μοντέλο και τα εξαγόμενά του αποτελούν εκτιμήσεις και πρέπει να αντιμετωπίζονται ως προσεγγίσεις του υπό μελέτη συστήματος και όχι σαν μία ακριβής αναπαράσταση της συμπεριφοράς του.



## Παράρτημα

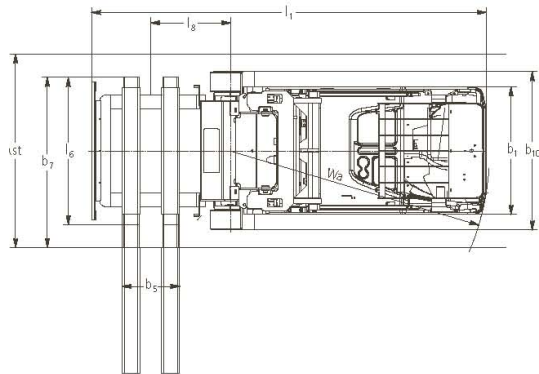
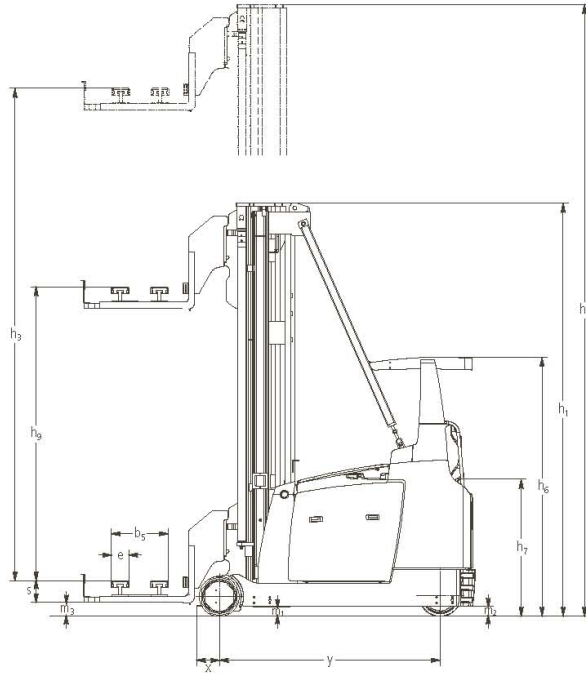


**A.**

Κάτοψη της Αποθήκης

# B.

## ETX 513-515



Standard values for working aisle widths (mm)					
with rail guidance					
Pallet size	Stacking-in depth	Ast	Ast <sub>y</sub> /VDI theoretical ETX 513 T/F	Ast <sub>y</sub> /VDI theoretical ETX 515 T/F	Ast <sub>y</sub> * practical
1200 x 800	1200	1400	3698	3986	+500
with inductive guidance					
Pallet size	Stacking-in depth	Ast	Ast <sub>y</sub> /VDI theoretical ETX 513 TG	Ast <sub>y</sub> /VDI theoretical ETX 515 T/F	Ast <sub>y</sub> * practical
1200 x 800	1200	1450	3698	3986	+1000

\* The practical transfer aisle width is a reference value.

# Technical Data in line with VDI 2198

Identification	1.1	Manufacturer (abbreviation)	Jungheinrich	Jungheinrich	1.1	
	1.2	Manufacturer's type designation	<b>ETX 513</b>	<b>ETX 515</b>	1.2	
	1.3	Drive (electric – battery or mains, diesel, petrol, fuel gas, manual)	electric	electric	1.3	
	1.4	Type of operation (hand, pedestrian, standing, seated, order picker)	bi-lateral stacker	bi-lateral stacker	1.4	
	1.5	Load capacity/ rated load	Q (t)	1.2 <sup>1)</sup>	1.2 <sup>1)</sup>	1.5
	1.6	Load centre distance	c (mm)	600	600	1.6
	1.8	Load distance, centre of drive axle to fork	x (mm)	171	213	1.8
	1.9	Wheelbase	y (mm)	1764	2094	1.9
	Weights	2.1	Service weight incl. battery (see line 6.5)	kg	6540	7530
2.2		Axle loading, laden front/rear	kg	5736/2058	6540/2490	2.2
2.3		Axle loading, unladen front/rear	kg	3810/2730	4350/3180	2.3
Wheels, Chassis	3.1	Tyres (solid rubber, superelastic, pneumatic, polyurethane)	Vulkollan®	Vulkollan®	3.1	
	3.2	Tyre size, front	295x144	380x152	3.2	
	3.3	Tyre size, rear	400x160	400x160	3.3	
	3.5	Wheels, number front/ rear (x = driven wheels)	2/1x	2/1x	3.5	
	3.6	Track width, front	b <sub>10</sub> (mm)	1306	1258	3.6
	4.2	Lowered mast height	h <sub>1</sub> (mm)	3820	3920	4.2
Basic Dimensions	4.4	Lift height	h <sub>3</sub> (mm)	5500	5500	4.4
	4.5	Extended mast height	h <sub>4</sub> (mm)	6650	6750	4.5
	4.7	Overhead load guard (cab) height	h <sub>6</sub> (mm)	2461	2461	4.7
	4.8	Seat height/standing height	h <sub>7</sub> (mm)	1360	1360	4.8
	4.19	Overall length (unladen)	l <sub>1</sub> (mm)	3423	3711	4.21
	4.21	Overall width	b <sub>1</sub> /b <sub>2</sub> (mm)	1210/1210	1210/1210	4.21
	4.22	Fork dimensions	s/e/l (mm)	60/180/1200	60/180/1200	4.22
	4.25	Width over forks	b <sub>5</sub> (mm)	540	540	4.25
	4.29	Reach, lateral	b <sub>7</sub> (mm)	1300	1300	4.29
	4.31	Ground clearance, laden, under mast	m <sub>1</sub> (mm)	20	20	4.31
	4.32	Ground clearance, centre of wheelbase	m <sub>2</sub> (mm)	90	90	4.32
	4.34	Aisle width for pallets 1200x800	Ast (mm)	1400	1400	4.32
	4.35	Turning radius	Wa (mm)	2135	2460	4.35
	4.42	Pallet width	b <sub>12</sub> (mm)	800	800	4.35
	4.43	Pallet length	l <sub>6</sub> (mm)	1200	1200	4.35
4.45	Clear driver compartment height inside	(mm)	1518	1518	4.45	
Performance Data	5.1	Travel speed, laden/ unladen	km/h	10.5/10.5	10.5/10.5	5.1
	5.2	Lift speed, laden/ unladen	m/s	0.45/0.46	0.45/0.46	5.2
	5.3	Lowering speed, laden/ unladen	m/s	0.48/0.48	0.48/0.48	5.3
	5.4	Reach speed, laden/ unladen	m/s	0.25/0.25	0.25/0.25	5.3
	5.10	Service brake		regenerative/ hydraulic	regenerative/ hydraulic	5.10
	5.11	Parking brake		electric spring-loaded	electric spring-loaded	5.11
E-Motor	6.1	Drive motor rating S <sub>2</sub> 60 min.	kW	7	7	6.1
	6.2	Lift motor rating at S <sub>3</sub> 15 %	kW	21	21	6.2
	6.3	Battery acc. to DIN 43531/35/36 A, B, C, no		4 EPzS 420	5 EPzS 700	6.3
	6.4	Battery voltage, nominal capacity K <sub>s</sub>	V/Ah	80/420	80/700	6.4
	6.5	Battery weight	kg	1238	1863	6.5
Others	8.1	Type of drive control		3-phase AC drive control	3-phase AC drive control	8.1
	8.4	Sound level at driver's ear according to EN 12053	dB(A)	73	73	8.4
	8.6	Steering		electric	electric	8.6
1) at FT 1000 kg						

This specification sheet according to VDI regulation 2198 only provides technical values for the standard truck. Non-standard tyres, different masts, additional equipment, etc. could produce other values. Right reserved for technical changes and improvements.





## Βιβλιογραφία



Altiok T. and Melamed B. (2007), *Simulation Modeling and Analysis with ARENA*, USA: Elsevier.

Banks, J. (1998), *Handbook of Simulation: Principles, Methodology, Advances, Applications, and Practice*, New York: Wiley.

Banks J., Carson J., Nelson B. and Nicol D. (2000), *Discrete - Event System Simulation*, Upper Saddle River, NJ: Prentice Hall.

Fishwick, P. (1995), *Simulation Model Design and Execution: Building Digital Worlds*, Upper Saddle River, NJ: Prentice Hall.

Garrido, J. (2009), *Object Orientated Simulation: A Modeling and Programming Perspective*, New York: Springer.

Law A. and Kelton D. (2000), *Simulation Modeling and Analysis*, USA: McGraw-Hill.

Sokolowski J. and Banks C. (2009), *Principles of Modeling and Simulation: A Multidisciplinary Approach*, New York: Wiley.

Pidd, M. (2009), *Systems Modeling: Theory and Practice*, New York: Wiley.

Zeigler B., Praehofer H. and Kim T.G. (2000), *Theory of Modeling and Simulation: Integrating Discrete Event and Continuous Complex Dynamic Systems*, USA: Academic Press.





