



Πανεπιστήμιο Πειραιώς
Τμήμα Ψηφιακών Συστημάτων
Π.Μ.Σ. “Ψηφιακά Συστήματα & Υπηρεσίες”

Διπλωματική Εργασία

Σχεδιασμός & Ανάπτυξη Ολοκληρωμένου Πληροφοριακού
Συστήματος επί της Διαδικασίας Ναυλώσεων

της Γώτα Κωνσταντίνας



Επιβλέπων : Γεώργιος Βασιλακόπουλος
Καθηγητής του Τμήματος Ψηφιακών Συστημάτων

Πειραιάς, Μάρτιος 2016



Πανεπιστήμιο Πειραιώς
Τμήμα Ψηφιακών Συστημάτων
Π.Μ.Σ. “Ψηφιακά Συστήματα & Υπηρεσίες”

Διπλωματική Εργασία

Σχεδιασμός & Ανάπτυξη Ολοκληρωμένου Πληροφοριακού
Συστήματος επί της Διαδικασίας Ναυλώσεων

της Γώτα Κωνσταντίνας (ΑΜ ΜΕ14023)

Επιβλέπων : Γεώργιος Βασιλακόπουλος
Καθηγητής του Τμήματος Ψηφιακών Συστημάτων

Πειραιάς, Φεβρουάριος 2016

.....
Γώτα Κωνσταντίνα

Διπλωματούχος Μηχανικός Ηλεκτρονικών

Μέσων και Τηλεπικοινωνιών

Σχολή Ικάρων - Μηχανικοί (ΣΜΑ)

(C) 2016 - Με την επιφύλαξη παντός δικαιώματος. All rights reserved.

Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας, εξ' ολοκλήρου ή τμήματος αυτής, για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, εκπαιδευτικής ή ερευνητικής φύσης, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να διατηρείται το παρόν μήνυμα. Ερωτήματα που αφορούν τη χρήση της εργασίας για κερδοσκοπικό σκοπό πρέπει να απευθύνονται προς τον συγγραφέα.

Οι απόψεις και τα συμπεράσματα που περιέχονται σε αυτό το έγγραφο εκφράζουν το συγγραφέα και δεν πρέπει να ερμηνευθεί ότι αντιπροσωπεύουν τις επίσημες θέσεις του Πανεπιστημίου Πειραιώς.

“There is no elevator to reach the peak, you have to take the stairs.”

Unknown

Ευχαριστίες

Σε αυτό το σημείο θα ήθελα να ευχαριστήσω όλους εκείνους που συνέβαλλαν και βοήθησαν στην πραγματοποίηση αυτής της διπλωματικής εργασίας.

Θα ήθελα αρχικά να ευχαριστήσω τον επιβλέποντα καθηγητή, κ. Γεώργιο Βασιλακόπουλο, που μου έδωσε τη δυνατότητα να ασχοληθώ με ένα τόσο ενδιαφέρον και σύγχρονο θέμα, καθώς επίσης και τη συνολική υποστήριξή του. Θα ήθελα, επίσης, να ευχαριστήσω την Διδάκτορα και συνεπιβλέποντα Δρ. Πουλυμενοπούλου Μικαέλα για τη συνεχή και αποτελεσματική βοήθειά της, καθώς και την αμέριστη συμπαράστασή της καθ' όλη τη διάρκεια εκπόνησης της διπλωματικής εργασίας. Επίσης ευχαριστώ τα υπόλοιπα μέλη της επιτροπής, τον κ. Βασιλακόπουλο, τον κ. Θεμιστοκλέους και την κα. Πρέντζα. Τέλος, θα ήθελα να ευχαριστήσω όλους εκείνους που έβαλαν έστω και ένα μικρό λιθαράκι στην περάτωση της διπλωματικής εργασίας και την οικογένεια μου για τη συμπαράσταση και την υπομονή της.

Περίληψη

Ο σκοπός της διπλωματικής εργασίας ήταν η μελέτη σχεδίασης, η ανάπτυξη και αξιολόγηση ενός ολοκληρωμένου πληροφοριακού συστήματος για τη διαδικασία ναυλώσεων. Η παρούσα διπλωματική εργασία έρχεται να αντιμετωπίσει τα πιο σημαντικά από τα ζητήματα υλοποίησης ναυλώσεων των πλοίων των ναυτιλιακών εταιρειών. Κύριο μέλημα της εργασίας ήταν αρχικά να εντοπιστούν καταρχήν τα τεχνολογικά ζητήματα εφαρμογής ενός τέτοιου πληροφοριακού συστήματος και να προταθούν τεχνολογικές προσεγγίσεις για την αντιμετώπισή τους.

Λέξεις - Κλειδιά: Κατανεμημένα Ναυτιλιακά Πληροφοριακά Συστήματα, Ναυλώσεις, Ηλεκτρονική Διαδικασία Ναυλώσεων, e-Brokering.

Abstract

The scope of this thesis was the design and the implementation of a WEB- Based Common Brokering Information System over the Shipping section. The present thesis deals with the more significant issues related to the implementation of a global e-Brokering procedure. Main focus of the thesis was to initially determine the technological aspects of the implementation of a unified Brokering System and to address those aspects by proposing technological approaches to deal with them. Emphasis was given in the architectural design for the system by combining current technological architectures based on a multi-tier architecture and web-based software development.

Keywords: Distributed Shipping Information System, Shipping Brokering, Electronic Brokering Procedure, e-Brokering.

1. Εισαγωγή	17
1.1 Η Έννοια Των Πληροφοριακών Συστημάτων	17
1.1.1 Πληροφοριακά συστήματα	17
1.1.2 Παραδείγματα πληροφοριακών συστημάτων	19
1.1.3 Η Πρόκληση των Πληροφοριακών Συστημάτων	20
1.1.4 Επιπτώσεις των Πληροφοριακών Συστημάτων	21
1.1.5 Πλεονεκτήματα ΠΣ	22
1.1.6 Μειονεκτήματα ΠΣ	24
1.2 Σύνοψη	24
2. Βάσεις Δεδομένων	27
2.1 Η Προσέγγιση των Βάσεων Δεδομένων	27
2.2 Συστήματα Διαχείρισης Βάσεων Δεδομένων	28
2.2.1 Σχεδίαση και Υλοποίηση της Βάσης Δεδομένων	28
2.2.2 Καταχώρηση των Δεδομένων στη Βάση του Συστήματος	30
2.2.3 Διαχείριση των Δεδομένων της Βάσης	31
2.3 Επιπτώσεις της Χρήσης των Βάσεων Δεδομένων	32
2.3.1 Πλεονεκτήματα Βάσεων Δεδομένων	32
2.3.2 Μειονεκτήματα Βάσεων Δεδομένων	33
3. Ανάγκη για Αυτοματοποίηση	35
3.1 Νησίδες Τεχνολογίας	35
3.2 Ανάγκες Ολοκλήρωσης	36
3.3 Εξέλιξη Υπηρεσιών Ολοκλήρωσης	37
3.3.1 Enterprise Resource Planning - ERP	37
3.4 Τεχνολογίες Ενδιάμεσου Λογισμικού	39
3.4.1 Κατηγορίες Ενδιάμεσου Λογισμικού	39
3.4.2. Ενδιάμεσο Λογισμικό Βάσεων Δεδομένων (Database Oriented Middleware)	40
3.4.3 Ενδιάμεσο Λογισμικό Μηνυμάτων (Message Based Technologies)	41
3.5 Τεχνολογία Ολοκλήρωσης Τεχνολογίας Εφαρμογών (Enterprise Application Integration-EAI)	46
3.5.1 Προσεγγίσεις EAI	47
3.5.2 Πλεονεκτήματα EAI	48
3.5.3 Μειονεκτήματα EAI	49

3.5.4 Κόστος EAI	50
3.5.5 Τεχνική Διάσταση της EAI	50
3.6 Υπηρεσιοστρεφείς Αρχιτεκτονικές (Service Oriented Architectures- SOA)	53
4. Μεθοδολογία RUP	57
4.1 Ροές Εργασίας	59
4.1.1 Η Ροή Εργασίας της Μοντελοποιημένης Επιχείρησης	60
4.1.2 Η Ροή Εργασίας των Απαιτήσεων	61
4.1.3 Η Ροή Εργασίας Ανάλυσης και Σχεδιασμού	61
4.1.4 Η Ροή Εργασίας της Υλοποίησης	61
4.1.5 Η Ροή Εργασίας του Ελέγχου	62
4.1.6 Η Ροή Εργασίας της Παράδοσης	62
4.1.7 Η Ροή Εργασίας Διαχείρισης και Διαμόρφωσης Αλλαγών	63
4.1.8 Η Ροή Εργασίας Διαχείρισης Έργου	63
4.1.9 Η Ροή Εργασίας του Περιβάλλοντος	64
4.2 Ενοποιημένη Γλώσσα Μοντελοποίησης - Unified Modeling Language (UML)	64
4.2.1 Τα διαγράμματα της UML	65
5. Ναυτιλία	73
5. 1 Ο ρόλος του Ναυλομεσίτη	74
5.2 Το Ναυλοσύμφωνο ως βασικό Συμβόλαιο Μεταφοράς	75
5.3 Διαδικασία Ναύλωσης (Case Study)	76
5.3.1 Περιγραφή Διαδικασιών και Εντοπισμός του Προβλήματος	77
6. Ανάπτυξη Πληροφοριακού Συστήματος	81
6.1 Υλοποίηση Πληροφοριακού Συστήματος	81
6.2 Σχεδίαση Διαγραμμάτων UML	82
6.2.1 Διάγραμμα Περίπτωση Χρήσης- Use Case Diagram	82
6.2.2 Διάγραμμα Δραστηριότητας - Activity Diagram	85
6.2.3 Διάγραμμα Συστατικών - Component Diagram	87
6.2.4 Διάγραμμα Κλάσεων - Class Diagram	88
6.3 Περιγραφή της Βάσης Δεδομένων	88
6.4 Περιγραφή της Διαδικασίας	92
6.4.1 Περίπτωση Χρήσης - Use Case Study Εύρεση πλοίου και Διαπραγμάτευση	94

6.4.2 Περίπτωση Χρήσης - Use Case Study Έλεγχος Πληροφοριών Πλοίων και Εταιρείας
108

7. Συμπεράσματα 116

8. Βιβλιογραφία 119

Πίνακας 1.1 : Θετικές και αρνητικές επιπτώσεις των Πληροφοριακών Συστημάτων.	26
Εικόνα 2.1: Τα επίπεδα στη Σχεδίαση μιας Βάσης Δεδομένων.	29
Εικόνα 3.1: Η εξέλιξη των Συστημάτων ERP μέσα στο χρόνο.	37
Εικόνα 3.2: ERP Συστήματα - Τεχνολογία.	38
Εικόνα 3.3: Enterprise Middleware Architecture.	40
Εικόνα 3.4: Εφαρμογές Δικτυακού Περιβάλλοντος Internet.	44
Εικόνα 3.5: Το EAI και οι Τεχνολογίες Ενδιάμεσου Λογισμικού.	47
Εικόνα 4.1: Ροές Εργασίας της Μεθοδολογίας RUP.	60
Εικόνα 5.1: Διάγραμμα Περίπτωσης Χρήσης του Ναυλωτή- Charterer.	83
Εικόνα 5.2: Διάγραμμα Περίπτωσης Χρήσης του Ναυλομεσίτη- Chartering Broker.	85
Εικόνα 5.3: Διάγραμμα Δραστηριότητας των Ναυλωτή- Charterer και Ναυλομεσίτη- Chartering Broker.	86
Εικόνα 5.4: Διάγραμμα Συστατικών του Πληροφοριακού Συστήματος.	87
Εικόνα 5.5: Διάγραμμα Κλάσεων του Πληροφοριακού Συστήματος.	88
Εικόνα 5.6: Διάγραμμα Πινάκων της Βάσης Δεδομένων.	89
Πίνακας 5.1: Πίνακας Βάσης Δεδομένων “bonita”. “ships”.	90
Πίνακας 5.2: Πίνακας Βάσης Δεδομένων “bonita”. “users”.	90
Πίνακας 5.3: Πίνακας Βάσης Δεδομένων “bonita”. “ship_personel”.	91
Πίνακας 5.4: Πίνακας Βάσης Δεδομένων “bonita”. “ship_users”.	91
Πίνακας 5.5: Πίνακας Βάσης Δεδομένων “bonita”. “ships_program”.	91
Πίνακας 5.6: Πίνακας Βάσης Δεδομένων “bonita”. “processon”.	92
Εικόνα 5.7: Ο χρήστης Ναυλωτής Favio Riviera εκκινεί τη διαδικασία Searching for a ship.	94
Εικόνα 5.8: Ο χρήστης Ναυλομεσίτης Helen Kelly εκκινεί τις διαδικασίες Monitor information of a Ship - Company & Negotiate.	95
Εικόνα 5.9: Ο χρήστης Ναυλωτής Favio Riviera συμπληρώνει τη φόρμα με τα επιθυμητά κριτήρια.	95
Εικόνα 5.10: Ο χρήστης Ναυλωτής Favio Riviera βλέπει όλα τα διαθέσιμα πλοία για τα κριτήρια που επιθυμεί με λιγότερες λεπτομέρειες.	96
Εικόνα 5.11: Ο χρήστης Ναυλωτής Favio Riviera βλέπει όλα τα διαθέσιμα πλοία για τα κριτήρια που επιθυμεί με περισσότερες λεπτομέρειες και επιλεγμένα τα επιθυμητά πλοία.	96
Εικόνα 5.12: Ο χρήστης Ναυλωτής Favio Riviera έχει στη διάθεση του όλα τα κριτήρια που έθεσε και έχει τη δυνατότητα να συμπληρώσει τις παρατηρήσεις του.	97
Εικόνα 5.13- 5.14 : Ο χρήστης Ναυλομεσίτης Helen Kelly έχει στη διάθεση του όλα τα κριτήρια που έθεσε ο Ναυλωτής και το όνομα του πλοίου που αφορά η αίτηση μεταφοράς.	97

Εικόνα 5.15- 5.16 : Ο χρήστης Ναυλομεσίτης Walter Bates έχει στη διάθεση του όλα τα κριτήρια που έθεσε ο Ναυλωτής και το όνομα του πλοίου που αφορά η αίτηση μεταφοράς. 98

Εικόνα 5.17- 5.18 : Ο χρήστης Ναυλομεσίτης Marc Marseau έχει στη διάθεση του όλα τα κριτήρια που έθεσε ο Ναυλωτής και το όνομα του πλοίου που αφορά η αίτηση μεταφοράς. 98

Εικόνα 5.19 : Ο χρήστης Ναυλομεσίτης Walter Bates έχει απορρίψει την αίτηση και ο Ναυλωτής βλέπει τους λόγους απόρριψης της αίτησης μεταφοράς. 99

Εικόνα 5.20: Ο Ναυλωτής Favio Riviera διαπραγματεύεται την αίτηση μεταφοράς για το Ναυλομεσίτη Marc Marseau. 100

Εικόνα 5.21: Ο Ναυλωτής Favio Riviera αποδέχεται την αίτηση μεταφοράς για το Ναυλομεσίτη Helen Kelly. 101

Εικόνα 5.22 : Ο χρήστης Ναυλομεσίτης Marc Marseau καταγράφει την προσφορά του και την αποστέλλει στο Ναυλωτή. 101

Εικόνα 5.23 : Ο χρήστης Ναυλομεσίτης Marc Marseau λαμβάνει ένα μήνυμα διακοπής της Διαπραγμάτευσης καθώς έχει προηγηθεί μια συμφωνία. Έτσι διακόπτεται και η Διαπραγμάτευση των δύο μερών. 102

Εικόνα 5.24 : Ο χρήστης Ναυλομεσίτης Helen Kelly λαμβάνει την αποδοχή του Ναυλωτή για την αίτηση μεταφοράς καθώς και ένα αίτημα σύνταξης του Ναυλοσύμφωνου. 102

Εικόνα 5.25: Ο Ναυλωτής Favio Riviera λαμβάνει το Ναυλοσύμφωνο του Ναυλομεσίτη Helen Kelly και μπορεί να ζητήσει αλλαγές σε περίπτωση διαφωνίας. 103

Εικόνα 5.25 : Ο χρήστης Ναυλομεσίτης Helen Kelly λαμβάνει τις αλλαγές του Ναυλωτή και το ζανα συντάσσει και το αποστέλλει εκ νέου. 103

Εικόνα 5.26 - 5.27 : Στη περίπτωση που το αποδεχτούν και οι δύο τότε το Ναυλοσύμφωνο αποστέλλεται από κοινού και στα δύο μέρη όπου μπορούν να το “κατεβάσουν” σε τοπικό φάκελο για αποθήκευση. 104

Εικόνα 5.28 : Ο χρήστης Favio Riviera βλέπει το ιστορικό της επικοινωνίας με το Ναυλομεσίτη Walter Bates. 104

Εικόνα 5.29 : Ο χρήστης Marc Marseau βλέπει το ιστορικό της επικοινωνίας με το Favio Riviera. 105

Εικόνα 5.30 : Ο χρήστης Helen Kelly βλέπει το ιστορικό της επικοινωνίας με το Favio Riviera. 106

Εικόνα 5.30 : Ο χρήστης Favio Riviera μπορεί να δει με ποιους είχε αρχίσει την επικοινωνία για Διαπραγμάτευση. 107

Εικόνα 5.31 : Ο χρήστης Favio Riviera μπορεί να δει όλες τις διαδικασίες που εκτελέστηκαν κατά τη διαδικασία της Διαπραγμάτευσης. 107

Εικόνα 5.32 : Ο χρήστης Helen Kelly μπορεί να εκκινήσει τις παραπάνω διαδικασίες. 108

Εικόνα 5.34 : Ο χρήστης Thomas Wallis μπορεί να επιλέξει μία από τις παραπάνω ενέργειες. 109

- Εικόνα 5.35 : Ο χρήστης Walter Bates έχει επιλέξει να δει το πρόγραμμα των πλοίων της εταιρείας του. 109
- Εικόνα 5.36 : Ο χρήστης Thomas Wallis έχει επιλέξει να δει το προσωπικό με το οποίο είναι επανδρωμένα τα πλοία της εταιρείας του και να κάνει αλλαγές. 110
- Εικόνα 5.37 : Ο χρήστης Thomas Wallis έχει επιλέξει να αλλάξει τα ονόματα των Α και Β Καπετάνιων. 110
- Εικόνα 5.38 : Ο χρήστης Walter Bates επιτυχώς έχει αλλάξει τα ονόματα του προσωπικού. 111
- Εικόνα 5.39: Ο χρήστης Walter Bates μπορεί να επιλέξει μια από τις παραπάνω ενέργειες. 111
- Εικόνα 5.40 : Ο χρήστης Walter Bates επιλέγει το πλοίο και εισάγει τα δεδομένα που επιθυμεί να αλλάξει στο πρόγραμμα των πλοίων από μία συμφωνία εκτός εφαρμογής. 112
- Εικόνα 5.41 : Ο χρήστης Walter Bates βλέπει τις αλλαγές που έγιναν στο πρόγραμμα των πλοίων. 112
- Εικόνα 5.42 : Η εισαγωγή στη Βάση Δεδομένων. 113
- Εικόνα 5.43 : Ο χρήστης Walter Bates επιλέγει ποιο πλοίο θέλει να αφαιρέσει από το πρόγραμμα. 113
- Εικόνα 5.44 : Ο χρήστης Walter Bates έχει με επιτυχία διαγράψει την εγγραφή από τη Βάση Δεδομένων. 114
- Εικόνα 5.45 : Ο χρήστης Helen Kelly έχει με επιτυχία ολοκληρώσει τη διαπραγμάτευση και επιλέγει να αλλάξει τα στοιχεία ενός πλοίου από τη συμφωνία που επιτεύχθηκε μέσω της εφαρμογής. 114
- Εικόνα 5.46 : Ο χρήστης Helen Kelly έχει με επιτυχία ολοκληρώσει την εισαγωγή του νέου ταξιδιού του πλοίου στο πρόγραμμα πλοίων. 115
- Εικόνα 5.47 : Η επιτυχής εισαγωγή των δεδομένων απεικονίζεται και στο πίνακα της Βάσης Δεδομένων. 115

1. Εισαγωγή

Για την αποδοτικότερη διοίκηση μιας επιχείρησης απαραίτητη είναι η ταχεία και αποτελεσματική λήψη αποφάσεων, οι οποίες θα επιτρέψουν σταθερή ανοδική πορεία στην εξέλιξη της επιχείρησης. Οι περισσότεροι οργανισμοί σήμερα είναι δέσμιοι των πληροφοριακών συστημάτων, καθώς η λειτουργία τους και η ύπαρξή τους είναι εξαρτημένη από το Πληροφοριακό Σύστημα το οποίο χρησιμοποιούν. Οι σημαντικότεροι παράγοντες που έκαναν επιτακτική τη χρήση πληροφοριακών συστημάτων είναι: η παγκοσμιοποίηση, η άνοδος της οικονομίας της πληροφορίας, ο μετασχηματισμός της επιχείρησης και η εμφάνιση της ψηφιακής επιχείρησης. Η οικονομία πλέον παύει να βασίζεται στην βιομηχανία κατά κύριο λόγο και βασίζεται πλέον στη γνώση και στην πληροφορία. Στη νέα ψηφιακή επιχείρηση οι βασικές λειτουργίες αλλά και οι σχέσεις γίνονται και ελέγχονται ψηφιακά. Στις ψηφιακές επιχειρήσεις οι τεχνολογίες πληροφορικής δεν είναι απλά βοηθητικές αλλά αποτελούν τη βάση για τη λειτουργία αυτών των επιχειρήσεων.

1.1 Η Έννοια Των Πληροφοριακών Συστημάτων

1.1.1 Πληροφοριακά συστήματα

Πληροφορία (Information) είναι κάθε μορφή επικοινωνίας, η οποία παρέχει κατανοητή και χρήσιμη γνώση στο πρόσωπο που την λαμβάνει (Tuomi, 1999-2000).

Πληροφοριακό σύστημα θεωρείται ένα οργανωμένο σύνολο από ανθρώπους, μηχανές και διαφορά μέσα, το οποίο έχει καθορισμένους σκοπούς και πάντοτε αποτελείται από τρία τμήματα:

- την είσοδο (input)
- την επεξεργασία (processing) και
- την έξοδο (output)

Γενικά ένα Πληροφοριακό Σύστημα (ΠΣ) είναι ένα τυποποιημένο σύστημα αλλαγής, διατηρήσεως στοιχείων (input data), επεξεργασίας αυτών (processing) και έκδοσης αποτελεσμάτων (output) με την μορφή αναφορών, καταστάσεων, εικόνας σε οθόνη, για την ικανοποίηση των αναγκών της επιχείρησης σε πληροφορίες (Brynjolfsson, Malone, Gurbaxani, Kambil, 1994).

Πληροφοριακό Σύστημα (Π.Σ) (Information System) είναι ένα σύνολο από αλληλεπιδρούσες συνιστώσες που συλλειτουργούν και συνεργάζονται για την συλλογή, επεξεργασία, αποθήκευση και διανομή της πληροφορίας με τελικό στόχο τη δημιουργία

πληροφοριών, που είναι αναγκαίες ή/και χρήσιμες σε ένα Οργανισμό ή μία Επιχείρηση, προκειμένου να επιτελέσει τους σκοπούς της, με στόχο την αποτελεσματικότερη λήψη αποφάσεων και εφαρμογή ελέγχου. Σε μία πιο ελεύθερη και απλοϊκή ερμηνεία, θα μπορούσαμε να πούμε ότι Πληροφοριακό Σύστημα είναι το σύστημα εκείνο, που δέχεται σαν είσοδο δεδομένα τα οποία επεξεργάζεται και τα αποδίδει στην έξοδο ως πληροφορίες. Πέρα από την υποστήριξη στη λήψη αποφάσεων, στο συντονισμό και στον έλεγχο, τα πληροφοριακά συστήματα μπορούν επίσης να βοηθούν τα στελέχη και το προσωπικό στην ανάλυση προβλημάτων, στην απεικόνιση σύνθετων θεμάτων και στη δημιουργία νέων προϊόντων.

Κάθε πληροφοριακό σύστημα πρέπει να προσδιορίζει αποδοτικά και αποτελεσματικά τις ανθρώπινες ανάγκες αυτών που χρησιμοποιούν το πληροφοριακό σύστημα, καθώς επίσης και να επεξεργάζεται όλες τις πληροφορίες, με αποτέλεσμα την ικανοποίηση των αναγκών αυτών.

Αυτό γίνεται πραγματικότητα με:

- την πιο αποτελεσματική ανάκτηση, αποθήκευση, επεξεργασία, παρουσίαση και διάδοση των πληροφοριών,
- την παροχή των απαραίτητων μέσων και του κατάλληλου περιβάλλοντος μάθησης στους εμπλεκόμενους χρήστες, ώστε να βελτιωθεί η αποτελεσματικότητα της διαδικασίας λήψης απόφασης,
- την υποστήριξη των διαδικασιών λειτουργίας, ελέγχου και στρατηγικού σχεδιασμού την επιχείρησης ή του οργανισμού.

Τέσσερις είναι οι δραστηριότητες με τις οποίες ένα πληροφοριακό σύστημα παράγει τις πληροφορίες που χρειάζεται ο οργανισμός για να παίρνει αποφάσεις, να ελέγχει λειτουργίες, να αναλύει προβλήματα και να δημιουργεί νέα προϊόντα ή υπηρεσίες. Αυτές οι δραστηριότητες είναι:

- Συλλογή δεδομένων. Τα δεδομένα συλλέγονται από διάφορες πηγές
 - από εσωτερικές πηγές (internal sources)
 - από εξωτερικές πηγές (external sources)
 - από το περιβάλλον
- Αποθήκευση δεδομένων. Με την αποθήκευση, τα δεδομένα φυλάσσονται με έναν οργανωμένο τρόπο για μελλοντική χρήση.
- Επεξεργασία δεδομένων. Η επεξεργασία των δεδομένων περιλαμβάνει υπολογισμούς, συγκρίσεις, ταξινομήσεις και κατηγοριοποιήσεις.
- Διάδοση πληροφοριών. Ο στόχος ενός Π.Σ είναι η διάδοση πληροφοριών. Η πληροφορία μπορεί να διαδοθεί σε διάφορες μορφές (μηνύματα, φόρμες, αναφορές, λίστες, γραφήματα, κλπ).
- Άνθρωποι: Το σύνολο των ανθρώπων που εργάζονται με το πληροφοριακό σύστημα, σε διάφορους ρόλους όπως χρήστες, διαχειριστές κ.τ.λ.

- Διαδικασίες: Το σύνολο των οδηγιών για τη χρήση και το συνδυασμό όλων των στοιχείων υποδομής ενός ΠΣ.
- Βάση Δεδομένων (Database)
- Λογισμικό (Software)
- Υλικός εξοπλισμός (Hardware)
- Δίκτυο (Network)

Όπως αναφέραμε και παραπάνω, τα πληροφοριακά συστήματα συλλέγουν, αποθηκεύουν, μεταδίδουν και επεξεργάζονται δεδομένα για την παροχή χρήσιμων, ολοκληρωμένων και έγκαιρων πληροφοριών, όπου αυτό είναι απαραίτητο. Τα πληροφοριακά συστήματα χρησιμοποιούνται από τις επιχειρήσεις για τους ακόλουθους λόγους:

- Για ταχύτατη και ακριβή επεξεργασία των δεδομένων
- Λόγω μεγάλης αποθηκευτικής ικανότητας
- Επιτρέπουν ταχύτατη επικοινωνία μεταξύ τοποθεσιών
- Προσφέρουν άμεση πρόσβαση σε πληροφορίες, που πρέπει να αντλήσει η επιχείρηση για την δραστηριότητά της
- Λόγω δυνατότητας συντονισμού ατόμων, ομάδων και οργανισμών
- Για την υποστήριξη των αποφάσεων που θα ληφθούν από την επιχείρηση
- Για αυτοματοποίηση και βελτίωση των διαδικασιών και των ροών εργασιών
- Για καλύτερη αξιοποίηση των πολύτιμων δεδομένων της επιχείρησης
- Για την αύξηση της αποτελεσματικότητας της επιχείρησης

1.1.2 Παραδείγματα πληροφοριακών συστημάτων

Υπάρχουν πολλά είδη πληροφοριακών συστημάτων που μπορούν να χρησιμοποιηθούν ανάλογα με τις ανάγκες και τις οικονομικές δυνατότητες της επιχείρησης. Τα σημαντικότερα πληροφοριακά συστήματα είναι τα εξής:

- SCMS (Supplier and Contract Management System / Συστήματα Διαχείρισης Αλυσίδας Εφοδιασμού)
- KMS (Knowledge Management Systems / Συστήματα Διαχείρισης Γνώσης)
- OAS (Office Automation Systems / Συστήματα Αυτοματοποίησης Γραφείου)
- TPS (Transaction Processing Systems / Συστήματα Επεξεργασίας Συναλλαγών)
- ERP (Enterprise resource planning / Συστήματα Ενδοεπιχειρησιακού Σχεδιασμού)
- ESS (Executive Support Systems / Συστήματα Υποστήριξης Διοίκησης)
- DSS (Decision Support Systems / Συστήματα Υποστήριξης Απόφασης)

- MIS (Management Information Systems / Διοικητικά Συστήματα Πληροφόρησης)

Τα πληροφοριακά συστήματα χρειάζονται επίσης αναπληροφόρηση (feedback), η οποία είναι έξοδος, που επιστρέφει στα κατάλληλα μέλη του οργανισμού για να τα βοηθήσει να αξιολογήσουν ή να διορθώσουν τη φάση της εισόδου. Τα πληροφοριακά συστήματα αποτελούν οργανωμένα σύνολα τα οποία περιλαμβάνουν έξι στοιχεία:

Οι Τεχνολογίες Πληροφορικής ή καλύτερα Τεχνολογίες Πληροφοριών (Information Technologies) είναι εργαλεία και τεχνικές οι οποίες υποστηρίζουν την ανάπτυξη των πληροφοριακών συστημάτων (information systems development). Λογισμικό, εξοπλισμός, τηλεπικοινωνίες κ.λπ. είναι μερικά από τα ανωτέρω εργαλεία και τεχνικές (Avison, 2003).

Από επιχειρηματική σκοπιά, ένα πληροφοριακό σύστημα τόσο για τις επιχειρήσεις γενικότερα, όσο και για εκείνες του λιανικού εμπορίου ειδικότερα, είναι ένα χρήσιμο εργαλείο για τη δημιουργία αξίας για την επιχείρηση. Τα πληροφοριακά συστήματα επιτρέπουν στην επιχείρηση να αυξήσει τα έσοδά της ή να μειώσει το κόστος προσφέροντας πληροφορίες που βοηθούν τους μάνατζερ να παίρνουν καλύτερες αποφάσεις ή να βελτιώνουν την εκτέλεση των επιχειρηματικών διεργασιών. Για παράδειγμα, ένα πληροφοριακό σύστημα για την ανάλυση των δεδομένων από τις ταμειακές μηχανές καταστημάτων που δραστηριοποιούνται στον κλάδο του λιανικού εμπορίου, μπορεί να αυξήσει την κερδοφορία της εταιρείας, βοηθώντας τους μάνατζερ να παίρνουν καλύτερες αποφάσεις σχετικά με το ποια προϊόντα να προωθήσουν στα καταστήματα της αλυσίδας με αποτέλεσμα να αυξάνουν την επιχειρηματική αξία (Gallupe, 2000).

1.1.3 Η Πρόκληση των Πληροφοριακών Συστημάτων

Αν και η τεχνολογία των πληροφοριών αναπτύσσεται με ιλιγγιώδη ταχύτητα, η ανάπτυξη και η χρήση των πληροφοριακών συστημάτων δεν είναι κάτι εύκολο ή μηχανικό. Τα στελέχη των επιχειρήσεων αντιμετωπίζουν πέντε βασικές προκλήσεις (Davern, 2000: Hartman, 2002: Orlikowski, 2001):

1. Η πρόκληση της επένδυσης σε πληροφοριακά συστήματα: Είναι φανερό ότι μια από τις μεγαλύτερες προκλήσεις που αντιμετωπίζουν οι μάνατζερ σήμερα είναι να εξασφαλίσουν ότι οι εταιρείες τους αποκομίζουν πράγματι ικανοποιητικές αποδόσεις από τα χρήματα που δαπανούν σε πληροφοριακά συστήματα. Οι περισσότερες εταιρείες δεν διαθέτουν μια ξεκάθαρη διαδικασία λήψης αποφάσεων για να αποφασίζουν ποιες τεχνολογικές επενδύσεις να επιδιώξουν και πως να τις διαχειριστούν.
2. Η στρατηγική επιχειρηματική πρόκληση: Για να επωφεληθούν από την τεχνολογία πληροφοριών, να αποκομίσουν πραγματικά πλεονεκτήματα παραγωγικότητας και να γίνουν ανταγωνιστικοί και αποτελεσματικοί, πολλοί οργανισμοί χρειάζονται να επανασχεδιαστούν.

3. Η πρόκληση της παγκοσμιοποίησης: Η γρήγορη ανάπτυξη του διεθνούς εμπορίου και η εξέλιξη της παγκόσμιας οικονομίας απαιτούν πληροφοριακά συστήματα τα οποία να υποστηρίζουν την παραγωγή και πώληση αγαθών σε πολλές διαφορετικές χώρες.
4. Η πρόκληση της αρχιτεκτονικής και της υποδομής των πληροφοριών: Η αντιμετώπιση νέων επιχειρηματικών και τεχνολογικών προκλήσεων μπορεί να απαιτεί ανασχεδιασμό του οργανισμού και οικοδόμηση μιας νέας αρχιτεκτονικής πληροφοριών και υποδομής τεχνολογίας πληροφοριών.
5. Η πρόκληση της ευθύνης και του ελέγχου: Μια από τις πιο σημαντικές προκλήσεις για τα στελέχη είναι να παίρνουν εμπεριστατωμένες αποφάσεις, οι οποίες να λαμβάνουν υπόψη τις αρνητικές αλλά και τις θετικές συνέπειες των πληροφοριακών συστημάτων. Θεωρώντας πολλή σημαντική τη δυνατότητα του ανθρώπου να μπορεί να ελέγχει και να καταλαβαίνει τα πληροφοριακά συστήματα, εγείρονται διαφορά θέματα ηθικά και κοινωνικά σχετικά με τη χρήση των πληροφοριακών συστημάτων.

1.1.4 Επιπτώσεις των Πληροφοριακών Συστημάτων

Οι τεχνολογίες του Internet και της ψηφιακής επιχείρησης που καθιστούν ευκολότερη παρά ποτέ τη συγκέντρωση, την ενοποίηση και τη διάδοση πληροφοριών, έχουν διεγείρει νέες ανησυχίες για την ορθή χρήση των πληροφοριών σχετικά με τους πελάτες, την προστασία του προσωπικού απορρήτου των ατόμων και την προστασία της πνευματικής ιδιοκτησίας. Το αρνητικό κοινωνικό κόστος της εισαγωγής τεχνολογιών και πληροφοριακών συστημάτων αρχίζει να διογκώνεται όσο αυξάνεται η δύναμη της τεχνολογίας. Πολλές από αυτές τις αρνητικές κοινωνικές συνέπειες δεν αποτελούν παραβιάσεις ατομικών δικαιωμάτων, ούτε εγκλήματα κατά της ιδιοκτησίας. Ωστόσο μπορούν να αποδειχθούν εξαιρετικά επιβλαβείς σε άτομα, κοινωνίες και πολιτικούς θεσμούς. Οι υπολογιστές μπορούν να καταστρέφουν πολύτιμα στοιχεία του πολιτισμού και της κοινωνίας μας, έστω και αν ταυτόχρονα μας προσφέρουν οφέλη.

Από τη μια, ο διπλασιασμός της υπολογιστικής ισχύος έδωσε τη δυνατότητα στους περισσότερους οργανισμούς να αξιοποιήσουν τα πληροφοριακά συστήματα στις βασικές παραγωγικές τους διεργασίες και να επιτυγχάνουν αποτελέσματα πολύ πιο άμεσα και ασφαλή σε σχέση με τον παραδοσιακό τρόπο. Από την άλλη, η αναδιοργάνωση της εργασίας θεωρείται, κατά κανόνα, από τον κόσμο των πληροφοριακών συστημάτων ως το μεγαλύτερο όφελος της νέας τεχνολογίας πληροφοριών. Πολύ σπανιότερα επισημαίνεται ότι ο ανασχεδιασμός των επιχειρηματικών διεργασιών είναι πιθανό να προκαλέσει την απώλεια εκατομμυρίων θέσεων εργασίας μεσαίων στελεχών και υπαλληλικού προσωπικού. Ένας οικονομολόγος ανέφερε την πιθανότητα να δημιουργήσουμε μια κοινωνία που θα διευθύνεται από μια μικρή «ελίτ επαγγελματιών υψηλής τεχνολογίας... σε ένα έθνος μονιμά ανέργων» (Rifkin, 1993). Άλλοι οικονομολόγοι είναι πολύ πιο αισιόδοξοι για τις πιθανές απώλειες θέσεων εργασίας. Αυτοί πιστεύουν ότι η αποδέσμευση ταλαντούχων και

μορφωμένων εργαζομένων από θέσεις που καταργούνται λόγω μιας αναδιοργάνωσης θα έχει ως αποτέλεσμα να μετακινηθούν αυτοί σε καλύτερες θέσεις γρηγορά αναπτυσσομένων κλαδών. Αυτοί οι υπολογισμοί όμως αγνοούν το εργατοτεχνικό προσωπικό και τα μεγαλύτερης ηλικίας και λιγότερο εκπαιδευμένα μεσαία στελέχη. Ο προσεκτικός σχεδιασμός και η ευαισθησία για τις ανάγκες των εργαζομένων μπορεί να βοηθήσουν τις επιχειρήσεις να επανασχεδιάσουν την εργασία ελαχιστοποιώντας όσο είναι δυνατό τις απώλειες θέσεων εργασίας.

Η πρόοδος της τεχνολογίας έχει βοηθήσει στην βελτίωση τεχνικών αποθήκευσης και ανάλυσης δεδομένων. Με τη βοήθεια της σύγχρονης τεχνολογίας πληροφοριακών συστημάτων, οι εταιρείες είναι σε θέση να συναρμολογούν και να συνδυάζουν πληθώρα αποσπασματικών πληροφοριών για τον καθένα από εμάς. Οι πληροφορίες αυτές «γεννιούνται» από τη χρήση πιστωτικών καρτών, τηλεφωνικών κλήσεων, αγορών από καταλόγους, τραπεζικές συναλλαγές και πολλές άλλες δραστηριότητες της καθημερινής μας ζωής. Αν αυτές οι πληροφορίες συνδυαστούν και αξιοποιηθούν καταλληλά, μπορούν να αποκαλύψουν όχι μόνο την πιστοληπτική ικανότητα των ανθρώπων αλλά και άλλες πολύτιμες πληροφορίες. Εταιρείες που διαθέτουν προϊόντα για πώληση, όπως είναι αυτές που ανήκουν στον κλάδο του λιανικού εμπορίου, αγοράζουν τις σχετικές πληροφορίες από πηγές που τις διαθέτουν, ώστε να διευκολύνονται στην καλύτερη στόχευση των εκστρατειών μάρκετινγκ. Έτσι δημιουργείται το προφίλ των πελατών και βοηθά τις εταιρείες να κάνουν με μεγαλύτερη σιγουριά τα επόμενα στρατηγικά τους βήματα.

Στον αντίποδα, τα συστήματα αυτά έχουν τη δυνατότητα να παρακολουθούν, να συλλαμβάνουν και να αποθηκεύουν δεδομένα γεγονός που οδηγεί εύλογα σε διαφορά ζητήματα ηθικής. Οι επιχειρήσεις λιανικών πωλήσεων στον Ιστό έχουν τώρα πρόσβαση σε λογισμικό που τους επιτρέπει να παρακολουθούν την αγοραστική συμπεριφορά ατόμων και ομάδων στο δίκτυο ενώ επισκέπτονται μια τοποθεσία στον Ιστό και κάνουν αγορές. Η εμπορική ζήτηση για τέτοιες προσωπικές πληροφορίες είναι πρακτικά ακόρεστη. Με αλλά λόγια οι εταιρείες κάνοντας χρήση διαφόρων τεχνολογιών (cookies, ιστοκοριοί, κατασκοπευτικά λογισμικά) παραβιάζουν την ιδιωτικότητα και το προσωπικό απόρρητο (Dejoie R, Fowler G, Paradise D, 1991).

1.1.5 Πλεονεκτήματα ΠΣ

Είναι γεγονός πως στην εποχή της ευρυζωνικότητας που μας διέπει, τα πληροφοριακά συστήματα παρέχουν στις επιχειρήσεις ποικίλα οφέλη και υπηρεσίες. Κάποια από αυτά τα οφέλη που προσκομίζει η εταιρεία, είναι η ταχύτερη και ακριβής επεξεργασία δεδομένων, η μεγάλη αποθηκευτική ικανότητα και η ταχύτερη επικοινωνία μεταξύ των τοποθεσιών. Επιπρόσθετα, με τη χρήση πληροφοριακών συστημάτων, παρέχεται η δυνατότητα καλύτερου συντονισμού ατόμων, ομάδων και υπηρεσιών, υποστήριξη αποφάσεων, αυτοματοποίηση και βελτίωση της ροής των εργασιών, αύξηση της

αποτελεσματικότητας του Οργανισμού και καλύτερη αξιοποίηση των πολύτιμων δεδομένων του.

Αξίζει, ωστόσο, να γίνει μια πιο ενδελεχής ανάλυση, που να έγκειται στα πλεονεκτήματα που προσφέρει το κάθε ΠΣ:

- Αρχικά, τα SCMS συμβάλλουν στην ελαχιστοποίηση των αποθεμάτων, στην αυτοματοποίηση των παραγγελιών από τους προμηθευτές και στην βελτίωση του προγραμματισμού των διαδικασιών της. Προσφέρουν, επίσης, καλύτερη αναζήτηση και επιλογή προμηθευτών και παράδοση των προϊόντων και των υπηρεσιών. Τμήμα των SCMS αποτελούν τα CSM (Συστήματα Διαχείρισης Εφοδιαστικής Αλυσίδας), τα οποία παρέχουν επίσης πολλά πλεονεκτήματα, καθώς συγχρονίζουν τις διαδικασίες στην αγορά και στην παραγωγή και διακινούν τα προϊόντα πιο γρήγορα, ολοκληρώνουν τα logistics του προμηθευτή και μειώνουν το χρόνο, την προσπάθεια και το κόστος αποθήκευσης.
- Τα OAS παρέχουν ένα είδος ολοκληρωμένου λογισμικού, που διευκολύνει κατά πολύ τις διαδικασίες μιας επιχείρησης και τον καθορισμό προτεραιοτήτων. Διευκολύνουν την επικοινωνία ανάμεσα στους χρήστες και στους διαφορετικούς χώρους της επιχείρησης. Την ίδια στιγμή, προσφέρουν έναν καταρτισμένο εργονομικό σχεδιασμό, ενώ σύμφωνα με τις περισσότερες αναφορές από εργαζόμενους σε επιχειρήσεις όπου χρησιμοποιήθηκαν τα OAS, παρέχουν και ευχάριστες συνθήκες εργασίας.
- Τα KMS ανακαλύπτουν και κωδικοποιούν τη γνώση, τη διανέμουν και κάνουν εφικτή την πρόσβαση σε αυτή σε όλους, ενώ παράλληλα, δημιουργούν γνώση σχετικά με τις αγοραστικές συνήθειες του καταναλωτικού κοινού.
- Τα TPS διαχειρίζονται τις συναλλαγές της επιχείρησης, διευκολύνοντας έτσι τον συντονισμό των εργασιών, καθώς παρέχουν εξειδικευμένες και λεπτομερειακές αναφορές στα ανώτερα στελέχη του Οργανισμού. Τα Συστήματα Υποστήριξης Επιτελικών Στελεχών στηρίζονται στη διαλογική επεξεργασία και καθορίζουν τις προβολές τους μέσω συγκεντρωτικών αναφορών.
- Τα Συστήματα Υποστήριξης Διοίκησης (ESS) καταφέρνουν να κρατούν ενήμερη τη διοίκηση και σε επαφή με τα υπόλοιπα στελέχη της επιχείρησης.
- Τα DSS συστήματα είναι ευέλικτα, προσαρμόσιμα και γρήγορα, ενώ υποστηρίζουν και τη διαδικασία των αποφάσεων.
- Τα Διοικητικά Συστήματα Πληροφόρησης (MIS) επικεντρώνονται σε διαδικασίες ελέγχου, ενώ συγχρόνως τις εκσυγχρονίζουν, παραδίδοντας και αναλυτικές αναφορές ελέγχου στα ανώτερα στελέχη.
- Τα ERP πληροφοριακά συστήματα εστιάζουν την προσοχή τους στην έγκαιρη και έγκυρη ενημέρωση, ενώ συγχρόνως μειώνουν το κόστος και επιταχύνουν τόσο τις διαδικασίες που εκκρεμούν, όσο και αυτές που ήδη ολοκληρώνονται. Συγκροτούν σε ενιαίο σύνολο τον προγραμματισμό της παραγωγής, των αποθεμάτων της επιχείρησης, τις πωλήσεις που αυτή επιτυγχάνει και το λογιστήριο. Επιπλέον, συγκροτούν μία σταθερή δομή και οργάνωση, δηλαδή έναν Οργανισμό-Επιχείρηση. Η τεχνολογία αποκτά μια Ομοιόμορφη

Ενοποιημένη Υποδομή (δηλαδή πλατφόρμα), οι λειτουργίες γίνονται πιο αποτελεσματικές και οι διαδικασίες της επιχείρησης στρέφονται περισσότερο στον πελάτη (πελατοκεντρική συμπεριφορά). Τέλος, τα ERP βοηθούν στη διαχείριση πολλών τμημάτων της επιχείρησης, όπως της αποθήκης της, της διοίκησης παραγωγής, της διαχείρισης των παγίων και της συντήρησης και επιδιόρθωσης του εξοπλισμού, ενώ επίσης, ρυθμίζουν τις πωλήσεις και τις διανομές, διαχειρίζονται τα παραστατικά, καθορίζουν ενιαία τιμολογιακή πολιτική και διαχειρίζονται τις προμήθειες και τα αποθέματα της.

1.1.6 Μειονεκτήματα ΠΣ

Εκτός όμως από τα πολλά και σημαντικά πλεονεκτήματα που διαθέτουν τα πληροφοριακά συστήματα, αναγκαίο είναι να αναφερθούμε και στα μειονεκτητά τους.

- Αρχικά, το MIS έχει έλλειψη δημιουργικότητας και ποιοτικής πληροφορίας. Τα συστήματα αυτά δεν προσαρμόζονται εύκολα και η αναλυτική ικανότητά τους δεν είναι επαρκής σε σχέση με άλλα συστήματα.
- Όσον αφορά το πληροφοριακό σύστημα DSS, κυρίως λόγω επιβάρυνσης του συστήματος με καινούργιους χρήστες, μειώνεται αισθητά η επίδοσή του, ενώ λειτουργεί μόνο στο ίδιο περιβάλλον της επιχείρησης.
- Ανάλογο μειονέκτημα υπάρχει και στο KMS, καθώς δεν δημιουργεί την εντύπωση ενός αυθεντικού συστήματος, κάτι το οποίο είναι πολύ σημαντικό στη λειτουργία του.
- Τέλος, η ανάπτυξη και η δημιουργία του ERP αποτελεί μια χρονοβόρα διαδικασία, ιδιαίτερα ακριβή, χωρίς να παρέχει ευελιξία στο σύστημα ενώ παράλληλα, εμφανίζει προβλήματα ολοκλήρωσης με άλλα πληροφοριακά συστήματα. Όσον αφορά στον τεχνολογικό τομέα του, υπάρχει μεγάλη δυσκολία στην προσαρμογή στις ανάγκες της κάθε επιχείρησης και χρειάζεται ανά τακτά χρονικά διαστήματα συντήρηση και αναβάθμιση. Επιπλέον, στον οικονομικό τομέα, απαραίτητη κρίνεται η προγενέστερη εκπαίδευση των χρηστών, προκειμένου να μπορέσουν να το χρησιμοποιήσουν, επιβαρύνοντας έτσι τον προϋπολογισμό της επιχείρησης, ενώ η ανάγκη για μακροπρόθεσμη απόδοση της επένδυσης γι' αυτό το σύστημα το καθιστά οικονομικά δαπανηρό.

1.2 Σύνοψη

Είναι σαφές ότι η πρόοδος της τεχνολογίας των πληροφοριακών συστημάτων έχει πολλαπλά οφέλη σε διάφορους τομείς, Βοηθήσαν στην αύξηση της παραγωγικότητας με υπηρεσίες όπως τα τηλεφωνικά συστήματα, οι εναέριες μεταφορές, τα γνωστά σε όλους μας ΑΤΜ και γενικότερα σε συστήματα που χρησιμοποιούμε αρκετά συχνά στην καθημερινή μας ζωή. Επιχειρήσεις, σχολεία, ιδιωτικές εταιρείες εξαρτώνται σε απίστευτο βαθμό από τα

πληροφορικά συστήματα και επομένως είναι ιδιαίτερα τρωτές σε αστοχίες αυτών των συστημάτων. Καθώς τα πληροφοριακά συστήματα είναι πανταχού παρόντα όσο και το τηλεφωνικό σύστημα, είναι εντυπωσιακό να θυμόμαστε ότι δεν υπάρχουν κανονιστικές αρχές και ρυθμιστικές δυνάμεις για αυτά, παρόμοιες με αυτές που υπάρχουν για την τηλεφωνία, την παροχή ηλεκτρισμού, το ραδιόφωνο, την τηλεόραση ή άλλες τεχνολογίες κοινής ωφέλειας. Η απουσία προτύπων και η κρισιμότητα ορισμένων από τις εφαρμογές των συστημάτων θα προκαλέσουν κατά πάσα πιθανότητα την απαίτηση για εθνικά και διεθνή πρότυπα και ίσως τη δημιουργία εποπτικών αρχών.

Τα πληροφοριακά συστήματα είναι βέβαιο πως βοηθήσαν με τρόπο απaráμιλλο στην πρόοδο της ιατρικής με συστήματα τα οποία χρησιμοποιούνται σε διάφορους κλάδους της υγείας με συνέπεια την αποτελεσματικότερη παρακολούθηση των ασθενών. Παρ' όλα τα οφέλη όμως στον τομέα της υγείας, η χρήση των νέων τεχνολογιών εγκυμονεί και κινδύνους για την υγεία. Πολλά είναι τα προβλήματα που αντιμετωπίζουν στις μέρες μας οι χρήστες των πληροφοριακών συστημάτων σημαντικότερα των οποίων είναι, σύμφωνα με ιατρικές πηγές, η επαναλαμβανόμενη εντατική κάκωση, το σύνδρομο του καρπιαίου σωλήνα, το σύνδρομο οφθαλμολογικής κόπωσης από υπολογιστή και το τεχνο-άγχος (Brod, 2000).

Το Internet και οι Web εφαρμογές έδωσαν τη δυνατότητα σε επιχειρήσεις να διαμοιράζονται και να διανέμουν πληροφορίες ακαριαία σε εκατομμύρια ανθρώπους ανά την υφήλιο. Όμως η σύγχρονη αυτή τεχνολογία δημιουργεί προκλήσεις για τα παραδοσιακά καθεστώτα προστασίας του προσωπικού απορρήτου των ατόμων και της πνευματικής ιδιοκτησίας. Η τεχνολογία αποθήκευσης και ανάλυσης δεδομένων επιτρέπει στις εταιρείες να συγκεντρώνουν ευκολά προσωπικά δεδομένα για άτομα από πολλές και διάφορες πηγές και να αναλύουν τα δεδομένα αυτά για να δημιουργούν λεπτομερή ηλεκτρονικά προφίλ για άτομα και τη συμπεριφορά τους. Οι παραδοσιακοί νόμοι περί πνευματικών δικαιωμάτων είναι ανεπαρκείς για την προστασία από την πειρατεία του λογισμικού επειδή το ψηφιακό υλικό μπορεί να αντιγραφεί πολύ εύκολα (Gopal & Sanders, 1997).

Όλα όσα περιγράφονται παραπάνω παρουσιάζονται με συνοπτικό τρόπο στον πίνακα που ακολουθεί:

Οφέλη Πληροφοριακών Συστημάτων	Αρνητικές Επιπτώσεις
Τα Πληροφοριακά Συστήματα μπορούν να εκτελούν υπολογισμούς και να επεξεργάζονται τις πληροφορίες πολύ πιο γρήγορα από τους ανθρώπους.	Ο αυτοματισμός δραστηριοτήτων που γινόταν προηγουμένως από ανθρώπους μπορεί να καταργήσει θέσεις εργασίας.
Τα Πληροφοριακά Συστήματα βοηθούν τις εταιρείες να μαθαίνουν περισσότερα για τις αγοραστικές συνήθειες και προτιμήσεις των πελατών τους.	Τα Πληροφοριακά Συστήματα δίνουν τη δυνατότητα σε οργανισμούς να συγκεντρώνουν προσωπικά δεδομένα που παραβιάζουν την ιδιωτικότητα των ανθρώπων.

Οφέλη Πληροφοριακών Συστημάτων	Αρνητικές Επιπτώσεις
Τα Πληροφοριακά Συστήματα αυξάνουν την παραγωγικότητα με υπηρεσίες όπως οι αυτόματες ταμειολογιστικές μηχανές (ΑΤΜ).	Τα Πληροφοριακά Συστήματα χρησιμοποιούνται εκτενώς σε επιχειρήσεις και σε περίπτωση διακοπής της λειτουργίας τους μπορεί να παραλύσει την αγορά.
Τα Πληροφοριακά Συστήματα βοήθησαν στην πρόοδο πολλών τομέων της ανθρώπινης δραστηριότητας όπως είναι ο τομέας της Ιατρικής.	-
Με το Διαδίκτυο διανέμονται πληροφορίες ακαριαία σε εκατομμύρια ανθρώπους σε όλο τον κόσμο.	-

Πίνακας 1.1 : Θετικές και αρνητικές επιπτώσεις των Πληροφοριακών Συστημάτων.

2. Βάσεις Δεδομένων

Τα Πληροφοριακά Συστήματα σήμερα στοχεύουν στην ικανοποίηση των πληροφοριακών αναγκών των οργανισμών. Συμβάλλουν σημαντικά στην επίτευξη των σκοπών τους με την ενσωμάτωση τεχνολογιών πληροφορικής και επικοινωνιών (information & communication technologies) στις επιχειρησιακές διαδικασίες (business processes) τους. Τέτοιου είδους Πληροφοριακά Συστήματα δίνουν ιδιαίτερη έμφαση στην «πληροφορία» και επομένως θεωρούν τις τεχνολογίες πληροφορικής και επικοινωνιών ως μέσα για την παραγωγή, επεξεργασία και παροχή πληροφοριών.

Με την πάροδο του χρόνου, τα Πληροφορικά Συστήματα αντιμετωπίζουν αυξημένες προκλήσεις. Οι κατανεμημένες τεχνολογίες βασισμένες στον Παγκόσμιο Ιστό (World Wide Web-WWW) παρέχουν την αναγκαία υποδομή για τη ψηφιακή διασύνδεση οργανισμών και τα σύγχρονα Πληροφοριακά Συστήματα πρέπει να ικανοποιούν τις πληροφοριακές ανάγκες των ψηφιακών διασυνδεδεμένων οργανισμών. Αυτό έχει ως αποτέλεσμα οι οργανισμοί να έχουν στη διάθεση τους μεγάλο όγκο πληροφορίας με αποτέλεσμα η τεχνολογία να ενσωματώνεται ολοένα και περισσότερο στις επιχειρησιακές διαδικασίες τους. Προκειμένου να οργανωθεί και να εκμεταλλευτεί η μεγάλη ποσότητα της πληροφορίας τα Πληροφορικά Συστήματα χρησιμοποιούν τις Βάσεις Δεδομένων.

Τι είναι όμως μια Βάση Δεδομένων; Βάση Δεδομένων είναι μια καλά ορισμένη και ολοκληρωμένη συλλογή δεδομένων που αφορούν σε ένα οργανισμό. Ορισμός των δεδομένων σημαίνει ότι τα λειτουργικά δεδομένα του οργανισμού αποθηκεύονται σε αρχεία συγκεκριμένου τύπου, σε προκαθορισμένες μορφές και σύμφωνα με προκαθορισμένους περιορισμούς. Μια ολοκληρωμένη συλλογή δεδομένων περιλαμβάνει όχι μόνο τα δεδομένα αλλά και τις σημασιολογικές ιδιότητες που υπάρχουν σ' αυτά καθώς και στις σχέσεις μεταξύ τους. Η υλοποίηση μίας βάσης δεδομένων γίνεται υπό μορφή αλληλο-σχετισμένων αρχείων δεδομένων και κατά τρόπο ώστε να διευκολύνεται η προσπέλασή τους με τη μεγαλύτερη δυνατή ταχύτητα.

Μια σημαντική απόφαση που πρέπει να ληφθεί κατά την ανάπτυξη ενός Πληροφοριακού Συστήματος είναι ο καθορισμός της αρχιτεκτονικής του. Αντοιχα πρέπει να καθορισθεί και η αρχιτεκτονική της Βάσης Δεδομένων. Οι κύριες αρχιτεκτονικές των Βάσεων Δεδομένων είναι:

- η Συγκεντρωτική (Centralised). Στη συγκεκριμένη αρχιτεκτονική, όλα τα δεδομένα του οργανισμού αποθηκεύονται σε ένα υπολογιστικό σύστημα και διαχειρίζονται συγκεντρωτικά.
- η Κατανεμημένη (Distributed). Στην κατανεμημένη αρχιτεκτονική, τα δεδομένα αποθηκεύονται σε διάφορα υπολογιστικά συστήματα, πιθανών διαφορετικών κατασκευαστών και δυνατοτήτων, που συνδέονται μεταξύ τους μέσω ενός τοπικού ή εκτεταμένου δικτύου και διαχειρίζονται κατανεμημένα.

2.1 Η Προσέγγιση των Βάσεων Δεδομένων

Η ανάπτυξη Πληροφοριακών Συστημάτων σε περιβάλλοντα Βάσεων Δεδομένων διαφέρει κατά πολύ από την ανάπτυξη Πληροφοριακών Συστημάτων σε περιβάλλοντα αρχείων. Η διαφορά τους στηρίζεται στο γεγονός ότι στην πρώτη περίπτωση η διαχείριση των δεδομένων γίνεται μέσω

ενός φεχωριστού συτήματος λογισμικού ενώ στη δεύτερη περίπτωση γίνεται μέσω των προγραμμάτων εφαρμογών.

Τα συστήματα λογισμικού που αφορούν στη διαχείριση των δεδομένων ονομάζονται Συστήματα Διαχείρισης Βάσεων Δεδομένων (Data Base Management Systems- DBMS). Με τον τρόπο αυτό οι τύποι των δεδομένων του οργανισμού και οι περιορισμοί επ' αυτων περιγράφονται ξεχωριστά από τα προγράμματα εφαρμογών ενώ τα δεδομένα μπορεί να αποθηκεύονται μοναδικά και ν χρησιμοποιούνται ταυτόχρονα από πολλά προγράμματα.

2.2 Συστήματα Διαχείρισης Βάσεων Δεδομένων

Τα Συστήματα Διαχείρισης Βάσεων Δεδομένων είναι ειδικά πακέτα λογισμικού που έχουν ως στόχο την αποτελεσματική ικανοποίηση των αναγκών των Πληροφοριακών Συστημάτων σχετικά με το σχεδιασμό, την υλοποίηση και την αξιοποίηση των Βάσεων Δεδομένων σύμφωνα με τις προδιαγραφές αποδοτικότητας και ασφάλειας που έχουν τεθεί. Τα περισσότερα ΣΔΒΔ είναι πακέτα λογισμικού αποτελούμενα από λογικά τμήματα (modules) και συνοδεύονται από διάφορα εργαλεία λογισμικού (software tools). Τα λογικά τμήματα και τα εργαλεία λογισμικού των ΣΔΒΔ παρέχουν διάφορες υπηρεσίες για την κάλυψη πολλαπλών απαιτήσεων των χρηστών. Μεταξύ αυτών περιλαμβάνονται οι ακόλουθες:

- Σχεδιασμός, δόμηση και υλοποίηση βάσεων δεδομένων.
- Λειτουργία Βάσεων Δεδομένων.
- Ανάπτυξη Εφαρμογών
- Υποστήριξη Αποφάσεων
- Διεπαφές με άλλα πακέτα λογισμικού
- Υλοποίηση σε Κατανεμημένο Περιβάλλον Υπολογιστών

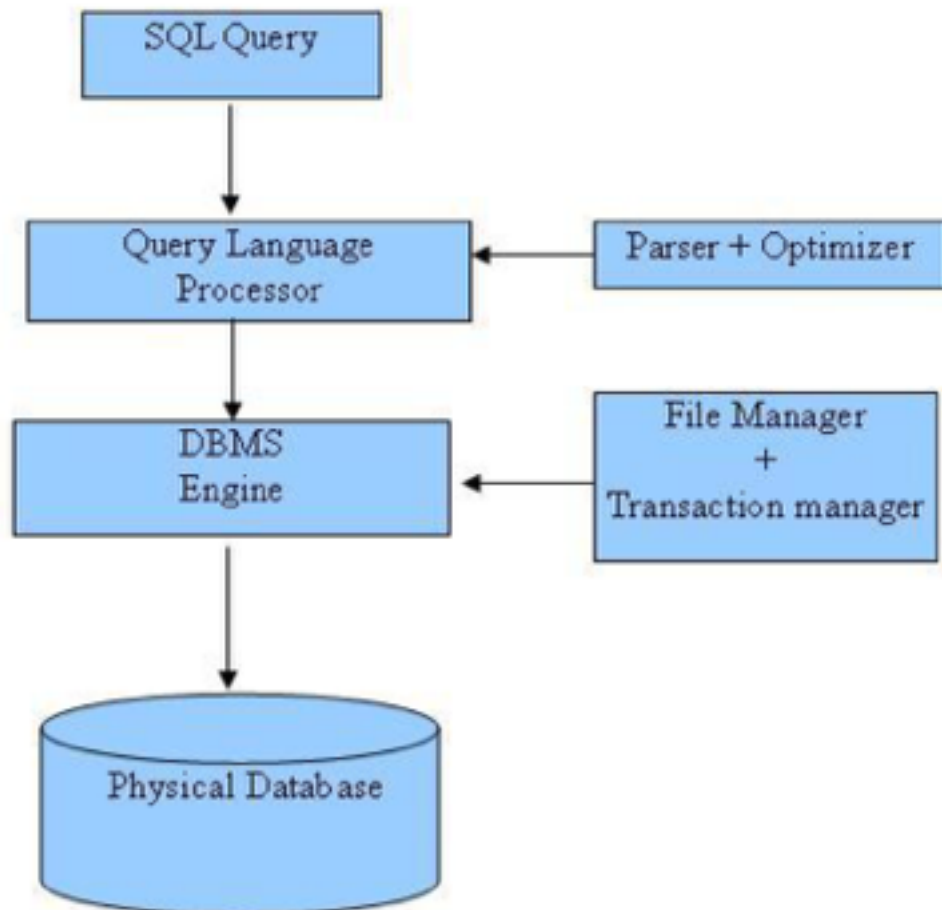
2.2.1 Σχεδίαση και Υλοποίηση της Βάσης Δεδομένων

Το στάδιο αυτό πραγματοποιείται από ειδικό προγραμματιστή ο οποίος αφού αναλύσει τις απαιτήσεις που έχουμε από τη βάση δεδομένων, προχωρεί στο σχεδιασμό και την υλοποίησή της, χρησιμοποιώντας τα κατάλληλα εργαλεία ανάπτυξης. Το στάδιο αυτό πραγματοποιείται μόνο μια φορά και απαιτεί ιδιαίτερη προσοχή, καθώς μια εσφαλμένη σχεδίαση, είναι δυνατόν να οδηγήσει σε αναποτελεσματική διαχείριση των δεδομένων, κάτι που μακροπρόθεσμα θα κοστίζει, τόσο χρόνο, όσο και χρήμα.

Η σχεδίαση μιας βάσης δεδομένων περιλαμβάνει τον καθορισμό της δομής της βάσης, των τύπων δεδομένων που θα καταχωρούνται στη βάση, και του πλήθους και του είδους των διαδικασιών που θα εφαρμόζονται στα δεδομένα της βάσης. Στο σημείο αυτό θα πρέπει να αναφέρεται ότι πριν το σχεδιασμό της βάσης, προηγείται πλήρης ανάλυση της συμπεριφοράς και των απαιτήσεων του συτήματος που πρόκειται να υλοποιήσουμε. Οι

τεχνικές λεπτομέρειες που αφορούν αυτή την ανάλυση είναι αδύνατο να αναφερθούν εδώ, και αποτελούν αντικείμενο μιας ολόκληρης θεματικής ενότητας, με τίτλο Ανάλυση και Σχεδιασμός Συστημάτων.

Μιλώντας γενικά, η σχεδίαση μιας βάσης δεδομένων, μπορεί να οριστεί ως μια διεργασία τριών επιπέδων. Τα επίπεδα αυτά είναι, το φυσικό επίπεδο (physical level), το λογικό επίπεδο (logical level), και το επίπεδο όψης (view level).



Εικόνα 2.1: Τα επίπεδα στη Σχεδίαση μιας Βάσης Δεδομένων.

Στο φυσικό επίπεδο, η κύρια μέριμνα είναι ο καθορισμός του τρόπου με τον οποίο τα δεδομένα της βάσης θα αποθηκεύονται στο σύστημα. Στο επίπεδο αυτό, καθορίζονται οι τύποι των δεδομένων της βάσης, τους αλγόριθμους διαχείρισης των δεδομένων, και γενικά, βασικό ρόλο παίζουν οι διαδικασίες οι οποίες χαρακτηρίζονται από μεγάλο βαθμό πολυπλοκότητας.

Στο λογικό επίπεδο, τα πράγματα είναι πιο απλά. Στο επίπεδο αυτό οι λεπτομέρειες και η πολυπλοκότητα του φυσικού επιπέδου δεν έχουν σημασία. Σημαντικό είναι να

καθορισθεί ποια δεδομένα θα αποθηκευτούν στη βάση του συστήματος, και ποιες είναι οι σχέσεις που υφίστανται ανάμεσά τους. Είναι λοιπόν εύκολα αντιληπτό, πως η διαφορά ανάμεσα στο φυσικό και στο λογικό επίπεδο, έχει να κάνει με τον τύπο διαχείρισης των δεδομένων: στο φυσικό επίπεδο καθορίζονται πως τα δεδομένα θα αποθηκευτούν στη βάση του συστήματος, ενώ στο λογικό επίπεδο, καθορίζονται ποια είναι αυτά τα δεδομένα.

Τέλος, στο επίπεδο όφης, καθορίζεται εκείνο το τμήμα της βάσης που θα είναι ορατό σε κάθε χρήστη. Αυτό κυρίως βρίσκει εφαρμογή σε μεγάλες βάσεις δεδομένων με πολύπλοκη σχεδίαση και μεγάλο αριθμό χρηστών. Τα δεδομένα που καταχωρούνται στη βάση αυτή, είναι κάθε είδους, και γενικά μη συσχετιζόμενα μεταξύ τους. Είναι κοινή πρακτική, κυρίως στις περιπτώσεις των μεγάλων βάσεων δεδομένων, να ορίζονται ομάδες χρηστών στους οποίους να καταχωρούνται διαφορετικά δικαιώματα πρόσβασης, και διαφορετικά δεδομένα προς διαχείριση.

2.2.2 Καταχώρηση των Δεδομένων στη Βάση του Συστήματος

Μετά τη σωστή σχεδίαση και υλοποίηση της βάσης, μπορούμε πλέον να καταχωρήσουμε σε αυτή τα δεδομένα του συστήματος για το οποίο υλοποιήθηκε. Η διαδικασία της καταχώρησης γίνεται με τη βοήθεια κατάλληλα σχεδιασμένων φορμών, και μπορεί να πραγματοποιηθεί από οποιονδήποτε.

Στο στάδιο αυτό, η σχεδίαση και υλοποίηση της βάσης έχει πλέον ολοκληρωθεί, και η βάση είναι έτοιμη να χρησιμοποιηθεί για την καταχώρηση των δεδομένων. Η καταχώρηση αυτή δεν απαιτεί ειδικές γνώσεις πληροφορικής, και μπορεί να γίνει από οποιοδήποτε χρήστη, με μικρή σχετικά εκπαίδευση πάνω στη λειτουργία του προγράμματος. Για την καταχώρηση των δεδομένων χρησιμοποιούνται ειδικές φόρμες που περιλαμβάνουν πλαίσια εισαγωγής δεδομένων, συνοδευόμενα από κάποιο τίτλο, που να περιγράφει τη λειτουργία τους.

Η διαδικασία καταχώρησης δεδομένων στη βάση του συστήματος, διευκολύνεται σε πολύ μεγάλο βαθμό, εάν οι φόρμες μέσα από τις οποίες ο χρήστης καταχωρεί αυτά τα δεδομένα, είναι λειτουργικές, έξυπνα σχεδιασμένες, και κυρίως, χωρίς περιττά στοιχεία. Μια εφαρμογή η οποία δεν χαρακτηρίζεται από απλότητα όσον αφορά τη λειτουργία της, είναι σίγουρο ότι θα τρομάζει τον απλό χρήστη και θα μειώσει σημαντικά την αποδοτικότητά του. Αντίθετα μια σωστά σχεδιασμένη εφαρμογή που ανα πάσα στιγμή θα δίνει στο χρήστη να καταλάβει τι ακριβώς πρέπει να κάνει, θα του επιτρέψει να ολοκληρώσει την εργασία του εύκολα και γρήγορα. Ο σχεδιασμός της αλληλεπίδρασης του προγράμματος με τους χρήστες (User Interface, UI), αποτελεί αναπόσπαστο στοιχείο κάθε εφαρμογής, και απαιτεί πολύ μεγάλη προσοχή.

2.2.3 Διαχείριση των Δεδομένων της Βάσης

Η διαχείριση αυτή περιλαμβάνει μια σειρά από δραστηριότητες όπως την ανάκτηση δεδομένων που πληρούν συγκεκριμένα κριτήρια αναζήτησης, τη δημιουργία αντιγράφων ασφαλείας, και τη μέριμνα για τη σωστή συντήρηση της βάσης εφ' όσον η δυνατότητα αυτή παρέχεται από το σύστημα διαχείρισης βάσεων δεδομένων που χρησιμοποιούμε.

Κατά καιρούς έχουν εμφανιστεί στην αγορά πολλά συστήματα διαχείρισης βάσεων δεδομένων, από πολλές εταιρείες, τα οποία ποικίλλουν, τόσο στη σχεδίαση και διαχείριση της βάσης, όσο και στις δυνατότητες που προσφέρουν. Τα πιο γνωστά και ευρέως χρησιμοποιούμενα συστήματα διαχείρισης βάσεων δεδομένων είναι η Microsoft Access, η Oracle, η Informix, η Sybase, και παλαιότερα η dBase III plus και η dBase IV. Από τα μοντέρνα συστήματα διαχείρισης, η Microsoft Access χρησιμοποιείται κυρίως για την ανάπτυξη και διαχείριση βάσεων δεδομένων σε προσωπικούς υπολογιστές και σε επιχειρήσεις μεσαίου μεγέθους, ενώ η Oracle και η Sybase, χρησιμοποιούνται σε μεγάλες επιχειρήσεις, και υποστηρίζουν πολλούς χρήστες, πολλές ταυτόχρονες προσπελάσεις πάνω στα ίδια δεδομένα, καθώς και απομακρυσμένη πρόσβαση μέσω δικτύου.

Η διαχείριση των δεδομένων της βάσης, πραγματοποιείται ανά πάσα στιγμή μετά την υλοποίηση αυτής, και εφαρμόζεται στα δεδομένα εκείνα που έχουν ήδη καταχωρηθεί. Η κυριότερη μορφή διαχείρισης των δεδομένων της εφαρμογής, έχει να κάνει με την επιλογή όσων από αυτά πληρούν κάποια κριτήρια αναζήτησης ή έχουν κάποια συγκεκριμένη τιμή. Ας πάρουμε για παράδειγμα την περίπτωση μιας βιβλιοθήκης, για την οποία έχουμε αναπτύξει μια βάση δεδομένων. Έστω ότι θέλουμε να μάθουμε ποια βιβλία είναι γραμμένα στην Ελληνική γλώσσα, αναφέρονται στην Ιστορία του Βυζαντινού Πολιτισμού, και έχουν εκδοθεί στη δεκαετία του 1990. Είναι προφανές ότι χωρίς τη βάση δεδομένων, η ανάκτηση αυτής της πληροφορίας, θα απαιτούσε πολλές ώρες δουλειάς και πολύ μεγάλη προσπάθεια. Με τη χρησιμοποίηση όμως της βάσης, αυτή η διαδικασία αναζήτησης, είναι δουλειά μόλις λίγων λεπτών. Η ικανότητα αυτή των βάσεων δεδομένων να ανακτούν τις πληροφορίες που θέλουμε σε ελάχιστο χρονικό διάστημα, είναι ο βασικός λόγος που οδήγησε στην αλματώδη ανάπτυξή τους τα τελευταία χρόνια.

Μετά την ανάκτηση των δεδομένων από τη βάση του συστήματος, μπορούμε να εφαρμόσουμε πάνω σε αυτά πολλούς τρόπους διαχείρισης, όπως για παράδειγμα, να τα εκτυπώσουμε. Οι μηχανογραφικές καταστάσεις με τις βαθμολογίες των μαθητών που αναρτώνται στα Γυμνάσια και στα Λύκεια στο τέλος κάθε εξεταστικής περιόδου, δεν αποτελούν παρά τέτοιου είδους εκτυπώσεις, οι οποίες δημιουργούνται από τα προγράμματα που διαχειρίζονται τα δεδομένα αυτού του είδους. Ένα άλλο είδος διαχείρισης είναι η ταξινόμησή τους με βάση κάποια κριτήρια, όπως για παράδειγμα, με αλφαβητική σειρά. Τέλος μπορούμε να παρέμβουμε στα ίδια τα δεδομένα και να αλλάζουμε τις τιμές τους, ή ακόμη και να τα διαγράψουμε από τη βάση.

Όπως είναι λοιπόν εύκολο να διαπιστωθεί, οι διαδικασίες που εφαρμόζονται πάνω στα δεδομένα της βάσης, μπορούν να διαχωρισθούν σε δύο κατηγορίες. Στην πρώτη

κατηγορία, ανήκουν διεργασίες οι οποίες περιορίζονται μόνο στην εμφάνιση των δεδομένων στην οθόνη του υπολογιστή και γενικά σε ενέργειες παρόμοιας μορφής, οι οποίες όμως δεν μεταβάλλουν τα δεδομένα, ενώ στη δεύτερη κατηγορία, ανήκουν διεργασίες, οι οποίες τροποποιούν τις τιμές των δεδομένων, ή ακόμη και τα διαγράφουν τελείως. Για το λόγο αυτό απαιτείται ιδιαίτερη προσοχή κυρίως για τις διεργασίες της δεύτερης κατηγορίας, διότι στις πιο πολλές περιπτώσεις τα αποτελέσματα αυτών των διεργασιών δεν είναι αναστρέψιμα, και δύνανται να οδηγήσουν ακόμη και σε οριστική απώλεια των δεδομένων από τη βάση της εφαρμογής.

Σε όλα τα συστήματα διαχείρισης βάσεων δεδομένων, η διαχείριση των δεδομένων γίνεται με τη βοήθεια ειδικών εργαλείων. Τα εργαλεία αυτά είναι όπως έχουμε ήδη περιγράψει η γλώσσα ορισμού δεδομένων και η γλώσσα χειρισμού δεδομένων. Υπάρχουν πολλές εκδόσεις και πολλές μορφές για αυτές τις γλώσσες, τα τελευταία όμως χρόνια έχει επικρατήσει ένα κοινό πρότυπο που φέρει το όνομα δομημένη γλώσσα ερωτοαποκρίσεων (Structured Query Language, SQL). Η γλώσσα αυτή απαντάται σε όλα ανεξαιρέτως τα συστήματα διαχείρισης βάσεων δεδομένων, και παρά το γεγονός ότι εμφανίζεται σε πολλές παραλλαγές, η βασική ιδέα που περιγράφει τη λειτουργία της, είναι ουσιαστικά η ίδια.

2.3 Επιπτώσεις της Χρήσης των Βάσεων Δεδομένων

Η ανάπτυξη ενός Πληροφοριακού Συστήματος σε περιβάλλον Βάσεων Δεδομένων παρέχει σημαντικά πλεονεκτήματα στον οργανισμό σε ότι αφορά τη διαχείριση και την ασφάλεια των δεδομένων του. Τα πλεονεκτήματα προκύπτουν από τη δημιουργία μια ολοκληρωμένης θεώρησης των δεδομένων του οργανισμού και από τη δυνατότητα τήρησης κοινών προτύπων για τα δεδομένα και την ικανοποίηση των πληροφοριακών απαιτήσεων ενός συστήματος. Τα μειονεκτήματα αφορούν λειτουργικά και οικονομικά θέματα που αναπόφευκτα προκύπτουν.

2.3.1 Πλεονεκτήματα Βάσεων Δεδομένων

Τα βασικά πλεονεκτήματα που προκύπτουν για τον οργανισμό από το σωστό σχεδιασμό είναι τα εξής:

- Μείωση της Περίσσειας των Δεδομένων. Περιττά δεδομένα ονομάζονται εκείνα που όταν διαγραφούν από τη βάση δεδομένων δεν μειώνεται το πληροφοριακό της περιεχόμενο. Τα περιττά δεδομένα μπορεί να είναι είτε σύνθετα δεδομένα που μπορεί να εξαχθούν από στοιχειώδη αρχεία είτε επαναλήψεις δεδομένων που αποθηκεύονται σε διαφορετικά αρχεία. Και ο κύριος λόγος για τον οποίο δεν είναι επιθυμητή η ύπαρξη περιττών δεδομένων είναι η ασυνέπεια που μπορεί να προκαλέσουν.

- **Ανεξαρτησία Δεδομένων.** Με τον τρόπο αυτό τα προγράμματα εφαρμογών δεν εξαρτώνται από τις λογικές και φυσικές δομές των αρχείων δεδομένων που χρησιμοποιούν. Επομένως οι εφαρμογές δεν επηρεάζονται από τυχόν αλλαγές στις δομές των αρχείων της Βάσης Δεδομένων.
- **Κοινοχρησία Δεδομένων.** Τα δεδομένα που έχουν αποθηκευτεί στη Βάση Δεδομένων είναι διαθέσιμα σε όλους τους χρήστες που διαθέτουν την απαιτούμενη δικαιοδοσία. Επόμενο είναι να μπορούν χρησιμοποιηθούν και από άλλες εφαρμογές.
- **Ασφάλεια Δεδομένων.** Τα σύγχρονα ΣΔΒΔ παρέχουν προηγμένες δυνατότητες για τον έλεγχο των προσβάσεων των σρηστών στα δεδομένα, τήρησης εφεδρικότητας αντιγράφων δεδομένων, ανάκαμψης κτλ. [3]

2.3.2 Μειονεκτήματα Βάσεων Δεδομένων

Τα μειονεκτήματα προκύπτουν κυρίως από τον οικονομικό παράγοντα. Αυτό συμβαίνει διότι κάθε εξέλιξη και επιπλέον δυνατότητα που δίνει ένα Πληροφοριακό Σύστημα απαιτεί και πιο εξειδικευμένο προσωπικό και τεχνολογία. Συνεπώς τα μειονεκτήματα που προκύπτουν από τη χρήση των Βάσεων Δεδομένων είναι:

- **Απόκτηση Εξειδικευμένου Λογισμικού.** Το κόστος απόκτησης των αδειών χρήσης ενός ΣΔΒΔ εξαρτάται από παράγοντες όπως πόσους χρήστες μπορεί να υποστηρίξει ταυτόχρονα, το χρόνο αποκρίσης κτλ. Επομένως το κόστος απόκτησης μπορεί να φτάσει σε αρκετά υψηλό επίπεδο.
- **Απαιτήσεις Εξειδικευμένου Προσωπικού.** Για την υποστήριξη σύγχρονων και απαιτητικών ΣΔΒΔ απαιτείται η στελέχωση του οργανισμού να γίνει με εξειδικευμένο προσωπικό που οδηγεί αυτόματα και στην αύξηση του κόστους.
- **Απαιτήσεις Υπολογιστικών Πόρων.** Οι απαιτήσεις αυτές είναι συνάρτηση διαφόρων παραγόντων όπως είναι η αρχιτεκτονική του Πληροφοριακού Συστήματος και της Βάσης Δεδομένων, ο προβλεπόμενος αριθμός των ταυτόχρονων χρηστών και το είδος των εφαρμογών που προσπελάζουν τη Βάση Δεδομένων. [3]

3. Ανάγκη για Αυτοματοποίηση

Από τη στιγμή που η Πληροφορική έκανε την είσοδό της δυναμικά στις επιχειρήσεις και τους οργανισμούς, έχουν γίνει πολλές προσπάθειες ανάπτυξης εφαρμογών που αποσκοπούν στην αυτοματοποίηση των διαδικασιών της επιχείρησης- του οργανισμού. Η αρχική ιδέα ήταν να αυτοματοποιηθούν συγκεκριμένες διαδικασίες στα τμήματα που ήταν απαραίτητο, και όχι όλες οι διαδικασίες μιας επιχείρησης. Έτσι, δεν ήταν αναγκαία η σκέψη για ολοκληρωμένο πληροφοριακό σύστημα. Όμως στη συνέχεια, οι επιχειρήσεις αρχίσαν να αυτοματοποιούν όλο και περισσότερα κομμάτια του πληροφοριακού τους συστήματος. Για αυτό το λόγο, οι εφαρμογές που αναπτύχθηκαν δεν είχαν ενταχθεί σε κάποιο πλάνο ανάπτυξης ολοκληρωμένης πληροφοριακής αρχιτεκτονικής, με αποτέλεσμα το σχηματισμό μιας «μάζας» από αυτόνομα πληροφοριακά συστήματα σε κάθε επιχείρηση. Όσο περνούσε ο καιρός η μάζα αυτή μεγάλωνε και ταυτόχρονα σχηματιζόταν προβλήματα διασύνδεσης των διαφορών και ανεξάρτητων εφαρμογών μιας επιχείρησης. Τα προβλήματα αυτά καθιστούσαν δύσκολη, ή ακόμη και αδύνατη, την επικοινωνία μεταξύ των ετερογενών εφαρμογών ενός τμήματος και κατά συνέπεια και όλης της επιχείρησης.

3.1 Νησίδες Τεχνολογίας

Το πρόβλημα στην αυτοματοποίηση των εφαρμογών προέρχεται από την έλλειψη αρχικού σκοπού επικοινωνίας μεταξύ των εφαρμογών. Τα πληροφοριακά συστήματα των οποίων οι εφαρμογές έπαιρναν μέρος στην αυτοματοποίηση, πολλές φορές έτρεχαν σε διαφορετικές πλατφόρμες και λειτουργικά συστήματα και βασιζόταν σε διαφορετικές γλώσσες προγραμματισμού. Το γεγονός αυτό, καθιστούσε πρακτικά αδύνατη την επικοινωνία και τη διαχείριση κοινών επιχειρησιακών δεδομένων και μετέτρεπε κάθε εφαρμογή σε μια αποκομμένη προγραμματιστική μονάδα που χαρακτηρίστηκε ως «Νησίδα Τεχνολογίας».

Λόγω της απουσίας αρχιτεκτονικού πλάνου, κάθε τμήμα της επιχείρησης δημιουργούσε νησίδες τέτοιου είδους που αυτοματοποιούσαν συγκεκριμένες διαδικασίες και είχαν πρόσβαση σε συγκεκριμένα επιχειρησιακά δεδομένα. Αυτό είχε ως αποτέλεσμα για το τμήμα την επεξεργασία παρόμοιων διαδικασιών και την αποθήκευση παρόμοιων δεδομένων από διαφορετικές εφαρμογές, γεγονός που εύκολα οδηγούσε σε πλεονασμό αποθήκευσης και ασυνέπεια των δεδομένων. Παρουσιάζονταν σημαντικά προβλήματα ακεραιότητας και επικαιροποίησης των δεδομένων και δεν ήταν δυνατόν να υπάρχει μία ξεκάθαρη εικόνα αυτών από όλα τα τμήματα της επιχείρησης.

Εκτός από το θέμα της ακεραιότητας, προέκυψε και θέμα με το λειτουργικό κόστος της επιχείρησης μετά τη δημιουργία των νησίδων. Βασιμμένοι στο γεγονός ότι κάθε εφαρμογή

λειτουργούσε σε διαφορετική πλατφόρμα ή λειτουργικό σύστημα και είχε υλοποιηθεί με διαφορετική γλώσσα προγραμματισμού, οι οργανισμοί έπρεπε να διαθέσουν πόρους (οικονομικούς και ανθρώπινους) για την υποστήριξη και την συντήρηση των πληροφοριακών του συστημάτων. Επιπλέον, ήταν απαραίτητη η ύπαρξη διαδικασίας εξαγωγής των δεδομένων από μία νησίδα, ώστε αυτά να επεξεργαστούν και να εισαχθούν σε μία άλλη που θα τα χρειαστεί για τη λειτουργία της. Όλες οι παραπάνω ενέργειες απαιτούσαν προσωπικό εξειδικευμένο και κατάλληλα εκπαιδευμένο ώστε να μπορέσει να τις εκτελέσει ορθά. Επειδή ήταν εξαιρετικά δύσκολο να βρεθεί κάποιος που να πληρεί όλες τις προϋποθέσεις που χρειάζονταν για να γίνουν οι ενέργειες αυτές, οι επιχειρήσεις ήταν αναγκασμένες να προσλαμβάνουν μία σειρά από υπαλλήλους, γεγονός που αύξανε τα έξοδα λειτουργίας και συντήρησης. Για αυτό, οι επιχειρήσεις αποφάσισαν να προχωρήσουν στην αναζήτηση νέων τρόπων ολοκλήρωσης ή διασύνδεσης των εφαρμογών τους. [5]

3.2 Ανάγκες Ολοκλήρωσης

Όσο περνούσε ο καιρός, οι νησίδες τεχνολογίας αυξάνονταν επικίνδυνα και έτσι δημιουργήθηκε η ανάγκη για εύρεση πιθανών τρόπων ολοκλήρωσης. Στις αρχές της δεκαετίας του 1990 ξεκίνησε ένας αγώνας για τη λύση του προβλήματος της ολοκλήρωσης των πληροφοριακών υποδομών των οργανισμών. Με τις Τεχνολογίες Ενδιάμεσου Λογισμικού (middleware), στην αρχή, και τα Συστήματα Διαχείρισης Επιχειρησιακών Πόρων (Enterprise Resource Planning-ERP) στη συνέχεια, ο αγώνας εξελίχθηκε αλλά όχι αρκετά ώστε τα αποτελέσματα να είναι ικανοποιητικά. Έτσι, στα τέλη της ίδιας δεκαετίας, η κατεύθυνση του αγώνα αλλάζει και στρέφεται σε μία νέα κατηγορία λογισμικού ολοκλήρωσης, γνωστή και ως Τεχνολογία Ολοκλήρωσης Επιχειρησιακών Εφαρμογών (Enterprise Application Integration-EAI). Η τεχνολογία προσπαθεί και καταφέρνει να ξεπεράσει με επιτυχία τα προβλήματα διασυνδεσιμότητας και διαλειτουργικότητας συνθέτοντας ολοκληρωμένες τεχνολογικές υποδομές που στυρίζονταν τη λειτουργικότητα των ήδη υπάρχοντων εφαρμογών. Οι υποδομές αυτές παίζουν το ρόλο της διασύνδεσης ανάμεσα στις εφαρμογές των επιχειρήσεων και απαιτούν μεγάλες οργανωτικές αλλαγές στην επιχείρηση, ανάλογα με τον αριθμό των εφαρμογών. Πολλοί οργανισμοί δεν είχαν τη δυνατότητα να προβούν σε αυτές τις αλλαγές, είτε λόγω οικονομικών παραγόντων είτε λόγω αντίδρασης του ανθρώπινου δυναμικού.

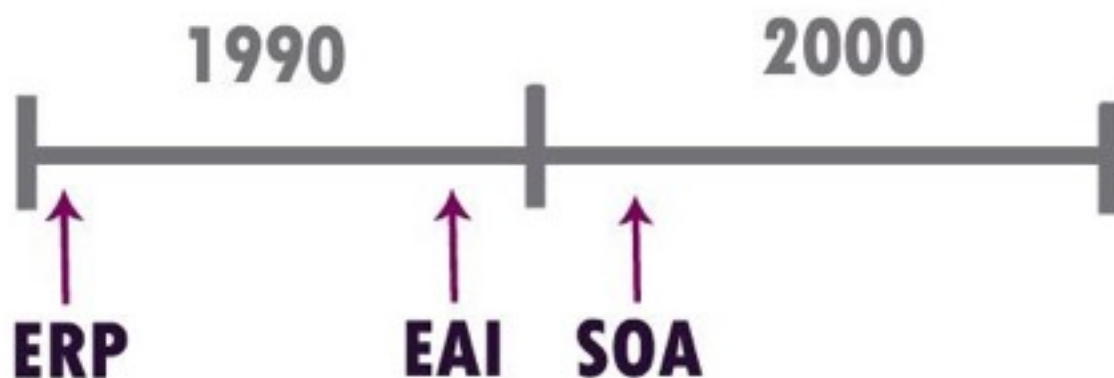
Αυτό οδήγησε, στις αρχές της δεκαετίας του 2000, σε μία εξέλιξη που δίνει στην Τεχνολογία Ολοκλήρωσης Επιχειρησιακών Εφαρμογών μία νέα προσέγγιση ολοκλήρωσης. Η εξέλιξη αυτή ονομάστηκε Υπηρεσιοστρεφής Αρχιτεκτονική (Service Oriented Architecture-SOA) και έχει ως στόχο τη δημιουργία λύσεων που λύνουν όλα τα προβλήματα της EAI. Σκοπός της SOA είναι η ολοκλήρωση των εφαρμογών με τρόπο τέτοιο ώστε να τις καθιστά επαναχρησιμοποιήσιμες. Επίσης, οι λύσεις που προκύπτουν δεν χρειάζονται μεγάλο εύρος

αλλαγών, γεγονός που σημαίνει ευκολότερη εφαρμογή από τις επιχειρήσεις και λιγότερες αντιδράσεις από το ανθρώπινο δυναμικό αυτών.

3.3 Εξέλιξη Υπηρεσιών Ολοκλήρωσης

3.3.1 Enterprise Resource Planning - ERP

Τα Επιχειρησιακά Συστήματα Διαχείρισης Πόρων εμφανίστηκαν στην δεκαετία του 1990 και προσπάθησαν να λύσουν το πρόβλημα της ολοκλήρωσης των οργανισμών. Χαρακτηριστικό των ERP συστημάτων είναι η αντικατάσταση των ετερογενών εφαρμογών και όχι η διασύνδεσή τους. Αυτό σημαίνει ότι οι νησίδες τεχνολογίας πλέον καταργούνται εντελώς και δημιουργείται ένα εσωτερικά ολοκληρωμένο πληροφοριακό σύστημα βασισμένο σε ERP.



Εικόνα 3.1: Η εξέλιξη των Συστημάτων ERP μέσα στο χρόνο.

«Τα ERP συστήματα είναι ολοκληρωμένα πακέτα που αποτελούνται από ένα δύνολο λειτουργικών μονάδων (modules) που αυτοματοποιούν και ολοκληρώνουν μεγάλο ποσοστό των επιχειρησιακών διαδικασιών των οργανισμών όπως η οικονομική διαχείριση, οι πωλήσεις, η διανομή, η διαχείριση παραγωγής, η διαχείριση ανθρώπινων πόρων, κλπ. Έχουν σχεδιαστεί με σκοπό να ολοκληρώσουν τις πληροφοριακές ροές ενός οργανισμού και να επιλύσουν το πρόβλημα του πλεονασμού των δεδομένων και της λειτουργικότητας των διαφορών ΠΣ» Μ.Θεμιστοκλέους, Β. Μαντζάνα (2010, σελ.20).

Από τον παραπάνω ορισμό είναι φανερό ότι τα ERP συστήματα λύνουν τα προβλήματα ασυνέπεια των δεδομένων που είχαν δημιουργηθεί από τις νησίδες τεχνολογίας. Πλέον είναι δυνατός ο χειρισμός και η επεξεργασία δεδομένων σε πραγματικό

χρόνο, γεγονός που συμβάλλει αρκετά στην αποτελεσματική διεκπεραίωση των επιχειρησιακών διαδικασιών.

Βασικό χαρακτηριστικό για την εφαρμογή της μεθόδου ERP είναι η υιοθέτηση της στρατηγικής της λειτουργίας. Ο οργανισμός αναγκάζεται να αποχωρηστεί τη δική του φιλοσοφία και να προσαρμόσει την λογική λειτουργίας του σε αυτή που ταιριάζει με το σύστημα ERP που έχει εφαρμόσει. Αυτό έχει ως αποτέλεσμα, πολλοί οργανισμοί με κοινά συστήματα να καταλήγουν με παρόμοιες στρατηγικές μεθόδους. Στην προσπάθεια των οργανισμών να αποφύγουν αυτό το φαινόμενο, προσαρμόζοντας το νέο σύστημα ERP στις ανάγκες τους, προκύπτει ένα μη επιθυμητό αποτέλεσμα. Τα συστήματα ERP χαρακτηρίζονται ως μονολυθικά και πολύπλοκα και η έλειψη ευελιξίας τους τα καθιστά ιδιαίτερα δύσκολα στην παραμετροποίηση, τόσο από τεχνικής όσο και από οικονομικής πλευράς. [5]



Εικόνα 3.2: ERP Συστήματα - Τεχνολογία.

Εναλλακτική λύση στο παραπάνω ζήτημα παρουσιάστηκε και η μερική αντικατάσταση των εφαρμογών με ERP συστήματα. Το φαινόμενο αυτό παρουσιάστηκε σε επιχειρήσεις που είχαν ως στόχο να κρατήσουν την προσωπικότητά τους, ολοκληρώνοντας όμως ταυτόχρονα τα πληροφοριακά τους συστήματα. Η εναλλαγή αυτή είχε ως αποτέλεσμα τη διατήρηση των νησίδων τεχνολογίας σε συνδιασμό με την εγκατάσταση ERP συστημάτων. Συμπαιρένουμε λοιπόν ότι οι ERP τεχνολογίες φέρουν μερική λύση στο μεγάλο

πρόβλημα της ολοκλήρωσης, το οποίο συνεχίζει να απασχολεί τους οργανισμούς τη δεκαετία του 1990.

3.4 Τεχνολογίες Ενδιάμεσου Λογισμικού

Η πρώτη προσπάθεια των οργανισμών να ξεπεράσουν το πρόβλημα της ολοκλήρωσης των συστημάτων τους οδήγησε στη δημιουργία των τεχνολογιών Ενδιάμεσου Λογισμικού (middleware), το οποίο χρησιμοποιήθηκε για τη διασύνδεση των εφαρμογών. Ο όρος middleware σημαίνει: *«ένας μηχανισμός που επιτρέπει σε μία οντότητα (πχ. Εφαρμογή ή βάση δεδομένων) να επικοινωνήσει με άλλη ή άλλες οντότητα/οντότητες»* Linticum (1999a, p.199).

Ο τρόπος ολοκλήρωσης μέσω τεχνολογιών Ενδιάμεσου Λογισμικού, όπως είναι προφανές, δεν είναι απλός και αυτό γιατί απαιτεί συνδυασμό διαφορετικών ειδών τεχνολογιών. Ο συνδυασμός αυτός δεν ακολουθεί συγκεκριμένες μεθόδους ή βήματα και αυτό γιατί κάθε επιχείρηση παρουσιάζει διαφορετικές ανάγκες ολοκλήρωσης, ακόμα και αν βρίσκεται σε ίδιο τομέα με άλλες που έχουν ήδη εφαρμόσει την ολοκλήρωσή τους με αυτή τη μέθοδο.

Ακόμη και αν οι επιχειρήσεις κατάφερναν να παραβλέψουν το κόστος και την πολυπλοκότητα της εφαρμογής της παραπάνω τεχνολογίας, ο αριθμός των συνδέσεων που απαιτούνταν για κάθε πληροφοριακό σύστημα πλησίαζε την απαγόρευση, ειδικά για αυτά με μεγάλο αριθμό εφαρμογών. Επίσης, κατά τη διασύνδεση έπρεπε να ληφθούν υπόψη συγκεκριμένοι περιορισμοί όσον αφορά τη δομή των δεδομένων και των μηνυμάτων, τη μορφή των διεπαφών και των βάσεων δεδομένων, το περιβάλλον υλοποίησης, κ.α.. Κατά συνέπεια, η διασύνδεση των εφαρμογών είναι μία πολύ περίπλοκη, χρονοβόρα και μη συμφέρουσα πρακτική η οποία δημιουργεί έναν αριθμό συνδέσεων που σε μερικές περιπτώσεις, αντί να βελτιώνει την πληροφοριακή υποδομή, κάνει την κατάσταση χειρότερη. Οι επιχειρήσεις με πολλές διασυνδέσεις φτάνουν στο σημείο δημιουργίας ενός «Σπαγγέτι Εφαρμογών», όρος που ορθά δώθηκε για το χαρακτηρισμό της κατάστασης, λόγω της περιπλοκής των διασυνδέσεων.

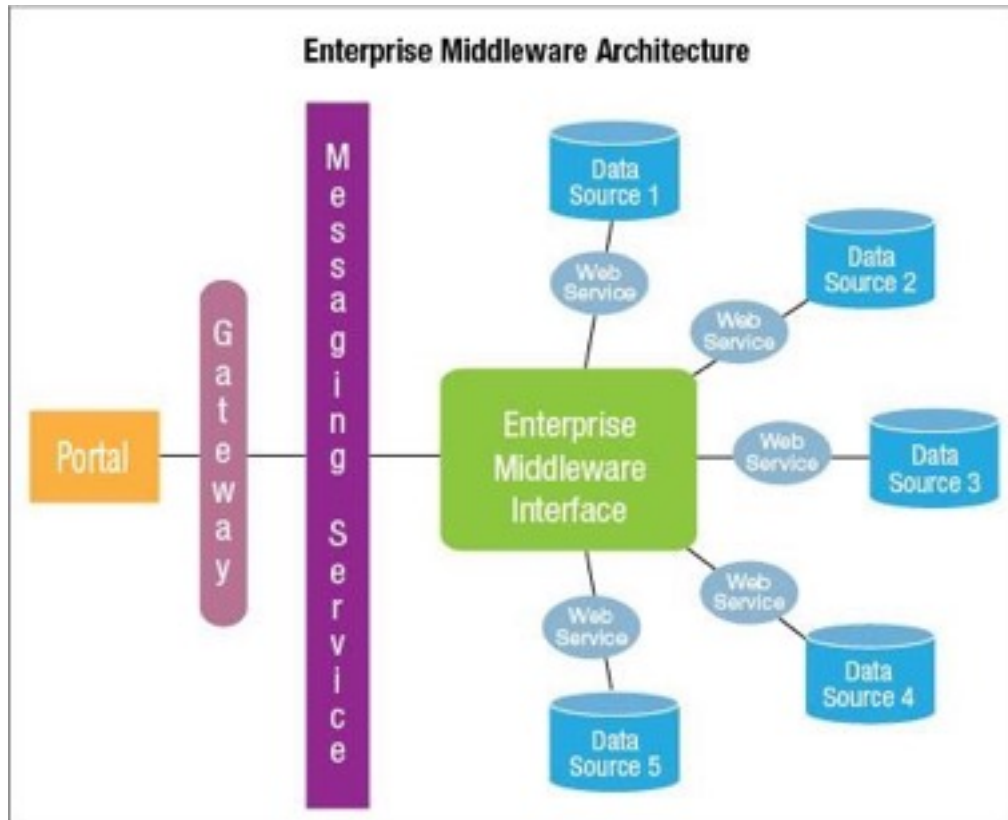
3.4.1 Κατηγορίες Ενδιάμεσου Λογισμικού

Οι πέντε βασικές κατηγορίες που ανήκουν στις τεχνολογίες Ενδιάμεσου Λογισμικού είναι οι εξής:

- Ενδιάμεσο Λογισμικό Βάσεων Δεδομένων (Database Oriented Middleware), Ενδιάμεσο Λογισμικό Μηνυμάτων (Message Based Middleware),

- Ενδιάμεσο Λογισμικό Συναλλαγών (Transaction Based Middleware), Ενδιάμεσο Λογισμικό Κατανεμημένων Αντικειμένων (Distributed Object Based Middleware),
- Ενδιάμεσο Λογισμικό Διεπαφών (Interface Oriented Technologies).

Κάθε μία από τις πέντε αυτές κατηγορίες περιέχει μία γκάμα από υπηρεσίες σχεδιασμένες για να ικανοποιούν κάθε ανάγκη ολοκλήρωσης που θα μπορούσε να έχει ένας οργανισμός. Ας δούμε αναλυτικά ποιες είναι αυτές.



Εικόνα 3.3: Enterprise Middleware Architecture.

3.4.2. Ενδιάμεσο Λογισμικό Βάσεων Δεδομένων (Database Oriented Middleware)

Στην κατηγορία αυτή ανήκει το λογισμικό που επιτρέπει τη διασύνδεση και την επικοινωνία μιας εφαρμογής με μία βάση δεδομένων. Είναι ένας από τους σημαντικότερους μηχανισμούς αφού πλέον όλα τα δεδομένα αποθηκεύονται σε βάσεις και ο μόνος τρόπος για να προσπεραστούν είναι μέσω αυτού. Καινοτομία αποτελεί το γεγονός ότι τα αρχεία και οι βάσεις που επικοινωνούν μπορούν να βρίσκονται είτε στον ίδιο είτε σε απομακρυσμένο υπολογιστή και παρόλα αυτά να λειτουργούν σωστά.

Οι πιο δημοφιλείς τεχνολογίες της κατηγορίας αυτής είναι οι παρακάτω:

- OPEN DATABASE CONNECTIVITY (ODBC)

«Ορίζεται ως μία πρότυπη διεπαφή για τη διασύνδεση σχεσιακών βάσεων δεδομένων» Ruh et al. (2000, p.54).

Ο μηχανισμός αυτός είναι μία καλά ορισμένη διεπαφή τύπου API και βασικό χαρακτηριστικό και πλεονέκτημά του είναι η ανεξαρτησία από το περιβάλλον και την πλατφόρμα υλοποίησης της βάσης δεδομένων. Ο μηχανισμός προσδιορισμού του κατάλληλου οδηγού (driver) ODBC για την μετάφραση των δεδομένων, καθιστά δυνατή αυτή την ανεξαρτησία. Επιπλέον, ο μηχανισμός έχει τη δυνατότητα να συνεργάζεται με άλλες τεχνολογίες Ενδιάμεσου Λογισμικού, γεγονός που βοηθάει στην αποτελεσματικότητά του.

- JAVA DATABASE CONNECTIVITY (JDBC)

Ο μηχανισμός αυτός είναι παρόμοιος με τον ODBC, με τη διαφορά ότι δημιουργήθηκε με βάση μια διεπαφή API για την γλώσσα προγραμματισμού Java. Παρέχει πρόσβαση κατά κύριο λόγο σε σχεσιακές βάσεις που λειτουργούν σε περιβάλλοντα Java και περιέχει συναρτήσεις για την επεξεργασία των δεδομένων των βάσεων.

Παρά τις ικανοποιητικές δυνατότητες των δύο παραπάνω τεχνολογιών, οι οποίες απλοποιούν κατά πολύ την διαδικασία επεξεργασίας, η μεταφορά των δεδομένων που επεξεργάζονται από τις βάσεις εξακολουθεί να αποτελεί πρόβλημα για τις τεχνολογίες Ενδιάμεσου Λογισμικού.

3.4.3 Ενδιάμεσο Λογισμικό Μηνυμάτων (Message Based Technologies)

Έρχεται για να λύσει το πρόβλημα μεταφοράς και διανομής μηνυμάτων από μία εφαρμογή (αποστολέα) σε μία άλλη (παραλήπτη). Οι τρόποι μετάδοσης μηνυμάτων μπορεί να είναι είτε σύγχρονοι είτε ασύγχρονοι. Στη σύγχρονη μετάδοση η εφαρμογή- αποστολέας παγώνει τις διεργασίες της μέχρι να λάβει απάντηση στο μήνυμά της ενώ στην ασύγχρονη συνεχίζει κανονικά, χωρίς η λειτουργία της να εξαρτάται από τις ενέργειες της εφαρμογής-παραλήπτη. Συνεπώς, η πρώτη τεχνική απαιτεί υψηλό βαθμό συνεργασίας κι από τα δύο άκρα.

Εκτός από αυτούς τους δύο τρόπους επικοινωνίας, υπάρχουν και οι παρακάτω:

- POINT-TO-POINT:

Οι εμπλεκόμενοι είναι άμεσα συνδεδεμένοι μεταξύ τους με ένα κανάλι επικοινωνίας.

- PUBLISH/SUBSCRIBE:

Περισσότερες από δύο εφαρμογές συνδέονται μεταξύ τους, και η κάθε μία από αυτές ορίζει τις πληροφορίες, τις δομές δεδομένων και τους τύπους των αιτημάτων που την ενδιαφέρουν να λαμβάνει καθώς και για ποιές κατηγορίες γεγονότων επιθυμεί να ενημερώνεται. Μέσα στις εφαρμογές αυτές υπάρχει ο εκδότης (publisher), που είναι η πρώτη εφαρμογή που αρχικά

κοινοποιεί πληροφορίες σε όλους τους συνδρομητές. Στη συνέχεια, αυτοί αποφασίζουν ποιες πληροφορίες τους ενδιαφέρουν, ενώ μπορούν με τη σειρά τους να δημοσιοποιήσουν πληροφορίες που διαθέτουν. Μία παρόμοια τεχνική είναι αυτή του Διαύλου Επικοινωνίας (Bus).

- **HUB AND SPOKE:**

Παρόμοια τεχνική με την παραπάνω, μόνο που ο κεντρικός κόμβος (Hub) διαθέτει και πληροφορίες διασύνδεσης για κάθε εφαρμογή που περιέχεται στη συνδρομητική λίστα. Σε αυτή την περίπτωση, οι εφαρμογές στέλνουν τις πληροφορίες τους στον κεντρικό κόμβο και αυτός αποφασίζει που θα τις διαμοιράσει, λαμβάνοντας υπόψη τους κανόνες διασύνδεσης που έχουν οριστεί για κάθε μια από τις εφαρμογές που είναι στο εύρος ελέγχου του. Η τεχνική Hub and Spoke είναι αποτελεσματικότερη από τις άλλες δύο αλλά όχι αποδοτική, αφού απαιτεί πολύ χρόνο για την επεξεργασία των μηνυμάτων.

Τα πέντε παραπάνω είδη επικοινωνίας χρησιμοποιούνται από τις τεχνολογίες Ενδιάμεσου Λογισμικού υποστήριξης μηνυμάτων. Οι κυριότερες τεχνολογίες που τα χρησιμοποιούν είναι:

- **REMOTE PROCEDURE CALLS (RPCS)**

Ο μηχανισμός αυτός καταφέρει να μεταφέρει μηνύματα καλύπτοντας την πολυπλοκότητα των λειτουργικών συστημάτων και των δικτύων. Εστιάζει στην ολοκλήρωση διαδικασιών για καταμεμημένες εφαρμογές και βασίζεται στη σύγχρονη επικοινωνία με point-to-point συνδέσεις. Ο προγραμματισμός για ένα τέτοιο μηχανισμό είναι απλός, το κόστος συντήρησης παρόλα αυτά είναι υψηλό. Η επαναχρησιμοποίησή του δεν είναι ιδιαίτερα εφυκτή και απαιτεί μεγάλη υπολογιστική ισχύ και υψηλών ταχυτήτων δίκτυα.

- **MESSAGE ORIENTED MIDDLEWARE (MOM)**

Σύμφωνα με τους Ruh et al.,(2000) το MOM είναι ένας τύπος ενδιάμεσου λογισμικού που χρησιμοποιεί την ανταλλαγή μηνυμάτων ως μέθοδο ολοκλήρωσης ενώ παράλληλα επιτρέπει τη δημιουργία, επεξεργασία, αποθήκευση και ανταλλαγή μηνυμάτων. Χρησιμοποιεί ασύγχρονη ανταλλαγή μηνυμάτων μέσω point-to-point συνδέσεων και είναι πιο ευέλικτο σε σχέση με το RPC. Το μοντέλο αυτό απαιτεί από τον αποστολέα και τον παραλήπτη την τροποποίηση του κώδικά τους ώστε να είναι δυνατή η επικοινωνία τους και έχει υψηλό κόστος συντήρησης.

- **MESSAGE BROKERS**

Αποτελούν μηχανισμούς που μεταφέρουν και μεταφράζουν δεδομένα μεταξύ δύο εφαρμογών. Χρησιμοποιώντας ασύγχρονη μεταφορά και Hub and Spoke μηχανισμούς, οι Message Brokers επιτρέπουν την ανάπτυξη μη επεμβατικών συνδέσεων ενώ παράλληλα μπορούν να δρομολογήσουν και να επεξεργαστούν κατάλληλα τα μηνύματα. Συχνά χρησιμοποιούνται ως Adapters μεταξύ διαφορετικών πλατφόρμων λειτουργίας.

- **EXTENSIBLE MARKUP LANGUAGE (XML)**

Η XML δίνει τη δυνατότητα στις εφαρμογές να ανταλλάσσουν μηνύματα μέσω ενός πρότυπου μηχανισμού ο οποίος παρέχει τις αντίστοιχες προδιαγραφές για το σχεδιασμό και

τη συγγραφή των μηνυμάτων. Σαν γλώσσα είναι εξαιρετικά απλή και μπορεί κανείς να διαβάσει το περιεχόμενό της ακόμα και αν δεν έχει δει ποτέ στη ζωή του XML αρχείο.

Ο Treese (1998) ορίζει την XML ως:

«ένα πρότυπο τρόπο δημιουργίας αρχείων μέσω του οποίου περιγράφεται ο μηχανισμός ερμηνείας, παρουσίασης και επεξεργασίας».

Αργότερα θα δούμε αναλυτικά πως δουλεύει σαν γλώσσα και τι κανόνες σύνταξης έχουν οριστεί για αυτήν. Επιγραμματικά, με τη χρήση της, η διαδικασία της ολοκλήρωσης γίνεται πιο απλή αφού ο παραλήπτης δεν χρειάζεται να γνωρίζει κάποια πληροφορία για τον αποστολέα προκειμένου να ανταλλάξουν επιτυχώς μηνύματα. Ο καθένας μεταφράζει το μήνυμα στη μορφή που επιθυμεί για να το διαβάσει και αποστέλει απάντηση σε XML.

Η XML παρέχει συνδέσεις με αντικειμενοστραφή μοντέλα και άλλες γλώσσες προγραμματισμού.

- Τεχνολογίες Ενδιάμεσου Λογισμικού Συναλλαγών (Transaction Based Technologies)

Μία επιχειρησιακή συναλλαγή αποτελείται από μία ή περισσότερες επιχειρησιακές λειτουργίες. Για να θεωρηθεί επιτυχημένη μία συναλλαγή, πρέπει όλες οι λειτουργίες της να ολοκληρωθούν με επιτυχία.

Για να μπορέσει να γίνει αυτό, οι τεχνολογίες που χρησιμοποιούνται είναι οι παρακάτω:

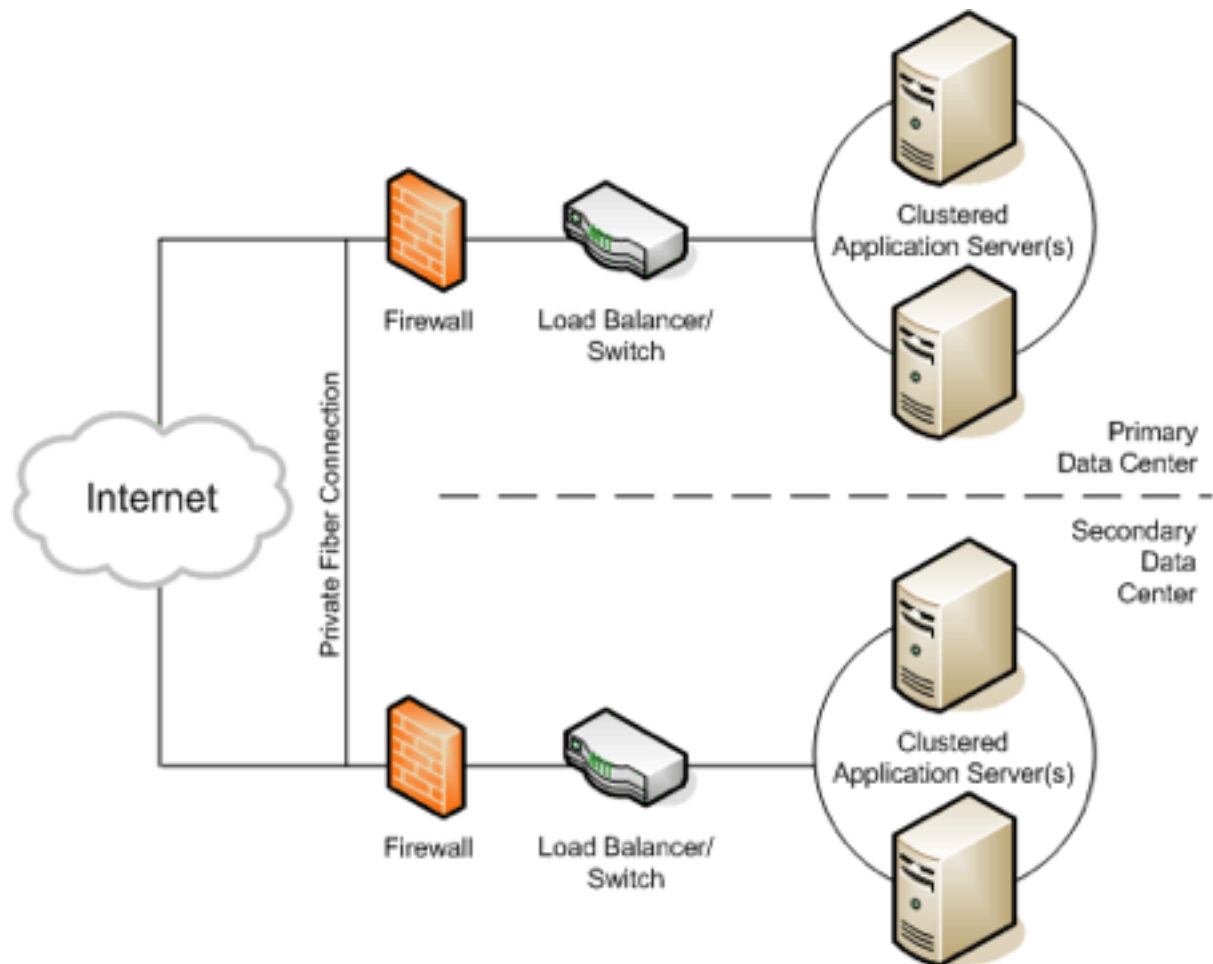
- Transaction Process Monitors (TP Monitors)

Αποτελούν προϊόντα που διασφαλίζουν την ακεραιότητα μιας συναλλαγής. Σύμφωνα με τον Bernstein (1990) ένας TP monitor είναι ένας μηχανισμός που συντονίζει τη ροή των αιτημάτων που γίνονται προς διάφορα προγράμματα προκειμένου να τα επεξεργαστούν για να εκτελέσουν μία συναλλαγή. Ένα αίτημα αφορά αποκλειστικά και μόνο μία συναλλαγή. Λόγω της λογικής εφαρμογών που περιέχουν, σε περίπτωση προβλημάτων μιας συναλλαγής, οι monitors την τερματίζουν και γίνονται rolls back των ενεργειών που είχαν ήδη πραγματοποιηθεί. Χρησιμοποιούνται για ολοκλήρωση υπηρεσιών, διασύνδεση εφαρμογών με βάσεις δεδομένων, ουρές μηνυμάτων ή με άλλες εφαρμογές, έχουν ενισχυμένη φορητότητα, επεκτασιμότητα, μηχανισμούς διαχείρισης σφαλμάτων, υψηλή επεξεργασία συναλλαγών και δρομολόγηση μηνυμάτων ανάλογα με την σειρά προτεραιότητάς τους. Η λειτουργία τους εξαρτάται άμεσα από το λογισμικό στο οποίο χρησιμοποιούνται και η φύση τους είναι ιδιαίτερα επεμβατική.

- Application Servers

Μπορούν να θεωρηθούν ως εξέλιξη των TP monitors αλλά διαφέρουν από αυτούς και γενικά από τις τεχνολογίες Ενδιάμεσου Λογισμικού. Οι διαφορές είναι ότι παρέχουν δομικά συστατικά συναλλαγών, είναι πιο εύχρηστοι και πιο φθηνοί, δεν έχουν όμως τόσο καλή απόδοση και αξιοπιστία. Επίσης, απαιτούν αρκετή προσπάθεια για να καταφέρουν να μεταφράσουν δεδομένα ή μηνύματα.

Μερικές από τις λειτουργίες τους είναι η επεξεργασία και κοινή χρήση της λογικής εφαρμογών, η διασύνδεση υπολογιστικών πόρων που βρίσκονται στο υπόβαθρο των συστημάτων, η δημιουργία διεπαφών και η εκτέλεση καταναμημένων εφαρμογών. Κατασκευάστηκαν με σκοπό τη χρήση τους σε εφαρμογές δικτυακού περιβάλλοντος Internet και επικοινωνούν με DOT, αυξάνοντας έτσι την λειτουργικότητά τους.



Εικόνα 3.4: Εφαρμογές Δικτυακού Περιβάλλοντος Internet.

- Τεχνολογία Καταναμημένων Αντικειμένων (Distributed Object Technology-DOT)

Η τεχνολογία αυτή παρέχει τη δημιουργία αντικειμενοστρεφών διεπαφών που βοηθούν υπάρχουσες ή νέες εφαρμογές να είναι προσβάσιμες από άλλες. Αυτό καταφέρνεται με την κοινή χρήση δεδομένων και λογικής εφαρμογής. Όπως προκύπτει από αυτό, οι αλλαγές στις εφαρμογές που συνδέονται με DOT είναι κάτι απαραίτητο και αναπόφευκτο. Η τεχνολογία συνεργάζεται με άλλες τεχνολογίες Ενδιάμεσου Λογισμικού όπως οι CORBA, COM, DCOM, EJB, κλπ. Και χρησιμοποιεί καταναμημένα αντικείμενα αντί για απλά.

Τα καταναμημένα αντικείμενα δεν είναι μέρος κάποιας εφαρμογής αλλά επικοινωνούν με αυτή μέσω διεπαφών. Χαρακτηρίζονται ως «φορητά» αντικείμενα που μπορούν να τρέχουν σε διάφορους εξυπηρετητές και να επικοινωνούν μέσω διεπαφών μηνυμάτων.

Στην τεχνολογία DOT ανήκουν οι παρακάτω τεχνικές:

- Common Object Request Broker Architecture (CORBA)

Πρόκειται για μια προδιαγραφή που ακολουθεί η εταιρία που έχει υιοθετήσει την τεχνική για την κατασκευή των προϊόντων. Στην ουσία, η CORBA παρέχει τους απαραίτητους κανόνες που πρέπει να ληφθούν υπόψην ώστε το λογισμικό που θα παραχθεί να είναι συμβατό με τα υπόλοιπα που δημιουργήθηκαν με βάση την ίδια τεχνική. Η τεχνική CORBA είναι διαθέσιμη σε παραπάνω από 25 πλατφόρμες λειτουργίας και, όπως καταλαβαίνει κανείς, από η φύση της έχει οριστεί να εξυπηρετεί πληροφοριακά συστήματα σε ετερογενή λειτουργικά περιβάλλοντα.

- Component Object Model (COM), COM+ and Distributed Component Object Model (DCOM)

Το πρότυπο COM προτάθηκε από την εταιρεία Microsoft για υποστήριξη των λειτουργικών της συστημάτων, παρόλα αυτά είναι κατάλληλο και για άλλα λειτουργικά, όπως τα Unix. Λειτουργεί όπως η CORBA, παρέχοντας κανόνες για τους προγραμματιστές ώστε αυτοί να δημιουργούν καταναμημένα αντικείμενα που είναι συμβατά με την COM. Στην πορεία της εξέλιξής της, προστέθηκαν ORB στη λειτουργία της και έτσι δημιουργήθηκε η COM+. Ένα βήμα παρακάτω έφερε την τεχνική DCOM η οποία ήταν συνέχεια των COM και COM+ αλλά είχε καλύτερες δυνατότητες χρήσης των απομακρυσμένων ORB.

- Enterprise Java Beans (EJB)

Υποστηρίζουν τη δημιουργία και χρήση εφαρμογών με καταναμημένα αντικείμενα και παρέχουν ένα σύνολο διεπαφών, όπως η COM και η CORBA. Η διαφορά από αυτές είναι ότι τα δομικά στοιχεία της τεχνικής αυτής είναι υλοποιημένα σε Java.

*«Τα EJB μπορούν να χαρακτηριστούν ως ένας μηχανισμός που είναι ανεξάρτητος από το πρωτόκολλο επικοινωνίας, υποστηρίζει ασύγχρονη επικοινωνία, παρέχει φορητότητα και απλοποιημένη ανάπτυξη εφαρμογών, είναι εύκολος στη χρήση και υποστηρίζει μηχανισμούς *publish/subscribe*. Είναι οι καλύτεροι μηχανισμοί για την υποστήριξη ομοιογενή Java περιβάλλοντων, παρέχουν την επαναχρησιμοποίηση των δομικών στοιχείων Java και γέφυρες επικοινωνίας με εφαρμογές που χρησιμοποιούν CORBA»* Μ. Θεμιστοκλέους, Β. Μαντζάνα (2010, σελ. 48).

- Τεχνολογίες Ενδιάμεσου Λογισμικού Διεπαφών (Interface Oriented Technologies)

Οι διεπαφές ορίζονται ως μονάδες λογισμικού που παρέχουν στους χρήστες αλληλεπίδραση με εφαρμογές. Πρέπει να είναι καλά ορισμένες και να περιγράφουν όλες τις ενέργειες που προσφέρει μία εφαρμογή. Επειδή δεν ανήκουν στο κύριο μέρος της εφαρμογής, αλλά την ολοκληρώνουν, είναι πιο εύκολα προσβάσιμες και προποποιήσιμες και απαιτούν λίγο σχετικά χρόνο για την επεξεργασία και αποκωδικοποίηση μηνυμάτων της εφαρμογής.

Οι διεπαφές χωρίζονται σε τρεις κατηγορίες:

- User Interfaces

Απευθύνονται σε εφαρμογές που δεν επικοινωνούν με άλλες, είναι αυτόνομες και ο μόνος που αλληλεπιδρά μαζί τους είναι ο χρήστης τους. Έτσι, το μοναδικό σημείο αλληλεπίδρασης είναι

η διεπαφή αυτού, μέσω της οποίας έχει πρόσβαση στα δεδομένα και τις ενέργειες της εφαρμογής. Πάνω στην τεχνική αυτή έχουν αναπτυχθεί εργαλεία ικανά να αντλήσουν πληροφορίες από τις εφαρμογές μέσω των διεπαφών. Οι Wrappers είναι ένα από τα εργαλεία αυτά, τα οποία χρησιμοποιούν μη επεμβατικές τεχνικές για να αντλήσουν απλώς δεδομένα (screens-as-data) ή ακόμη και να τα μετατρέψουν σε αντικείμενα CORBA, COM ή Java (screens-as-objects).

- **Application Programming Interfaces (APIS)**

Μηχανισμός που παρέχει πρόσβαση στα δεδομένα και την λειτουργικότητα μιας εφαρμογής, ορίζοντας το σύνολο των υπηρεσιών που αυτή προσφέρει σε άλλα Πληροφοριακά Συστήματα. Η πρόσβαση στον κώδικα που υλοποιεί τις υπηρεσίες δεν είναι απαραίτητη για να γίνει αυτός ο ορισμός. Έτσι προστατεύεται η τεχνική υλοποίησης της εφαρμογής, ενώ παράλληλα αυτή γνωστοποιεί τις λειτουργίες της σε άλλες εφαρμογές. Όπως καταλαβαίνει κανείς, ο μηχανισμός των APIs είναι εξαιρετικά χρήσιμος αλλά η φύση του τον αναγκάζει να χρειάζεται συνεχείς ενημερώσεις, αφού τα λογισμικά αλλάζουν αρκετά συχνά.

- **Adapters**

Είναι μονάδες λογισμικού που διασυνδέουν εφαρμογές αντιστοιχίζοντας τα στοιχεία των διεπαφών τους ώστε αυτά να μπορούν να είναι προσβάσιμα και από τις δύο. Λειτουργεί δηλαδή σαν ένας μεταφραστής μεταξύ των εφαρμογών χωρίς να επεμβαίνει στα δεδομένα υλοποίησής τους.

Συμπερασματικά, οι τεχνολογίες Ενδιάμεσου Λογισμικού χρησιμοποιούνται για εισαγωγή, εξαγωγή και διαχείριση στοιχών σε βάσεις δεδομένων, απόκτηση πρόσβασης σε εφαρμογές μέσω διεπαφών για την άντληση στοιχείων από αυτές, μετάφραση δεδομένων μεταξύ εφαρμογών και ολοκλήρωση λογισμικού που σχετίζεται με αντικείμενα ή συναλλαγές. Κάθε μία από τις τεχνικές δεν κατάφερε να επιλύσει μόνη της το πρόβλημα της ολοκλήρωσης και ο συνδυασμός τους ήταν κάτι εξαιρετικά δύσκολο, δαπανηρό και ριψοκίνδυνο έργο. [5]

3.5 Τεχνολογία Ολοκλήρωσης Τεχνολογίας Εφαρμογών (Enterprise Application Integration- EAI)

Η τεχνολογία EAI δημιουργήθηκε για να προσφέρει λύσεις πιο ευέλικτες και πιο οικονομικές από αυτές που υπήρχαν μέχρι τότε. Κύρια φιλοσοφία της ήταν η εστίαση στην ολοκλήρωση των επιχειρησιακών διαδικασιών και όχι των πληροφοριακών συστημάτων που είχαν αναπτυχθεί για την αυτοματοποίησή τους. Έτσι, η ολοκλήρωση δεν αφορούσε πλέον την διασύνδεση των εφαρμογών αλλά την εκ νέου δημιουργία ενός πληροφοριακού συστήματος που θα αυτοματοποιεί πλήρως τις διαδικασίες της επιχείρησης. Η EAI ορίζεται ως:

«ένα σύνολο από τεχνολογίες που επιτρέπει την μεταφορά και ανταλλαγή πληροφοριών μεταξύ εφαρμογών και διαδικασιών που βρίσκονται εντός ή εκτός οργανισμού» Linthicum(1999, p.354).

Η ΕΑΙ χρησιμοποιεί Τεχνολογίες Ενδιάμεσου Λογισμικού προκειμένου να καταφέρει να ενσωματώσει τη λειτουργικότητα που προέρχεται από ομογενή ή ετερογενή πληροφοριακά περιβάλλοντα. Ορίζει έναν ενδιάμεσο μηχανισμό ολοκλήρωσης ο οποίος αναλαμβάνει το συντονισμό των αυτοματοποιημένων επιχειρησιακών διαδικασιών. Αυτό συνεπάγεται την εξωτερική ολοκλήρωση των συστημάτων, χωρίς να απαιτείται κάποια αλλαγή στην υλοποίηση των εφαρμογών. Οι διεπαφές του χρήστη προσαρμόζονται κατάλληλα ώστε να συνδέσουν την εφαρμογή με το μηχανισμό που την ολοκληρώνει. Επομένως, ο αριθμός των συνδέσεων και των διεπαφών μειώνεται κατακόρυφα σε σύγκριση με τις προηγούμενες τεχνολογίες ολοκλήρωσης.



Εικόνα 3.5: Το ΕΑΙ και οι Τεχνολογίες Ενδιάμεσου Λογισμικού.

3.5.1 Προσεγγίσεις ΕΑΙ

Οι επιχειρήσεις που επιλέγουν να εφαρμόσουν την τεχνολογία ΕΑΙ πρέπει να αποφασίσουν και ποια προσέγγιση ολοκλήρωσης θα ακολουθήσουν. Οι επιλογές είναι δύο:

- ΣΤΡΑΤΗΓΙΚΗ ΠΡΟΣΕΓΓΙΣΗ

Ο οργανισμός αρχικά καταγράφει τις διαδικασίες του και αν επιθυμεί να τις βελτιώσει ή να τις αναδιοργανώσει. Έπειτα, δημιουργείται ένα πλάνο αντιστοίχισης των διαδικασιών στα πληροφοριακά συστήματα που υπάρχουν στον οργανισμό ώστε να διαπιστωθεί ποιες από αυτές χρειάζονται αυτοματοποίηση. Κατά τη διαδικασία της αυτοματοποίησης, μπορούν να αποσυρθούν συστήματα που παρουσιάζουν πλεονασμό λειτουργικότητας, να μεταφερθούν δεδομένα ακόμη και να τροποποιηθούν ήδη αυτοματοποιημένες διαδικασίες που δεν

ανταποκρίνονται στις ανάγκες του οργανισμού. Με αυτό τον τρόπο επιτυγχάνεται η πλήρης ολοκλήρωση των πληροφοριακών υποδομών της επιχείρησης, η οποία πλέον μπορεί να παρακολουθεί και να ελέγχει τις διαδικασίες τις ευκολότερα.

- **ΕΥΚΑΙΡΙΑΚΗ ΠΡΟΣΕΓΓΙΣΗ**

Όταν ο οργανισμός χρειάζεται επείγοντως ολοκλήρωση σε ένα συγκεκριμένο τμήμα του λογισμικού του, ή όταν δεν έχει την οικονομική δυνατότητα να ολοκληρώσει όλη την πληροφοριακή του υποδομή, επιλέγει την ευκαιριακή προσέγγιση και εφαρμόζει τεχνικές ολοκλήρωσης ΕΑΙ στο τμήμα/τμήματα που επιθυμεί. Επομένως, η προσέγγιση αυτή έχει περιορισμένο εύρος εφαρμογής και επιλέγεται όταν πρέπει να γίνουν άμεσες αλλαγές.

Πέρα από τις προσεγγίσεις, οι οργανισμοί επιλέγουν και μέθοδο ολοκλήρωσης μεταξύ της Δεδομενοκεντρικής και Διαδικασιοκεντρικής. Στην πρώτη, η ολοκλήρωση επικεντρώνεται μόνο στα δεδομένα ενώ στη δεύτερη επικεντρώνεται στις διαδικασίες, οι οποίες για να ολοκληρωθούν θα πρέπει να έχουν ολοκληρωθεί και τα δεδομένα. Το θέμα της επιλογής μεθόδου πρακτικά τίθεται σε αυτούς που επιλέγουν την ευκαιριακή προσέγγιση, αφού στην στρατηγική απαιτείται η Διαδικασιοκεντρική ολοκλήρωση. [5]

3.5.2 Πλεονεκτήματα ΕΑΙ

Σύμφωνα με τους Shang και Seddon(2000), τα πλεονεκτήματα της ΕΑΙ κατηγοριοποιούνται ως εξής:

- **ΛΕΙΤΟΥΡΓΙΚΑ:**

- Μείωση κόστους
- Βελτίωση παραγωγικότητας
- Βελτίωση ποιότητας
- Βελτίωση παρεχόμενων υπηρεσιών

- **ΔΙΑΧΕΙΡΙΣΤΙΚΑ:**

- Καλύτερη διαχείριση πόρων
- Ενίσχυση μηχανισμού λήψης αποφάσεων
- Βελτίωση απόδοσης

- **ΣΤΡΑΤΗΓΙΚΑ:**

- Ανάπτυξη επιχειρησιακών δραστηριοτήτων
- Υποστήριξη επιχειρησιακών συμμαχιών

- Δημιουργία επιχειρηματικής καινοτομίας
- Δημιουργία διαφοροποίησης προϊόντων
- Ενίσχυση δεσμών με πελάτες και προμηθευτές

- **ΠΛΗΡΟΦΟΡΙΑΚΗΣ ΥΠΟΔΟΜΗΣ:**

- Επίτευξη ειχειρηματικής ευελιξίας
- Μείωση κόστους πληροφοριακής υποδομής
- Αύξηση δυνατοτήτων πληροφοριακής υποδομής
- ΟΡΓΑΝΩΣΙΑΚΑ:
 - Υποστήριξη οργανωσιακών αλλαγών
 - Υποστήριξη εκπαίδευσης
 - Δημιουργία κοινών οραμάτων

3.5.3 Μειονεκτήματα ΕΑΙ

Σύμφωνα με τους Shang και Seddon(2000), τα μειονεκτήματα της ΕΑΙ κατηγοριοποιούνται ως εξής:

- ΛΕΙΤΟΥΡΓΙΚΑ:
 - Συνοδεύεται από επιπλέον κόστος για τον ανασχεδιασμό και την αλλαγή της οργανωσιακής δομής και των διαδικασιών
 - Απαιτεί υψηλό αρχικό κόστος
- ΔΙΑΧΕΙΡΙΣΤΙΚΑ:
 - Έλλειψη εργαζόμενων με γνώσεις ΕΑΙ
 - Προηγούμενες προσπάθειες ολοκλήρωσης είχαν αποτύχει
- ΣΤΡΑΤΗΓΙΚΑ:
 - Αντίδραση στην αλλαγή
 - Οι οργανισμοί είναι διστακτικοί στο να μοιραστούν τα δεδομένα τους με τους επιχειρηματικούς τους εταίρους
- ΠΛΗΡΟΦΟΡΙΑΚΗΣ ΥΠΟΔΟΜΗΣ:
 - Έλλειψη plug and play ΕΑΙ λύσεων
 - Καμιά τεχνολογία δεν λύνει όλα τα προβλήματα
 - Οι τεχνολογίες ολοκλήρωσης συχνά προκαλούν σύγχυση
 - Απαιτείται συνδυασμός τεχνολογιών και ΕΑΙ πακέτων
 - Έλλειψη εργαζόμενων με πρόσοντα ΕΑΙ
 - Υψηλή πολυπλοκότητα
 - Απουσία πληροφοριακής αρχιτεκτονικής
 - Έλλειψη κοινών ορισμών και προτύπων
 - Περιορισμοί ολοκλήρωσης από τα υπάρχοντα ΠΣ
 - Έλλειψη εγγράφων τεκμηρίωσης των παλιών ΠΣ
 - Πολύπλοκα και ασύμβατα μεταξύ τους ΠΣ

- ΟΡΓΑΝΩΣΙΑΚΑ:
 - Πολιτική επίδραση στις σχέσεις εργαζόμενων-τμημάτων
 - Πολυπλοκότητα επιχειρησιακών διαδικασιών
 - Προβλήματα κουλτούρας
 - Έλλειψη χρόνου για εκπαίδευση σε ΕΑΙ.

3.5.4 Κόστος ΕΑΙ

Σύμφωνα με τους Irani et al. (1998), το κόστος κατατάσσεται σε:

- ΑΜΕΣΟ:
 - Υλικού
 - Λογισμικού
 - Ανάπτυξης
 - Συντήρησης
 - Συμβούλων
- ΎΜΜΕΣΟ ΑΝΘΡΩΠΙΝΟ:
 - Εκπαίδευση εργαζομένων
 - Αλλαγή κουλτούρας εργαζομένων
 - Διαχειριστικών ενεργειών
- ΎΜΜΕΣΟ ΟΡΓΑΝΩΣΙΑΚΟ:
 - Αναδιοργάνωσης επιχειρησιακών διαδικασιών
 - Αλλαγής υποδομής οργανισμού
 - Ενεργειών μείωσης αντίδρασης στην αλλαγή
 - Ανασχεδιασμού επιχειρηματικής στρατηγικής
 - Υλικού
 - Λογισμικού
 - Ανάπτυξης
 - Συντήρησης
 - Συμβούλων

3.5.5 Τεχνική Διάσταση της ΕΑΙ

3.5.5.1 Τα στρώματα και τα Επίπεδα της ΕΑΙ

Η ολοκλήρωση με ΕΑΙ πραγματοποιείται με τέσσερα επίπεδα και επτά στρώματα. Τα επίπεδα είναι:

- Επίπεδο Δεδομένων, που είναι υπεύθυνο για τη διαχείριση των δεδομένων από τις εφαρμογές,

- Επίπεδο Διαδικασιών, που είναι υπεύθυνο για την αυτοματοποίηση των διαδικασιών,
- Επίπεδο Διαχείρισης Γνώσης, το οποίο αναλαμβάνει την αποθήκευση πληροφοριών σχετικά με τη λειτουργία του οργανισμού και
- Επίπεδο Εφαρμογής, που περιέχει όλες τις εφαρμογές που έχουν προκύψει μετά την ολοκλήρωση.

Τα στρώματα στα οποία στηρίζεται η ΕΑΙ είναι:

- Σύνδεση, όπου γίνεται η εξαγωγή των δεδομένων από την εφαρμογή- αποστολέα,
- Μεταφορά, όπου μεταφέρονται τα δεδομένα στην αρχιτεκτονική ολοκλήρωσης,
- Μετάφραση, όπου γίνεται η απαραίτητη επεξεργασία και μετάφραση, ανάλογα με τις προδιαγραφές του παραλήπτη,
- Ολοκλήρωση Διαδικασιών, όπου γίνεται η ολοκλήρωση και η πλήρης κατανόηση των διαδικασιών που επεξεργάστηκαν,
- Ολοκλήρωση Αρχών και Κανόνων, όπου αποθηκεύονται όλοι οι κανόνες και οι αρχές των υπηρεσιών και η λογική των διαδικασιών,
- Σημασιολογική Ολοκλήρωση, όπου δημιουργούνται σημασιολογικές ερμηνείες και συνθέτονται πληροφορίες για την παραγωγή γνώσης και
- Ολοκλήρωση Διαχείρισης Γνώσης, όπου χρησιμοποιούνται όλα τα στρώματα πριν από αυτό ώστε να ολοκληρωθεί η γνώση.

3.5.5.2 Αξιολόγηση Τεχνολογιών Ολοκλήρωσης

Τα κριτήρια αξιολόγησης, με βάση δεκάδες οργανισμούς που μελετήθηκαν από συγγραφείς, είναι τα εξής:

- ΣΤΟΙΧΕΙΑ ΕΦΑΡΜΟΓΩΝ:
 - Δεδομένα
 - Αντικείμενα
 - Διαδικασίες
- ΣΤΡΩΜΑΤΑ ΟΛΟΚΛΗΡΩΣΗΣ:
 - Στρώμα Μεταφοράς
 - Στρώμα Μετάφρασης
 - Στρώμα Ολοκλήρωσης Διαδικασιών
- ΣΥΝΔΥΑΣΜΟΣ ΤΥΠΩΝ ΠΣ:

- ΠΣ κατά παραγγελία προς ΠΣ κατά παραγγελία ○ ΠΣ κατά παραγγελία προς Πακέτο
- ΠΣ κατά παραγγελία προς Ηλεκτρονικό Επιχειρείν ○ Πακέτο προς Πακέτο
- Πακέτο προς Ηλεκτρονικό Επιχειρείν
- Custom προς Πακέτο προς Ηλεκτρονικό Επιχειρείν
- ΑΠΑΙΤΗΣΕΙΣ ΟΛΟΚΛΗΡΩΣΗΣ:
 - Συντηρισινότητα ○ Ευελιξία
 - Επεκτασιμότητα ○ Φορητότητα
 - Επαναχρησιμοποιησιμότητα ○ Ωριμότητα Τεχνολογίας
 - Πολυπλοκότητα
 - Μη-επεμβατική Τεχνολογία ○ Απόδοση
 - Ολοκλήρωση Πραγματικού Χρόνου ○ Συμβατότητα με Mainframe

3.5.2.3 Κριτήρια Αξιολόγησης ΕΑΙ

Προκειμένου να διευκολυνθεί η αξιολόγηση των ΕΑΙ πακέτων, έχει προταθεί από συγγραφείς το παρακάτω πλαίσιο αξιολόγησης με τα εξής κριτήρια:

- Ολοκληρωμένο ή Σειρά Εργασιών
- Χαλαρή ή Σφικτή Σύνδεση
- Τυποποιημένη ή Εξατομηκευμένη ΕΑΙ
- Ενδοεπιχειρησιακή ή Διεπιχειρησιακή Ολοκλήρωση

3.5.2.4 Στάδια Ανάπτυξης ΕΑΙ

Τα στάδια ανάπτυξης εφαρμογών ΕΑΙ, σύμφωνα με την μεθοδολογία των Themistocleous και Irani (2007), είναι τα εξής:

- Μελέτη Σκοπιμότητας και Ανάλυσης Παραγόντων
- Δημιουργία Εναλλακτικών Σεναρίων Υλοποίησης και Αξιολόγηση
- Αναδιοργάνωση Επιχειρησιακών Διαδικασιών
- Αναδρόμηση Υπάρχουσας Υποδομής
- Ανάλυση Απαιτήσεων και Αξιολόγηση Λύσεων ΕΑΙ
- Δημιουργία Νέων Εφαρμογών

- Ολοκλήρωση και Έλεγχος
- Λειτουργία και Συντήρηση

3.5.2.5 Ιδιαιτερότητες EAI

Όπως κάθε τεχνική, έτσι και η EAI παρουσιάζει αρκετές ιδιαιτερότητες που πολλοί χαρακτηρίζουν ως μειονεκτήματα. Τα κυριότερα από αυτά είναι:

- Κανένα πακέτο EAI δεν λύνει όλα τα Προβλήματα Ολοκλήρωσης
- Ακριβό Λογισμικό
- Μεγάλη Γκάμα Γνώσεων
- Μεγάλες Αλλαγές
- Μεγάλη Αντίδραση
- Υψηλό Κόστος
- Χρονοβόρα Διαδικασία
- Υψηλό Πολυπλοκότητα
- Αρκετά Οργανωσιακά Προβλήματα
- Συγκεντρωτική Λύση
- Απώλεια Αυτονομίας
- Μειωμένη Διεσδυτικότητα σε Εφαρμογές Επιχειρησιακών Εταιριών

3.6 Υπηρεσιοστρεφείς Αρχιτεκτονικές (Service Oriented Architectures- SOA)

Οι Υπηρεσιοστρεφείς Αρχιτεκτονικές αποτελούν μετεξέλιξη των EAI και αποτελούνται από ένα σύνολο αρχών, τεχνολογιών, μεθόδων και πρακτικών που αποσκοπούν στο σχεδιασμό επαναχρησιμοποιήσιμων και ευέλικτων πληροφοριακών συστημάτων. Ο λόγος που οδήγησε στην εξέλιξη αυτή ήταν η έλλειψη της EAI σε επαναχρησιμοποίηση του κώδικα των εφαρμογών. Οι ΥΑ εστιάζουν στο θέμα της επαναχρησιμοποίησης, συγκεκριμένα όλη τους η δομή και λειτουργία βασίζεται σε αυτό. Αντί να δίνουν έμφαση στις διαδικασίες και στην ολοκλήρωση ορισμένων από αυτές, διασπούν τις διαδικασίες σε μικρότερα λογικά κομμάτια και ολοκληρώνουν το καθένα από αυτά. Αυτό καθιστά ευκολότερη την επαναχρησιμοποίηση των υποδιαδικασιών και την διασύνδεση των εφαρμογών.

Η ΥΑ θα μπορούσε να θεωρηθεί ένα αρχιτεκτονικό υπόδειγμα που χρησιμοποιεί ως βασικό συστατικό ολοκλήρωσης τις υπηρεσίες παγκόσμιου ιστού. Σύμφωνα με τους Rosen et al, (2008), για την υλοποίηση μιας τέτοιας αρχιτεκτονικής απαιτούνται υπηρεσίες που στηρίζονται στις ίδιες αρχές και έχουν κοινά χαρακτηριστικά. Δηλαδή έχουν παρόμοιο μέγεθος και δομή, κοινά πρότυπα, επικοινωνούν σε σημασιολογικό και τεχνικό επίπεδο και δεν παρουσιάζουν κενά στην λειτουργικότητά τους.

Εκτός από τα παραπάνω, βασικές αρχές για τις ΥΑ είναι και οι:

- Διαχωρισμός Λειτουργικών Μονάδων Λογισμικού
- Συνέπεια και Επαναχρησιμοποίηση
- Λογική Αφαίρεση
- Ευελιξία Αρχιτεκτονικής

Οι πρακτικές που προτείνονται για την τις ΥΑ είναι:

- Δημιουργία Οράματος και Στρατηγικής για την Αρχιτεκτονική
- Αποφυγή Φαινόμενου Big Bang
- Επιλογή Κατάλληλου Έργου για Αρχή
- Ανάπτυξη Μικρού Μέρους της Αρχιτεκτονικής
- Δημιουργία Βέλτιστων Πρακτικών
- Δημοσιοποίηση Αποτελεσμάτων
- Επιτυχία ή Αποτυχία

Οι ΥΑ βασίζονται στους Διαύλους Επιχειρησιακών Υπηρεσιών (ESB) και τις Υπηρεσίες Παγκόσμιου Ιστού (Web Services). Οι πρώτοι αφορούν την εξέλιξη των μηχανισμών Hub and Spoke της ΕΑΙ και αποτελούν βασικό δομικό συστατικό των ΥΑ γιατί σχετίζονται με την αυτοματοποίηση και την ολοκλήρωση των επιχειρησιακών υπηρεσιών και διαδικασιών και συνδέουν τις Υπηρεσίες Παγκόσμιου Ιστού μεταξύ τους. Σύμφωνα με τους Θεμιστοκλέους και Μαντζάνα (2010, σελ 138), οι ESB υποστηρίζουν διαλειτουργικότητα, ενέργειες διασύνδεσης υπηρεσιών, μεταφορά και μετάφραση δεδομένων των υπηρεσιών πελάτη και παραγωγού, έξυπνη δρομολόγηση μηνυμάτων ανάμεσα στα web services, λειτουργίες ασφαλείας, διαχείριση υπηρεσιών και παροχή διαδικασιών ελέγχου και σύνδεσης υπηρεσιών. Οι δεύτεροι αποτελούν μικρές, δικτυοκεντρικές, αυτόνομες, ανεξάρτητες και αυτοπεριγραφικές εφαρμογές που βοηθούν άλλα προγράμματα να πραγματοποιήσουν μία σειρά από ενέργειες που αυτά έχουν ορίσει. Κάθε web service μπορεί να επικοινωνήσει με άλλα, να ανταλλάξει δεδομένα με αυτά (μέσω XML), να πραγματοποιήσει ενέργειες για λογαριασμό ενός άλλου. Επίσης, τα web services έχουν την δυνατότητα να αλληλεπιδρούν με εφαρμογές, χωρίς αυτές να βρίσκονται στον ίδιο υπολογιστή με αυτά. Η απόσταση, επομένως, δεν παίζει κανένα ρόλο στην επικοινωνία των web services.

Για να αναπτυχθεί μια Υπηρεσιοστρεφής Αρχιτεκτονική, πρέπει να οριστούν οι επιχειρησιακές διαδικασίες, οι εφαρμογές και τα τμήματα λογισμικού που θα αποτελέσουν μέρος αυτής, οι πιο σημαντικές υπηρεσίες σε σύγκριση με όλες, οι σχέσεις και οι

αλληλεπιδράσεις μεταξύ των τμημάτων λογισμικού, ο τρόπος σύνθεσης των εφαρμογών, ο τρόπος ολοκλήρωσης, τα δεδομένα που θα ανταλλάσσονται, οι μεταχρηματισμοί των δεδομένων και η σημασιολογία και η ερμηνεία αυτών. [5]

4. Μεθοδολογία RUP

Το 1998 ο Jacobson δήλωσε ότι ο πόλεμος μεθόδων τελείωσε (Jacobson, 2000). Πίστευε ότι είχε επιτευχθεί ένα πρότυπο και αυτό το πρότυπο ήταν η ενοποιημένη γλώσσα μοντελοποίησης (UML). Ο Jacobson είχε δηλώσει ότι «όλες οι διαφορετικές μέθοδοι που είχαν ανακαλυφθεί στη βιομηχανία λογισμικού μετακινούνται τώρα σε μια γλώσσα μοντελοποίησης: την UML (Unified Modeling Language) και ότι αυτό το νέο πρότυπο περιγράφεται πολύ καλύτερα και πληρέστερα από όλες τις προηγμένες γλώσσες μοντελοποίησης. Αυτό ίσως να ήταν αλήθεια, αλλά ίσως και όχι. Παρ' όλα αυτά, για τις αντικειμενοστραφείς προσεγγίσεις η τυποποίηση της UML ήταν μια πρόοδος. [1]

Αναγνωρίστηκε ότι το να υπάρχει μια τυποποιημένη γλώσσα μοντελοποίησης, όπως η UML, δεν ήταν αρκετό. Όπως υποστηρίζει «χρειάζεται να γνωρίζεις κανείς πώς να τη χρησιμοποιήσει». Αυτό οδήγησε στην ανάπτυξη των συσχετιζόμενων διαδικασιών ανάπτυξης λογισμικού ή στην ανάπτυξη των μεθοδολογιών. Η διαδικασία με την οποία αναπτύσσεται ένα σύστημα, για παράδειγμα οι ενέργειες που πρέπει να γίνουν, το πώς προσδιορίζονται οι απαιτήσεις, τα στάδια, οι στόχοι κλπ δεν αποτελούν μέρος της UML. Η UML είναι απλά μια γλώσσα μοντελοποίησης. Είναι διαφορετικό πράγμα να ανακαλύπτεις τι πρέπει να μοντελοποιηθεί. Έτσι αναπτύχθηκε η εξελίχθηκε η διαδικασία η οποία είναι γνωστή ως Ενοποιημένη Διαδικασία (Unified Process) και η οποία αξιοποιεί τη μοντελοποίηση με χρήση UML.

Ο Jacobson εργάστηκε αρχικά στη βιομηχανία τηλεπικοινωνιών στην εταιρεία Ericson, όπου ανέπτυξε κάποιες ιδέες του για την Ενοποιημένη Διαδικασία, αλλά εγκατέλειψε την Ericson το 1987 για να ιδρύσει μια εταιρεία στην Στοκχόλμη, την Objectory. Το όνομά της προέρχεται από τις λέξεις “Object Factory” και το προϊόν της εταιρείας ήταν μια διαδικασία για την ανάπτυξη συστημάτων. Το 1995 η Rational Software Corporation εξαγόρασε την Objectory. Η Rational είχε επίσης αναπτύξει ένα πλήθος από πρακτικές ανάπτυξης λογισμικού σχετικά κυρίως με την αρχιτεκτονική με την αρχιτεκτονική λογισμικού και την επαναληπτική ανάπτυξη. Το αποτέλεσμα της συγχώνευσης ήταν αυτό που καλείται Rational Objectory Process και αργότερα το 1998 μετονομάστηκε σε Rational Unifies Process (RUP). Η RUP περιγράφεται ως μια ολοκληρωμένη διαδικασία ικανή να υποστηρίξει ολόκληρο τον κύκλο ζωής της ανάπτυξης λογισμικού.

Ο Jacobson δε χρησιμοποιεί τον όρο «μέθοδος» ή μεθοδολογία που συνηθίζεται για να περιγράψει την Ενοποιημένη Διαδικασία. Υποστηρίζει ότι «μία μέθοδος είναι ένα σύνολο από ιδέες που παρουσιάζουν ενδιαφέρον και γενικά αποτελούν βήμα προς βήμα περιγραφές. Όμως τυπικά δε βοηθά τους σχεδιαστές στο πώς να τη χρησιμοποιήσουν για την ανάπτυξη εμπορικού λογισμικού. Για να επιτύχει αυτό η μέθοδος πρέπει να είναι «προσαρμοσμένη» σε μια πραγματική διαδικασία τεχνολογίας λογισμικού. Επομένως, η Ενοποιημένη Διαδικασία είναι ουσιαστικά οι θεμελιώδεις και εννοιολογικές ιδέες της μεθοδολογίας του, ενώ η

Διαδικασία (για παράδειγμα RUP) αποτελεί κάποια τεχνολογία λογισμικού, που μετατρέπεται και παραδίδεται ως προϊόν.

Η Ενοποιημένη Διαδικασία περιγράφεται ως «κατευθυνόμενη από την περίπτωση χρήσης, επικεντρωμένη στην αρχιτεκτονική, επαναληπτική και με σταδιακή ανάπτυξη» (Jacobson, 1999). Αυτή η προοπτική υποστηρίζεται ότι αρκεί για να καταστήσει τη διαδικασία μοναδική. Στην Ενοποιημένη Διαδικασία χρησιμοποιούνται για να συλλάβουν τις απαιτήσεις του χρήστη. Μια περίπτωση χρήσης περιγράφει τη λειτουργικότητα ενός στοιχείου συστήματος, το οποίο δίνει στο χρήστη ένα «αποτέλεσμα κάποιας αξίας». Γίνεται εστίαση σε στοιχεία ιδιαίτερης αξίας για έναν συγκεκριμένο χρήστη (ή μια ομάδα χρηστών) και υπερνικούν μ' αυτόν τον τρόπο την αύξηση των «καταλόγων επιθυμίας» ή της ασαφούς γενικής λειτουργίας που είναι μεν συμπαθητική για το σύστημα αλλά δεν είναι ουσιαστική. Το σύνολο αυτών των περιπτώσεων χρήσης απεικονίζει τη συνολική λειτουργία του συστήματος. Υποστηρίζονται όμως ότι αποτελούν κάτι περισσότερο από αυτό επειδή χρησιμοποιούνται για τη διαδικασία ανάπτυξης, μέσω της σχεδίασης, της υλοποίησης και του ελέγχου. Επομένως το σύστημα σχεδιάζεται με βάση τις περιπτώσεις χρήσης, υλοποιείται για να υποστηρίξει τις περιπτώσεις χρήσης και εξετάζεται με βάση το περιεχόμενο των περιπτώσεων χρήσης.

Η RUP είναι μια διαδικασία επικεντρωμένη στην αρχιτεκτονική. Η αρχιτεκτονική λογισμικού συγκρίνεται με την αρχιτεκτονική ενός κτιρίου, δεδομένου ότι είναι ένα σχεδιάγραμμα του πλάνου οικοδόμησης το οποίο επιτρέπει στους ανθρώπους να δουν το κτίριο πριν πραγματικά χτιστεί. Η αρχιτεκτονική ενός συστήματος είναι παρόμοια αλλά παρέχει διαφορετικές απόψεις του συστήματος. Περιλαμβάνει τις λεπτομέρειες του υλικού, το λειτουργικό σύστημα, τη βάση δεδομένων, το δίκτυο κ.α. συν τις μη λειτουργικές απαιτήσεις (που μπορεί να αφορούν την αξιοπιστία, την απόδοση, την προσαρμογή, τα πρότυπα αλληλεπίδρασης με το χρήστη κλπ.). Η αρχιτεκτονική καθορίζεται στην αρχή επιγραμματικά αλλά εξελίσσεται και αξιοποιείται μαζί με τις ανάγκες του λογισμικού καθώς το σύστημα αναπτύσσεται. Πολλές μεθοδολογίες αγνοούν την ανάπτυξη της αρχιτεκτονικής και επικεντρώνεται μόνο στην εξέλιξη του λογισμικού μέρους του έργου.

Η RUP επίσης περιγράφεται ως επαναληπτική και ανάπτυξη σε στάδια επειδή οι συντάκτες της θεωρούν ότι οι απαιτήσεις χρηστών αρχικά δεν μπορούν απ' ευθείας να είναι πλήρως και ακριβώς καθορισμένες. Οι απαιτήσεις εξελίσσονται με την καλύτερη κατανόηση του συστήματος και τις αλλαγές που συμβαίνουν με την πάροδο του χρόνου. Επομένως, το έργο δε θεωρείται ότι αποτελεί μια μεγάλη δραστηριότητα που εκτελείται με ένα θεαματικό αποτέλεσμα στο τέλος, αλλά ως μια σειρά ελεγχόμενων σταδιακών ενεργειών που βοηθούν να ελαχιστοποιηθεί ο κίνδυνος και να μειωθεί η πιθανότητα αποτυχίας του συστήματος. Αυτό σημαίνει ότι δεν είναι μια προσέγγιση μοντέλου καταρράκτη ή κύκλου ζωής. Στην πραγματικότητα μοιάζει με το σπινάλ μοντέλο.

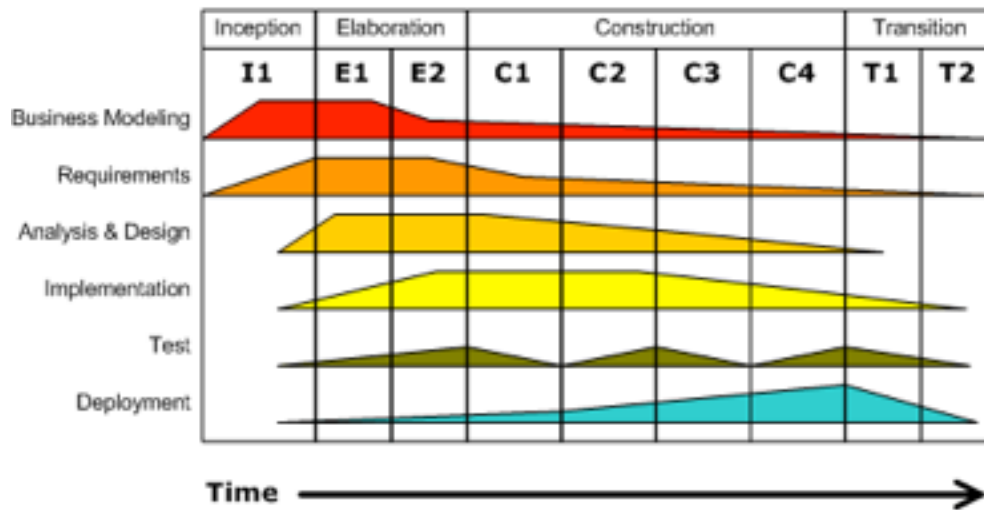
4.1 Ροές Εργασίας

Η RUP έχει ένα αριθμό «κύκλων» που όλοι μαζί συμβάλλουν στην ανάπτυξη του έργου και «εκτελούνται» καθ' όλη τη διάρκεια ζωής τους. Κάθε κύκλος περιλαμβάνει τέσσερις φάσεις: έναρξη, επεξεργασία, κατασκευή και μετάβαση. Η **εικόνα** δείχνει τις φάσεις και τις ροές εργασίας της RUP. Η έμφαση στη φάση της κατασκευής δίνεται στην υλοποίηση αλλά υπάρχουν επίσης και στοιχεία άλλων ροών εργασίας. Κάθε φάση αποτελείται από έναν αριθμό επαναλήψεων που περιλαμβάνουν όλες τις σημαντικές ροές εργασίας. Ο αριθμός θα καθοριστεί από τις συνθήκες. Έτσι η ανάπτυξη δεν αποτελεί ένα πέρασμα μέσα από τις ροές εργασίες αλλά μια σειρά επαναλήψεων γύρω από τις ροές για κάθε μία από τις τέσσερις φάσεις.

Μια ροή εργασίας είναι μια ακολουθία δραστηριοτήτων η οποία «παράγει ένα αποτέλεσμα σημαντικής αξίας». Υπάρχουν εννέα βασικοί πυρήνες διαδικασίας ροών εργασίας RUP. Οι ροές εργασίας **στην παραπάνω εικόνα** με τις πρώτες έξι να καλούνται ροές εργασίας της εφαρμοσμένης μηχανικής και τις υπόλοιπες τρεις ροές εργασίας υποστήριξης (π.χ. διαμόρφωση και διαχείριση αλλαγών, διαχείριση έργου και περιβάλλον).

Μια ενδιαφέρουσα πτυχή της RUP είναι η έννοια ενός εργαζόμενου, Δεν είναι στην πραγματικότητα ένα άτομο, αλλά κάποιος που παίζει αυτό το ρόλο. Η RUP ορίζει μια λίστα από όλους τους εργαζομένους που ενδεχομένως περιλαμβάνονται στη διαδικασία, από τον αρχιτέκτονα, τον αναλυτή συστημάτων, και το σχεδιαστή έως το μέτοχο, τον υπεύθυνο έργου και τον υπεύθυνο ελέγχου αλλαγών. Ένας εργαζόμενος σχετίζεται με ένα σύνολο «συνεκτικών» δραστηριοτήτων, δηλαδή δραστηριοτήτων που εκτελούνται από ένα άτομο, σε σχέση με το χειρισμό ενός εργαλείου. Ένα εργαλείο μπορεί να είναι μια περίπτωση χρήσης, ένα μοντέλο, ένα κομμάτι κώδικα κλπ. Είναι μέρος της φιλοσοφίας, τα εργαλεία να αναπτύσσονται και να διατηρούνται σε μια CASE εργαλειοθήκη. Η RUP περιλαμβάνει επίσης ένα σύνολο από οδηγίες οι οποίες σχετίζονται με τις διαδικασίες, τις δραστηριότητες ή τα βήματα. Μπορεί να καθορίζουν ποια είναι η καλύτερη πρακτική, πώς κατασκευάζεται ένα καλό εργαλείο, ή μια ευρετική μέθοδο. Έτσι, δίνουν συμβουλές για το πώς θα διεξαχθεί μια αξιολόγηση, πώς να διαμορφωθεί μια περίπτωση χρήσης, πώς να ελεγχθεί κάτι, κλπ.

Iterative Development
Business value is delivered incrementally in time-boxed cross-discipline iterations.



Εικόνα 4.1: Ροές Εργασίας της Μεθοδολογίας RUP.

4.1.1 Η Ροή Εργασίας της Μοντελοποιημένης Επιχείρησης

Ξεκινά με την αξιοποίηση του Επιχειρησιακού Μοντέλου. Ουσιαστικά, θέτει το πλαίσιο για το σύστημα που αναπτύσσεται και τη μορφή της οργάνωσης στην οποία το σύστημα πρόκειται να επεκταθεί. Μπορεί να συμπεριλάβει τον προσδιορισμό των επιχειρησιακών κανόνων, κλπ. Ανάλογα με τη φύση της προτεινόμενης διαδικασίας ανάπτυξης,

Αυτή η ροή εργασίας δεν είναι πάντα απαραίτητη, για παράδειγμα, εάν η επέκταση αφορά απλά την προσθήκη ενός νέου χαρακτηριστικού σ' ένα υπάρχον σύστημα. Στην προκειμένη περίπτωση της διπλωματικής εργασίας, τίθεται το ζήτημα της ανάπτυξης ενός καινούριου συστήματος εξ' ολοκλήρου, επομένως και η ροή εργασίας καθίσταται πλήρως απαραίτητη. Οι αντικειμενικοί στόχοι αυτής της ροής εργασίας είναι σχεδόν παρόμοιοι μ' αυτούς των άλλων μεθοδολογιών, αλλά στη RUP οι ίδιες τεχνικές χρησιμοποιούνται σε όλα τα επόμενα στάδια της ανάπτυξης (εργαζόμενοι, διαδικασίες, δραστηριότητες και ροές εργασίας), εξασφαλίζοντας κατά συνέπεια μια άκρη- προς- άκρη διαδικασία όπου όλοι μαζί μιλούν μια κοινή γλώσσα. Οι βασικές διαδικασίες της μοντελοποίησης της επιχείρησης είναι ως εξής:

- Ένα έγγραφο του οράματος της επιχείρησης, όπου διευκρινίζονται οι στόχοι ανάπτυξης.

- Ένα μοντέλο περίπτωσης χρήσης της επιχείρησης, που απεικονίζει τις λειτουργίες της προτεινόμενης ανάπτυξης.
- Ένα μοντέλο επιχειρησιακών στόχων, που περιγράφει την πραγματοποίηση των επιχειρησιακών περιπτώσεων χρήσης.

4.1.2 Η Ροή Εργασίας των Απαιτήσεων

Οι στόχοι αυτής της ροής εργασίας είναι να καθοριστούν μαζί με όσους συμμετέχουν στο σύστημα, τι πρέπει να κάνει το σύστημα και γιατί, ώστε να καθοριστούν τα όρια του συστήματος και να υπολογιστούν οι δαπάνες και να καθοριστούν τα όρια του συστήματος και να υπολογιστούν οι δαπάνες και τα χρονοδιαγράμματα. Αναπτύσσεται σε ένα «όραμα» του συστήματος το οποίο στη συνέχεια μεταφράζεται σε ένα μοντέλο περίπτωσης χρήσης με μερικές συμπληρωματικές προδιαγραφές απαιτήσεων. Επίσης, συλλέγονται και αναλύονται οι λειτουργικές και μη λειτουργικές απαιτήσεις. Καθορίζονται οι βασικές ανάγκες μετόχων και χρηστών και τα υψηλού επιπέδου χαρακτηριστικά γνωρίσματα και στη συνέχεια μετατρέπονται σε συγκεκριμένες απαιτήσεις λογισμικού. Και γι' αυτό χρησιμοποιείται ένα μοντέλο περίπτωσης χρήσης.

4.1.3 Η Ροή Εργασίας Ανάλυσης και Σχεδιασμού

Ο στόχος αυτής της ροής εργασίας είναι να μετατραπούν οι απαιτήσεις από το προηγούμενο στάδιο σε μια προδιαγραφή υλοποίησης. Η ανάλυση εξασφαλίζει ότι οι λειτουργικές απαιτήσεις καλύπτονται, αγνοώντας χαρακτηριστικά μη- λειτουργικές απαιτήσεις και το περιβάλλον της εκτέλεσης. Έτσι η ανάλυση είναι μια λογική άποψη του σχεδίου του συστήματος. Ο σχεδιασμός παίρνει το αποτέλεσμα της ανάλυσης και το προσαρμόζει στους περιορισμούς της αρχιτεκτονικής και των μη λειτουργικών απαιτήσεων. Περιλαμβάνει τις δραστηριότητες του καθορισμού και της βελτίωσης της αρχιτεκτονική, της ανάλυσης της συμπεριφοράς (λειτουργίες), του σχεδιασμού των δομικών συστατικών και του σχεδιασμού της βάσης δεδομένων.

4.1.4 Η Ροή Εργασίας της Υλοποίησης

Αυτή ροή εργασίας πρόκειται να μετατρέψει τα σχέδια σε μια εφαρμογή. Περιλαμβάνει τον προγραμματισμό της διαδικασίας, τη μετατροπή των κλάσεων και των αντικειμένων από το παραπάνω στάδιο στα δομικά συστατικά τους, τον έλεγχο των επιμέρους συστατικών και την κατασκευή μιας πρώτης λειτουργικής έκδοσης του

συστήματος. Τα χωριστά τμήματα λογισμικού διαδοχικά ενσωματώνονται στο πλήρες σύστημα μέσα σε μια εύλογη χρονική περίοδο.

4.1.5 Η Ροή Εργασίας του Ελέγχου

Αυτή ροή εργασίας εξετάζει και ελέγχει την αλληλεπίδραση των δομικών συστατικών, ότι δηλαδή όλες οι απαιτήσεις έχουν εφαρμοστεί και ότι ένα ποιοτικό προϊόν έχει αναπτυχθεί από την άποψη ότι δεν έχει λάθη και ότι είναι κατάλληλο για τον σκοπό για τον οποίο κατασκευάστηκε. Το σύστημα εξετάζεται για την αξιοπιστία, τη λειτουργικότητα και την απόδοσή του. Ο έλεγχος δεν είναι μία ενιαία φάση του έργου. Πραγματοποιείται σε όλο τον κύκλο ζωής και σε όλα τα στάδια. Σαφώς διαφοροποιείται από τα αρχικά στάδια στα μεταγενέστερα. Ο έλεγχος μπορεί να περιλαμβάνει ελέγχους δοκιμασίας επιδόσεων, ελέγχους εγκατάστασης, ελέγχους λειτουργίας, ελέγχους αξιοπιστίας, ελέγχους φόρτου, ελέγχους απόδοσης, κλπ. Οι έλεγχοι για να είναι επιτυχείς μπορεί να περιλαμβάνουν τη δημιουργία περιπτώσεων χρήσης, διαδικασιών, σεναρίων κλπ., τα οποία βοηθούν στην πραγματοποίηση ενός ολοκληρωμένου ελέγχου.

4.1.6 Η Ροή Εργασίας της Παράδοσης

Αυτή η ροή εργασίας παραδίδει το ολοκληρωμένο λογισμικό στους χρήστες και περιλαμβάνει:

- Έλεγχο της λειτουργίας του λογισμικού στην επιχείρηση (έλεγχος- Βήτα).
- Κατάρτιση των τελικών χρηστών.
- Μετάβαση από το υπάρχον λογισμικό (περιλαμβάνει μετατροπή των βάσεων δεδομένων)
- Εγκατάσταση λογισμικού.

Η ακριβής φύση της ροής εργασίας της παράδοσης θα εξαρτηθεί από το εάν το σύστημα είναι ένα παραδοσιακό, το οποίο κατασκευάστηκε ειδικά για την εταιρεία, ένα εσωτερικά αναπτυγμένο κομμάτι λογισμικού ή ένα έτοιμο πακέτο που παραδίδεται μέσω του διαδικτύου. Η ροή εργασίας της παράδοσης είναι σημαντική στη RUP και τονίζεται ότι οι δραστηριότητες της παράδοσης συχνά αγνοούνται από άλλες μεθοδολογίες και προσεγγίσεις.

4.1.7 Η Ροή Εργασίας Διαχείρισης και Διαμόρφωσης Αλλαγών

Αυτή η ροή εργασίας παρακολουθεί και συντηρεί την ακεραιότητα του έργου. Τα εργαλεία που αναπτύσσονται στα πλαίσια του έργου αντιπροσωπεύουν μια σημαντική επένδυση και η χρήση τους πρέπει να μεγιστοποιηθεί. Τα εργαλεία πρέπει να προσδιοριστούν και να αποθηκευτούν και οι διάφορες εκδόσεις να ελεγχθούν από την εταιρία. Η ροή εργασίας περιλαμβάνει την παρακολούθηση και διαχείριση των αιτημάτων αλλαγής, των δαπανών αλλαγής και τη διατήρηση επαφής με τις διάφορες εκδόσεις των προϊόντων και των διαδικασιών. Περιλαμβάνει επίσης τη διαχείριση της σύνθεσης του υλικού και του λογισμικού. Η χρήση εργαλείων προτείνεται για αυτό που περιγράφεται ως «βαρετές πτυχές» της ροής εργασίας.

4.1.8 Η Ροή Εργασίας Διαχείρισης Έργου

Αυτή η ροή εργασίας παρέχει ένα πλαίσιο για τη διαχείριση των έργων λογισμικού και τη διαχείριση του κινδύνου. Παρέχει επίσης τις οδηγίες για τον προγραμματισμό, τη στελέχωση, τον έλεγχο και γενικά τη διαχείριση του έργου. Η RUP αναγνωρίζει ότι η διαχείριση έργου είναι μια ιδιαίτερα δύσκολη πτυχή της ανάπτυξης λογισμικού και αφιερώνει έτσι μια ροή εργασίας ώστε να εξασφαλιστεί ότι ένα έργο θα είναι επιτυχές από αυτή την άποψη και ειδικότερα ως προς τους κινδύνους κατά την εκτέλεσή του.

Στον προγραμματισμό υπάρχουν δυο επίπεδα σχεδίων, ένα «ακατέργαστο σχέδιο» (το σχέδιο φάσης) και ένα σύνολο «καλοδεχόμενων σχεδίων» (οι φάσεις που επαναλαμβάνονται). Το σχέδιο φάσης αφορά τα σημαντικότερα κύρια σημεία του έργου, από την έγκριση του έργου έως την παραγωγή προϊόντων, ενώ τα σχέδια επανάληψης εξετάζουν τη λεπτομέρεια της τρέχουσας επανάληψης και ενδεχομένως της επόμενης επανάληψης. Μέρος αυτής της ροής εργασίας εξετάζει του πιθανούς κινδύνους για το έργο, π.χ. αλλαγές απαιτήσεων ή δυσκολίες στη στρατολόγηση των βασικών ειδικοτήτων. Αυτοί οι κίνδυνοι αξιολογούνται ως προς την πιθανότητά τους και τον πιθανό αντίκτυπό τους στο έργο. Εάν ένας κίνδυνος αξιολογείται ως σοβαρός τότε γίνονται προσπάθειες για να περιοριστεί ο κίνδυνος και καταστρώνονται σχέδια αντιμετώπισής του, σε περίπτωση που ο κίνδυνος εμφανιστεί τελικά. Ως τμήμα της διαχείρισης του έργου η RUP συστήνει την καθιέρωση και διατήρηση λεπτομερών μετρικών για αυτό. Οι μετρικές αυτές μπορεί να αναφέρονται στην πρόοδο, στην παραγωγικότητα, στα επίπεδα επαναχρησιμοποίησης, στην ικανοποίηση των πελατών, κλπ. και είναι σημαντικές όχι μόνο για να διαχειριστούν αποτελεσματικά το τρέχον έργο αλλά και για την εκμάθηση και τη βελτίωση της διαχείρισης των μελλοντικών έργων.

[1]

4.1.9 Η Ροή Εργασίας του Περιβάλλοντος

Αυτή ροή εργασίας ασχολείται με την υποστήριξη του έργου από τις σχετικές διαδικασίες, τις μεθόδους και τα εργαλεία σε ένα οργανισμό. Όπως έχουμε δει η χρήση των εργαλείων είναι ένα βασικό στοιχείο στη RUP, έτσι οι δραστηριότητες αφορούν την επιλογή, την προμήθεια, την εφαρμογή και τη διαχείριση των κατάλληλων εργαλείων και διαδικασιών υποστήριξης.

Η RUP, σύμφωνα με τη Rational, βασίζεται στην εμπειρία και τις καλύτερες πρακτικές και είναι κατάλληλη για ένα ευρύ φάσμα έργων και οργανισμών. Είναι προσανατολισμένη στις περιπτώσεις χρήσης, εστιάζει στην ανάπτυξη του λογισμικού με επαναληπτικό τρόπο και παρέχει τις απαραίτητες φάσεις, ροές εργασίας, οδηγίες και πλαίσια για την ανάπτυξη του λογισμικού. Σχεδιάζεται για να χρησιμοποιεί και να αξιοποιεί την UML για τα στοιχεία μοντελοποίησής της και είναι βασισμένη στην ακέραια χρήση των εργαλείων για να υποστηρίξει τις διαδικασίες της.

4.2 Ενοποιημένη Γλώσσα Μοντελοποίησης - Unified Modeling Language (UML)

Σύμφωνα με το επίσημο εγχειρίδιο αναφοράς, η Ενοποιημένη Γλώσσα Μοντελοποίησης (Unified Modeling Language ή UML) είναι μια γραφική γλώσσα γενικού σκοπού, η οποία χρησιμοποιείται για τον προσδιορισμό, οπτικοποίηση, ανάπτυξη και τεκμηρίωση των κατασκευασμάτων (artifacts) ενός συστήματος λογισμικού [2]. Η UML αποτελεί industry standard για τη μοντελοποίηση συστημάτων λογισμικού και χρησιμοποιείται στη μοντελοποίηση συστημάτων βασισμένων σε αντικείμενα (αντικειμενοστρεφή συστήματα).

Βασικό χαρακτηριστικό της είναι ότι αποτελεί μια γλώσσα μοντελοποίησης ανεξάρτητη από τις μεθοδολογίες που χρησιμοποιούνται κατά την ανάπτυξη συστημάτων λογισμικού. Επιπλέον, δεν πρόκειται μόνο για μια τυποποίηση και ανακάλυψη μιας κοινής σημειολογίας. Περιέχει νέες και ενδιαφέρουσες έννοιες που δεν υπάρχουν εν γένει στο πεδίο της αντικειμενοστρεφούς ανάπτυξης, όπως για παράδειγμα η περιγραφή και χρησιμοποίηση προτύπων (patterns) σε μια γλώσσα μοντελοποίησης, η χρησιμοποίηση της έννοιας του στερεοτύπου (stereotype) για την επέκταση της γλώσσας, η παροχή πλήρους ιχνηλασιμότητας (traceability) από τα εννοιολογικά μοντέλα ενός συστήματος στα εκτελέσιμα συστατικά της φυσικής αρχιτεκτονικής. Συνεπώς, η κατανόηση της UML δεν περιορίζεται στην εκμάθηση των συμβόλων και της σημασίας τους, αλλά εκτείνεται σε ευρύτερο πλαίσιο στη μάθηση της αντικειμενοστρεφούς μοντελοποίησης.

Η UML χρησιμοποιείται για τη μοντελοποίηση μεγάλου εύρους συστημάτων. Ο στόχος της UML είναι να περιγράψει κάθε τύπο συστήματος, μέσα από αντικειμενοστρεφή

διαγράμματα. Η πιο συνήθης χρήση της είναι η παραγωγή μοντέλων συστημάτων λογισμικού, πέρα από αυτό όμως χρησιμοποιείται για την περιγραφή συστημάτων που δεν αφορούν λογισμικό, όπως για παράδειγμα μηχανικών συστημάτων. Οι κυριότερες κατηγορίες συστημάτων στα οποία χρησιμοποιείται η UML είναι οι εξής: πληροφοριακά συστήματα, τεχνολογικά συστήματα, συστήματα λογισμικού, ενσωματωμένα συστήματα πραγματικού χρόνου, καταναμημένα συστήματα, καθώς και συστήματα επιχειρήσεων.[6]

4.2.1 Τα διαγράμματα της UML

Τα κυριότερα διαγράμματα της UML είναι τα εξής:

- **Διάγραμμα Περιπτώσεων Χρήσης (Use Case Diagram)**

Το διάγραμμα περιπτώσεων χρήσης στη UML χρησιμοποιείται για την μοντελοποίηση της λειτουργικότητας ενός συστήματος, όπως αυτή γίνεται αντιληπτή από τον εξωτερικό χρήστη. Τα διαγράμματα αυτά διαμερίζουν τη λειτουργικότητα του συστήματος σε συναλλαγές που έχουν νόημα για τους χρήστες του συστήματος ή αλλιώς χειριστές (actors). Τα επιμέρους τμήματα της λειτουργικότητας ονομάζονται περιπτώσεις χρήσης (use cases). Το σύνολο των περιπτώσεων χρήσης συνιστούν τη συμπεριφορά του συστήματος. Τα βασικά διαγραμματικά στοιχεία του διαγράμματος περιπτώσεων χρήσης είναι το σύστημα, ο χειριστής, η περίπτωση χρήσης και οι σχέσεις μεταξύ τους. Η αξία του διαγράμματος περιπτώσεων χρήσης είναι ιδιαίτερα σημαντική, διότι καθορίζει τις λειτουργικές απαιτήσεις, οι οποίες θα αποτελέσουν σημείο αναφοράς καθ' όλη τη διάρκεια ανάπτυξης του συστήματος. Ο σημαντικότερος ρόλος του συγκεκριμένου διαγράμματος είναι ότι αποτελεί ένα μέσο επικοινωνίας μεταξύ πελατών και σχεδιαστών, όσον αφορά στη λειτουργικότητα του συστήματος. Η απλότητα των συμβολισμών το καθιστά ιδανικό για αυτό το σκοπό, παρέχοντας τη δυνατότητα εύκολης αντίληψης του συνόλου των λειτουργιών καθώς και εύκολης τροποποίησής τους.

Ο χειριστής αντιπροσωπεύει μια εξωτερική οντότητα, άνθρωπο ή σύστημα, η οποία αλληλεπιδρά με το σύστημα. Ο χειριστής αναπαριστά ένα ρόλο, όχι έναν μεμονωμένο χρήστη του συστήματος, μιας και ο ίδιος χρήστης μπορεί να αλληλεπιδρά με το σύστημα με πολλαπλούς ρόλους. Οι χειριστές είναι κλάσεις με το στερεότυπο «actor», όπου το όνομα της κλάσης γενικά αναπαριστά το ρόλο του χειριστή. Στο διάγραμμα περιπτώσεων χρήσης χρησιμοποιείται μόνο η σχέση γενίκευσης ανάμεσα σε χειριστές, προκειμένου να περιγραφεί η κοινή συμπεριφορά ανάμεσα τους, την οποία και κληρονομούν από μια πρόγονο κλάση χειριστή.

Ο τυπικός ορισμός μιας περίπτωσης χρήσης είναι μια ακολουθία ενεργειών που πραγματοποιείται από το σύστημα για την παραγωγή μετρήσιμων αποτελεσμάτων που έχουν νόημα για τον χρήστη. Η περίπτωση χρήσης ορίζει ένα συγκεκριμένο τρόπο χρησιμοποίησης του συστήματος, προσδιορίζοντας την αλληλεπίδραση ανάμεσα σε έναν ή περισσότερους χειριστές και το σύστημα. Το στιγμιότυπο μιας περίπτωσης χρήσης ονομάζεται σενάριο

(scenario), και αναπαριστά ένα συγκεκριμένο μονοπάτι εκτέλεσης (execution path) μέσα στο σύστημα.

Η περίπτωση χρήσης έχει τα ακόλουθα χαρακτηριστικά:

6. Ξεκινάει πάντα από ένα χειριστή.
7. Πρέπει να επιστρέφει κάποιου είδους απτή πληροφορία στο χρήστη.
8. Μια περίπτωση χρήσης είναι πλήρης, με την έννοια ότι αποτελεί μια πλήρη περιγραφή. Μια περίπτωση χρήσης δε θεωρείται ότι έχει ολοκληρωθεί μέχρις ότου η τελική πληροφορία παραχθεί, ακόμη κι αν απαιτούνται γι' αυτό πολλαπλές αλληλεπιδράσεις μεταξύ των αντικειμένων. Ένα σύνθημα λάθος είναι η διαίρεση μιάς περίπτωσης χρήσης σε μικρότερες, οι οποίες παράγουν ενδιάμεσα αποτελέσματα.

Ανάμεσα στις περιπτώσεις χρήσης υπάρχουν τρία είδη σχέσεων: η επέκταση (extends), η συμπερίληψη (uses ή includes) και η ομαδοποίηση (grouping). Η σχέση της επέκτασης είναι μια σχέση γενίκευσης που χρησιμοποιείται στην περίπτωση όπου μια περίπτωση χρήσης συμπεριλαμβάνει ένα τμήμα, όχι απαραίτητα ολόκληρη, την συμπεριφορά της περίπτωσης χρήσης που επεκτείνει. Τέτοιου είδους περιπτώσεις χρήσης χρησιμοποιούνται στο χειρισμό εξαιρέσεων. Η σχέση της συμπερίληψης είναι και αυτή μια σχέση γενίκευσης που χρησιμοποιείται στην περίπτωση όπου μια περίπτωση χρήσης συμπεριλαμβάνει την πλήρη λειτουργικότητα μιας άλλης. Όταν ένα σύνολο περιπτώσεων χρήσης παρουσιάζουν σε κάποια τμήματα κοινή συμπεριφορά, η σχέση αυτή χρησιμοποιείται για τη μοντελοποίηση αυτής της κοινής συμπεριφοράς σε μια περίπτωση χρήσης που χρησιμοποιείται από τις υπόλοιπες. Τέλος με τη σχέση της ομαδοποίησης, περιπτώσεις χρήσης, οι οποίες διαθέτουν παρόμοια συμπεριφορά ή σχετίζονται με κάποιο τρόπο μεταξύ τους, οργανώνονται σε πακέτα. Ωστόσο, για λόγους απλότητας των διαγραμμάτων περιπτώσεων χρήσης η τελευταία σχέση συνήθως δε χρησιμοποιείται.

Η περιγραφή των περιπτώσεων χρήσης γίνεται με τη μορφή κειμένου στην ορολογία του χρήστη και αποτελεί μια απλή και συνεπή τεκμηρίωση. Η τεκμηρίωση κάθε περίπτωσης χρήσης, για να είναι πλήρης, θα πρέπει να περιλαμβάνει μια ακολουθία γεγονότων που λαμβάνουν χώρα για την υλοποίηση της επιθυμητής συμπεριφοράς. Επικεντρώνεται στην εξωτερική συμπεριφορά του συστήματος, αγνοώντας τον τρόπο υλοποίησης και την εσωτερική δομή του. Μερικά σημεία τα οποία θα πρέπει να συμπεριλαμβάνονται στην περιγραφή είναι ο στόχος της περίπτωσης χρήσης, από ποιόν χειριστή ξεκινάει, η ακολουθία των μηνυμάτων μεταξύ χειριστή και συστήματος, εναλλακτική ροή γεγονότων σε περιπτώσεις εξαιρέσεων, και τέλος το πώς η περίπτωση χρήσης τερματίζεται επιστρέφοντας κάποια τιμή στο χειριστή.

- **Διάγραμμα Κλάσεων (Class Diagram)**

Το διάγραμμα κλάσεων είναι ο πρώτος τύπος διαγράμματος της UML, και ο οποίος έχει άμεση σχέση με τα αντικειμενοστρεφή συστήματα. Σε ένα αντικειμενοστρεφές σύστημα τα δομικά στοιχεία του είναι οι κλάσεις και οι σχέσεις μεταξύ των κλάσεων, οι οποίες επιτρέπουν τη συνεργασία αντικειμένων που δημιουργούνται ως στιγμιότυπα των κλάσεων.

Το διάγραμμα κλάσεων αποτελείται από τις κλάσεις του συστήματος και τις μεταξύ τους συσχετίσεις, περιγράφοντας με αυτό τον τρόπο τη στατική δομή του συστήματος.

Το διάγραμμα κλάσεων μπορεί να χρησιμοποιηθεί σε διάφορες φάσεις της ανάπτυξης του συστήματος. Στο αρχικό στάδιο της ανάλυσης απαιτήσεων οι κατασκευαστές αρχίζουν να αποκτούν γνώση για το πεδίο του προβλήματος του συστήματος. Αυτή η αρχική κατανόηση των εννοιών του πεδίου του προβλήματος καταγράφεται σε ένα διάγραμμα κλάσεων, το οποίο ονομάζεται μοντέλο του πεδίου προβλήματος (problem domain model). Στο μοντέλο αυτό καταγράφονται ως κλάσεις οι έννοιες του πεδίου του προβλήματος και οι μεταξύ τους συσχετίσεις. Έπειτα, στο στάδιο της ανάλυσης, με οδηγό το μοντέλο του πεδίου προβλήματος, κατασκευάζεται ένα διάγραμμα κλάσεων, το οποίο αναπαριστά τη βασική αρχιτεκτονική δομή του συστήματος. Σε αυτό το στάδιο οι κλάσεις πρέπει να επιδιώκουν την αναπαράσταση του συστήματος που μοντελοποιείται με την ελάχιστη δυνατή πληροφορία, χωρίς να επιχειρείται αναφορά σε θέματα υλοποίησης. Στη συνέχεια, μεταβαίνοντας στο στάδιο της σχεδίασης, η περιγραφή των κλάσεων συμπληρώνεται με τις λειτουργίες που υλοποιούν τη συμπεριφορά των αντικειμένων και με επιπρόσθετες ιδιότητες ή συσχετίσεις, που επιβάλλονται από το περιβάλλον υλοποίησης. Τέλος, κατά την υλοποίηση του συστήματος, είναι δυνατόν να επέλθουν τροποποιήσεις στη δομή των κλάσεων λόγω απαιτήσεων που σχετίζονται με απόκρυφη πληροφορία, ορατότητα και άλλες μη λειτουργικές απαιτήσεις, όπως π.χ. απόδοση και ασφάλεια.

Σε μερικές περιπτώσεις, το διάγραμμα κλάσεων είναι το μόνο είδος διαγράμματος της UML που χρησιμοποιείται, λόγω των πληροφοριών που παρέχει σχετικά με τον πηγαίο κώδικα. Υπάρχει η δυνατότητα αυτόματης παραγωγής τμημάτων κώδικα από το διάγραμμα κλάσεων, καθώς και η αυτόματη δημιουργία διαγραμμάτων κλάσεων λαμβάνοντας ως είσοδο τον πηγαίο κώδικα. Για το λόγο αυτό, ο κάθε συμβολισμός είναι σημαντικός, ακόμα κι αν υποδηλώνεται με ένα στοιχειώδες σύμβολο στο διάγραμμα κλάσεων.

- Διάγραμμα Ακολουθίας (Sequence Diagram)

Το διάγραμμα ακολουθίας παρουσιάζει την αλληλεπίδραση μεταξύ αντικειμένων σε δύο διαστάσεις. Η κάθετη διάσταση αντιστοιχεί στην κλίμακα του χρόνου, ενώ στην οριζόντια διάσταση συμβολίζονται τα ανεξάρτητα αντικείμενα. Τα αντικείμενα συμβολίζονται με παραλληλόγραμμα μέσα στα οποία μπορεί να σημειωθεί το όνομα του στιγμιότυπου του αντικειμένου που συμμετέχει στο σενάριο που απεικονίζεται και ακολουθεί μετά από άνω-κάτω τελεία το όνομα της κλάσης στην οποία ανήκει το αντικείμενο.

Σε κάθε αντικείμενο αντιστοιχεί μια κάθετη γραμμή που ονομάζεται γραμμή ζωής (lifeline). Τα αντικείμενα ανταλλάσσουν μηνύματα, τα οποία στην επίσημη ορολογία της UML ονομάζονται ερεθίσματα (stimuli). Ένα μήνυμα που αποστέλλεται μεταξύ των αντικειμένων συμβολίζεται ως ένα βέλος από τη γραμμή ζωής ενός αντικειμένου προς τη γραμμή ζωής ενός άλλου. Μήνυμα μπορεί να είναι οτιδήποτε από τα εξής:

1. Κλήση μιας λειτουργίας: όταν ένα αντικείμενο καλεί μια λειτουργία ενός άλλου αντικειμένου. Πρόκειται για σύγχρονο μήνυμα, δηλαδή ο αποστολέας του μηνύματος θα πρέπει να περιμένει την ολοκλήρωση της λειτουργίας για να συνεχίσει. Η κεφαλή του

βέλους είναι γεμισμένη με μαύρο χρώμα. Πάνω από το βέλος αναγράφεται το όνομα της λειτουργίας που καλείται, με τις ενδεχόμενες παραμέτρους σε παρενθέσεις. Ειδική περίπτωση κλήσης είναι η αυτοκλήση, η οποία ξεκινάει από το αντικείμενο και καταλήγει πάλι σε αυτό.

2. Σήμα: όταν ένα αντικείμενο αποστέλλει ένα ασύγχρονο μήνυμα σε ένα άλλο αντικείμενο. Τυπικά ασύγχρονα μηνύματα συναντάμε σε πολυνηματικές εφαρμογές, όπου ένα μήνυμα τοποθετείται σε κάποια ουρά ενός νήματος εκτέλεσης ενώ το ενεργό αντικείμενο-παραλήπτης θα επεξεργαστεί το μήνυμα σε κάποια επόμενη χρονική στιγμή. Η διαφορά με την κλήση λειτουργίας στον συμβολισμό είναι πως η κατάληξη είναι ένα ανοιχτό βέλος.
3. Επιστροφή κλήσης: είναι ένα διακεκομμένο βέλος το οποίο συμβολίζει την επιστροφή από μία κλήση λειτουργίας. Πάνω στο διακεκομμένο βέλος αναγράφεται συνήθως η τιμή επιστροφής, αν υπάρχει..
4. Μηνύματα υπό συνθήκη: Στα μηνύματα υπό συνθήκη τοποθετούνται αγκύλες μέσα στις οποίες αναγράφεται μία συνθήκη που μπορεί να είναι αληθής ή ψευδής. Η σημασία του συμβολισμού είναι ότι το μήνυμα θα αποσταλεί μόνο αν η συνθήκη είναι αληθής. Αν θέλουμε ταυτόχρονα να δείξουμε μια αποστολή εναλλακτικού μηνύματος στην περίπτωση που η συνθήκη είναι ψευδής, τότε δείχνουμε τα δύο αμοιβαία αποκλειόμενα μηνύματα σαν μηνύματα με το ίδιο σημείο εκκίνησης και γράφουμε στο πρώτο τη συνθήκη και στο δεύτερο τη φράση [else].

Σε ένα διάγραμμα ακολουθίας είναι δυνατόν να προστεθούν και περισσότεροι συμβολισμοί που υποδηλώνουν βρόχους επανάληψης, μηνύματα που αποστέλλονται πολλαπλές φορές, συγχρονισμό νημάτων κ.ο.κ. Ωστόσο, ο σκοπός των διαγραμμάτων ακολουθίας δεν είναι να αποτυπώσουν τις λεπτομέρειες ενός αλγορίθμου αλλά να αναπαραστήσουν με απλό και κατανοητό τρόπο τα σενάρια συνεργασίας μεταξύ αντικειμένων.

- **Διάγραμμα Συνεργασίας (Collaboration Diagram)**

Σε ένα διάγραμμα συνεργασίας απεικονίζονται τα συνεργαζόμενα αντικείμενα και οι συσχετίσεις μεταξύ τους. Ενώ τα διαγράμματα ακολουθίας απεικονίζουν κυρίως τη ροή των μηνυμάτων σε ένα σενάριο μιας περίπτωσης χρήσης, τα διαγράμματα συνεργασίας χρησιμοποιούνται για να παρουσιάσουν τις σχέσεις μεταξύ αντικειμένων.

Πλησίον των συνδέσεων εμφανίζονται ως μικρότερες ακμές τα μηνύματα που αποστέλλονται. Για να απεικονιστεί η ακολουθία των μηνυμάτων που ανταλλάσσονται χρησιμοποιείται αρίθμηση των μηνυμάτων. Τα διαγράμματα ακολουθίας και συνεργασίας θεωρούνται συμπληρωματικά, καθώς περιέχουν την ίδια πληροφορία αλλά κάθε ένα δίνει μια διαφορετική οπτική γωνία (σε πολλά εργαλεία το ένα είδος διαγράμματος παράγεται αυτόματα από το άλλο). Από ένα τέτοιο διάγραμμα είναι εύκολο να αντιληφθεί κανείς την ομάδα των συνεργαζόμενων αντικειμένων και την ύπαρξη των συνδέσεων μεταξύ τους αλλά είναι δυσκολότερο να οπτικοποιηθεί η ροή των μηνυμάτων, η οποία φαίνεται καλύτερα στο διάγραμμα ακολουθίας.

- **Διάγραμμα Καταστάσεων (Statechart Diagram)**

Το διάγραμμα καταστάσεων χρησιμοποιείται για την περιγραφή της ροής του ελέγχου σε ένα σύστημα εστιάζοντας στις αλλαγές κατάστασης που λαμβάνουν χώρα σε ένα αντικείμενο. Πολύ συχνά οι προδιαγραφές ενός συστήματος μπορούν να καθοριστούν βάσει μιας μηχανής πεπερασμένων καταστάσεων (finite state machine) ή απλά μηχανής καταστάσεων. Συνήθως, μια μηχανή καταστάσεων περιγράφεται ως ένας γράφος όπου οι κόμβοι αντιστοιχούν σε καταστάσεις και τα βέλη υποδηλώνουν τη μετάβαση από μια κατάσταση σε μια άλλη. Εν γένει, οι μηχανές πεπερασμένων καταστάσεων είναι κατάλληλες για την περιγραφή σύγχρονων συστημάτων.

Συνήθως, ένα διάγραμμα καταστάσεων είναι προσαρτημένο σε μια κλάση και αποτελεί ένα μοντέλο όλων των δυνατών κύκλων ζωής ενός αντικειμένου της κλάσης. Κάθε αντικείμενο αντιμετωπίζεται ως ξεχωριστή οντότητα που επικοινωνεί με το περιβάλλον ανιχνεύοντας γεγονότα και αντιδρώντας σε αυτά. Όταν λαμβάνει χώρα ένα ανιχνεύσιμο γεγονός, το αντικείμενο αποκρίνεται με βάση την κατάσταση στην οποία βρίσκεται. Η εκτέλεση μιας ενέργειας μπορεί να οδηγήσει σε μετάβαση σε μια άλλη κατάσταση.

Σε ένα διάγραμμα καταστάσεων της UML απεικονίζονται γεγονότα, καταστάσεις και μεταβάσεις:

1. Ένα γεγονός (event) έχει χωρική και χρονική θέση στο σύστημα αλλά δεν έχει διάρκεια. Ένα γεγονός συμβολίζεται σημειώνοντας το όνομα του στις μεταβάσεις τις οποίες προκαλεί. Στην περίπτωση που κάποια ενέργεια πραγματοποιείται ταυτόχρονα με την εμφάνιση ενός γεγονότος, σημειώνεται μετά το όνομα του γεγονότος διαχωρισμένη με κάθετο “/”.
2. Μια κατάσταση (state) περιγράφει μια χρονική περίοδο κατά τη διάρκεια ζωής ενός αντικειμένου. Μπορεί να χαρακτηριστεί ως ένα σύνολο τιμών για τις ιδιότητες του αντικειμένου που είναι παρόμοιες από κάποια άποψη, ως μια περίοδος κατά την οποία ένα αντικείμενο αναμένει την εμφάνιση ενός γεγονότος, ή ως μία περίοδος κατά την οποία ένα αντικείμενο εκτελεί μια εργασία. Μια κατάσταση συμβολίζεται ως ένα ορθογώνιο με καμπύλες γωνίες. Ειδικά για το συμβολισμό της αρχικής κατάστασης ενός συστήματος χρησιμοποιείται ένας «γεμισμένος κύκλος».
3. Μια μετάβαση καθορίζει την απόκριση ενός αντικειμένου που βρίσκεται σε μια κατάσταση όταν λάβει χώρα ένα γεγονός. Εν γένει, μια μετάβαση περιλαμβάνει το γεγονός που την ενεργοποιεί, προαιρετικά μια συνθήκη ελέγχου έτσι ώστε η μετάβαση να πραγματοποιείται μόνο όταν η συνθήκη είναι αληθής, μια ενέργεια και μια τελική κατάσταση.

- **Διάγραμμα Δραστηριότητας (Activity Diagram)**

Ένας γράφος δραστηριότητας είναι μια ειδική μορφή μηχανής καταστάσεων που έχει ως στόχο τη μοντελοποίηση των υπολογισμών και της ροής της εργασίας. Οι καταστάσεις του γράφου δραστηριότητας αναπαριστούν τις καταστάσεις εκτέλεσης ενός υπολογισμού, όχι τις καταστάσεις των αντικειμένων που συμμετέχουν. Υπό κανονικές συνθήκες, ένας γράφος

δραστηριότητας προϋποθέτει ότι οι υπολογισμοί πραγματοποιούνται χωρίς εξωτερικές γεγονοδηγούμενες διακοπές, αλλιώς είναι προτιμότερο ένα διάγραμμα καταστάσεων. Μια κατάσταση δραστηριότητας δεν αναμένει την εμφάνιση ενός γεγονότος, αλλά την ολοκλήρωση της διαδικασίας που περιγράφει, για τη μετάβαση στην επόμενη δραστηριότητα. Ένας γράφος δραστηριότητας μπορεί να περιλαμβάνει διακλάδωση της δραστηριότητας σε ταυτόχρονα νήματα εκτέλεσης.

Ένα διάγραμμα δραστηριότητας είναι ο συμβολισμός ενός γράφου δραστηριότητας στη UML. Περιγράφει τις συνθήκες που καθορίζουν ποιές δραστηριότητες θα εκτελεστούν σε κάθε σημείο του προγράμματος, ποιές δραστηριότητες μπορούν να λαμβάνουν χώρα παράλληλα καθώς και τυχόν επαναληπτικές δομές που περιλαμβάνονται.

Οι διακλαδώσεις συμβολίζονται είτε με συνθήκες φρουρούς επί των μεταβάσεων είτε με κόμβους απόφασης (ρόμβους) με πολλαπλές εξερχόμενες ακμές. Μια ένωση συμβολίζει συνένωση πολλών εισερχόμενων μεταβάσεων σε μία εξερχόμενη, ενώ μια διχάλα την ανάλυση μιας εισερχόμενης μετάβασης σε πολλές παράλληλες εξερχόμενες μεταβάσεις.

Τα διαγράμματα δραστηριότητας είναι χρήσιμα για την ανάλυση μιας περίπτωσης χρήσης (συνοδεύοντας την τεκμηρίωση της) όταν πρέπει να γίνει κατανοητό ποιές ενέργειες πρέπει να πραγματοποιηθούν υπό διάφορες δυνατές συνθήκες. Επιπρόσθετα, τα διαγράμματα δραστηριότητας είναι χρήσιμα για την περιγραφή πολύπλοκων αλγορίθμων, οι οποίοι πρόκειται να υλοποιηθούν από μία ή και περισσότερες μεθόδους μιας κλάσης.

- **Διάγραμμα Συστατικών (Component Diagram)**

Ένα συστατικό (component) είναι μια φυσική μονάδα υλοποίησης κώδικα με σαφώς προσδιορισμένες διασυνδέσεις, η οποία αποτελεί επαναχρησιμοποιήσιμο τμήμα του συστήματος. Σε ένα αντικειμενοστρεφές σύστημα ένα συστατικό ενσωματώνει την υλοποίηση μίας ή περισσότερων κλάσεων.

Καλά σχεδιασμένα συστατικά δε θα πρέπει να εξαρτώνται άμεσα από άλλα συστατικά αλλά μόνο από διασυνδέσεις. Οι εξαρτήσεις έχουν επίδραση στη συντήρηση ενός συστήματος λογισμικού. Αν κάποιο συστατικό A εξαρτάται από κάποιο άλλο συστατικό B, οποιαδήποτε αλλαγή στο B μπορεί να επηρεάσει το A. Υπό την ίδια έννοια, οι εξαρτήσεις καθορίζουν την ευκολία επαναχρησιμοποίησης ενός συστατικού. Στην περίπτωση όπου ένα συστατικό στο σύστημα μπορεί να αντικατασταθεί από κάποιο άλλο, που υποστηρίζει τις ίδιες διασυνδέσεις, δεν επιφέρονται αλλαγές στο υπόλοιπο σύστημα.

Ένα διάγραμμα συστατικών απεικονίζει το δίκτυο των εξαρτήσεων μεταξύ των συστατικών του συστήματος. Μια εξάρτηση μεταξύ δύο συστατικών υποδηλώνει ότι για την ορθή λειτουργία του ενός συστατικού απαιτείται η ύπαρξη ενός άλλου. Το συστατικό συμβολίζεται ως ένα ορθογώνιο ενώ οι εξαρτήσεις συμβολίζονται ως διακεκομμένες ακμές με κατεύθυνση από το εξαρτώμενο συστατικό προς αυτό που παρέχει τις λειτουργίες.

Στα διαγράμματα συστατικών υπάρχει η δυνατότητα απεικόνισης λογικών τμημάτων ενός συστήματος με τη χρήση των πακέτων. Ένα πακέτο περιλαμβάνει ένα σύνολο από συστατικά τα οποία έχουν λειτουργική συνάφεια.

- **Διάγραμμα Ανάπτυξης (Deployment Diagram)**

Το διάγραμμα ανάπτυξης περιγράφει την οργάνωση των επεξεργαστικών πόρων (κόμβων) του συστήματος και την αντιστοίχιση των συστατικών λογισμικού στους κόμβους αυτούς. Κατά κύριο λόγο απεικονίζουν την τοπολογία του υλικού επί του οποίου εκτελείται το σύστημα λογισμικού.

Ένας κόμβος (node) είναι ένα φυσικό αντικείμενο που αναπαριστά έναν υπολογιστικό πόρο, ο οποίος στη γενική περίπτωση έχει τουλάχιστον μνήμη και δυνατότητα επεξεργασίας. Οι κόμβοι μπορούν να αντιστοιχίζονται σε στερεότυπα ώστε να διακρίνονται διαφορετικά είδη πόρων, όπως Κεντρικές μονάδες επεξεργασίας, μνήμες, εξυπηρετητές για βάσεις δεδομένων και συσκευές διασύνδεσης με άλλα συστήματα. Ένας κόμβος συμβολίζεται ως ένας τρισδιάστατος κύβος με το όνομα του κόμβου και ενδεχομένως ένα στερεότυπο που εκφράζει την κατηγορία στην οποία ανήκει. Σε κάθε κόμβο μπορούν προαιρετικά να αναφερθούν (υπό μορφή σημειώσεων) και τα συστατικά τα οποία εκτελούνται σε αυτόν.

Η τοπολογία του συστήματος απεικονίζεται συνδέοντας τους κόμβους με γραμμές συσχέτισης, οι οποίες μπορούν να υποδηλώνουν ρητά το πρωτόκολλο επικοινωνίας ή να χαρακτηρίζουν το σύστημα μεταφοράς δεδομένων με κάποιο τρόπο.

Το διάγραμμα ανάπτυξης δεν προσφέρει σημαντική πληροφορία για μια αυτόνομη αντικειμενοστρεφή εφαρμογή που εκτελείται αποκλειστικά σε έναν υπολογιστή. Χρησιμοποιείται από μηχανικούς συστημάτων για τη μοντελοποίηση ενσωματωμένων συστημάτων, συστημάτων πελάτη/εξυπηρετητή, όπου υπάρχει σαφής διαχωρισμός μεταξύ των εφαρμογών που εκτελούνται στο σύστημα του πελάτη και των μονίμων δεδομένων που φιλοξενούνται στον εξυπηρετητή, καθώς και πλήρως καταναμημένων συστημάτων που περιλαμβάνουν συνήθως πολλαπλά επίπεδα εξυπηρετητών και συνήθως φιλοξενούν πολλαπλές εκδόσεις των συστατικών λογισμικού στους κόμβους τους. [6]

5. Ναυτιλία

Ως ναυτιλία ορίζεται ο κλάδος εκείνος που ασχολείται με τη μεταφορά αγαθών διαμέσου της θαλάσσιας οδού. Σε αυτήν την ενέργεια βασικά εργαλεία – μέσα μεταφοράς είναι τα πλοία κάθε λογής τα οποία καλύπτουν ανάγκες μεταφοράς στερεών, υγρών ή άλλων μορφών αγαθών. Πλέον η ναυτιλία είναι μια παγκόσμια αγορά, στην οποία παρέχεται μια πλήρης γκάμα υπηρεσιών μεταφοράς για μεγάλη ποικιλία φορτίων, τα οποία μεταφέρονται όχι μόνο σε κοντινές αποστάσεις όπως τα όρια μια χώρας ή μιας θάλασσας αλλά σε όλο τον κόσμο, από ήπειρο σε ήπειρο. Ο παγκόσμιος χαρακτήρας της ναυτιλίας όμως, δεν στηρίζεται μόνο στη μεταφορά αγαθών σε διάφορα μέρη του κόσμου. Έγκειται και στη δυνατότητα διαφόρων εμπλεκόμενων (πλοιοκτήτες, εταιρείες που θέλουν να μεταφέρουν αγαθά) να επικοινωνούν και να συνάπτουν εμπορικές συμφωνίες ανεξάρτητα από το που βρίσκονται. Θα μπορούσε να πει κανείς ότι η ναυτιλία είναι ένα σύνολο από αλληλοσχετιζόμενες βιομηχανίες οι οποίες καλύπτουν το εύρος της υφιλίου.

Ο παγκόσμιος χαρακτήρας της ναυτιλίας προσφέρει δυνατότητες δραστηριοποίησης σε όλο τον κόσμο εμφανίζοντας συνεχώς ευκαιρίες και νέες αγορές. Παρόλα αυτά πρόκειται για έναν ιδιαίτερα περίπλοκο κλάδο καθώς οι εμπορικές ενέργειες στο νέο περιβάλλον της ναυτιλίας επηρεάζονται από τις κατά τόπους οικονομικές, πολιτικές και άλλες εξελίξεις. Ταυτόχρονα σε κάθε μεριά του κόσμου υπάρχουν διαφορετικές οικονομικές και εμπορικές συνήθειες οι οποίες συνοδεύονται από ποικίλα νομικά πλαίσια. Κάτι τέτοιο απαιτεί την συνεχή εγρήγορση των εμπλεκόμενων ώστε να γνωρίζουν αλλά και να παρακολουθούν τις αλλαγές που μπορεί να επηρεάζουν τις εμπορικές δραστηριότητές τους σε κάθε πλευρά του κόσμου. Εκτός από αυτό η ίδια η αγορά από τότε που μεταλλάχτηκε σε ένα παγκόσμιο στίβο, προσφέρει άπειρους συνδυασμούς πλοίων και αγαθών οπότε είναι ιδιαίτερα περίπλοκη η ανεύρεση του κατάλληλου συνεργάτη για την πραγματοποίηση μια εμπορικής συναλλαγής που περιλαμβάνει μεταφορές δια μέσου της θάλασσας.

Σημαντικό στοιχείο μιας επιτυχημένης συμφωνίας είναι να κατέχει κάποιος τη κατάλληλη πληροφορία στη κατάλληλη στιγμή για να μπορέσει να εντοπίσει ευκαιρίες αλλά και πιθανούς κινδύνους που μπορεί να ελλοχεύουν ακόμα και σε μία άλλη μεριά του κόσμου. Το πιο καίριο από όλα είναι ο εντοπισμός του κατάλληλου πλοιοκτήτη ή ναυλωτή για συνεργασία, τη κατάλληλη στιγμή. Σε αυτή τη κατεύθυνση είναι μεγάλη και αξιοπρόσεκτη η συμβολή μιας μερίδας ανθρώπων που αποτελούν αναπόσπαστο κομμάτι της ναυτιλίας, οι ναυλομεσίτες (Ship Broker).

Οι ναυλομεσίτες έχουν έναν ρόλο διττό, εκείνο του επιχειρηματία και εκείνον του εξυπηρετή που στην ουσία μεσολαβεί μεταξύ των πλοιοκτητών και των ναυλωτών, βρίσκει τις κατάλληλες ευκαιρίες για τον καθένα από αυτούς και διαχειρίζεται τις μεταξύ τους σχέσεις ώστε να επιτευχθεί η σύναψη μίας συμφωνίας για τη μεταξύ τους συνεργασία. Είναι εκείνος που γνωρίζει την αγορά και ανακαλύπτει ευκαιρίες για τον εκάστοτε πελάτη του, ενώ

ταυτόχρονα αναλαμβάνει τη διεκπεραίωση όλων των διαπραγματεύσεων για την τελική επισφράγιση μιας συνεργασίας με ένα συμβόλαιο, το ναυλοσύμφωνο.

5. 1 Ο ρόλος του Ναυλομεσίτη

Ο ναυλομεσίτης (Chartering Broker) μεσολαβεί μεταξύ ενός πλοιοκτήτη που διαθέτει το πλοίο για εκναύλωση και ενός ναυλωτή που επιθυμεί να ναυλώσει το πλοίο για πραγματοποίηση μεταφοράς φορτίου, ρυθμίζοντας τις σχετικές λεπτομέρειες της σύναψης συμβάσεων ναύλωσης, έναντι αμοιβής που καλείται προμήθεια (commission). Ειδικότερα, οι δραστηριότητες ενός ναυλομεσίτη είναι οι εξής:

- **Αναζήτηση φορτίου ή πλοίου.** Ανάλογα με τα συμφέροντα που εκπροσωπεί, αναζητεί το κατάλληλο φορτίο που θα μεταφερθεί στο πλοίο του πελάτη του (shipbroker ή ναυλομεσίτης του πλοιοκτήτη) ή το κατάλληλο πλοίο με το οποίο θα μεταφερθεί το φορτίο του πελάτη του (cargobroker ή ναυλομεσίτης του ναυλωτή).
- **Διαπραγματεύσεις και συμφωνία σχέσης ναύλωσης.** Καθώς κατά τη διαδικασία διαπραγμάτευσης ναύλωσης δεν υπάρχει άμεση επικοινωνία μεταξύ πλοιοκτήτη και ναυλωτή, ο ναυλομεσίτης είναι υπεύθυνος για την οργάνωση της όλης διαδικασίας και ενεργεί προς φύλαξη των συμφερόντων που εκπροσωπεί με σκοπό το «κλείσιμο» της ναύλωσης- συμφωνίας.
- **Σύνταξη ναυλοσυμφώνου.** Μόλις επιτύχει τη σχέση συνεργασίας μεταξύ των δύο μερών- πλοιοκτήτη και ναυλωτή- ο μεσίτης συντάσσει το ναυλοσύμφωνο, φροντίζοντας να συμπεριλάβει σε αυτό ό,τι συμφωνήθηκε κατά τη διάρκεια της διαπραγμάτευσης.
- **Υπογραφή ναυλοσυμφώνου.** Μόλις συνταχθεί το ναυλοσύμφωνο, ο μεσίτης υπογράφει (κατόπιν σχετικής εξουσιοδότησης) για λογαριασμό των συμβαλλόμενων μερών που εκπροσωπεί και δεσμεύει με αυτό τον τρόπο τον πελάτη του (πλοιοκτήτη ή ναυλωτή).
- **Διαχείριση οικονομικών θεμάτων ναύλωσης.** Όταν πλέον το ναυλοσύμφωνο είναι σε ισχύ, ο μεσίτης αναλαμβάνει να διευθετήσει τα οικονομικά ζητήματα της ναύλωσης, όπως για παράδειγμα την προώθηση του τιμολογίου με το ύψος του ναύλου που πρέπει να καταβληθεί από το ναυλωτή για την πραγματοποίηση της μεταφοράς από τον πλοιοκτήτη καθώς επίσης και την παρακολούθηση της έγκυρης πληρωμής και λήφειας του ναύλου.

5.2 Το Ναυλοσύμφωνο ως βασικό Συμβόλαιο Μεταφοράς

Όταν το βασικό συμβόλαιο μεταφοράς είναι το ναυλοσύμφωνο, οι όροι του καθορίζουν τα δικαιώματα και τις υποχρεώσεις των συμβαλλομένων μερών. Το δίκαιο που διέπει αυτή τη συμφωνία των μερών είναι το δίκαιο που προβλέπεται ρητά στο ναυλοσύμφωνο. Τις περισσότερες φορές, συμφωνείται και εφαρμόζεται το αγγλικό δίκαιο, συχνά αποκαλούμενο ως εθιμικό δίκαιο (common law). Το δίκαιο αυτό, που διέπει τη σχέση ναυλωτή-εκναυλωτή, βασίζεται στη νομολογία, δηλαδή στο προηγούμενο δεδικασμένο παρόμοιων αντιδικιών που εκδικάστηκαν κατά το παρελθόν στα αγγλικά δικαστήρια, οι αποφάσεις των οποίων κατευθύνουν την επιχειρηματολογία και την κρίση των δικαστών στην εκδίκαση σημερινών αντίστοιχων υποθέσεων. Τα συμβαλλόμενα μέρη απαλλάσσονται από τις υποχρεώσεις και τα δικαιώματά τους, που απορρέουν από τους όρους του ναυλοσυμφώνου, εφ' όσον αυτό ματαιωθεί. Η ματαίωση του ναυλοσυμφώνου (frustration of the charter party) οφείλεται σε μεγάλη καθυστέρηση (delay) ή σε αδυναμία εκτέλεσης του ναυλοσυμφώνου (impossibility of performance) λόγω κάποιου απρόβλεπτου γεγονότος, χωρίς να ευθύνεται κανένα από τα μέρη.

Σύμφωνα με το αγγλικό εθιμικό δίκαιο οι όροι των ναυλοσυμφώνων ταξινομούνται σε:

- Ρητοί και υπονοούμενοι όροι (express & implied terms)

Ρητοί όροι (express terms) είναι οι όροι που περιγράφονται ρητώς στα ναυλοσύμφωνα, είτε στις τυποποιημένες φόρμες ή στο πρόσθετο παράρτημα (rider). Υπονοούμενοι όροι (implied terms) είναι οι όροι που δεν περιέχονται γραπτώς στα ναυλοσύμφωνα, αλλά είναι τόσο φανεροί και ισχυροί που γίνονται σιωπηλά αποδεκτοί από τα συμβαλλόμενα μέρη. Συνήθως, οι υπονοούμενοι όροι αφορούν ζητήματα μεγαλύτερης βαρύτητας και γι' αυτό η παραβίασή τους επιφέρει σημαντικές κυρώσεις. Ωστόσο, όταν υπάρχει ασυμφωνία μεταξύ ενός υπονοούμενου και ενός ρητού όρου, θα υπερισχύσει ο ρητός όρος που θεωρείται πάντα ότι εκφράζει καλύτερα τις προθέσεις των μερών. Οι πιο σημαντικοί υπονοούμενοι όροι του αγγλικού δικαίου αφορούν την αξιοπλοΐα (seaworthiness), τη δέουσα επιμέλεια εκτέλεσης της ναύλωσης (due dispatch) και τη μη παρέκκλιση (proper route).

- Περιγραφές (representations)

Περιγραφές (representations) είναι οι όροι που αφορούν τις παρουσιάσεις που δίνονται κατά τη διάρκεια των διαπραγματεύσεων και αποτελούν υποσχέσεις που δίνουν τα συμβαλλόμενα μέρη. Σε περίπτωση παροχής ανακριβών στοιχείων (misrepresentation), θα εξετασθεί η πρόθεση ή όχι του συμβαλλόμενου. Εάν η παροχή των ανακριβών στοιχείων έγινε με πρόθεση και επηρέασαν το άλλο συμβαλλόμενο μέρος να υπογράψει το ναυλοσύμφωνο, τότε ο τελευταίος έχει το δικαίωμα να ακυρώσει το συμβόλαιο (repudiate the charter party). Εάν η παροχή ανακριβών στοιχείων έγινε χωρίς πρόθεση, τότε ο συμβαλλόμενος που έδωσε αυτά τα στοιχεία υποχρεούται να καταβάλλει αποζημιώσεις, εκτός εάν αποδείξει ότι είχε λόγους να πιστεύει και πίστευε μέχρι τη στιγμή που υπογράφηκε

το ναυλοσύμφωνο ότι τα στοιχεία που έδωσε ήταν αληθινά. Παραδείγματα όρων "representations" είναι τα χαρακτηριστικά του πλοίου (το όνομα, το διακριτικό σήμα, ο χρόνος ναυπήγησης κ.α.) που έχουν περιγραφεί από τον πλοιοκτήτη και τα χαρακτηριστικά του φορτίου (όγκος, βάρος, τύπος κ.α.) που έχουν περιγραφεί από το ναυλωτή.

- Προϋποθέσεις (conditions)

Προϋποθέσεις (conditions) είναι οι όροι που η παραβίαση τους από το ένα συμβαλλόμενο μέρος δίνει το δικαίωμα στο άλλο μέρος να ακυρώσει το ναυλοσύμφωνο και να αξιώσει αποζημιώσεις (repudiate the charter party and sue for damages). Εάν ο θιγόμενος επιλέξει να μην ακυρώσει το ναυλοσύμφωνο, αλλά να συνεχίσει αυτό να ισχύει, τότε δεσμεύεται από την πράξη του αυτή και δεν μπορεί να ισχυριστεί αργότερα ότι το άλλο συμβαλλόμενο μέρος δεν εκτέλεσε τις υποχρεώσεις του που απορρέουν από το ναυλοσύμφωνο (waiver of the breach of condition). Παραδείγματα όρων "condition" είναι η γεωγραφική θέση του πλοίου κατά τη στιγμή της υπογραφής του ναυλοσυμφώνου, ο χρόνος αναχώρησης για το λιμάνι φόρτωσης, η εθνικότητα του πλοίου, η κλάση του πλοίου που δηλώνεται από το νηογνώμονα, η μεταφορική ικανότητα του πλοίου για ορισμένο φορτίο, η ημερομηνία κατά την οποία το πλοίο αναμένεται να είναι έτοιμο προς φόρτωση κ.λ.π.

- Εγγυήσεις (warranties)

Εγγυήσεις (warranties) είναι οι όροι που η παραβίαση τους από το ένα συμβαλλόμενο μέρος δίνει το δικαίωμα στο άλλο μέρος να αξιώσει αποζημίωση (sue for damages). Παραδείγματα τέτοιων όρων είναι η συντήρηση του πλοίου, τα καύσιμα του, η υποχρέωση του ναυλωτή να φέρει το φορτίο στο λιμάνι φόρτωσης, η επαναπαράδοση του πλοίου από το ναυλωτή στον εκναυλωτή, η ταχύτητα του πλοίου κ.λ.π.

- Απροκαθόριστοι όροι (innominate terms)

Απροκαθόριστοι όροι (innominate terms) είναι οι όροι που αντιμετωπίζονται από το δικαστήριο άλλοτε ως "conditions" και άλλοτε ως "warranties", ανάλογα με τη σοβαρότητα των συνεπειών της αθέτησης. Παράδειγμα όρου "innominate" είναι η αξιοπλοΐα του πλοίου.

5.3 Διαδικασία Ναύλωσης (Case Study)

Θα γίνει μία παρουσίαση της διαδικασίας ναύλωσης με τη βοήθεια μίας πραγματικής περίπτωσης. Όλα όσα περιγράφονται είναι αληθινά, ωστόσο, το πραγματικό όνομα της πλοιοκτήτριας εταιρείας, της ναυλώτριας εταιρείας και του πλοίου δεν είναι πραγματικά.

Η εταιρεία "FullChart Corp, Panama", η οποία είναι μια "First Class" ναυλώτρια εταιρεία, θέλει να μεταφέρει 700 σωλήνες από τη Σαγκάη στη Βενεζουέλα. Για να το επιτύχει αυτό εκδηλώνει το ενδιαφέρον της στην αγορά με του παρακάτω "order" μέσου του ναυλωτικού γραφείου με το οποίο συνεργάζεται.

EX CHINA
ABT 60.000 CBM – PC 700 PIPES ABT 8.600 MT
(700 PCS X 2.6 M OD 12.2 M L 12.3 T UW – MAX 5 TIERS)
DECK OK
SHANGHAI / 1 SP VENEZUELA
NOV try DEC
8 TTLDAPS C
BOFFER FIOS try abt usd 100/m³

Με τη σύνταξη του παραπάνω “order” ξεκινάει το στάδιο διερεύνησης. Το “order” περιλαμβάνει τις εξής πληροφορίες για το φορτίο:

1. Αποτελείται από 700 σωλήνες οι οποίοι καταλαμβάνουν εμβαδό 8.600 μ² και όγκο 60.000 μ³. Δίνεται επίσης αναλυτική περιγραφή των διαστάσεων του κάθε σωλήνα καθώς επίσης και ο μέγιστος αριθμός «στρώσεων» από σωλήνες που μπορούν να φορτωθούν (5 στρώσεις).
2. Οι σωλήνες μπορούν φορτωθούν και στο κατάστρωμα.
3. Το λιμάνι φόρτωσης είναι η Σαγκάη και το λιμάνι εκφόρτωσης βρίσκεται στη Βενεζουέλα.
4. Το πλοίο θα πρέπει να βρίσκεται στο λιμάνι φόρτωσης Νοέμβριο με Δεκέμβριο.
5. Απαιτούνται συνολικά 8 μέρες για τη φόρτωση και την εκφόρτωση του φορτίου.
6. Ο προσφερόμενος ναύλος κυμαίνεται περί τα 100 \$ / μ³.
7. Το ναυλοσύμφωνο που θα υπογραφεί προτείνεται να είναι τύπου “Gencon”.
8. Η προμήθεια μέχρι στιγμής είναι 3,75%.

5.3.1 Περιγραφή Διαδικασιών και Εντοπισμός του Προβλήματος

Η διαδικασία των ναυλώσεων ξεκινά από τον Charter. Ο Charter παίρνει εντολή από τις Εταιρείες που επιθυμούν να μεταφέρουν τα προϊόντα τους μέσω της θαλάσσιας οδού να εντοπίσει το κατάλληλο είδος Πλοίου, το οποίο θα διαθέτει το απαιτούμενο μέγεθος για τη μεταφορά και θα είναι διαθέσιμο τις επιθυμητές ημερομηνίες του εντολέα. Ο Charter ξεκινά την επικοινωνία με τους αντίστοιχους Ναυλομεσίτες των Ναυτιλιακών Εταιρειών που διαθέτουν τα Πλοία προκειμένου να βρουν Πλοία διαθέσιμα παραθέτοντας τις προδιαγραφές που απαιτεί η μεταφορά του Φορτίου καθώς και το ποσοστό προμήθειας που επιθυμεί ο ίδιος ο Ναυλωτής. Η επικοινωνία αυτή γίνεται μέσω Ηλεκτρονικού Ταχυδρομείου, E-Mail.

Μια τέτοιου είδους επικοινωνία έρχεται να εκσυγχρονίσει το Πληροφοριακό Σύστημα το οποίο αποτελεί και το παρόν θέμα της διπλωματικής εργασίας. Η επικοινωνία μέσω

Ηλεκτρονικού Ταχυδρομείου στη σημερινή εποχή θεωρείται “αρχαία”. Στην εποχή της πλήρης πλέον ψηφιοποίησης όλων των διαδικασιών μέσω του Παγκόσμιου Ιστού, ο χώρος της Ναυτιλίας δεν θα πρέπει να μένει πίσω και οφείλει να εξελίσσεται. Το μεγαλύτερο μερίδιο της παγκόσμιας οικονομίας εξαρτάται από την πορεία και την εξέλιξη στο χώρο της Ναυτιλίας. Εδώ λοιπόν έρχεται και το εν λόγω Πληροφοριακό Σύστημα προκειμένου η εύρεση και η ναύλωση των πλοίων να γίνεται μέσα από μία Παγκόσμια έρευνα μέσω μίας διαδικτυακής εφαρμογής.

Εν συνεχεία της διαδικασίας, οι Ναυλομεσίτες των Ναυτιλιακών Εταιρειών εξετάζουν ποια από τα Πλοία του Στόλου τους είναι διαθέσιμα τις επιθυμητές ημερομηνίες και έχουν τη χωρητικότητα που απαιτείται προκειμένου να γίνει η μεταφορά του Φορτίου. Μόλις εντοπίσουν οι Ναυλομεσίτες διαθέσιμο Πλοίο στέλνουν τις απαραίτητες πληροφορίες κατασκευής, χωρητικότητας και προσωπικού με το οποίο είναι επανδρωμένο το επιλεγμένο Πλοίο καθώς και την τιμή ναύλωσής του. Η τιμή ναύλωσης μπορεί να χωριστεί σε τιμή ναύλωσης ανά ημέρα ή τιμή ναύλωσης ανά Φορτίο.

Από τη στιγμή αυτή και μετέπειτα ξεκινά η διαδικασία των διαπραγματεύσεων επί της τιμής ναύλωσης καθώς και των ημερομηνιών αναχώρησης και άφιξης του Πλοίου από τους επιθυμητούς προορισμούς. Ένα άλλο αντικείμενο διαπραγμάτευσης αποτελεί και το αν το πλήρωμα πρόκειται να εργαστεί τις ημέρες των Αργιών ή όχι. Έτσι η διαδικασία ναυλώσεων μπαίνει στη φάση του Accept (Αποδοχή) & Recap (Αναθεώρηση) του Subject (Θέματος). Στη φάση αυτή τα δύο μέρη κατόπιν διαπραγματεύσεων αποδέχονται ή αναθεωρούν τη Συμφωνία που επρόκειτο να κλείσουν. Στη συνέχεια ακολουθεί η φάση του Lift (Ανάληψη) & Reject (Απόρριψη) του Subject. Στη περίπτωση που και τα δύο μέρη αποδέχονται τους όρους τότε πραγματοποιείται και η Ανάληψη της Συμφωνίας και ακολουθεί η σύνταξη του Ναυλοσυμφώνου (Contract) διαφορετικά έπεται η Απόρριψη του αιτήματος της χρήσης του Πλοίου για μεταφορά του Φορτίου.

Επομένως ο Ναυλωτής μπορεί να:

1. Κάνει έρευνα (Searching) τις πληροφορίες μιας Ναυτιλιακής Εταιρείας
2. Κάνει αίτηση για τη ναύλωση ενός πλοίου παραθέτοντας τα δεδομένα που απαιτούνται για την εύρεση του επιθυμητού Πλοίου
 - a. Μπορεί να διαπραγματευτεί μια αίτηση
 - b. Μπορεί να απορρίψει μια αίτηση
 - c. Μπορεί να αποδεχτεί και να αναλάβει την εξεταζόμενη Συμφωνία
3. Μπορεί να λάβει το Ναυλοσύμφωνο
 - a. Μπορεί να υπογράψει το Ναυλοσύμφωνο
 - b. Μπορεί να αποθηκεύσει το Ναυλοσύμφωνο
 - c. Μπορεί να αποστείλει το Ναυλοσύμφωνο
4. Μπορεί να κάνει αναζήτηση το ιστορικό των Ναυλοσυμφώνων που έχει υπογράψει συνολικά και με κάθε Ναυτιλιακή Εταιρεία ξεχωριστά

Στη συνέχεια ο Ναυλομεσίτης μπορεί να:

1. Να διαβάσει μια αίτηση που έχει γίνει για ναύλωση ενός Πλοίου
 - a. Να διαπραγματευτεί μια αίτηση
 - b. Να συγκρίνει τις αιτήσεις που έχει από τους ναυλωτές για ένα πλοίο και να επιλέξει την καλύτερη συμφωνία
 - c. Να απορρίψει μια αίτηση
2. Μπορεί να κάνει αναζήτηση το ιστορικό των Ναυλοσυμφώνων που έχει υπογράψει συνολικά και με κάθε Εταιρεία Ναυλωτών ξεχωριστά
3. Να μεταβάλλει τις πληροφορίες ενός Πλοίου
4. Μπορεί να συντάξει το Ναυλοσύμφωνο
 - a. Μπορεί να επεξεργαστεί το Ναυλοσύμφωνο
 - b. Μπορεί να υπογράψει το Ναυλοσύμφωνο
 - c. Μπορεί να αποθηκεύσει το Ναυλοσύμφωνο
 - d. Μπορεί να αποστείλει το Ναυλοσύμφωνο.

6. Ανάπτυξη Πληροφοριακού Συστήματος

Όπως ήδη έχει αναφερθεί και στην παραπάνω ενότητα, το πρόβλημα στο χώρο της Ναυτιλίας και συγκεκριμένα στη διαδικασία των ναυλώσεων έγκειται στο γεγονός ότι η επικοινωνία των δύο μερών πραγματοποιείται μέσω του Ηλεκτρονικού Ταχυδρομείου. Αξίζει να σημειωθεί ότι η διαδικασία των ναυλώσεων βασίζεται κυρίως στις γνωριμίες των Ναυλωτών και Ναυλομεσιτών. Αυτό σημαίνει αυτόματα ότι και το εύρος της έρευνας που κάνουν για την έρευνα ενός πλοίου για μεταφορά προϊόντων, άλλα και άλλων υλικών και εμπορευμάτων, είναι αρκετά περιορισμένο. Συνεπώς οι Ναυλωτές δεν γνωρίζουν αν εν τέλει επιτεύχθηκε η καλύτερη προσφορά και συμφωνία προς όφελός τους.

6.1 Υλοποίηση Πληροφοριακού Συστήματος

Σκοπός του εν λόγω Πληροφοριακού Συστήματος είναι η διαδικτυακή εύρεση της καλύτερης προσφοράς και συμφωνίας μέσω αυτοματοποιημένων διαδικασιών. Για την υλοποίηση του Πληροφοριακού Συστήματος χρησιμοποιήθηκαν οι εξής τεχνολογίες:

- Η πλατφόρμα ανάπτυξης Ενωποιημένης Γλώσσας Προγραμματισμού Rational Systems Developer (RSD) της εταιρείας IBM & έκδοση 6.0.1

Η συγκεκριμένη πλατφόρμα σχεδιάστηκε και αναπτύχθηκε από της εταιρεία IBM και αποτελεί ένα εργαλείο σχεδίασης και οπτικής μοντελοποίησης βασισμένο στη γλώσσα Ενοποιημένης Μοντελοπίησης- UML. Οι δυνατότητες που προσφέρει το συγκεκριμένο εργαλείο είναι:

1. Βασίζεται στη γλώσσα UML.
 2. Υποστηρίζει αυτόματη μετατροπή ενός διαγράμματος σε ένα άλλο.
 3. Διαχείριση του μοντέλου ανάπτυξης για ταυτόχρονη ανάπτυξη και αρχιτεκτονικής σχεδίασης του υπό-ανάπτυξης Συστήματος.
- Το σχεσιακό Σύστημα Διαχείρισης Βάσεων Δεδομένων MySQL Workbench έκδοση 6.3.4

Το σχεσιακό Σύστημα Διαχείρισης Βάσεων Δεδομένων είναι ένα από τα πλέον γνωστά ανοιχτού κώδικα ΣΔΒΔ. Η πλατφόρμα MySQL δεν είναι τέλεια άλλα είναι μια καλή επιλογή από την άποψη ότι χαρίζει αρκετή ευελιξία στις απαιτήσεις ενός Πληροφοριακού Συστήματος.

- Η πλατφόρμα υλοποίησης Πληροφοριακών Συστημάτων Bonitasoft BPM έκδοση 7.1.5

Η πλατφόρμα Bonita BPM αποτελεί μια ανοιχτού λογισμικού πλατφόρμα διαχείρισης και ροής επιχειρησιακών διαδικασιών. Το Bonita BPM αποτελείται από τρία κυρίως συστατικά :

1. Το Bonita Studio, το οποίο επιτρέπει το χρήστη να τροποποιεί τις επιχειρησιακές διαδικασίες ακολουθώντας πιστά πάντα τους περιορισμούς που θέτει η λογική Business Process Model & Notation- BPMN. Ο χρήστης επιπλέον έχει τη δυνατότητα να συνδέει τις διαδικασίες με άλλες τεχνολογίες των Πληροφοριακών Συστημάτων όπως είναι οι Βάσεις Δεδομένων προκειμένου να δημιουργούν μία αυτοματοποιημένη διαδικτυακή εφαρμογή. Η σχεδίαση των Πληροφοριακών Συστημάτων γίνεται γραφικά και η τεχνολογία του Bonita Studio στηρίζεται στη πλατφόρμα Προγραμματισμού Eclipse.
2. Το Bonita BPM Engine, το οποίο είναι μια JAVA εφαρμογή και εκτελεί τις διαδικασίες που σχεδιάστηκαν στο Bonita Studio. Το Engine API επιτρέπει τον χρήστη να αλληλεπιδρά προγραμματιστικά με τις διαδικασίες.
3. Το Bonita Portal, είναι ένα είδος ιστοσελίδας μέσω της οποίας οι τελικοί χρήστες μπορούν να διαχειρίζονται και να εκτελούν τις διαδικασίες τις οποίες όμως επιτρέπεται να εκτελέσουν. Επιπλέον το Bonita Portal δίνει τη δυνατότητα διαχείρισης αλλά και έκδοσης αναφορών σχετικά με τις διαδικασίες που έχουν ήδη εκτελεστεί.

6.2 Σχεδίαση Διαγράμματος UML

6.2.1 Διάγραμμα Περίπτωση Χρήσης- Use Case Diagram

Όπως ήδη έχουμε αναφέρει το διάγραμμα περιπτώσεων χρήσης είναι ο πιο απλός τρόπος απεικόνισης του Συστήματος που επιθυμεί κάποιος να αναπτύξει. Είναι ένας εύκολος τρόπος περιγραφής του υπ' άναπτυξη Πληροφοριακού Συστήματος μεταξύ του Προγραμματιστή και της αιτούσας εταιρείας που επιθυμεί την υλοποίησή του.

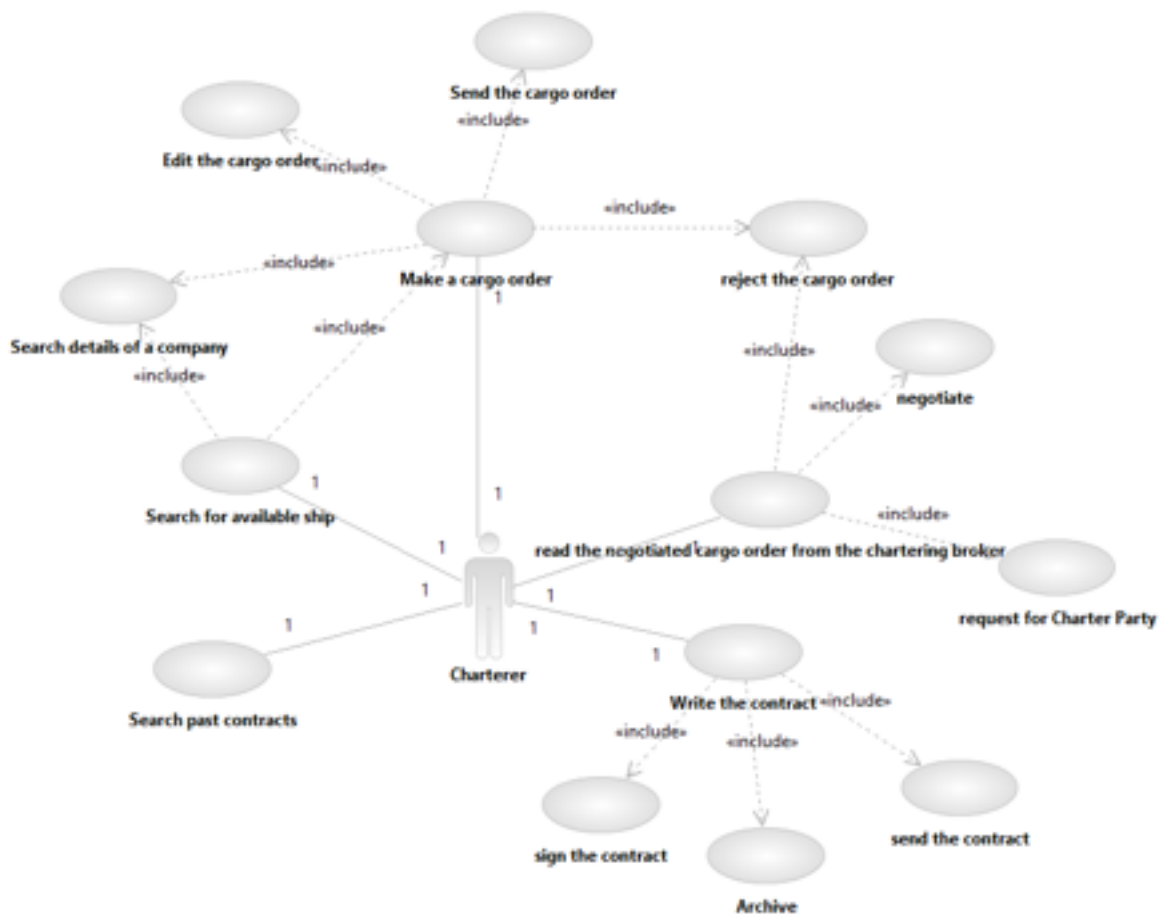
Στην εξετάζουσα περίπτωση οι κύριοι χρήστες- Actors του Πληροφοριακού Συστήματος είναι δύο:

1. Ο Charterer- Ναυλωτής και
2. Ο Chartering Broker- Ναυλομεσίτης.

Οι περιπτώσεις χρήσης κάθε ενός από τους παραπάνω χρήστες είναι οι εξής:

- Ο Ναυλωτής μπορεί να:
 1. Κάνει έρευνα (Searching) τις πληροφορίες μιας Ναυτιλιακής Εταιρείας
 2. Κάνει αίτηση για τη ναύλωση ενός πλοίου παραθέτοντας τα δεδομένα που απαιτούνται για την εύρεση του επιθυμητού Πλοίου

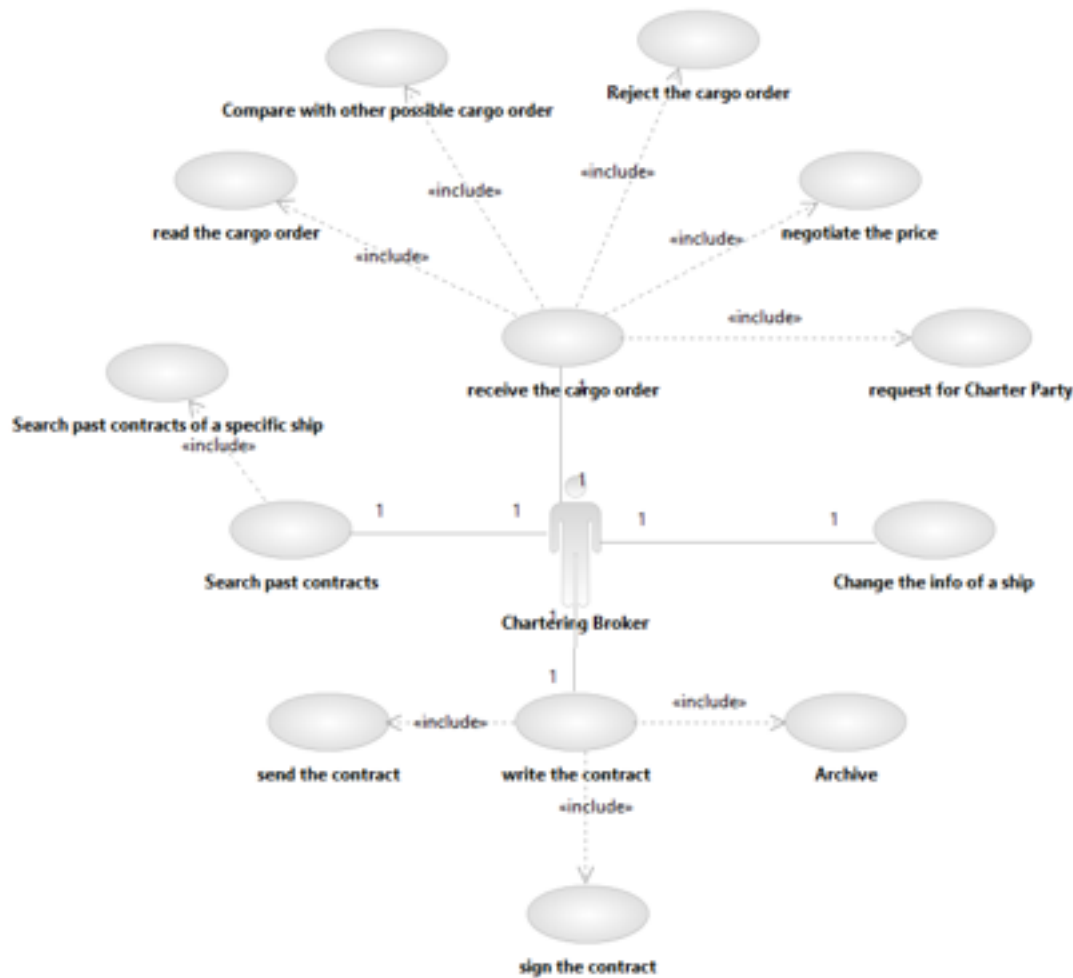
- a. Μπορεί να διαπραγματευτεί μια αίτηση
 - b. Μπορεί να απορρίψει μια αίτηση
 - c. Μπορεί να αποδεχτεί και να αναλάβει την εξεταζόμενη Συμφωνία
3. Μπορεί να λάβει το Ναυλοσύμφωνο
 - a. Μπορεί να υπογράψει το Ναυλοσύμφωνο
 - b. Μπορεί να αποθηκεύσει το Ναυλοσύμφωνο
 - c. Μπορεί να αποστείλει το Ναυλοσύμφωνο
 4. Μπορεί να κάνει αναζήτηση το ιστορικό των Ναυλοσυμφώνων που έχει υπογράψει συνολικά και με κάθε Ναυτιλιακή Εταιρεία ξεχωριστά



Εικόνα 5.1: Διάγραμμα Περίπτωσης Χρήσης του Ναυλωτή- Charterer.

- Ο Ναυλομεσίτης μπορεί να:
 1. Να διαβάσει μια αίτηση που έχει γίνει για ναύλωση ενός Πλοίου
 - a. Να διαπραγματευτεί μια αίτηση
 - b. Να συγκρίνει τις αιτήσεις που έχει από τους ναυλωτές για ένα πλοίο και να επιλέξει την καλύτερη συμφωνία
 - c. Να απορρίψει μια αίτηση
 2. Μπορεί να κάνει αναζήτηση το ιστορικό των Ναυλοσυμφώνων που έχει υπογράψει συνολικά και με κάθε Εταιρεία Ναυλωτών ξεχωριστά
 3. Μπορεί να συντάξει το Ναυλοσύμφωνο
 - a. Μπορεί να λάβει το Ναυλοσύμφωνο
 - b. Μπορεί να υπογράψει το Ναυλοσύμφωνο
 - c. Μπορεί να αποθηκεύσει το Ναυλοσύμφωνο
 - d. Μπορεί να αποστείλει το Ναυλοσύμφωνο
 4. Να μεταβάλλει τις πληροφορίες ενός Πλοίου

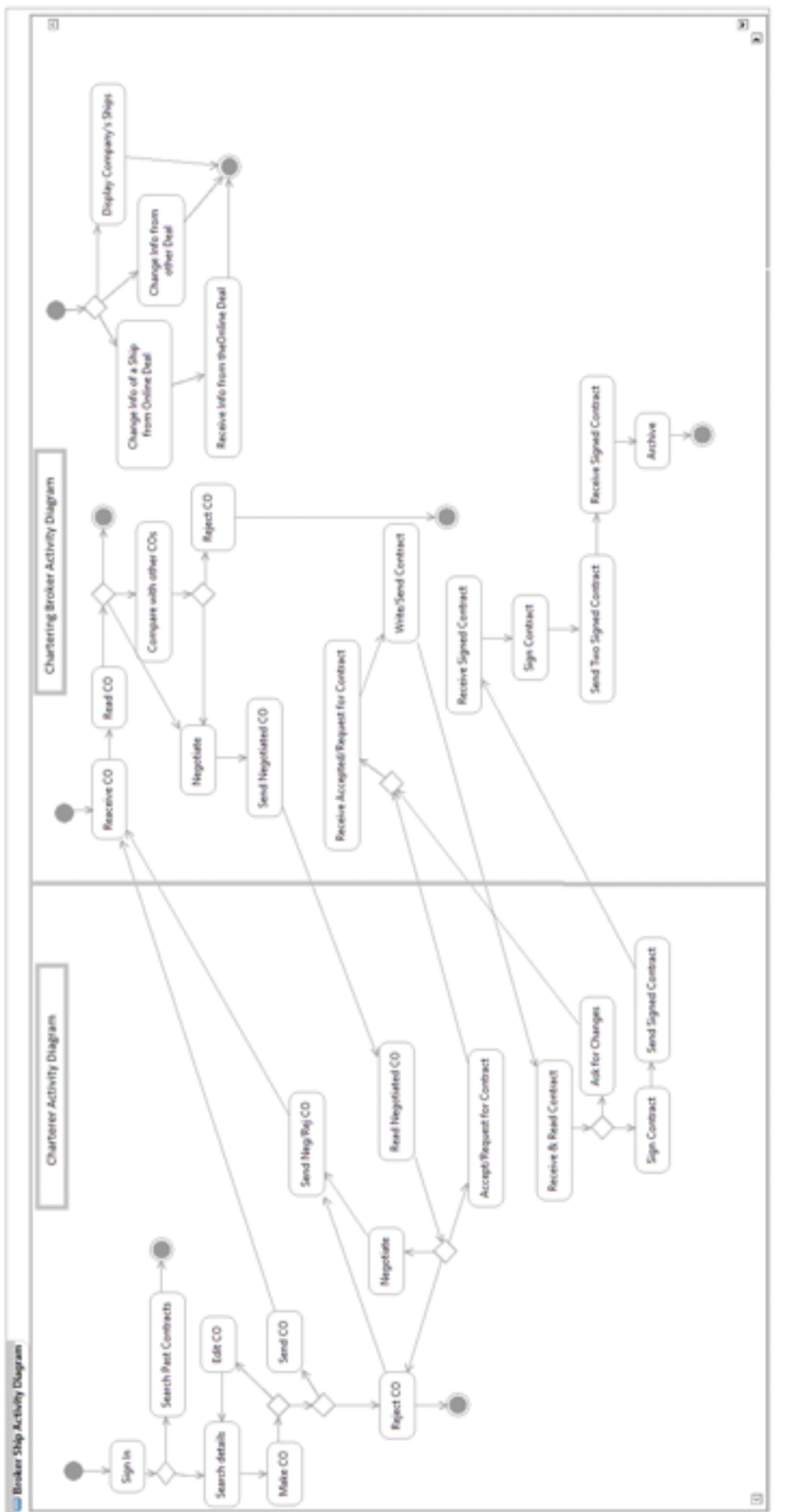
Οι παράπανω περιπτώσεις χρήσης φαίνονται αναλυτικά και στην παρακάτω εικόνα που παρατίθεται:



Εικόνα 5.2: Διάγραμμα Περίπτωσης Χρήσης του Ναυλομεσίτη- Chartering Broker.

6.2.2 Διάγραμμα Δραστηριότητας - Activity Diagram

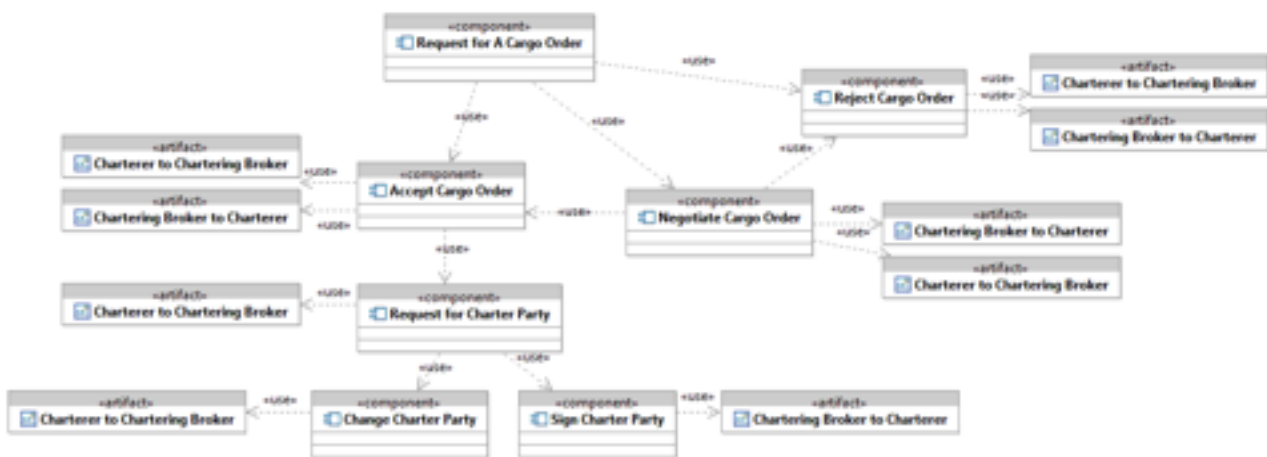
Το Διάγραμμα Δραστηριότητας είναι ένα ακόμα σημαντικό διάγραμμα στη Γλώσσα Ενοποιημένης Μοντελοποίησης, το οποίο περιγράφει τις δυναμικές πτυχές του Πληροφοριακού Συστήματος. Στην ουσία είναι ένα διάγραμμα ροής που αναπαριστά τη ροή από την μία ενέργεια στην άλλη. Η ενέργεια μπορεί να περιγραφεί και ως λειτουργία του συστήματος. Ο σκοπός του Διαγράμματος Δραστηριότητας είναι να αναπαριστά τη ροή των λειτουργιών του συστήματος, να περιγράφει την αλληλουχία σπό τη μία ενέργεια στην άλλη και την παράλληλη, πολύπτυχη και συντρέχουσα ροή του Συστήματος. Η παρακάτω εικόνα μας δείχνει το Διάγραμμα Δραστηριότητας του υπ' άναπτυξη Πληροφοριακού Συστήματος.



Εικόνα 5.3: Διάγραμμα Δραστηριότητας των Ναυλωτή-Charterer και Ναυλομεσίτη-Chartering Broker.

6.2.3 Διάγραμμα Συστατικών - Component Diagram

Το Διάγραμμα Συστατικών μας απεικονίζει τον τρόπο με τον οποίο τα βασικά συστατικά του Πληροφοριακού Συστήματος είναι συνδεδεμένα μεταξύ τους. Πιο συγκεκριμένα μας απεικονίζει τη δομή των αυθαίρετα πολύπλοκων συστημάτων. Το συστατικό είναι αυτό το οποίο είναι απαραίτητο προκειμένου να εκτελεστεί μια έιτουργία του Πληροφοριακού Συστήματος. Οι συνδέσεις που υπάρχουν μεταξύ των Components υποδηλώνουν τη σχέση του παρόχου και του καταναλωτή της υπηρεσίας.



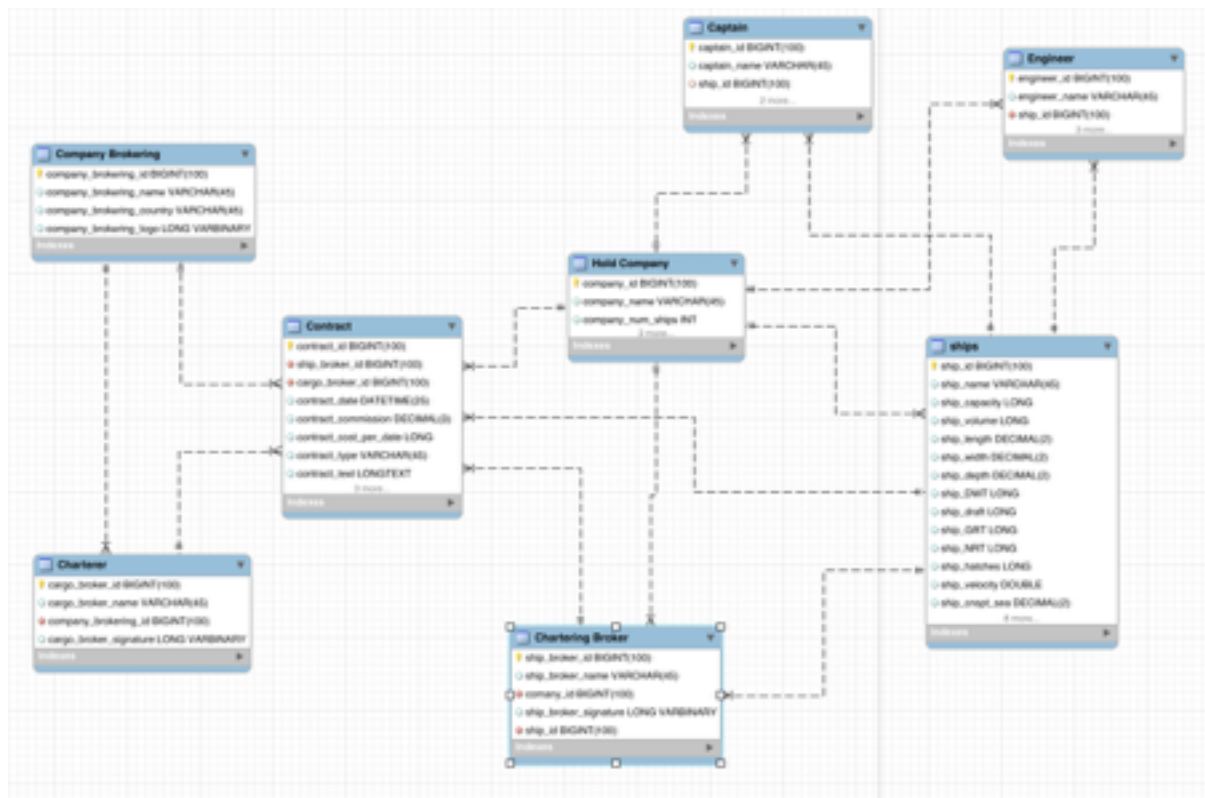
Εικόνα 5.4: Διάγραμμα Συστατικών του Πληροφοριακού Συστήματος.

Στην παραπάνω εικόνα XX παρατηρούμε ότι το πρωταρχικό συστατικό αποτελεί η Αίτηση για ενοικίαση ενός πλοίου, το οποίο με τη σειρά του υλοποιείται μέσω των συστατικών Αποδοχή (Accept), Διαπραγμάτευση (Negotiate) και Απόρριψη (Reject) του αιτήματος. Και τα τρία συστατικά εμπεριέχουν τις εξής σχέσεις :

1. Ναυλωτής σε Ναυλομεσίτη (Charterer to Chartering Broker)
2. Ναυλομεσίτης σε Ναυλωτή (Chartering Broker to Charterer)

Μετάπειτα το συστατικό Διαπραγμάτευση (Negotiate) περιλαμβάνει το συστατικό Απόρριψη (Reject) και Αποδοχή (Accept) με τις σχέσεις που αναφέρθηκαν παραπάνω. Κατόπιν το συστατικό Αποδοχή περιλαμβάνει το συστατικό Αίτηση για Σύνταξη του Ναυλοσυμφώνου (Charter Party) με μοναδική σχέση Ναυλωτής σε Ναυλομεσίτη (Charterer to Chartering Broker). Με τη σειρά του το συστατικό αυτό εμπεριέχει άλλα δύο συστατικά την Αλλαγή (Change) και την Υπογραφή (Sign) του Ναυλοσυμφώνου. Τα δύο αυτά συστατικά έχουν ως μοναδική σχέση Ναυλωτής σε Ναυλομεσίτη (Charterer to Chartering Broker).

προτεινόμενες δομές δεδομένων (datatypes) που υλοποιούν το προτεινόμενο σχήμα



δεδομένων.

Εικόνα 5.6: Διάγραμμα Πινάκων της Βάσης Δεδομένων.

Στην παραπάνω εικόνα παρατηρούμε τις σχέσεις μεταξύ των πινάκων για τις οποίες ισχύει:

- Ναυτιλιακή εταιρεία - Ναυλομεσίτη (Hold Company - Chartering Broker)= 1:N
- Ναυτιλιακή εταιρεία - Πλοίο (Hold Company - Ship) = 1:N
- Ναυτιλιακή εταιρεία - Ναυλοσύμφωνο (Hold Company - Contract) = 1:N
- Ναυτιλιακή εταιρεία - Καπετάνιος (Hold Company - Captain) = 1:N
- Ναυτιλιακή εταιρεία - Μηχανικός (Hold Company - Engineer) = 1:N
- Πλοίο - Καπετάνιος (Ship - Captain)= 1:N
- Πλοίο Μηχανικός (Ship - Engineer)= 1:N
- Πλοίο - Ναυλοσύμφωνο (Ship - Contract)=1:N
- Ναυλομεσίτης - Πλοίο (Chartering Broker - Ship)= 1:N
- Ναυλομεσίτης - Ναυλοσύμφωνο (Chartering Broker - Contract)= 1:N
- Εταιρεία Ναυλωτών - Ναυλωτής (Brokering Company - Charterer)= 1:N
- Εταιρεία Ναυλωτών - Ναυλοσύμφωνο (Brokering Company - Contract)= 1:N
- Ναυλωτής - Ναυλοσύμφωνο (Charterer - Contract)= 1:N

Αναλυτικά οι πίνακες που χρησιμοποιήθηκαν ήταν:

COLUMN	DATATYPE
shipId	INT(10)
shipName	VARCHAR(45)
shipType	VARCHAR(45)
shipFlag	VARCHAR(45)
shipDateBuilt	YEAR
shipLOA	DECIMAL(5,2)
shipLBP	DECIMAL(5,2)
shipBreadth	DECIMAL(5,2)
shipDepth	DECIMAL(5,2)
shipDraft	VARCHAR(20)
shipDWT	INT(11)
shipAvailableSpace	INT(11)
shipGRT	INT(11)
shipNRT	INT(11)
LightShip	INT(11)
shipTEU	DECIMAL(5,2)
shipCompany	VARCHAR(45)

Πίνακας 5.1: Πίνακας Βάσης Δεδομένων “bonita”. “ships”.

COLUMN	DATATYPE
userId	VARCHAR(5)
userName	VARCHAR(45)
userLastName	VARCHAR(45)
userRole	VARCHAR(45)
userCompany	VARCHAR(45)
userEmail	VARCHAR(45)

Πίνακας 5.2: Πίνακας Βάσης Δεδομένων “bonita”. “users”.

COLUMN	DATATYPE
shipId	INT(11)
captainA	VARCHAR(45)
captainB	VARCHAR(45)
captainC	VARCHAR(45)
engineerA	VARCHAR(45)
engineerB	VARCHAR(45)
engineerC	VARCHAR(45)

Πίνακας 5.3: Πίνακας Βάσης Δεδομένων “bonita”. “ship_personel”.

COLUMN	DATATYPE
shipId	INT(11)
shipUserId	VARCHAR(45)
shipUserName	VARCHAR(45)
shipComapny	VARCHAR(45)

Πίνακας 5.4: Πίνακας Βάσης Δεδομένων “bonita”. “ship_users”.

COLUMN	DATATYPE
no_travel	INT(100)
shipId	INT(10)
dateDeaprture	DATE
dateArrival	DATE
arrivalPort	VARCHAR(45)
departurePort	VARCHAR(45)
shipCompany	VARCHAR(45)

Πίνακας 5.5: Πίνακας Βάσης Δεδομένων “bonita”. “ships_program”.

COLUMN	DATATYPE
id	INT(10)
processId	VARCHAR(45)
processONforID	VARCHAR(45)
flag	VARCHAR(1)

Πίνακας 5.6: Πίνακας Βάσης Δεδομένων “bonita”. “processon”.

6.4 Περιγραφή της Διαδικασίας

Οι συμμετέχοντες (ρόλοι) στη διαδικασία είναι δύο: ο Ναυλωτής (Charterer) και ο Ναυλομεσίτης (Chartering Broker). Η ροή του διαγράμματος ξεκινάει από το Ναυλωτή συνεχίζει στο Ναυλομεσίτη και τέλος καταλήγει στο Ναυλομεσίτη να αλλάζει τα δεδομένα ενός πλοίου σε περίπτωση επίτευξης μιας συμφωνίας.

Για να γίνει το Πληροφοριακό Σύστημα πιο κατανοητό, η Διαδικασία ξεκινάει από το Ναυλωτή ο οποίος αρχικά εισάγει τις ημερομηνίες αναχώρησης και άφιξης, αντίστοιχα κατόπιν τα λιμάνια προορισμού, τη σημαία που επιθυμεί να έχει το καράβι αλλά και το μέγεθος και το είδος της ποσότητας που επιθυμεί να μεταφέρει. Επίσης εδώ στο σημείο αυτό έχει τη δυνατότητα να παρουσιαστούν επιπλέον λεπτομέρειες ή και όχι. Έπειτα μετά από έλεγχο της Βάσης Δεδομένων με την οποία είναι συνδεδεμένο το Σύστημα, απεικονίζεται μια λίστα με διαθέσιμα πλοία τα οποία ανταποκρίνονται στα κριτήρια που έχει θέσει ο Ναυλωτής μέσω της φόρμας. Μέσα από τη λίστα αυτή ο Ναυλωτής έχει τη δυνατότητα να επιλέξει περισσότερα από ένα πλοία και για τα οποία να συμπληρώσει μια αίτηση στην οποία θα αναγράφονται και οι απαιτήσεις της μεταφοράς. Στο σημείο αυτό ο Ναυλωτής μπορεί να μεταβάλλει τα στοιχεία της φόρμας προκειμένου να εκτελέσει εκ νέου νέα αναζήτηση. Η αίτηση μεταφοράς για τα επιθυμητά πλοία συμπληρώνεται στο επόμενο βήμα όπου ο Ναυλωτής έχει τη δυνατότητα να καταγράψει λεπτομερώς όλες τις ιδιαιτερότητες του φορτίου καθώς και το είδος του ναύλου που επιθυμεί. Ο Ναυλωτής από τη στιγμή που θα αποστείλει την αίτηση, τότε αυτόματα η αίτηση αυτή αποστέλλεται αυτόματα σε όλους τους αποδέκτες των ενδιαφερόμενων πλοίων. Σημειώνεται ότι κάθε πλοίο είναι αντιστοιχισμένο σε ένα και μοναδικό Ναυλομεσίτη προκειμένου να επιτυγχάνεται καλύτερη διαχείριση αυτού.

Η αποστολή της αίτησης ενδιαφέροντος για ένα πλοίο γίνεται μέσα από ένα είδος επανάληψης. Το Σύστημα επικοινωνεί με τη Βάση Δεδομένων προκειμένου να βρει και να αποθηκεύσει τον αριθμό ταυτότητας κάθε χρήστη με τον οποίο είναι συνδεδεμένο το κάθε πλοίο. Εδώ είναι το σημείο στο οποίο ο Ναυλομεσίτης ξεκινά τη δεύτερη διαδικασία, στην

οποία αν έχει γίνει αίτηση για ένα πλοίο της αρμοδιότητάς του, τότε λαμβάνει ένα μήνυμα μέσω της διαδικασίας το οποίο του επιτρέπει να δει την αίτηση του Ναυλομεσίτη. Στην αίτηση αυτή εμφανίζονται όλα τα στοιχεία τα οποία έχει συμπληρώσει ο Ναυλωτής καθώς και όλες οι περαιτέρω λεπτομέρειες που καταγράφηκαν από τον συντάκτη της αίτησης. Στη φάση αυτή της διαδικασίας ο Ναυλομεσίτης μπορεί είτε να Απορρίψει και να τελειώσει στο σημείο αυτό η διαδικασία είτε να ξεκινήσει τη Διαπραγμάτευση. Στη φάση της Διαπραγμάτευσης ο Ναυλομεσίτης έχει το δικαίωμα να συμπληρώσει ένα κείμενο με τις δικές του παρατηρήσεις, σχόλια και απαιτήσεις όσων αφορά το πλοίο, τον τρόπο πληρωμής καθώς και άλλες λεπτομέρειες.

Στο σημείο αυτό το Σύστημα εξετάζει τη Βάση Δεδομένων έτσι ώστε να ελέγξει εάν έχει ήδη επιτευχθεί κάποια συμφωνία για τη συγκεκριμένη αίτηση με κάποια άλλη Ναυτιλιακή Εταιρεία. Σε περίπτωση που έχει επιτευχθεί ήδη άλλη συμφωνία τότε εμφανίζεται ένα μήνυμα στο παρών Ναυλομεσίτη ότι “Η συμφωνία επιτεύχθηκε με άλλη Ναυτιλιακή Εταιρεία”. Διαφορετικά αν εκκρεμεί ακόμα η συμφωνία τότε η Διαπραγματευτική Αίτηση του Ναυλομεσίτη αποστέλλεται στο συντάκτη- Ναυλωτή της αίτησης όπου με τη δική του σειρά Απορρίπτει, Αποδέχεται είτε Επαναδιαπραγματεύεται την αίτηση Ναύλωσης. στην περίπτωση της Επαναδιαπραγμάτευσης ο Ναυλωτής συμπληρώνει εκ νέου τα σχόλια του και την αποστέλλει στον ένα και μοναδικό αυτή τη φορά Ναυλομεσίτη με τον οποίο γίνεται η Διαπραγμάτευση. Η διαδικασία αυτή επαναλαμβάνεται έως ότου ένα από τους δύο την αποδεχτεί είτε την απορρίψει. Στην περίπτωση της Αποδοχής τότε αποστέλλεται στον Ναυλομεσίτη ένα αίτημα σύνταξης του Ναυλοσύμφωνου μέσα από ένα πρότυπο έγγραφο που έχει αποθηκευτεί στο Σύστημα.

Στη συνέχεια ο Ναυλομεσίτης αποστέλλει το Ναυλοσύμφωνο στο Ναυλωτή ,υπογεγραμμένο από τη Ναυτιλιακή Εταιρεία. Στο σημείο αυτό ο Ναυλωτής έχει τη δυνατότητα να ζητήσει να γίνουν αλλαγές σε περίπτωση που κάτι έχει καταγραφεί και δεν είναι σύμφωνο με τις απαιτήσεις του. Αν συμβεί αυτό τότε αποστέλλονται τα σχόλια στο Ναυλομεσίτη προκειμένου να γράψει εκ νέου το Ναυλοσύμφωνο. Η επανάληψη αυτή συνεχίζεται έως ότου και τα δύο μέρη συμφωνήσουν απόλυτα. Αν υπογράψει και ο Ναυλωτής τότε επιτεύχθηκε συμφωνία, ενημερώνεται η Βάση Δεδομένων προκειμένου να ενημερώσει με τη σειρά της και τις άλλες ίδιες αιτήσεις και αρχειοθετείται και από τα δύο μέρη.

Κατόπιν ο Ναυλομεσίτης οφείλει να μεταβάλλει τα δεδομένα αναχώρησης και άφιξης ενός πλοίου προκειμένου με αυτό τον τρόπο να ενημερωθεί και η Βάση Δεδομένων. Στο σημείο αυτό ο Ναυλομεσίτης μπορεί να μεταβάλλει τα δεδομένα ενός πλοίου και για μία συμφωνία που επιτεύχθηκε εκτός του Συστήματος, είτε τηλεφωνικά είτε μέσω Ηλεκτρονικού Ταχυδρομείου. Επιπλέον μπορεί να παρακολουθεί και το πρόγραμμα αφίξεων και αναχωρήσεων των πλοίων της Ναυτιλιακής του Εταιρείας καθώς και να μεταβάλλει κάθε φορά και το προσωπικό επάνδρωσης των πλοίων.

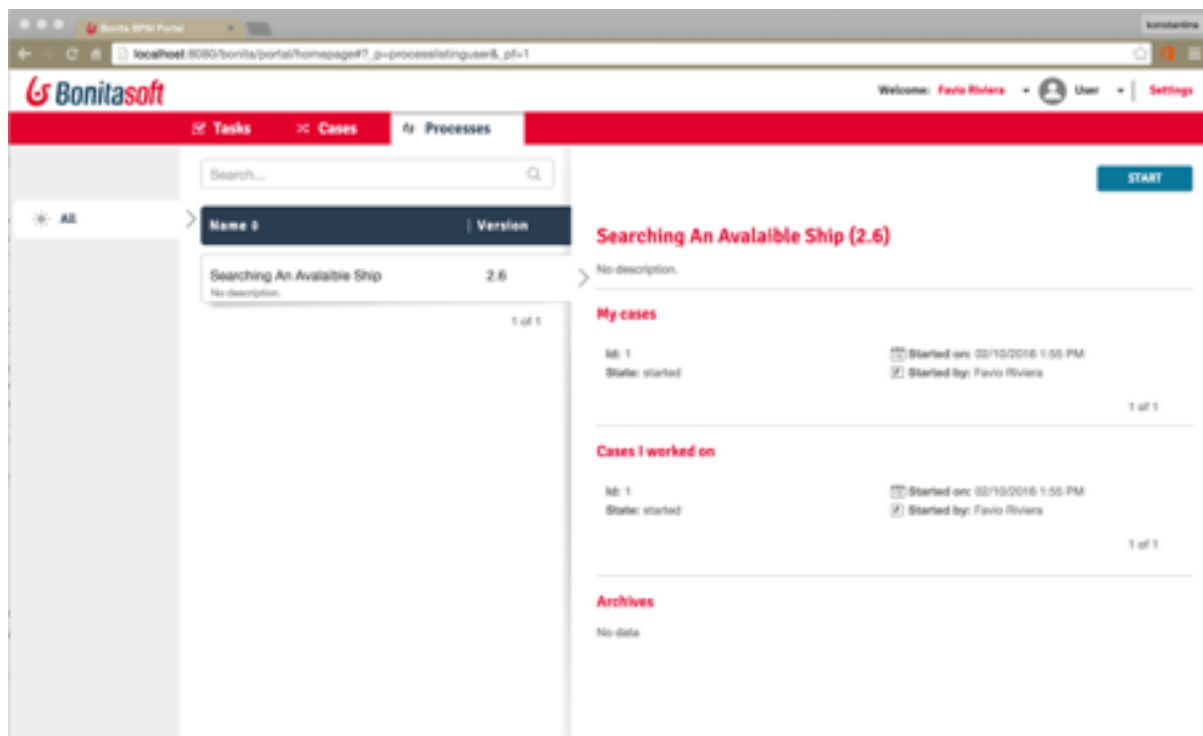
Κάθε ένα στάδιο από τα παραπάνω απεικονίζονται στις εικόνες της επόμενης παραγράφου.

6.4.1 Περίπτωση Χρήσης - Use Case Study Εύρεση πλοίου και Διαπραγμάτευση

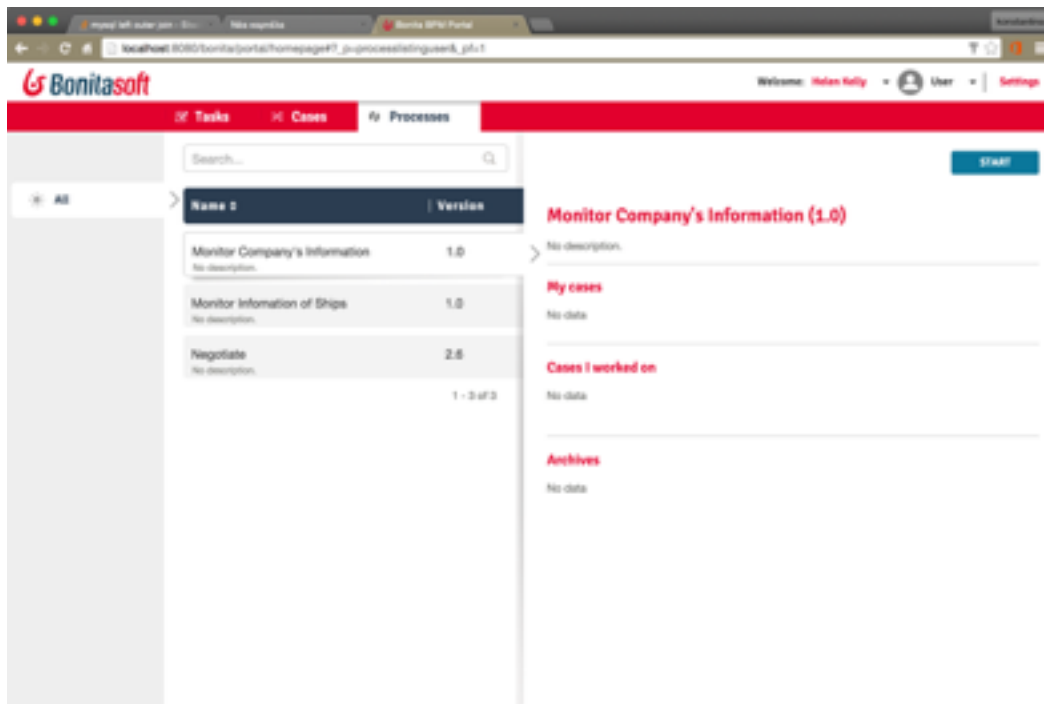
Οι χρήστες οι οποίοι μπορούν να χρησιμοποιήσουν την εφαρμογή έχουν χωριστεί σε δύο κατηγορίες:

- οι Ναυλωτές - Charterers (π.χ Favio Riviera, Patrick Gardenier, April Sanchez κτλ)
- οι Ναυλομεσίτες - Chartering Brokers (π.χ Walter Bates, Helen Kelly, William Jobs, Thomas Wallis κτλ)

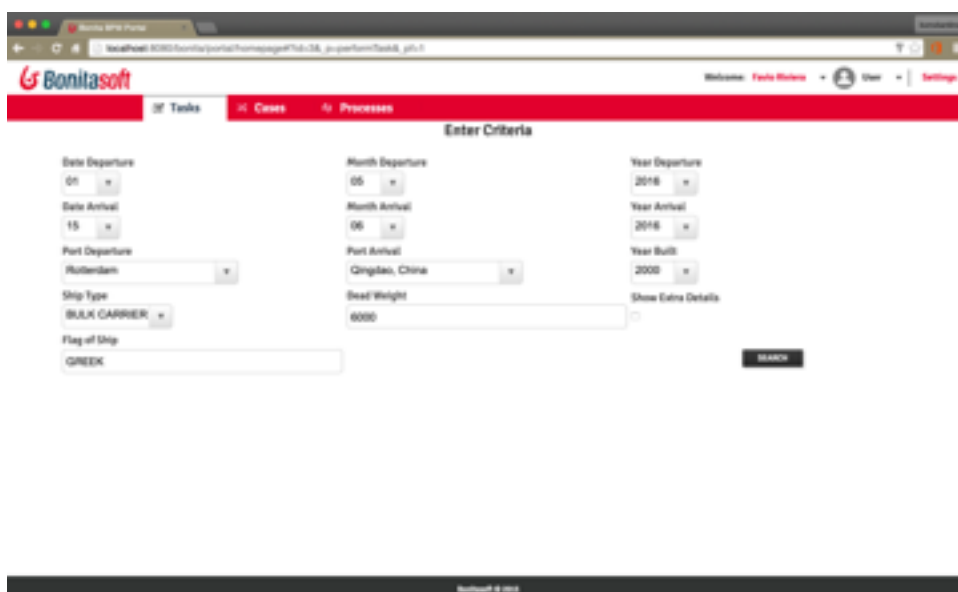
Η κάθε μία κατηγορία έχει περιορισμένα δικαιώματα. Για παράδειγμα ο Ναυλωτής μπορεί να ξεκινήσει τη διαδικασία για Έρευση πλοίου ενώ ο Ναυλομεσίτης όχι. Το ίδιο ισχύει και για τις άλλες δύο διαδικασίες δηλαδή τον Έλεγχο των πλοίων και τη Διαπραγμάτευση, όπως φαίνεται ξεκάθαρα και στις παρακάτω εικόνες.



Εικόνα 5.7: Ο χρήστης Ναυλωτής Favio Riviera εκκινεί τη διαδικασία Searching for a ship.



Εικόνα 5.8: Ο χρήστης Ναυλομεσίτης Helen Kelly εκκινεί τις διαδικασίες Monitor information of a Ship - Company & Negotiate.



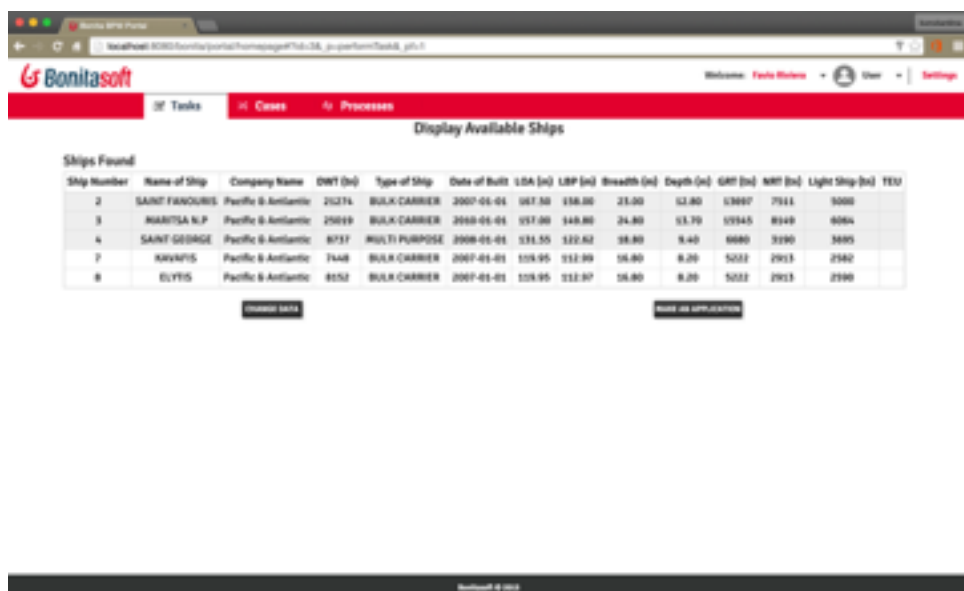
Εικόνα 5.9: Ο χρήστης Ναυλωτής Favió Riviera συμπληρώνει τη φόρμα με τα επιθυμητά κριτήρια.

Όπως ήδη έχουμε αναφέρει τη διαδικασία εκκινεί ο Ναυλωτής από την Έρευνα για πλοίο. Επομένως ο Ναυλωτής Favió Riviera εκκινεί τη διαδικασία και του εμφανίζεται μια φόρμα στην οποία συμπληρώνει τα επιθυμητά του στοιχεία για τη μεταφορά που θέλει να πραγματοποιήσει. Από την παρακάτω εικόνα που παρατίθεται βλέπουμε ότι ο Ναυλωτής έχει επιλέξει ημερομηνία αποχώρησης στις 2016-Μαΐου-01 και ημερομηνία άφιξης 2016-

Ιούνιος-15 με λιμάνι αποχώρησης το λιμάνι Rotterdam και λιμάνι άφιξης το λιμάνι Qingdao στη Κίνα. Η σημαία που επιθυμεί να έχει το καράβι είναι ελληνική με κατασκευή πλοίου μετά το έτος 2000 με τύπο πλοίου “Bulk Carrier” και νεκρό φορτίο να είναι στους 6000 τόνους. Επιπλέον στο σημείο αυτό ο Ναυλωτής έχει τη δυνατότητα να επιλέξει άμα θέλει λεπτομέρειες στη λίστα με τα διαθέσιμα πλοία ή όχι. Ο χρήστης όταν πατήσει το κουμπί Search τότε του εμφανίζεται μια λίστα με τα διαθέσιμα πλοία. Στο σημείο αυτό ο χρήστης μπορεί να αλλάξει πάλι τα κριτήρια σε περίπτωση λάθους.

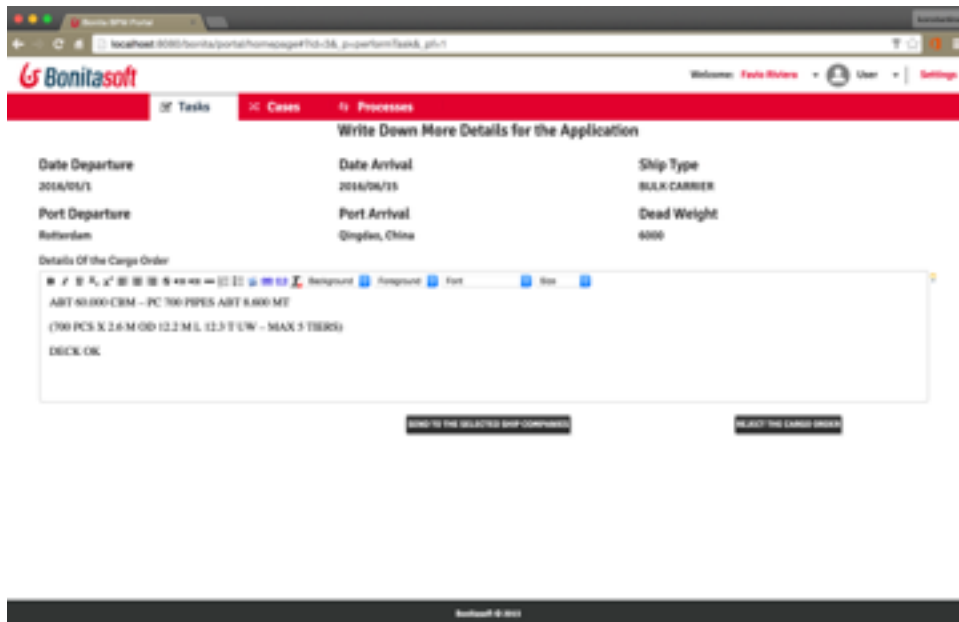


Εικόνα 5.10: Ο χρήστης Ναυλωτής Fανίο Riviera βλέπει όλα τα διαθέσιμα πλοία για τα κριτήρια που επιθυμεί με λιγότερες λεπτομέρειες.



Εικόνα 5.11: Ο χρήστης Ναυλωτής Fανίο Riviera βλέπει όλα τα διαθέσιμα πλοία για τα κριτήρια που επιθυμεί με περισσότερες λεπτομέρειες και επιλεγμένα τα επιθυμητά πλοία.

Μέσα από τη λίστα που του εμφανίζεται ο χρήστης μπορεί να επιλέξει όσα πλοία επιθυμεί και για τα οποία θα μπορεί να κάνει αίτηση μεταφοράς. Έστω ότι επιλέγει τα πλοία υπ' αριθμόν Νο 2, 3 & 4 για να κάνει αίτηση. Στην επόμενη σελίδα εμφανίζεται η αίτηση και στο σημείο αυτό μπορεί να συμπληρώσει οποιοσδήποτε λεπτομέρειες επιθυμεί να είναι γνωστές στο Ναυλομεσίτη και θα συμβάλλουν σημαντικά στην εύρεση της καλύτερης τιμής.



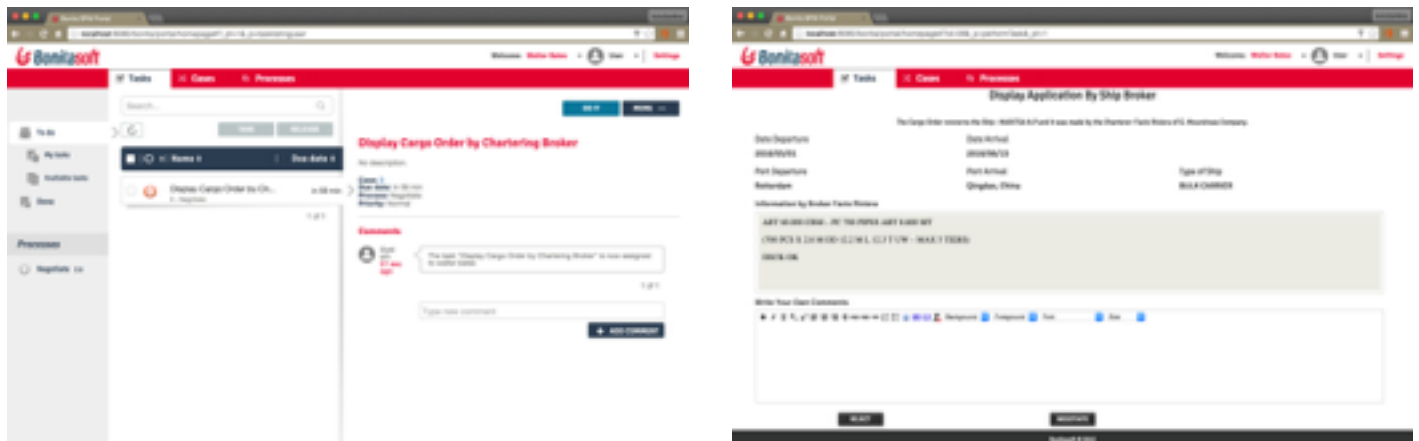
Εικόνα 5.12: Ο χρήστης Ναυλωτής Favió Rivièra έχει στη διάθεση του όλα τα κριτήρια που έθεσε και έχει τη δυνατότητα να συμπληρώσει τις παρατηρήσεις του.

Στο σημείο αυτό ο Ναυλωτής μπορεί να απορρίψει την αίτηση για μεταφορά για οποιοσδήποτε λόγους. Τη στιγμή που ο Ναυλωτής πατήσει για αποστολή της αίτησης τότε

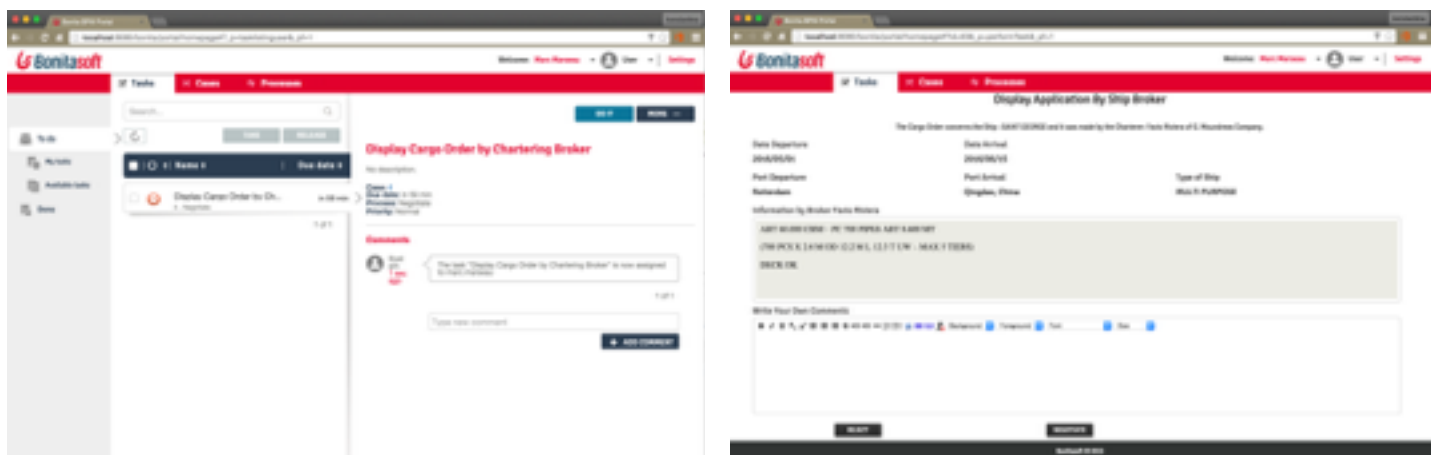


Εικόνα 5.13- 5.14 : Ο χρήστης Ναυλομεσίτης Helen Kelly έχει στη διάθεση του όλα τα κριτήρια που έθεσε ο Ναυλωτής και το όνομα του πλοίου που αφορά η αίτηση μεταφοράς.

ταυτόχρονα οι αιτήσεις αποστέλλονται σε όλους τους Ναυλομεσίτες τους οποίους αφορούν τα πλοία. Από τη Βάση Δεδομένων έχουμε ορίσει ότι το πλοίο υπ' αριθμόν 2 είναι χρεωμένο στο χρήστη Helen Kelly, το πλοίο υπ' αριθμόν 3 είναι χρεωμένο στο χρήστη Walter Bates και το πλοίο υπ' αριθμόν 4 είναι χρεωμένο στο χρήστη Marc Marseau. Εάν μπούμε στο προφίλ του κάθε χρήστη θα δούμε αντίστοιχα και τις αιτήσεις.



Εικόνα 5.15- 5.16 : Ο χρήστης Ναυλομεσίτης Walter Bates έχει στη διάθεση του όλα τα κριτήρια που έθεσε ο Ναυλωτής και το όνομα του πλοίου που αφορά η αίτηση μεταφοράς.

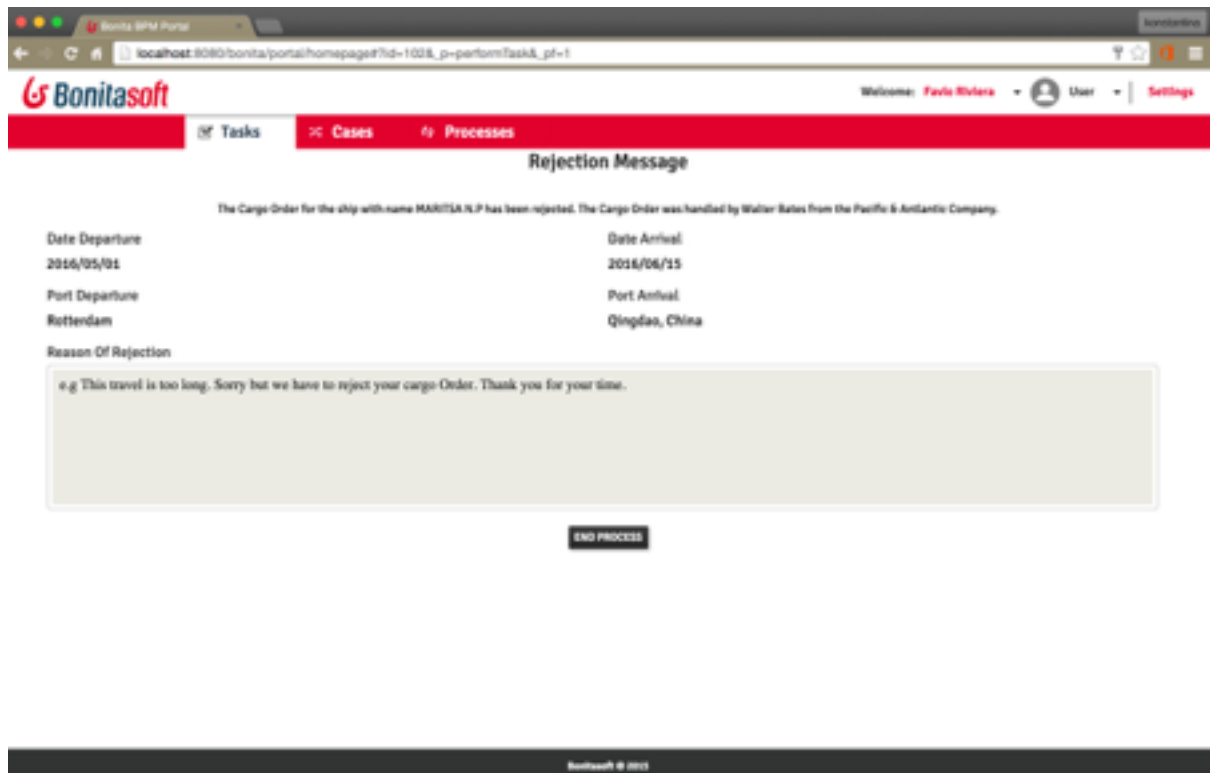


Εικόνα 5.17- 5.18 : Ο χρήστης Ναυλομεσίτης Marc Marseau έχει στη διάθεση του όλα τα κριτήρια που έθεσε ο Ναυλωτής και το όνομα του πλοίου που αφορά η αίτηση μεταφοράς.

Στις παραπάνω αιτήσεις αναγράφεται το όνομα του πλοίου για το οποίο έχει ενδιαφερθεί ο Ναυλωτής καθώς και το όνομα του Ναυλωτή και από ποια εταιρεία Ναυλώσεων προέρχεται. Επιπλέον καταγράφονται οι λεπτομέρειες για τις ημερομηνίες και τα λιμάνια αναχώρησης και άφιξης αλλά και το μέγεθος της μεταφοράς. Τέλος παρουσιάζονται και τα επιπλέον σχόλια του Ναυλωτή.

Για κάθε μία αίτηση τώρα θα εξετάσουμε διαφορετικά σενάρια περιπτώσεων. Έστω για την παραπάνω αίτηση μεταφοράς ενδιαφέρονται μόνο οι χρήστες Helen Kelly και Marc

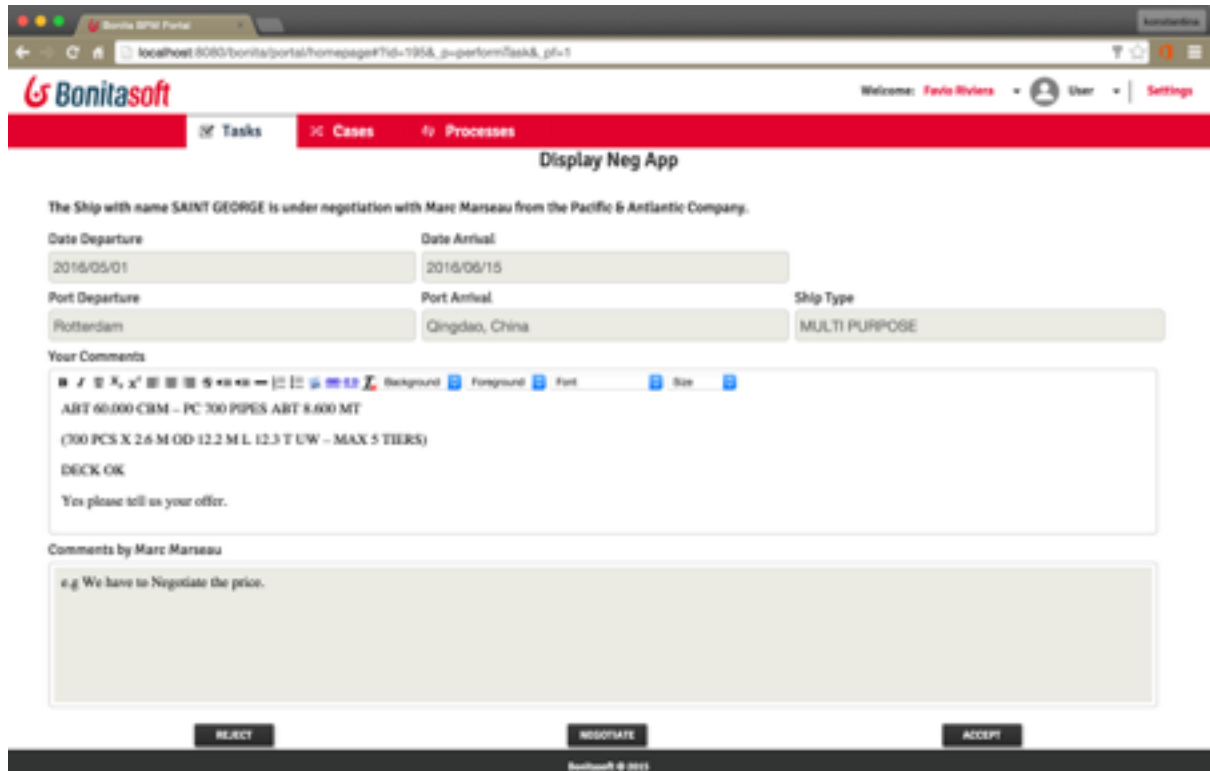
Marseau ενώ ο χρήστης Walter Bates την απορρίπτει για δικούς του λόγους. Ο χρήστης Walter Bates θα απαντήσει στην αίτηση μεταφοράς μέσω του κουμπι απόρριψης καταγράφοντας και τους λόγους απόρριψης στο Ναυλωτή. Στο προφίλ του Ναυλωτή εμφανίστηκε η απόρριψη της αίτησης μεταφοράς και στην οποία αναγράφονται οι λεπτομέρειες απόρριψης καθώς και το όνομα του Ναυλομεσίτη που τη διαχειρίστηκε. Μετά από αυτό το βήμα της διαδικασίας τελειώνει και η αίτηση μεταφοράς για το πλοίο του Ναυλομεσίτη.



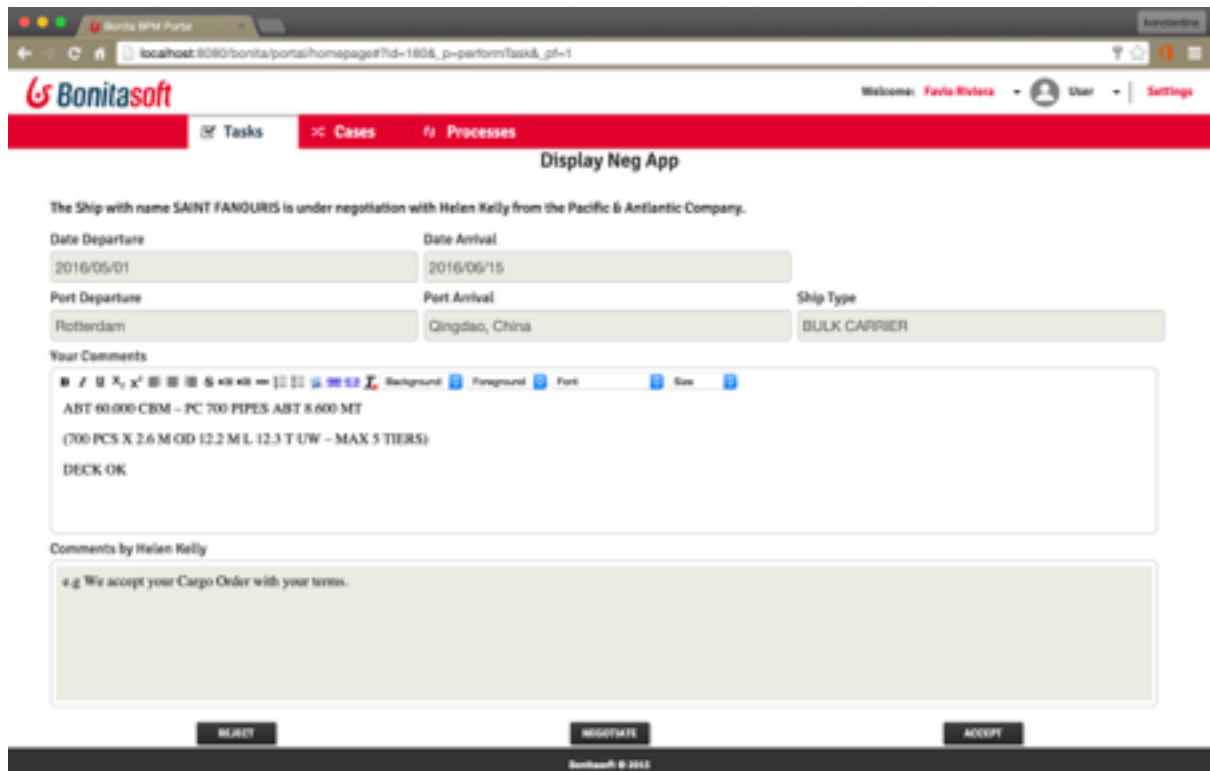
Εικόνα 5.19 : Ο χρήστης Ναυλομεσίτης Walter Bates έχει απορρίψει την αίτηση και ο Ναυλωτής βλέπει τους λόγους απόρριψης της αίτησης μεταφοράς.

Για τους χρήστες Helen Kelly και Marc Marseau ισχύει ότι συμπληρώνουν τα σχόλιά τους για την αίτηση μεταφοράς την αποστέλλουν στο Ναυλωτή και ο Ναυλωτής βλέπει και τις δύο απαντήσεις και αποφασίζει ποια αίτηση θα απορρίψει, ποια θα αποδεχτεί ή με ποια θα συνεχίσει να διαπραγματεύεται. Αυτή η διαδικασία της διαπραγμάτευσης μπορεί να συνεχιστεί και για τις δύο αιτήσεις έως ότου βρεθεί μια αποδεκτή συμφωνία είτε με το Ναυλομεσίτη Helen Kelly είτε με το Marc Marseau. Υποθέτουμε ότι ο Ναυλωτής Favio Riviera διαπραγματεύεται και με τις δυο αιτήσεις και στο τέλος αποδέχεται αυτή της Helen Kelly. Έστω ο Ναυλομεσίτης Marc Marseau έχει παράλβει την αίτηση για διαπραγμάτευση και συντάσσει τους όρους της διαπραγμάτευσης. Όμως ταυτόχρονα ο Ναυλωτής έχει ήδη αποδεχτεί την αίτηση μεταφοράς με το Ναυλομεσίτη Helen Kelly. Τότε αυτόματα ενημερώνεται η Βάση Δεδομένων ότι έχει υπάρξει συμφωνία και θα πρέπει να ενημερωθούν τυχόν εκκρεμούσες αιτήσεις μεταφοράς προκειμένου να ακυρωθούν. Στην προκειμένη

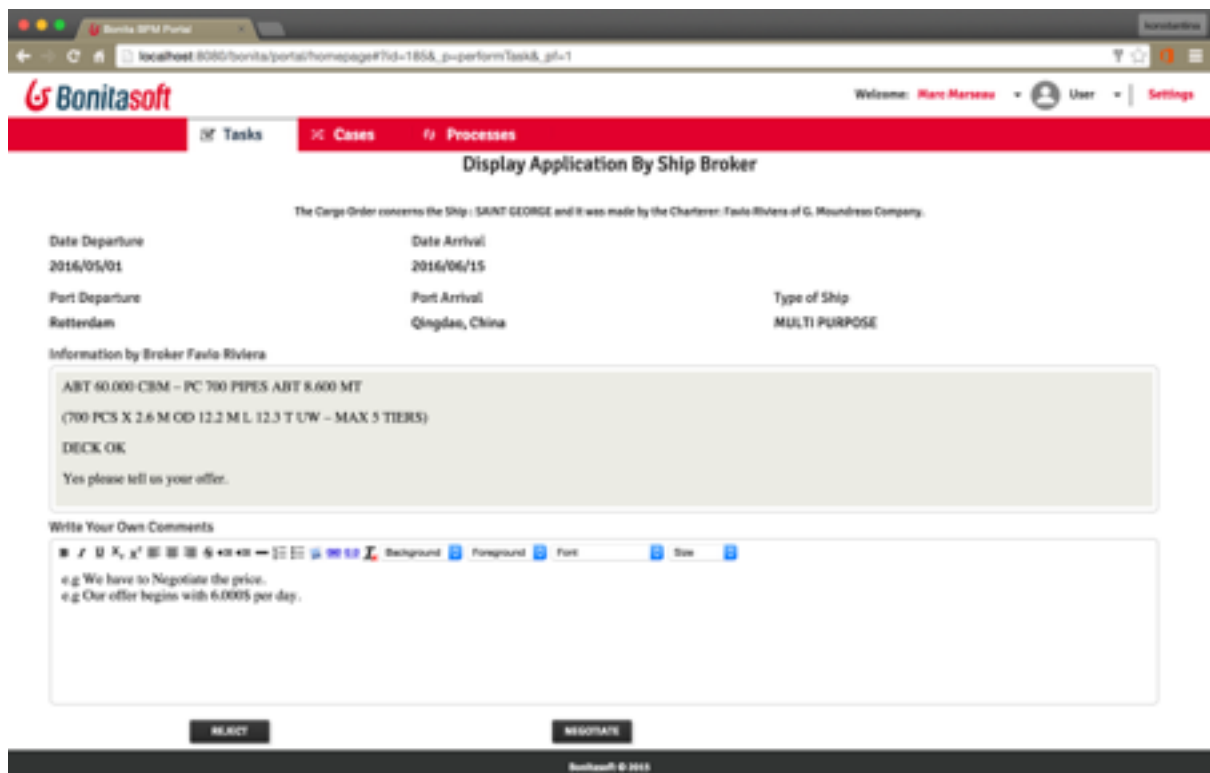
περίπτωση η αίτηση του Ναυλομεσίτη Marc Marseau. Στην εικόνα 5.21 είναι και η στιγμή που ο Ναυλωτής Favio Riviera αποδέχεται την αίτηση μεταφοράς και αποστέλλει στο Ναυλομεσίτη αίτημα για σύνταξη του ναυλοσυμφώνου.



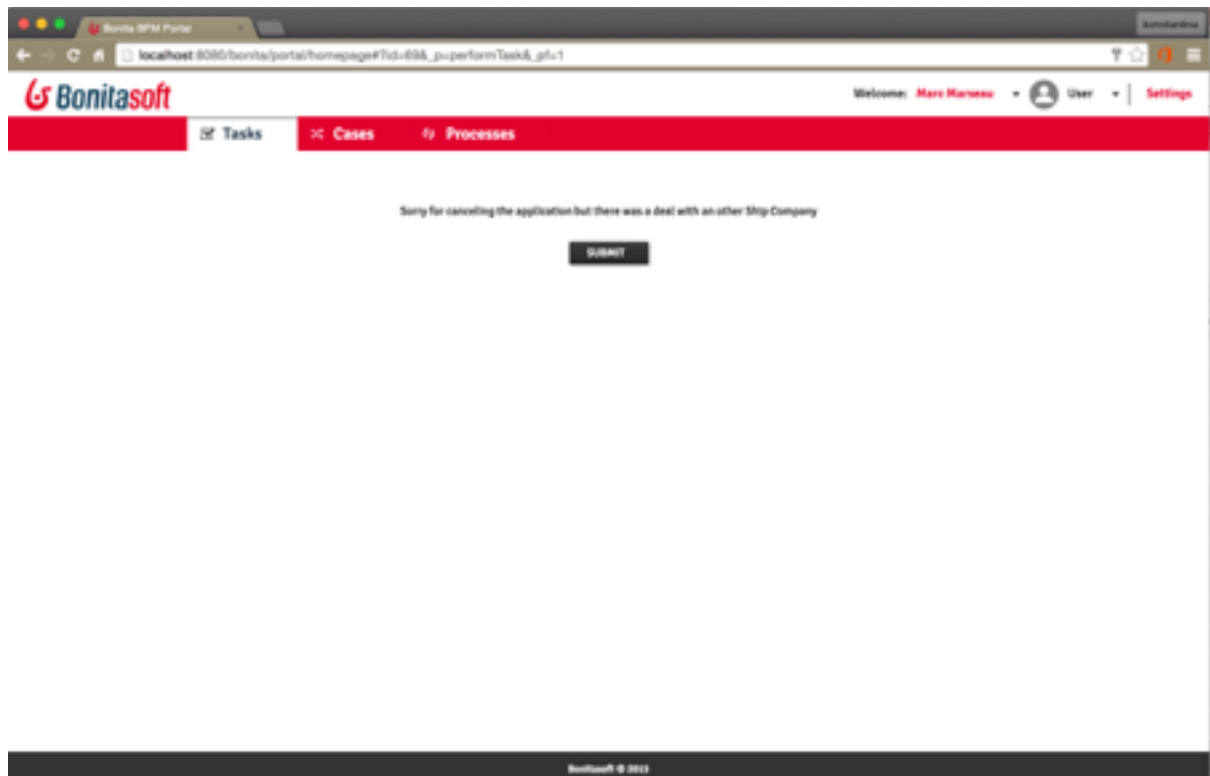
Εικόνα 5.20: Ο Ναυλωτής Favio Riviera διαπραγματεύεται την αίτηση μεταφοράς για το Ναυλομεσίτη Marc Marseau.



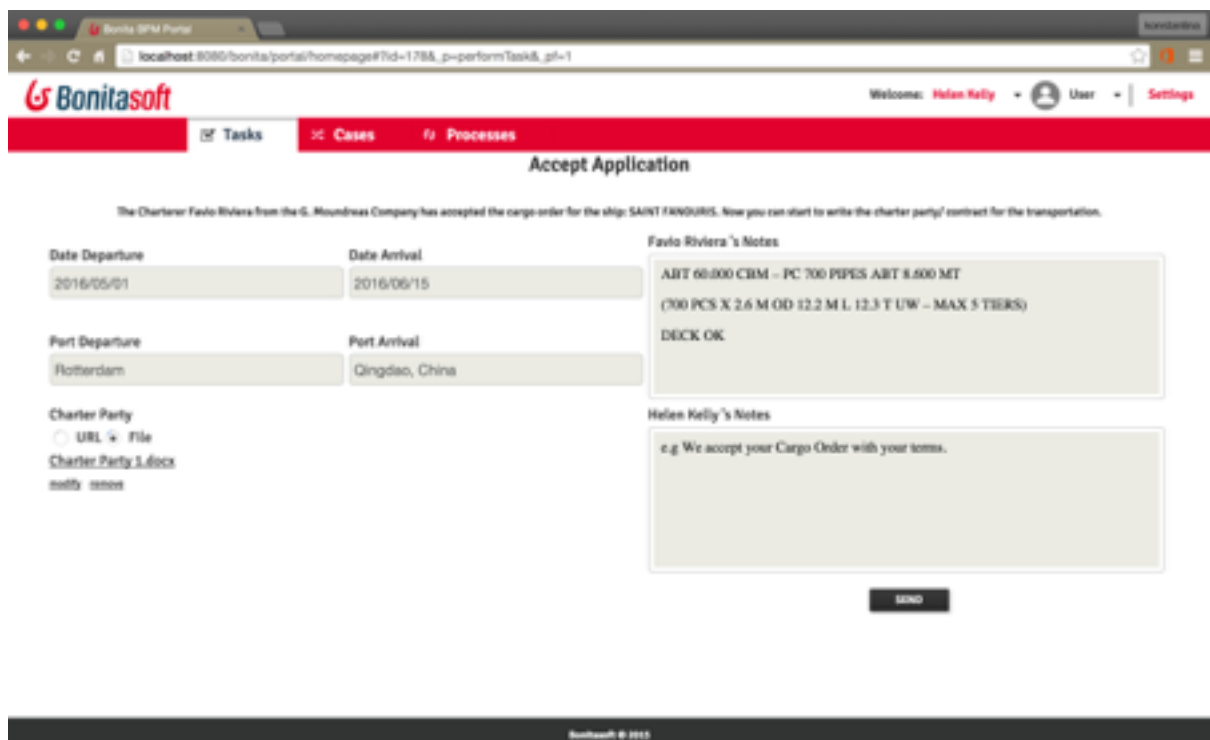
Εικόνα 5.21: Ο Ναυλωτής Fazio Riviera αποδέχεται την αίτηση μεταφοράς για το Ναυλομεσίτη Helen Kelly.



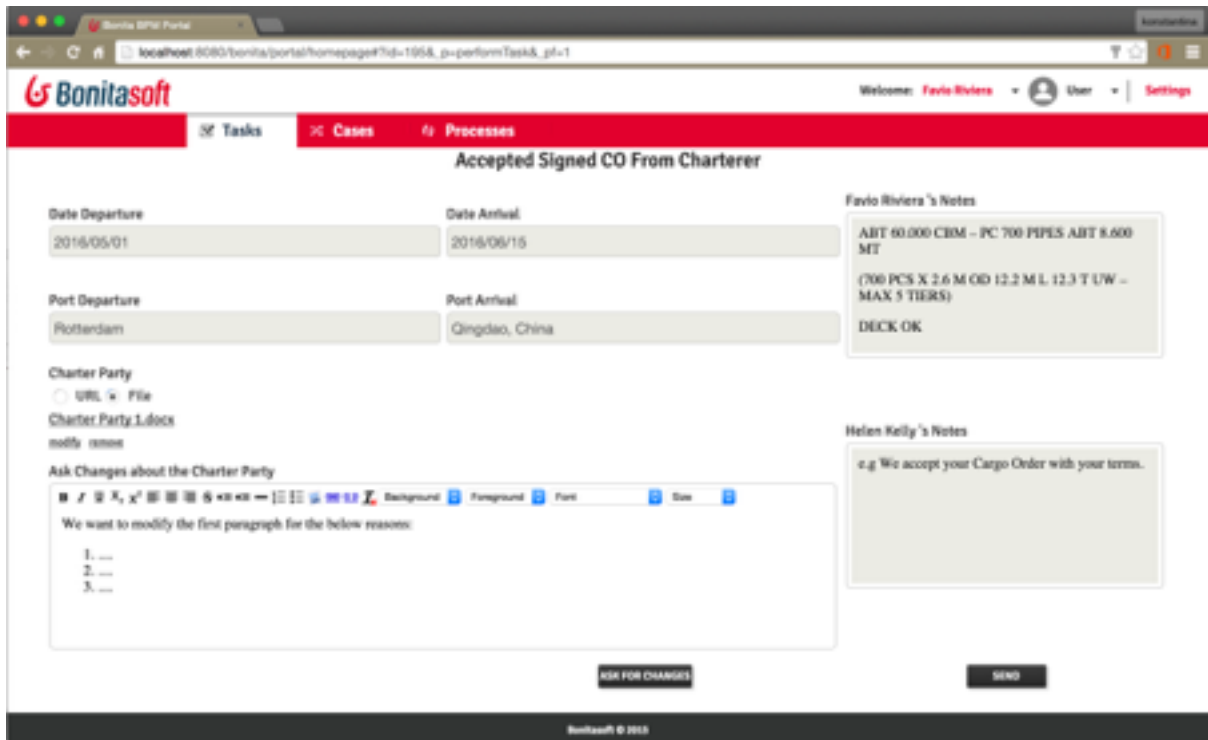
Εικόνα 5.22 : Ο χρήστης Ναυλομεσίτης Marc Marseau καταγράφει την προσφορά του και την αποστέλλει στο Ναυλωτή.



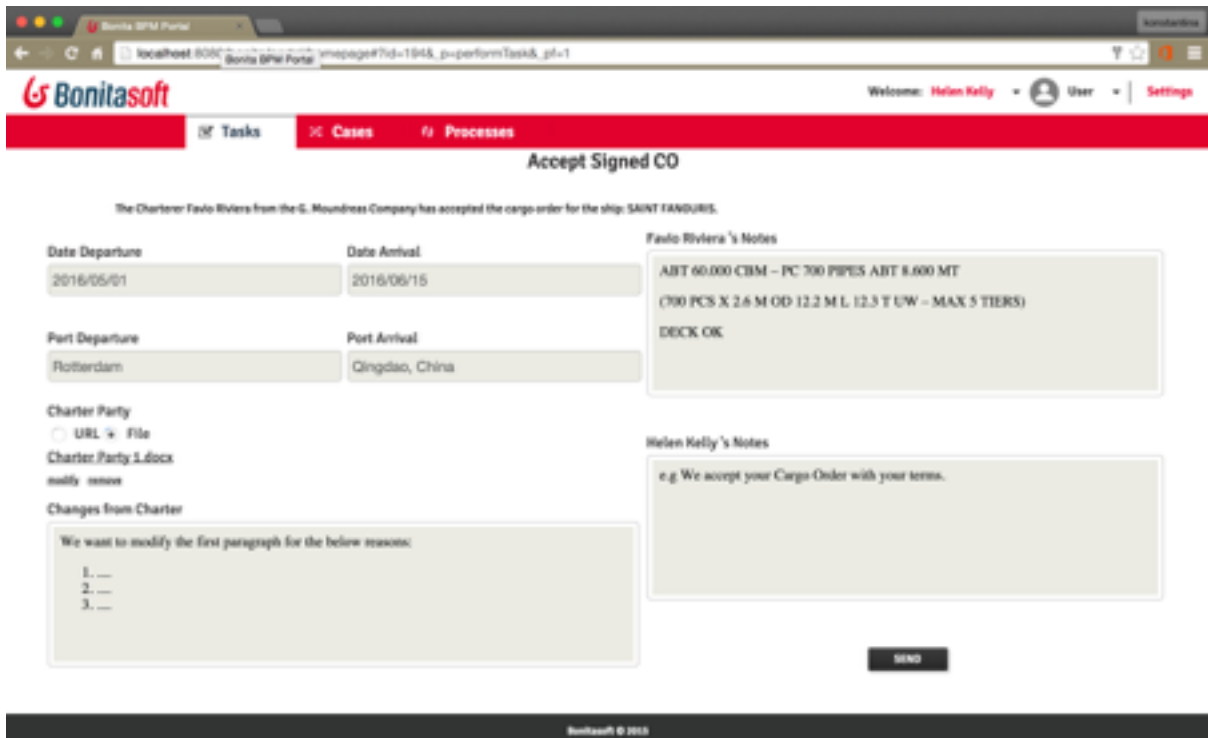
Εικόνα 5.23 : Ο χρήστης Ναυλομεσίτης Marc Marseau λαμβάνει ένα μήνυμα διακοπής της Διαπραγμάτευσης καθώς έχει προηγηθεί μια συμφωνία. Έτσι διακόπτεται και η Διαπραγμάτευση των δύο μερών.



Εικόνα 5.24 : Ο χρήστης Ναυλομεσίτης Helen Kelly λαμβάνει την αποδοχή του Ναυλωτή για την αίτηση μεταφοράς καθώς και ένα αίτημα σύνταξης του Ναυλοσυμφώνου.



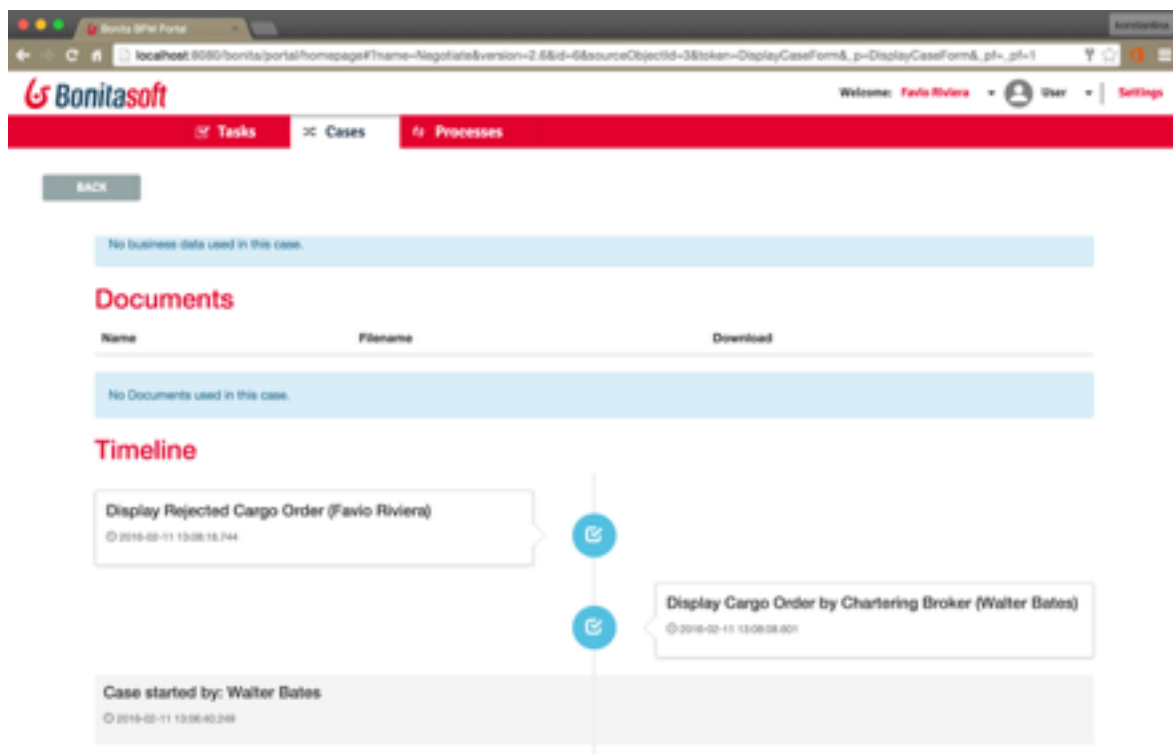
Εικόνα 5.25: Ο Ναυλωτής Favió Riviera λαμβάνει το Ναυλοσύμφωνο του Ναυλομεσότη Helen Kelly και μπορεί να ζητήσει αλλαγές σε περίπτωση διαφωνίας.



Εικόνα 5.25 : Ο χρήστης Ναυλομεσότης Helen Kelly λαμβάνει τις αλλαγές του Ναυλωτή και το ζανα συντάσσει και το αποστέλλει εκ νέου.



Εικόνα 5.26 - 5.27 : Στη περίπτωση που το αποδεχτούν και οι δύο τότε το Ναυλοσύμφωνο αποστέλλεται από κοινού και στα δύο μέρη όπου μπορούν να το “κατεβάσουν” σε τοπικό φάκελο για αποθήκευση.



Εικόνα 5.28 : Ο χρήστης Favio Riviera βλέπει το ιστορικό της επικοινωνίας με το Ναυλομεσίτη Walter Bates.

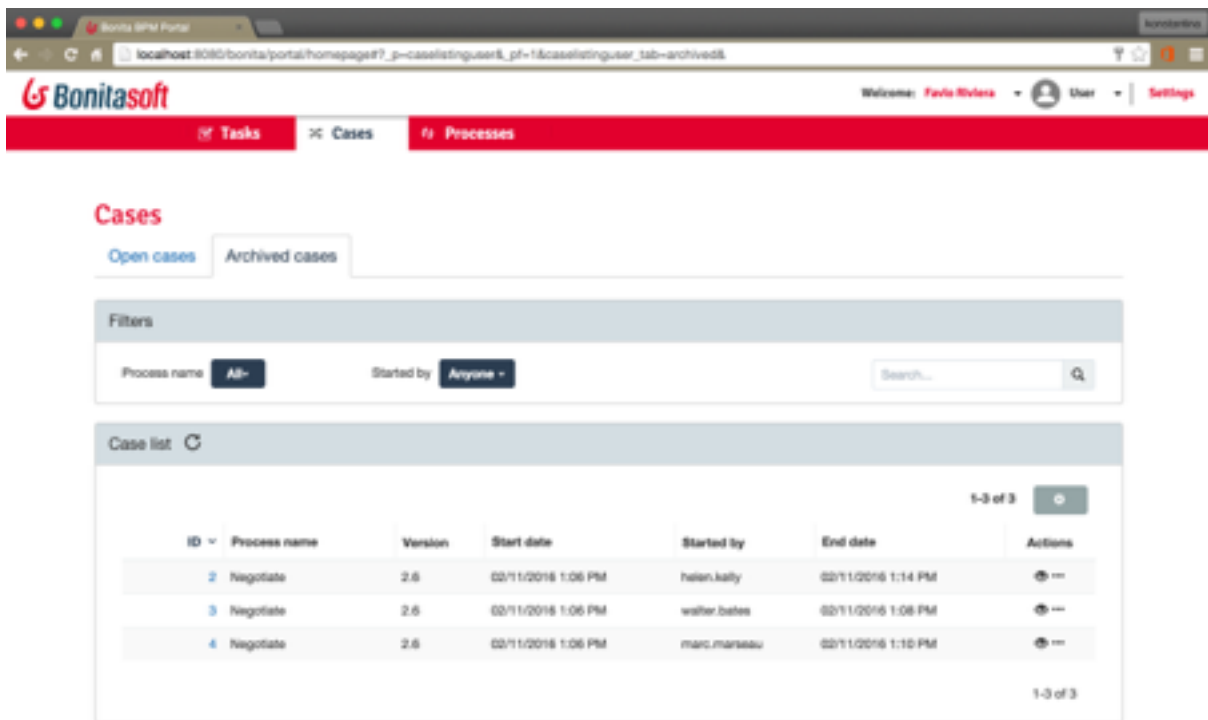
The screenshot shows the Bonitasoft BPM Portal interface. At the top, there is a navigation bar with 'Tasks', 'Cases', and 'Processes' tabs. The 'Cases' tab is active. Below the navigation bar, there is a 'BACK' button. The main content area is divided into three sections: 'Business data', 'Documents', and 'Timeline'. The 'Business data' section shows 'No business data used in this case.' The 'Documents' section shows 'No Documents used in this case.' The 'Timeline' section displays a vertical timeline of events. The events are: 'Case started by: Marc Marseau' (2016-02-11 13:08:54.842), 'Display Rejection (Marc Marseau)' (2016-02-11 13:10:02.267), 'Display Negotiated Cargo Order (Favio Riviera)' (2016-02-11 13:09:35.202), 'Display Cargo Order by Chartering Broker (Marc Marseau)' (2016-02-11 13:10:02.294), and 'Display Cargo Order by Chartering Broker (Marc Marseau)' (2016-02-11 13:09:01.123). The timeline is centered around a vertical line with circular markers.

Εικόνα 5.29 : Ο χρήστης Marc Marseau βλέπει το ιστορικό της επικοινωνίας με το Favio Riviera.

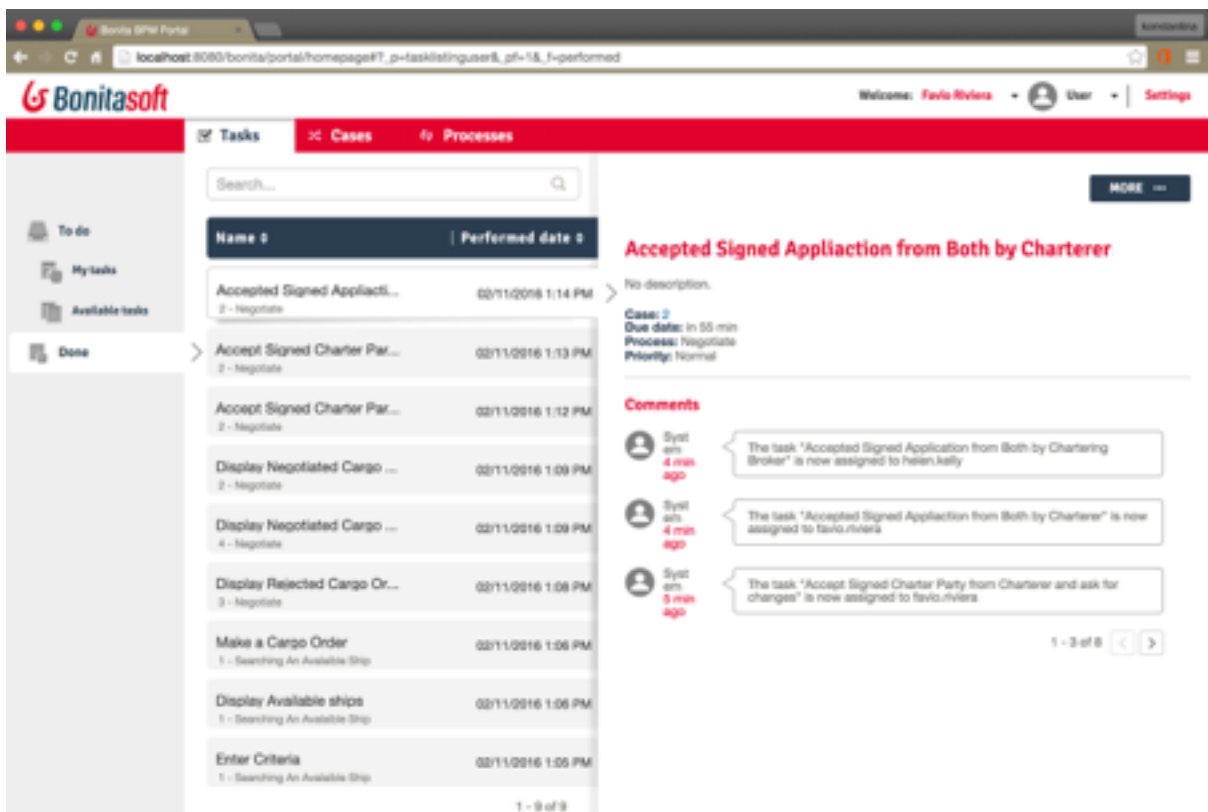
The screenshot shows a web browser window displaying the Bonitasoft portal. The page title is "Timeline". The navigation bar includes "Tasks", "Cases", and "Processes". The user is logged in as "Helen Kelly". The timeline consists of the following events:

- Accepted Signed Application from Both by Chartering Broker (Helen Kelly) - 2016-02-11 13:14:45.529
- Accepted Signed Application from Both by Charterer (Favio Riviera) - 2016-02-11 13:14:51.412
- Accept Signed Charter Party from Charterer and ask for changes (Favio Riviera) - 2016-02-11 13:13:58.822
- Accepted Cargo Order (Helen Kelly) - 2016-02-11 13:13:24.191
- Accept Signed Charter Party from Charterer and ask for changes (Favio Riviera) - 2016-02-11 13:12:56.575
- Accepted Cargo Order (Helen Kelly) - 2016-02-11 13:11:25.809
- Display Negotiated Cargo Order (Favio Riviera) - 2016-02-11 13:09:43.298
- Display Cargo Order by Chartering Broker (Helen Kelly) - 2016-02-11 13:08:15.042
- Case started by: Helen Kelly - 2016-02-11 13:06:27.636

Εικόνα 5.30 : Ο χρήστης Helen Kelly βλέπει το ιστορικό της επικοινωνίας με το Favio Riviera.



Εικόνα 5.30 : Ο χρήστης Favió Riviera μπορεί να δει με ποιους είχε αρχίσει την επικοινωνία για Διαπραγμάτευση.



Εικόνα 5.31 : Ο χρήστης Favió Riviera μπορεί να δει όλες τις διαδικασίες που εκτελέστηκαν κατά τη διαδικασία της Διαπραγμάτευσης.

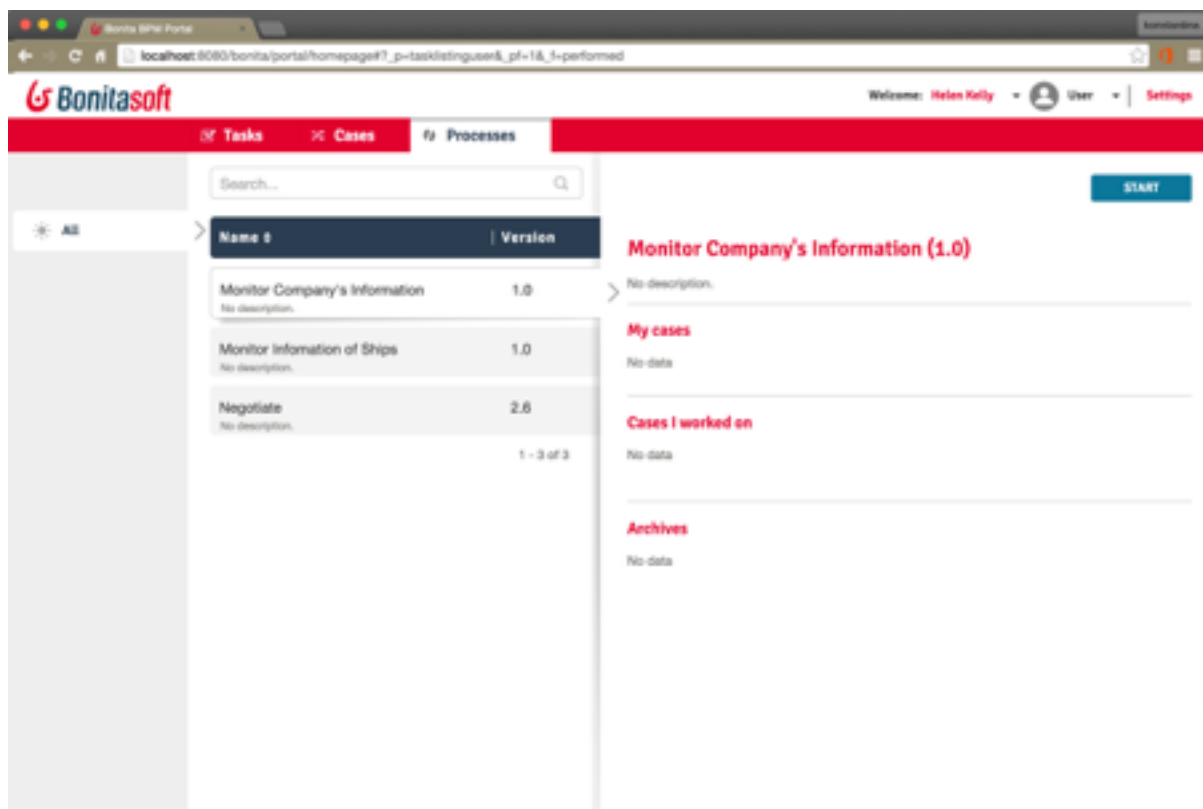
6.4.2 Περίπτωση Χρήσης - Use Case Study Έλεγχος Πληροφοριών Πλοίων και Εταιρείας

Τις δύο συγκεκριμένες διαδικασίες μπορεί να εκκινήσει μόνο ο χρήστης με δικαιώματα Ναυλομεσίτη - Chartering Broker. Όπως ήδη έχουμε αναφέρει παραπάνω τέτοιοι χρήστες για παράδειγμα είναι οι Walter Bates, Helen Kelly, Thomas Wallis κτλ.

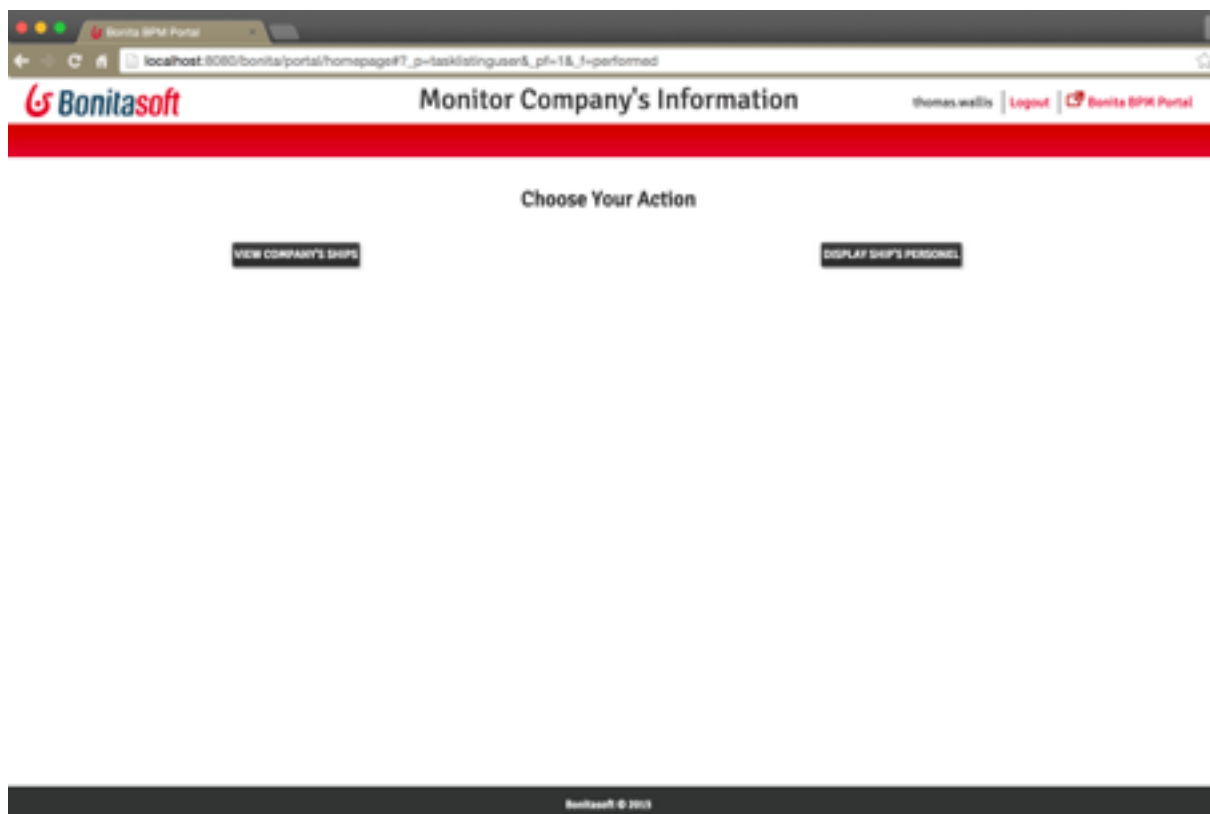
Στη διαδικασία Έλεγχος Πληροφοριών της Εταιρείας ο Ναυλομεσίτης έχει τη δυνατότητα:

- Να δει το προωπικό του πλοίου πάνω σε κάθε πλοίο
- Να δει το πρόγραμμα των πλοίων
- Να αλλάξει το προσωπικό με το οποίο είναι επανδρωμένο το πλοίο.

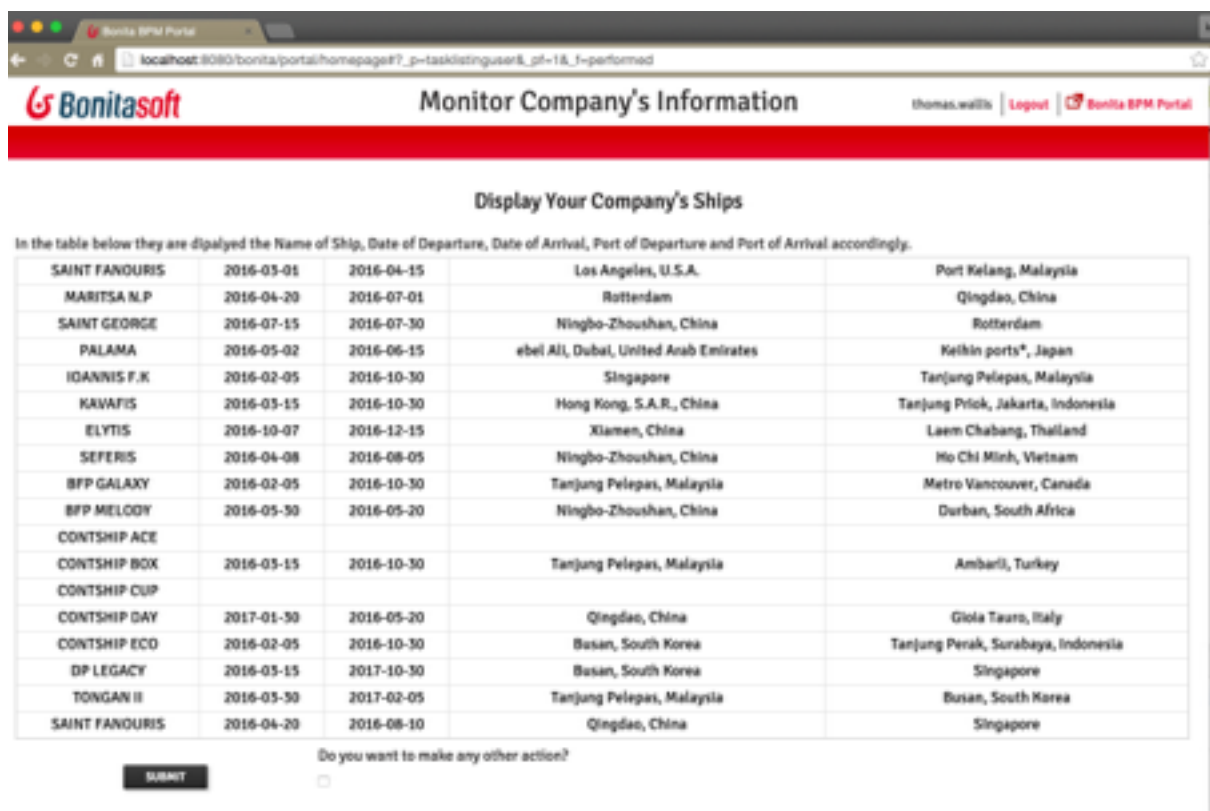
Οι ενέργειες φαίνονται και στις παρακάτω εικόνες.



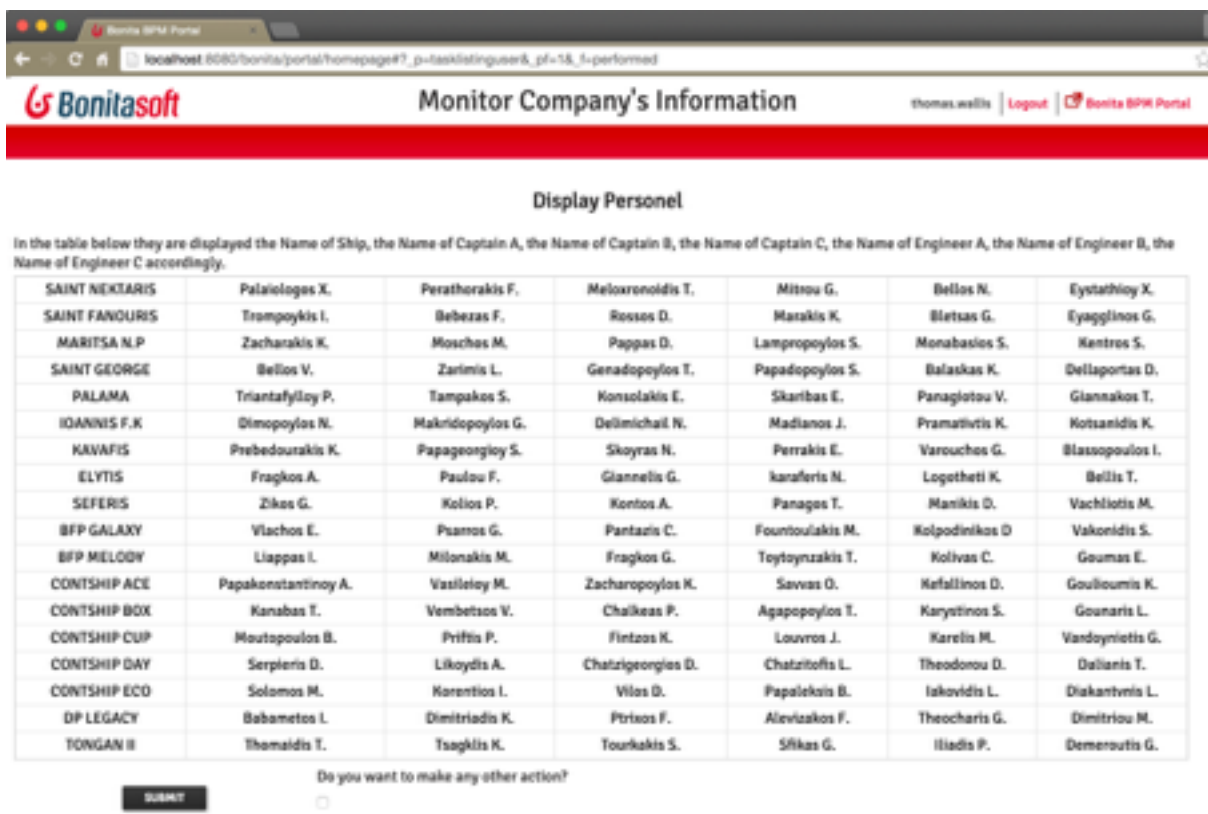
Εικόνα 5.32 : Ο χρήστης Helen Kelly μπορεί να εκκινήσει τις παραπάνω διαδικασίες.



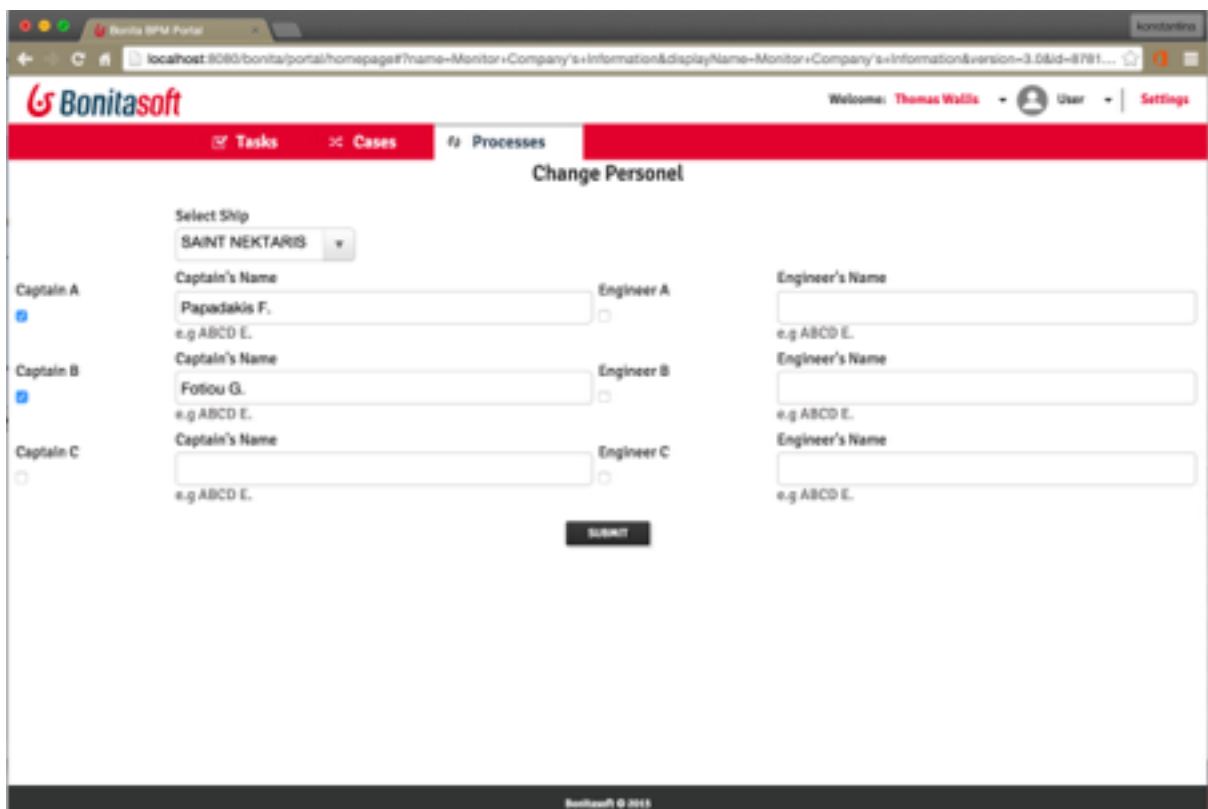
Εικόνα 5.34 : Ο χρήστης Thomas Wallis μπορεί να επιλέξει μία από τις παράπανω ενέργειες.



Εικόνα 5.35 : Ο χρήστης Walter Bates έχει επιλέξει να δει το πρόγραμμα των πλοίων της εταιρείας του.



Εικόνα 5.36 : Ο χρήστης Thomas Wallis έχει επιλέξει να δει το προσωπικό με το οποίο είναι επανδρωμένα τα πλοία της εταιρείας του και να κάνει αλλαγές.



Εικόνα 5.37 : Ο χρήστης Thomas Wallis έχει επιλέξει να αλλάξει τα ονόματα των A και B Καπετάνιων.

shipId	captainA	captainB	captainC	engineerA	engineerB	engineerC
1	Papadakis F.	Fotiou G.	Melaxronoidis T.	Mitrou G.	Bellos N.	Eystathioy X.
2	Trompoykis I.	Bebezias F.	Rossos D.	Marakis K.	Blatsas G.	Eyagglinos G.

Εικόνα 5.38 : Ο χρήστης *Walter Bates* επιτυχώς έχει αλλάξει τα ονόματα του προσωπικού.

Στη διαδικασία Έλεγχος Πληροφοριών των Πλοίων ο Ναυλομεσίτης έχει τη δυνατότητα:

- Να αλλάξει τα δεδομένα από μία συμφωνία που έκλεισε μέσω της εφαρμογής
- Να αλλάξει τα δεδομένα από μία συμφωνία που έκλεισε εκτός της εφαρμογής
- Να διαγράψει το πρόγραμμα ενός πλοίου όταν αυτό φτάσει στον προορισμό του.

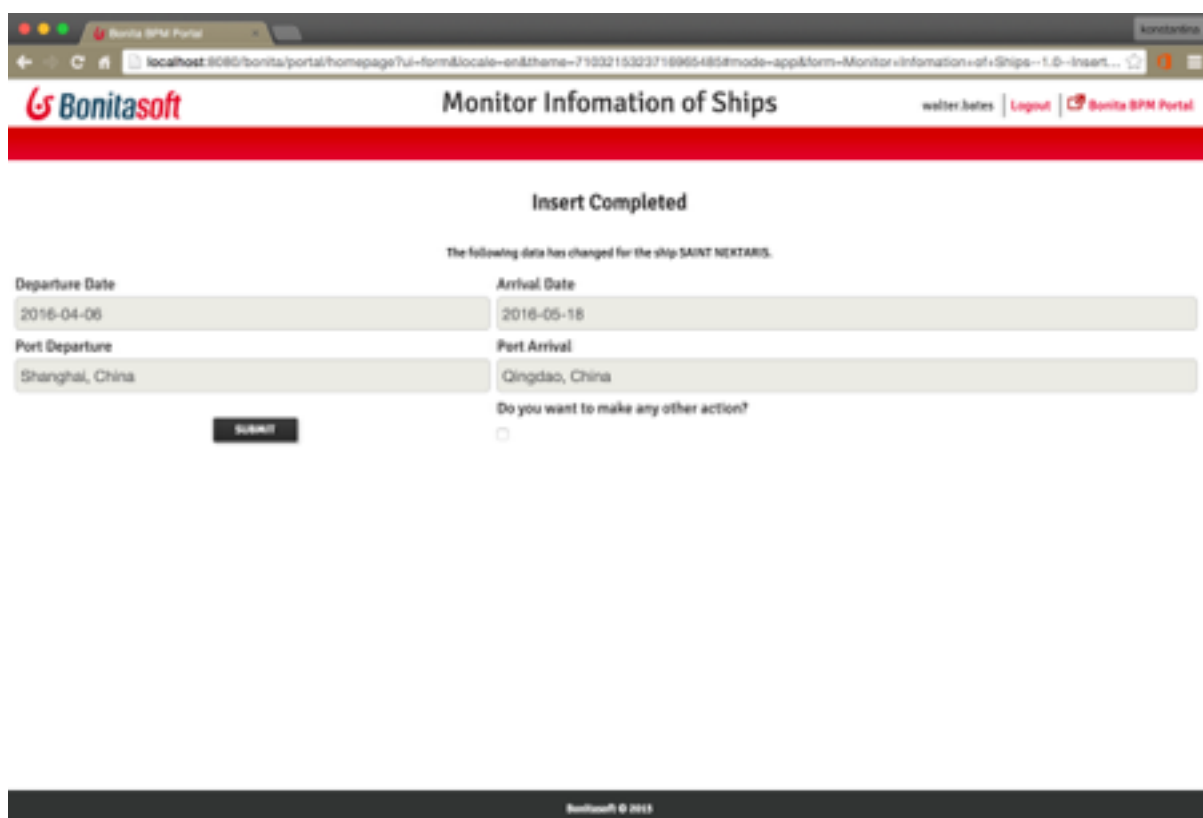
Οι ενέργειες φαίνονται και στις παρακάτω εικόνες.



Εικόνα 5.39: Ο χρήστης *Walter Bates* μπορεί να επιλέξει μια από τις παραπάνω ενέργειες.



Εικόνα 5.40 : Ο χρήστης Walter Bates επιλέγει το πλοίο και εισάγει τα δεδομένα που επιθυμεί να αλλάξει στο πρόγραμμα των πλοίων από μία συμφωνία εκτός εφαρμογής.



Εικόνα 5.41 : Ο χρήστης Walter Bates βλέπει τις αλλαγές που έγιναν στο πρόγραμμα των πλοίων.

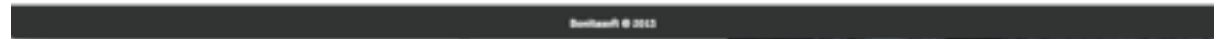
18	18	2016-03-30	2017-02-05	Tanjung Pelepas, Malaysia	Busan, South Korea	Pacific & Atlantic
19	1	2016-04-06	2016-05-18	Shanghai, China	Qingdao, China	Pacific & Atlantic

Εικόνα 5.42 : Η εισαγωγή στη Βάση Δεδομένων.

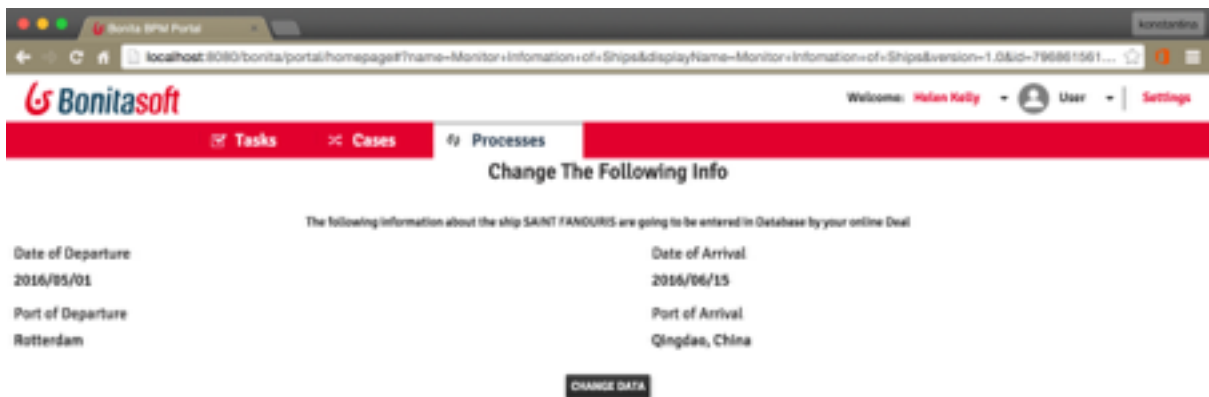
The screenshot shows the Bonitasoft BPM Portal interface. At the top, there is a navigation bar with 'Tasks', 'Cases', and 'Processes' tabs. Below this, a red banner reads 'Select the Ship that has reached at its destination'. Underneath, a message says 'Choose from the list below the ship that have arrived to its destination'. A table lists 19 ships with columns for ID, Ship Name, Start Date, End Date, Origin, and Destination. At the bottom of the table is a 'SUBMIT' button.

ID	Ship Name	Start Date	End Date	Origin	Destination
2	SAINT FANOURIS	2016-03-01	2016-04-15	Los Angeles, U.S.A.	Port Kelang, Malaysia
3	MARITSA NLP	2016-04-20	2016-07-01	Rotterdam	Qingdao, China
4	SAINT GEORGE	2016-07-15	2016-07-30	Ningbo-Zhoushan, China	Rotterdam
5	PALAMA	2016-05-02	2016-06-15	ebel Ali, Dubai, United Arab Emirates	Keihin ports*, Japan
6	IOANNIS F.K	2016-02-05	2016-10-30	Singapore	Tanjung Pelepas, Malaysia
7	KAVAFIS	2016-03-15	2016-10-30	Hong Kong, S.A.R., China	Tanjung Priok, Jakarta, Indonesia
8	ELYTIS	2016-10-07	2016-12-15	Xiamen, China	Laem Chabang, Thailand
9	SEFERIS	2016-04-08	2016-08-05	Ningbo-Zhoushan, China	Ho Chi Minh, Vietnam
10	BFP GALAXY	2016-02-05	2016-10-30	Tanjung Pelepas, Malaysia	Metro Vancouver, Canada
11	BFP MELODY	2016-05-30	2016-05-20	Ningbo-Zhoushan, China	Durban, South Africa
12	CONTSHIP ACE				
13	CONTSHIP BOX	2016-03-15	2016-10-30	Tanjung Pelepas, Malaysia	Ambarli, Turkey
14	CONTSHIP CLIP				
15	CONTSHIP DAY	2017-01-30	2016-05-20	Qingdao, China	Gioia Tauro, Italy
16	CONTSHIP ECO	2016-02-05	2016-10-30	Busan, South Korea	Tanjung Perak, Surabaya, Indonesia
17	DP LEGACY	2016-03-15	2017-10-30	Busan, South Korea	Singapore
18	TONGAN II	2016-03-30	2017-02-05	Tanjung Pelepas, Malaysia	Busan, South Korea
19	SAINT NEKTARIS	2016-04-06	2016-05-18	Shanghai, China	Qingdao, China

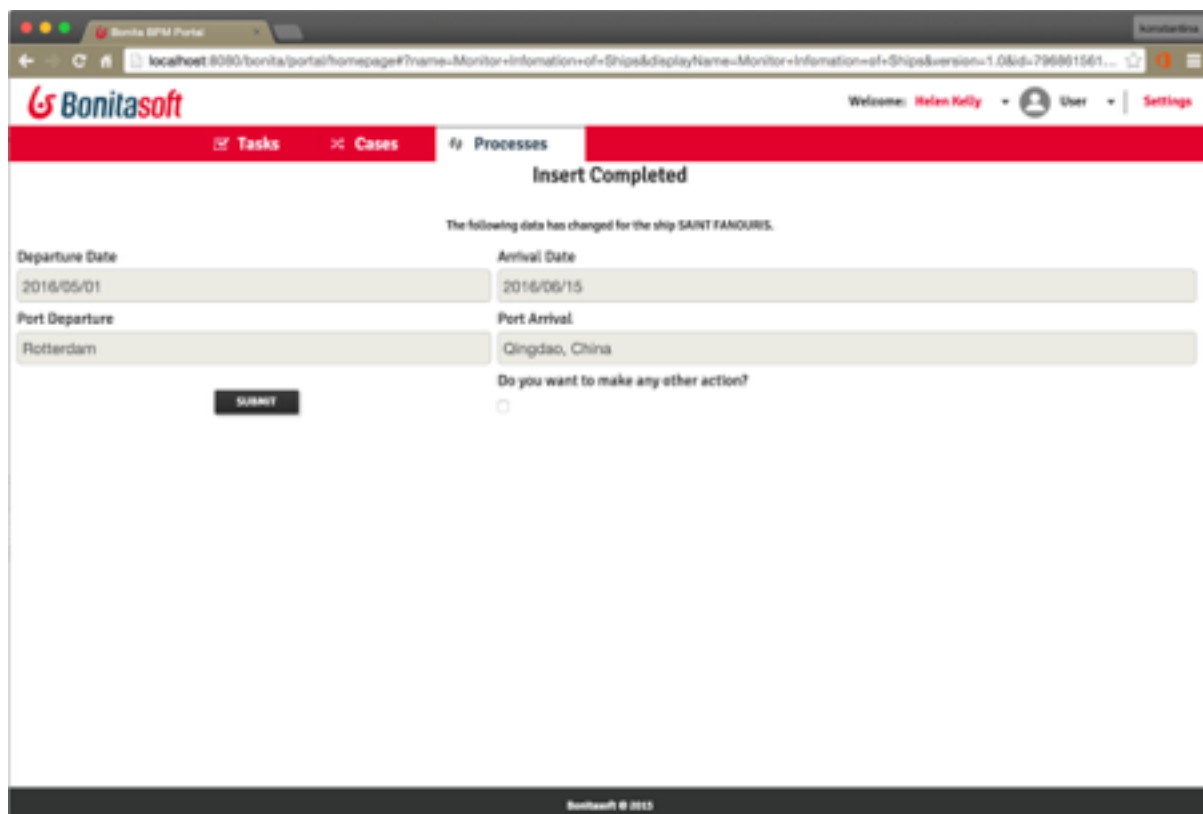
Εικόνα 5.43 : Ο χρήστης Walter Bates επιλέγει ποιο πλοίο θέλει να αφαιρέσει από το πρόγραμμα.



Εικόνα 5.44 : Ο χρήστης Walter Bates έχει με επιτυχία διαγράψει την εγγραφή από τη Βάση Δεδομένων.



Εικόνα 5.45 : Ο χρήστης Helen Kelly έχει με επιτυχία ολοκληρώσει τη διαπραγμάτευση και επιλέγει να αλλάξει τα στοιχεία ενός πλοίου από τη συμφωνία που επιτεύχθηκε μέσω της εφαρμογής.



Εικόνα 5.46 : Ο χρήστης Helen Kelly έχει με επιτυχία ολοκληρώσει την εισαγωγή του νέου ταξιδιού του πλοίου στο πρόγραμμα πλοίων.

18	18	2016-03-30	2017-02-05	Tanjung Pelepas, Malaysia	Busan, South Korea	Pacific & Atlantic
20	1	2016-01-01	2016-01-02	Shanghai, China	Shanghai, China	Pacific & Atlantic
21	2	2016-05-01	2016-06-15	Rotterdam	Qingdao, China	Pacific & Atlantic

Εικόνα 5.47 : Η επιτυχής εισαγωγή των δεδομένων απεικονίζεται και στο πίνακα της Βάσης Δεδομένων.

7. Συμπεράσματα

Ολοκληρώνοντας τη διπλωματική εργασία, αξίζει να τονισθούν και να επισημανθούν τα βασικά συμπεράσματα της ανάπτυξης του εν λόγω Πληροφοριακού Συστήματος. Αντικείμενο της παρούσης εργασίας ήταν η μελέτη και η ανάπτυξη ενός Πληροφοριακού Συστήματος που δύναται να χρησιμοποιηθεί στο κλάδο της Ναυτιλίας.

Παρουσιάστηκε ένα μοντέλο διαδυκτιακής πρακτόρευσης ενός πλοίου, σύμφωνα με το οποίο μέσα από τον Παγκόσμιο Ιστό όλες οι Εταιρείες Ναυλώσεων και οι Ναυτηλιακές Εταιρείες έχουν τη δυνατότητα να ελέγχουν και να εντοπίζουν ανά πάσα στιγμή διαθέσιμα πλοία για τις διάφορες επιθυμητές μεταφορές που επιδιώκουν να πραγματοποιήσουν. Η ανάπτυξη του μοντέλου έγινε σύμφωνα με τα πιο βασικά συστήματα και λογισμικά που μπορεί να διαθέτει μια Εταιρεία, όπως είναι συστήματα ERP και Business Intelligence. Όλες αυτές οι τεχνολογίες παρουσιάζονται με εκτενή τρόπο, τονίζοντας τα οφέλη και τα πλεονεκτήματα που αποκομίζει μια Εταιρεία στο Ναυτιλιακό χώρο. Τέλος παρουσιάστηκε μία περίπτωση χρήσης ανάμεσα σε ένα Ναυλωτή που επιθυμεί την εύρεση ενός πλοίου και τη διαδικασία διαπραγμάτευσης και επίτευξης μιας συμφωνίας.

Γενικότερα, θα μπορούσε κανείς να σημειώσει ότι όλες οι επενδύσεις στην τεχνολογία πληροφοριών, είτε είναι έργα συστημάτων είτε υποδομής, παράγουν αξία για τις επιχειρήσεις κατά δύο κυρίως τρόπους. Η πιο προφανής συνεισφορά στην αξία είναι μέσω βελτιώσεων στις υφιστάμενες επιχειρηματικές διεργασίες, ή με τη δημιουργία εντελώς νέων επιχειρηματικών διεργασιών, καθαρό αποτέλεσμα των οποίων είναι να αυξηθεί η αποδοτικότητα της εργασίας και του κεφαλαίου που χρησιμοποιεί η Εταιρεία. Δεύτερον, τα πληροφοριακά συστήματα συμβάλλουν σε βελτιώσεις στη λήψη διοικητικών αποφάσεων, με αύξηση της ταχύτητας και με βελτίωση της ακρίβειας στη λήψη αποφάσεων. Τόσο οι βελτιώσεις στις επιχειρηματικές διεργασίες όσο και εκείνες στη λήψη διοικητικών αποφάσεων μπορούν να μετρηθούν με τη χρήση παραδοσιακών μεθόδων κατάρτισης προϋπολογισμού κεφαλαιουχικών επενδύσεων.

Τα πληροφοριακά συστήματα μπορούν επίσης να δημιουργήσουν αξία επειδή ενισχύουν στρατηγικά την Εταιρεία. Η αξία για την Εταιρεία μπορεί να μην είναι η άμεση αύξηση του ποσοστού απόδοσης της επένδυσης, αλλά, αντιθέτως, μια πιο μακροπρόθεσμη απόδοση που απορρέει από την καλύτερη στρατηγική θέση στον κλάδο. Τα συστήματα μπορούν να βοηθήσουν μία Εταιρεία να αποκτήσει διαφορετικά και διατηρήσιμα πλεονεκτήματα, για παράδειγμα ενισχύοντας τους δεσμούς της με πελάτες, διαφοροποιώντας υπηρεσίες, και αυξάνοντας την ευελιξία και την προσαρμοστικότητα σε πιο μακρές περιόδους.

Είναι επίσης σημαντικό να έχουμε κατά νου ότι τα συστήματα μπορούν να έχουν αξία, αλλά μπορεί η Εταιρεία να μην αποκομίζει το σύνολο ή ακόμη και μέρος της αξίας. Αν και τα έργα συστημάτων μπορεί να οδηγήσουν σε οφέλη για την Εταιρεία, όπως κερδοφορία και

παραγωγικότητα, ορισμένα ή όλα από τα οφέλη μπορεί να πηγαίνουν απευθείας στον άμεσα ενδιαφερόμενο Εταιρεία-καταναλωτή με τη μορφή χαμηλότερων τιμών ή πιο αξιόπιστων υπηρεσιών και προϊόντων. Η κοινωνία μπορεί να επιβραβεύει τις Εταιρείες που αυξάνουν το καταναλωτικό πλεόνασμα επιτρέποντάς τους να επιβιώσουν ή ανταμείβοντάς τις με αυξήσεις των εταιρικών εσόδων τους. Αλλά από διοικητική άποψη, η πρόκληση για την Εταιρεία είναι να διατηρήσει όσο μεγαλύτερο μέρος μπορεί υπό τις τρέχουσες συνθήκες της αγοράς από το όφελος που προκύπτει από τις επενδύσεις σε συστήματα.

Η αξία των συστημάτων από χρηματοοικονομική οπτική γωνία περιστρέφεται ουσιαστικά γύρω από το ζήτημα της απόδοσης του επενδυμένου κεφαλαίου. Παράγει μια συγκεκριμένη επένδυση σε πληροφοριακά συστήματα απόδοση ικανή για να δικαιολογήσει το κόστος της; Υπάρχουν πολλά προβλήματα με αυτή την προσέγγιση, και ένα από τα μεγαλύτερα είναι πώς να εκτιμηθούν τα οφέλη και να μετρηθεί το κόστος.

Τέλος, υιοθέτηση νέων τεχνολογιών οδηγεί και στην αναδιοργάνωση πολλών διεργασιών μιας Εταιρεία. Πολλά έργα επιχειρησιακών εφαρμογών και ανασχεδιασμού διεργασιών έχουν υπονομευτεί από κακές πρακτικές υλοποίησης και διαχείρισης αλλαγής, οι οποίες δεν αντιμετώπισαν τις ανησυχίες των εργαζομένων σχετικά με την αλλαγή. Ο μεγαλύτερος κίνδυνος για τον ανασχεδιασμό δεν προέρχεται από τις δυσκολίες της εταιρείας να οραματιστεί και να σχεδιάσει επαναστατικές αλλαγές στις επιχειρηματικές διεργασίες της, αλλά από την αντιμετώπιση του φόβου και της αγωνίας σε ολόκληρο τον οργανισμό, το ξεπέραςμα της αντίστασης στελεχών σε καίριες θέσεις, την αλλαγή θέσεων εργασίας, σταδιοδρομιών και μεθόδων πρόσληψης και την εκπαίδευση. Όλες οι επιχειρησιακές εφαρμογές απαιτούν στενότερο συντονισμό μεταξύ των διάφορων λειτουργικών ομάδων, καθώς και εκτεταμένες αλλαγές επιχειρηματικών διεργασιών.

8. Βιβλιογραφία

- [1] “Ανάπτυξη Προηγμένων Πληροφοριακών Συστημάτων: Μεθολογίες & Εργαλεία”, David Avison & Guy Fitzgerald
- [2] “Σχεδιασμός & Υλοποίηση Ολοκληρωμένου Ηλεκτρονικού Φακέλου Υγείας Ασθενούς Για Κατανεμημένα Ιατρικά Πληροφοριακά Συστήματα”, Αλεξάνδρου α. Χριστοδουλάκη, Εμμανουήλ Α. Χριστοδουλάκη, Αθήνα, Μάιος 2013
- [3] “Σχεδιασμός Βάσεων Δεδομένων”, Γεώργιος Βασιλακόπουλος, Πειραιάς 2009
- [4] “Σχεσιακές Βάσεις Δεδομένων- Θεωρία και Εφαρμογές”, Αθανάσιος Ι. Μάργαρης
- [5] “Υπηρεσίες Παγκόσμιου Ιστού και Υπηρεσιοστρεφείς Αρχιτεκτονικές”, Μαρίνος Γ. Θεμιστοκλέους, Βασιλική Γ. Μαντζάνα
- [6] “Η UML στην ανάπτυξη ενσωματωμένων συστημάτων”, Αρετάκη Αικατερίνη, Πάτρα, Ιούνιος 2009