

**ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ**

**ΤΜΗΜΑ ΔΙΔΑΚΤΙΚΗΣ ΤΗΣ ΤΕΧΝΟΛΟΓΙΑΣ  
ΚΑΙ ΨΗΦΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ**

**ΜΕΤΑΠΤΥΧΙΑΚΟ ΠΡΟΓΡΑΜΜΑ ΣΠΟΥΔΩΝ**

**ΚΑΤΕΥΘΥΝΣΗ “ΔΙΚΤΥΟΚΕΝΤΡΙΚΑ ΣΥΣΤΗΜΑΤΑ”**



**Πτυχιακή Εργασία**

**Ανάπτυξη Συστήματος βασισμένο  
σε Ροές Εργασίας**

Επιβλέπων: Δρ. Γεώργιος Βασιλακόπουλος

**Μεταπτυχιακός Φοιτητής: Μίαρης Αλέξανδρος ΜΕ/0468**

Πειραιάς 2008

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΡΔΙΑ

*Αφιερώνεται στους γονείς μου  
για τη συμπαράστασή τους.*

## Περίληψη

Στην εργασία αυτή αναλύονται οι κύριες έννοιες ενός συστήματος ροής εργασίας. Αναπτύσσεται η έννοια της Ροής εργασίας καθώς και οι οντότητες των διαδικασιών και των δραστηριοτήτων που συνάγουν στη δημιουργία ενός τέτοιου συστήματος.

Εν συνεχεία, θα αναλυθεί η έννοια του συστήματος ελέγχου Ροής Εργασίας. Θα γίνει αναφορά στους τύπους ροών εργασίας καθώς θα αναλυθούν οι έννοιες και οι πολιτικές ασφαλείας που διέπουν τέτοιου είδους συστήματα.

Για να μπορέσει να εκτιμηθεί όλο το θεωρητικό υπόβαθρο, θα παρουσιαστούν τα υποστηρικτικά συστήματα ροής εργασίας ενός εμπορικού συστήματος διαχείρισης στοιχείων χρηματοπιστωτικών οφειλών. Το όλο σύστημα στηρίζεται στα συστήματα ροής εργασίας και θα γίνει τμηματική αναφορά στα στοιχεία του.

Το τελευταίο κομμάτι της εργασίας αυτής σχετίζεται με το πρότυπο έγκρισης Ροής Εργασίας .

## Ευχαριστίες

Θερμές ευχαριστίες εκφράζω στον Καθηγητή κο Γεώργιο Βασιλακόπουλο για την επίβλεψη και τη βοήθεια που μου παρείχε για την ολοκλήρωση της διπλωματικής μου. Ευχαριστίες επίσης οφείλονται και στους συνεξεταστές Λέκτορα κο. Παραβάντη Ιωάννη και Αναπληρωτή Καθηγητή κο. Δεμέστιχα Παναγιώτη για την βοήθεια και την υποστήριξή τους.

Ιδιαίτερες ευχαριστίες οφείλονται στον κο. Παναγιώτου Στέφανο για την αμέριστη βοήθειά του , για τις συμβουλές και την καθοδήγησή του τόσο σε επίπεδο προσωπικό όσο και επαγγελματικό. Όλες του οι συμβουλές με οδήγησαν στην άρτια αντιμετώπιση ενός τέτοιου θέματος που ενασχολήθηκα στην εργασία αυτή.

## Περιεχόμενα

|  |    |
|--|----|
| Περίληψη.....  | 3  |
| Ευχαριστίες.....   | 4  |
| Περιεχόμενα.....   | 5  |
| Κατάλογος Εικόνων.....   | 7  |
| Κεφάλαιο 1.....  | 8  |
| Κεφάλαιο 1.....  | 8  |
| 1.1    Εισαγωγή.....   | 8  |
| 1.1.1    Ροή εργασίας.....   | 8  |
| 1.1.2    Ασφάλεια.....   | 9  |
| 1.2    Προβληματισμός.....   | 10 |
| 1.3    Προτεινόμενο Μοντέλο.....                                       | 10 |
| Κεφάλαιο 2 – Έννοιες Ροής Εργασίας.....                                | 12 |
| 2.1    Εισαγωγή.....   | 12 |
| 2.2    Τι είναι Ροή Εργασίας;.....                                     | 12 |
| 2.2.1    Επιχειρησιακή διαδικασία.....                                 | 14 |
| 2.2.2    Διαδικασίες.....  | 14 |
| 2.2.3    Οντότητες Διαδικασιών.....                                    | 15 |
| 2.2.4    Οντότητες Δραστηριοτήτων.....                                 | 15 |
| 2.3    Workflow Management System (WFMS).....                          | 16 |
| 2.3.1    Built-time Λειτουργίες.....                                   | 16 |
| 2.3.2    Λειτουργίες ελέγχου run-time.....                             | 17 |
| 2.3.3    Αλληλεπιδράσεις σε διαδικασίες run-time.....                  | 18 |
| 2.4    Τύποι Ροών Εργασίας.....  | 19 |
| 2.4.1    Παραγωγικές Ροές εργασίας.....                                | 19 |
| 2.4.2    Διοικητικές Ροές εργασίας.....                                | 19 |
| 2.4.3    Συνεργατικές Ροές εργασίας.....                               | 19 |
| 2.4.4    Ειδικές Ροές εργασίας.....                                    | 19 |
| 2.5    Διοικητικός συνασπισμός Ροής εργασίας.....                      | 20 |
| 2.6    Συμπεράσματα.....   | 21 |
| Κεφάλαιο 3 - Επισκόπηση των Εννοιών Ασφάλειας.....                     | 22 |
| 3.1    Εισαγωγή.....   | 22 |
| 3.2    Οι 5 στυλοβάτες της ασφάλειας πληροφοριών.....                  | 23 |
| 3.2.1    Προσδιορισμός και Επικύρωση.....                              | 23 |
| 3.2.2    Έλεγχος πρόσβασης (έγκριση).....                              | 23 |
| 3.2.3    Εμπιστευτικότητα.....   | 24 |
| 3.2.4    Ακεραιότητα.....  | 24 |
| 3.2.5    Non-repudiation (μη-άρνηση).....                              | 25 |
| 3.3    Προσεγγίσεις ελέγχου πρόσβασης για τις πολιτικές ασφάλειας..... | 26 |
| 3.3.1    Διακριτικός Έλεγχος Πρόσβασης (DAC).....                      | 26 |
| 3.3.2    Υποχρεωτικός Έλεγχος Πρόσβασης (MAC).....                     | 27 |
| 3.3.3    Ρολοκεντρικός Έλεγχος Πρόσβασης (RBAC).....                   | 31 |
| 3.4    Ασφάλεια στις Ροές Εργασίας.....                                | 31 |
| 3.4.1    Τρία επίπεδα ασφαλείας WfMS.....                              | 33 |
| 3.4.2    Διευκρίνιση επιπέδων εφαρμογών πολιτικών ασφάλειας.....       | 33 |
| 3.5    Συμπέρασμα.....   | 35 |
| Κεφάλαιο 4 – Τρέχων Παράδειγμα.....                                    | 37 |

|   |     |
|---|-----|
| Εισαγωγή .....  | 37  |
| 4.1 Περιγραφή Συστήματος.....   | 37  |
| 4.2 Βασική δομή Συστήματος .....  | 37  |
| 4.2.1 Εισαγωγή Στοιχείων – Loading Workflow .....                       | 37  |
| 4.2.2 Σύστημα Αυτόματης Διαχείρισης Κινήσεων – Production Workflow..... | 40  |
| 4.2.3 Σύστημα Διαχείρισης Κινήσεων Λογαριασμών (State Machine) .....    | 43  |
| 4.2.4 Σύστημα Εξαγωγής Αναφορών .....                                   | 46  |
| 4.3 Υποστηρίζουσα Βάση Συστήματος.....                                  | 48  |
| 4.3.1 Πίνακας Customers .....   | 48  |
| 4.3.2 Πίνακας Packs .....   | 49  |
| 4.3.3 Πίνακας Client.....   | 49  |
| 4.3.4 Πίνακας Cases .....   | 53  |
| 4.3.5 Πίνακας Locations.....  | 55  |
| 4.3.6 Πίνακας Contact .....   | 55  |
| 4.3.7 Πίνακας Payments.....   | 57  |
| Κεφάλαιο 5 – Role-based Access Control (RBAC).....                      | 60  |
| 5.1 Εισαγωγή .....  | 60  |
| 5.2 Ορισμός Ρόλων.....  | 60  |
| 5.3 Σύνοψη του RBAC .....   | 60  |
| 5.4 Τέσσερα εννοιολογικά πρότυπα RBAC .....                             | 61  |
| 5.4.1 Βασικό μοντέλο .....  | 61  |
| 5.4.2 Ιεραρχίες ρόλων .....   | 62  |
| 5.4.3 Περιορισμοί.....  | 63  |
| 5.4.4 Παγιωμένο πρότυπο .....   | 64  |
| 5.5 Σύνοψη .....  | 64  |
| Κεφάλαιο 6 - Πρότυπο Έγκρισης Ροής Εργασίας (WAM).....                  | 66  |
| 6.1 Εισαγωγή .....  | 66  |
| 6.2 Το πρότυπο έγκρισης (AT).....                                       | 66  |
| 6.3 Χωρισμός Περιορισμών Καθηκόντων επιβαλλόμενοι από το WAM... ..      | 68  |
| 6.4 Χορήγηση και Ανάκληση εγκρίσεων .....                               | 69  |
| 6.5 Συμπέρασμα .....  | 70  |
| Παράρτημα .....   | 72  |
| 1. Κώδικας Loading Workflow .....                                       | 72  |
| 2. Κώδικας Production Workflow .....                                    | 98  |
| 3. Κώδικας StateMachine Workflow .....                                  | 117 |
| 4. Κώδικας Workflow Αναφορών .....                                      | 127 |
| 5. Σκαρίφμα Διαχείρισης Συνεργασίας .....                               | 134 |
| 6. Σχεσιακό Σχήμα Βάσης.....  | 135 |
| Βιβλιογραφία.....   | 136 |

## Κατάλογος Εικόνων

|  |    |
|--|----|
| Εικόνα 1 - Γενική Ορολογία.....                            | 11 |
| Εικόνα 2 - Ορολογία Workflow .....                         | 13 |
| Εικόνα 3 - Σημείωση καθορισμού διαδικασίας.....            | 17 |
| Εικόνα 4 - Τρόπος Λειτουργίας Μοντέλου Bell Lapadula ..... | 30 |
| Εικόνα 5 - Loading Workflow.....                           | 38 |
| Εικόνα 6 - Επιλογή Δημιουργίας Πακέτων.....                | 39 |
| Εικόνα 7 - Production Workflow .....                       | 41 |
| Εικόνα 8 - State Machine Workflow .....                    | 45 |
| Εικόνα 9 Επιλογή ημερημνίας.....                           | 46 |
| Εικόνα 10 - Workflow Αναφορών .....                        | 47 |
| Εικόνα 11 Μορφή Αρχείου Αναφοράς .....                     | 48 |
| Εικόνα 12 - Πίνακας Customers.....                         | 49 |
| Εικόνα 13 - Πίνακας Packs .....                            | 49 |
| Εικόνα 14 - Πίνακας Client.....                            | 50 |
| Εικόνα 15 - Πίνακας Client_Addresses .....                 | 50 |
| Εικόνα 16 - Πίνακας Client_Phones .....                    | 51 |
| Εικόνα 17 - Πίνακας Client_EX_Data.....                    | 51 |
| Εικόνα 18 - Πίνακας Sec_Person.....                        | 52 |
| Εικόνα 19 - Πίνακας Sec_Person_Addresses.....              | 52 |
| Εικόνα 20 - Πίνακας Sec_Person_Phones .....                | 53 |
| Εικόνα 21 - Πίνακας Sec_Person_Ex_Data.....                | 53 |
| Εικόνα 22 - Πίνακας Cases .....                            | 54 |
| Εικόνα 23 - Πίνακας Case_Ex_Data.....                      | 55 |
| Εικόνα 24 - Πίνακας Locations.....                         | 55 |
| Εικόνα 25 - Πίνακας Contact .....                          | 56 |
| Εικόνα 26 - Πίνακας Payments .....                         | 57 |
| Εικόνα 27 - Πίνακας Installments .....                     | 58 |
| Εικόνα 28 - Πίνακας Costs.....                             | 58 |
| Εικόνα 29 - Πίνακας Tolls .....                            | 59 |

## Κεφάλαιο 1

### 1.1 Εισαγωγή

Οι επιχειρήσεις τείνουν να γίνουν πιο ανταγωνιστικές με την καλύτερη χρησιμοποίηση του χρόνου τους. Ο χρόνος είναι η αιτία που πολλές άλλες εταιρείες έχουν αναδιοργανώσει τις εσωτερικές διαδικασίες τους, καθώς έτσι θα μπορούν να λάβουν καλύτερα αποτελέσματα. Η επιδιόρθωση των επιχειρηματικών διαδικασιών (Business process re-engineering - BPR ) είναι υποχρεωτική αν μια επιχείρηση επιθυμεί να παραμείνει ενεργή και ανταγωνιστική.

Η εξάπλωση του διαδικτύου και οι κάθε λογής ψηφιακές ευκολίες, έκαναν εύκολη την ανάπτυξη μικρών και μεγάλων συναφών επιχειρήσεων σε κάθε τομέα. Τα επιτεύγματα στον τεχνολογικό τομέα, έκαναν εύκολη υπόθεση τον τρόπο με τον οποίο αναπτύσσονται οι κάθε λογής επιχειρήσεις.

Οι πληροφορίες και οι αξιοποίησή τους, είναι το άλλο κλειδί στην επιτυχία των επιχειρήσεων. Μέσω των πληροφοριών αυτών και του τρόπου που επεξεργάζονται, δίνονται οι λύσεις για τα προβλήματα που παρουσιάζονται. Η διαχείριση κάθε σωστά ανεπτυγμένης επιχείρησης μπορεί να εξάγει πληροφορίες από τα δεδομένα , μόνο μέσω της σωστής εκμετάλλευσης των πληροφοριών που τους παρέχονται.

Ο χρόνος και οι πληροφορίες είναι οι σημαντικότεροι παράγοντες για την επιτυχία μιας σημερινής επιχείρησης. Το δεδομένο αυτό έκανε σαφή την ανάγκη των συστημάτων ροών εργασίας μέσα στις μεγάλες επιχειρήσεις.

#### 1.1.1 Ροή εργασίας

Οι ροές εργασίας είναι αυτοματισμοί διαδικασιών. Η ύπαρξή τους οφείλετε στην ανάγκη να γίνουν οι διαδικασίες πιο αποδοτικές. Ακόμα, υποστηρίζουν και την ασφάλεια των δεδομένων και των πληροφοριών , που υπάρχει στο σύστημά τους.

Μια ροή εργασίας μπορεί να οριστεί σαν μια σειρά από διαφορετικά έργα μέσα σε μια διαδικασία, που διαδραματίζονται για να επιτευχθεί ένας απώτερος σκοπός [WFMC99]. Είναι η διαδικασία μιας ροής εργασίας διοίκησης συστημάτων (WfMS), για το συντονισμό διαφορετικών έργων μέσα σε μια διαδικασία. Μέσω των διαδικασιών αυτών, δημιουργούνται και εξαρτήσεις μεταξύ των διαφόρων έργων. Υπάρχουν δύο κύρια είδη εξαρτήσεων, οι στατικές και οι δυναμικές. Οι στατικές εξαρτήσεις ορίζονται από την αρχή και παραμένουν σταθερές κατά τη διαδικασία του έργου. Οι δυναμικές ορίζονται κατά τη διαδικασία του έργου, όπου είναι αναγκαίες και αλλάζουν από παράγοντες που δεν μπορούν να οριστούν εξ αρχής.



### 1.1.2 Ασφάλεια

Η αξία και η σημασία της πληροφορίας είναι τόσο μεγάλη, που πολλές φορές η χρησιμοποίηση αυτής είναι που κάνει τη διαφορά μεταξύ των αντιπάλων. Η διασφάλιση των πληροφοριών περιλαμβάνει και τα έξι βασικά στοιχεία ασφάλειας, την εμπιστευτικότητα, την ακεραιότητα, τη διαθεσιμότητα, η ωφέλεια, η αυθεντικότητα και η κτήση.

Ανεξάρτητα από το πόσο αποδοτική είναι μια αυτοματοποίηση ροής εργασίας, εάν δεν είναι ασφαλές είναι πιο καταστρεπτικό από την αξία του. Με άλλα λόγια, η ασφάλεια πληροφοριών δεν μπορεί να συμβιβαστεί σε μια προσπάθεια να προωθηθεί η αποδοτικότητα των αυτοματοποιήσεων επιχειρησιακής διαδικασίας.

Ο έλεγχος πρόσβασης είναι αρχικής σπουδαιότητας στα περιβάλλοντα ροής εργασίας. Απαιτείται να ελεγχθεί ποιός έχει πρόσβαση σε ποιές πληροφορίες για τους διάφορους στόχους που περιλαμβάνουν τη ροή της εργασίας. Οι εξαρτήσεις στόχου πρέπει να συντονιστούν και η ακεραιότητα αντικειμένου πρέπει να διατηρηθεί σε όλες τις εκτελέσεις διαδικασίας. Η πρόκληση της ασφάλειας ροής εργασίας είναι να εφαρμοστεί ένα πρότυπο έγκρισης που επιτρέπει τη δυναμική φύση της διαδικασίας ροής εργασίας. Συγκεκριμένα, οι ακόλουθες τρεις αρχές πρέπει να επιβληθούν:

- Διαχωρισμός Καθηκόντων: Αυτός ο τύπος περιορισμού απαιτεί ότι ένα άτομο αποκλείεται αμοιβαία από την εκτέλεση ενός συνδυασμού στόχων. Η επιβολή του χωρισμού των περιορισμών καθηκόντων στη ροή εργασίας εξασφαλίζει ότι η σημασιολογική ακεραιότητα των πληροφοριών είναι συνεχής μέσω της πρόληψης της απάτης. Ο χωρισμός των καθηκόντων εφαρμόζεται μέσω της προδιαγραφής των περιορισμών που απεικονίζουν χαρακτηριστικά τους επιχειρησιακούς κανόνες της οργάνωσης.
- Λιγότερο προνομιούχοι: Η επιβολή αυτής της αρχής επιτρέπει σε έναν χρήστη για να λάβει τις λιγότερες πιθανές άδειες που απαιτούνται για να εκτελέσουν έναν στόχο. Είναι βασισμένο στη λογική ότι η παρεμπόδιση των χρηστών από την απόκτηση των περιττών προνομίων μειώνει τον κίνδυνο ότι η ακεραιότητα πληροφοριών μπορεί να συμβιβαστεί κατά τη διάρκεια των εκτελέσεων στόχου.
- Σειρά γεγονότων: Οι εξαρτήσεις στόχου στη ροή της δουλειάς δίνουνε αφορμή για την ανάγκη για μια συγκεκριμένη διαταγή των εκτελέσεων στόχου. Επιπλέον, η δυναμική φύση των ροών εργασίας σημαίνει ότι αυτή η διαταγή των εκτελέσεων στόχου δεν μπορεί πάντα να προκαθοριστεί. Είναι σημαντικό για την ισχύ των εγκρίσεων να κρατηθεί σύμφωνος με τις εκτελέσεις των στόχων. Αυτό καλείται συγχρονισμός της ροής έγκρισης με τη ροή εργασίας.

Η προδιαγραφή περιορισμού είναι θεμελιώδης στην επιβολή των προαναφερθεισών αρχών. Οι περιορισμοί είναι πρώτιστα δύο τύπων: στατικός ή δυναμικός [BFA99]. Οι στατικοί περιορισμοί καθορίζονται πριν από την εκτέλεση διαδικασίας, και δεν αλλάζουν κατά τη διάρκεια της πορείας της

εκτέλεσης διαδικασίας. Δυναμικοί περιορισμοί, εντούτοις, καθορίζονται σύμφωνα με την ιστορία της εκτέλεσης διαδικασίας.

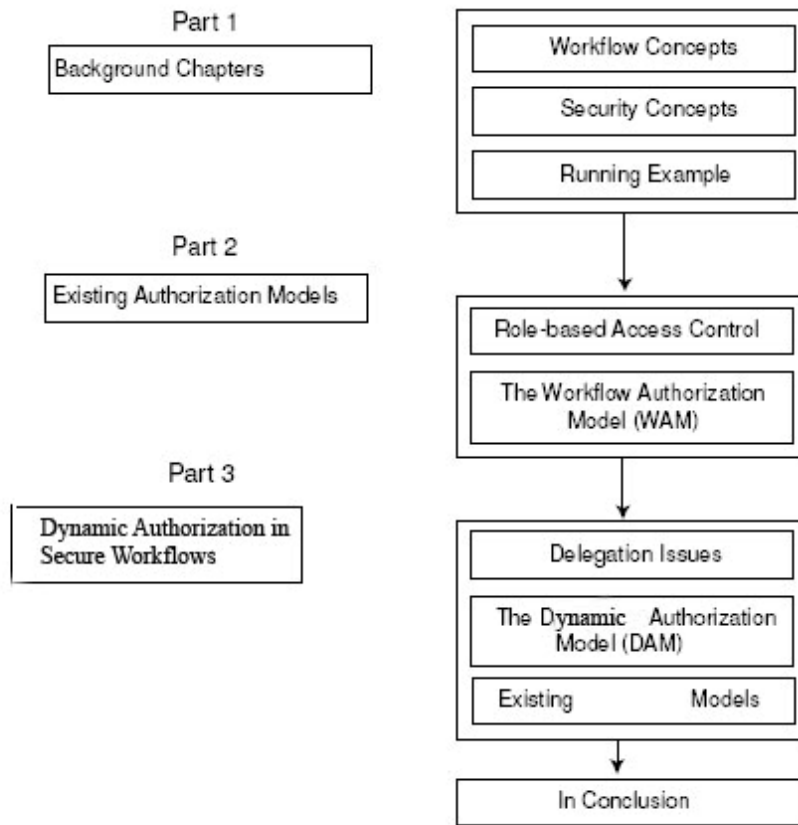
Η σημασία της δυναμικής προδιαγραφής περιορισμού στα περιβάλλοντα ροής εργασίας δεν μπορεί να μειωθεί. Οι βασισμένοι σε ρόλους μηχανισμοί ελέγχου πρόσβασης υπολείπονται της επιβολής των δυναμικών περιορισμών. Συνεπώς, η ανάγκη προέκυψε για ένα ροοεργασιακό πρότυπο έγκρισης να αναπτυχθεί. Οι Atluri και Huang δημιούργησαν το Μοντέλο Επικύρωσης Ροής Εργασίας (Workflow Authentication Model - WAM) [AH96b, HA99], το οποίο εξετάζει τη χορήγηση και την ανάκληση των εγκρίσεων στο WfMS. Το WAM ικανοποιεί τις περισσότερες απαιτήσεις για ένα ιδεατό μοντέλο ελέγχου. Αυτές είναι η επιβολή του χωρισμού των καθηκόντων, ο χειρισμός των χρονικών περιορισμών, μια βασισμένη σε ρόλους εφαρμογή και ο συγχρονισμός της ροής εργασίας με την ροή αυθεντικοποίησης.

## **1.2 Προβληματισμός**

Η ασφάλεια είναι ο πρωταρχικός ρόλος σε κάθε σύστημα. Ο στόχος εδώ είναι να αναπτυχθεί ένα μοντέλο, τέτοιο ώστε να διατηρηθεί η ασφάλεια ενός συστήματος ροής εργασίας. Για να είναι ασφαλές το μοντέλο αυτό, είναι αναγκαίο να είναι σε θέση να ικανοποιήσει τις δυναμικές καταστάσεις.

## **1.3 Προτεινόμενο Μοντέλο**

Στόχος της εργασίας αυτής, είναι η ανάλυση ενός συστήματος που στηρίζεται στα συστήματα ροής εργασίας. Μεγάλο μέρος πολλών πρόσφατων συστημάτων στηρίζονται σε ροές εργασίας, καθώς δίνουν ευελιξία και ταχύτητα στην επικοινωνία συστημικών διεπαφών.



Εικόνα 1 - Γενική Ορολογία

## Κεφάλαιο 2 – Έννοιες Ροής Εργασίας

### 2.1 Εισαγωγή

Ο ρυθμός στον οποίο η επιχείρηση διευθύνεται έχει αυξηθεί εντυπωσιακά στα πρόσφατα έτη. Ένα μεγάλο μέρος αυτού μπορεί να αποδοθεί στο Διαδίκτυο και την εμφάνιση του ηλεκτρονικού εμπορίου. Σε ένα περιβάλλον ηλεκτρονικού εμπορίου, οι μικρές επιχειρήσεις μπορούν να ανταγωνιστούν με τις μεγάλες εταιρίες σε μια καθολική πλατφόρμα μάρκετινγκ. Αυτό έχει συμβάλει σε μια ουσιαστική αύξηση στον ανταγωνισμό αγοράς και έχει αναγκάσει πολλές επιχειρήσεις για να επαναξιολογήσει την αποδοτικότητα των επιχειρησιακών διαδικασιών τους. Η αναδιάρθρωση των οργανωτικών πολιτικών και των μεθόδων για την επιχείρηση έχει κληθεί «Επιδιόρθωση επιχειρηματικών διαδικασιών -BPR» και διάφορα υπάρχοντα εργαλεία σχεδίου συστημάτων βοηθούν σε αυτόν τον στόχο. Αυτές οι καθαρισμένες επιχειρησιακές διαδικασίες είναι αυτοματοποιημένες στις ροές εργασίας που εξασφαλίζουν την ασφάλεια και αποδοτική ροή των πληροφοριών μεταξύ των δραστηριοτήτων και των συμμετεχόντων που αποτελούν την επιχειρησιακή διαδικασία. Κατά συνέπεια, είναι εμφανές ότι η τεχνολογία ροής εργασίας έχει μια μεγάλη επιρροή στις επιχειρησιακές διαδικασίες μιας οργάνωσης.

Οι έννοιες ροής εργασίας δεν είναι νέες. Έχουν καλυφθεί στους διάφορους τύπους τεχνολογιών, εκτός από το BPR, για πολλά χρόνια. Μερικοί από αυτούς περιλαμβάνουν τα εργαλεία επεξεργασίας εικόνας, διοικητική τεχνολογία εγγράφων, εγκαταστάσεις ηλεκτρονικού ταχυδρομείου και συναλλαγές βασισμένες στις εφαρμογές [WFMC95].

Αυτό το κεφάλαιο εισάγει τις έννοιες θεμελιώδεις σε μια κατανόηση των ροών εργασίας. Θα αρχίσει με τον καθορισμό των συστατικών μιας ροής εργασίας και μετά το WfMS, με το οποίο διαχειρίζονται οι ροές εργασίας. Μια συνοπτική περιγραφή των διαφορετικών τύπων ροών εργασίας δίνεται έπειτα. Τέλος, εισάγεται ο διοικητικός συνασπισμός της ροής εργασίας. Η εργασία του συνασπισμού θα χρησιμοποιηθεί εκτενώς για να καθοδηγήσει τη συζήτηση της ροής της δουλειάς.

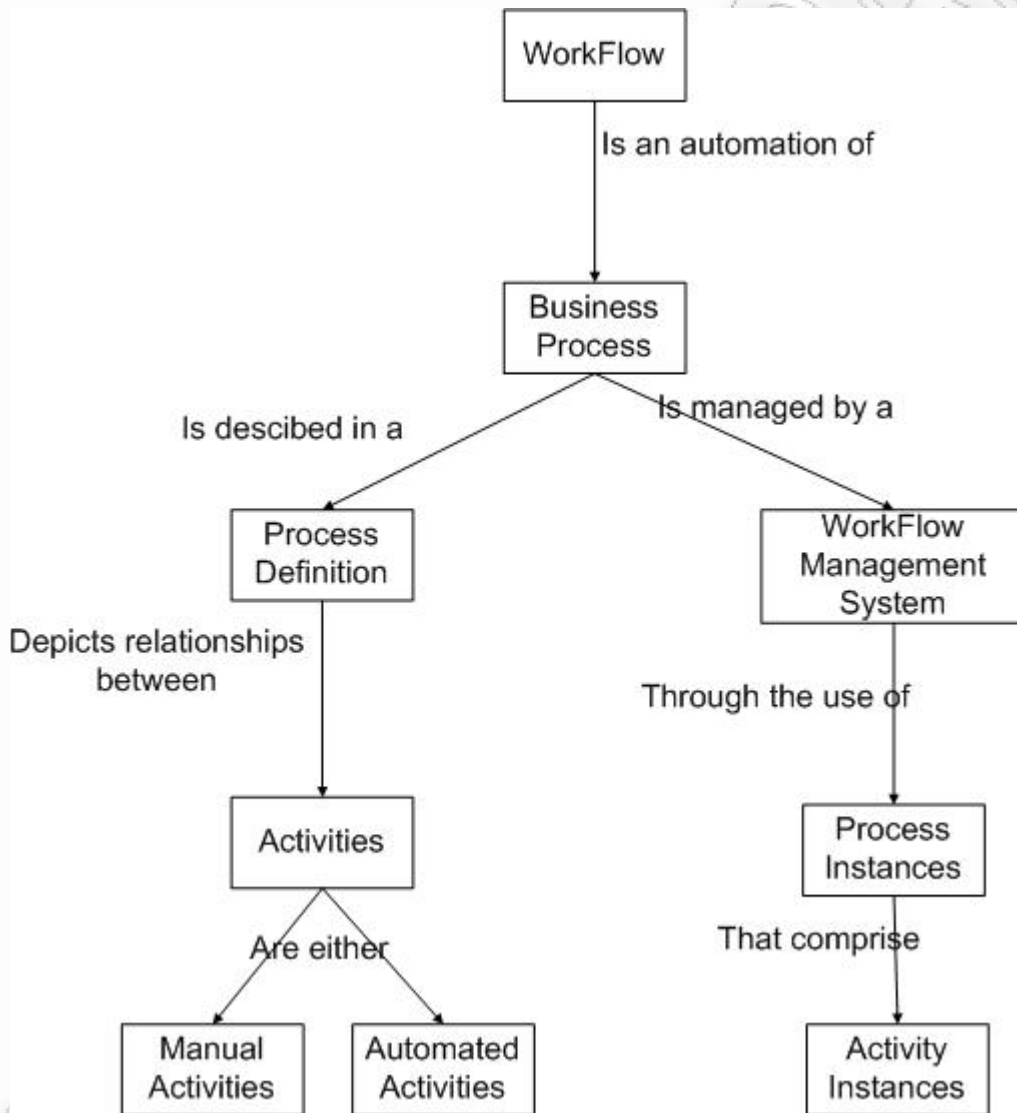
### 2.2 Τι είναι Ροή Εργασίας;

Μια ροή εργασίας είναι «η αυτοματοποίηση μιας επιχειρησιακής διαδικασίας, γενικά ή εν μέρει, κατά τη διάρκεια κάποιων βημάτων, και οι πληροφορίες ή οι στόχοι περνούν από έναν συμμετέχοντα σε άλλον για τη δράση, σύμφωνα με ένα σύνολο διαδικαστικών κανόνων» [WFMC99].

Μερικοί από τους όρους που είναι θεμελιώδεις σε μια κατανόηση των ροών της δουλειάς είναι οι έννοιες μιας επιχειρησιακής διαδικασίας, δραστηριότητα,

περίπτωση διαδικασίας και περίπτωση δραστηριότητας. Ο διοικητικός συνασπισμός ροής εργασίας έχει καθορίσει τυπικά αυτούς και άλλους όρους και πρότυπα σχετικά με τις ροές της δουλειάς [WFMC99].

Το σχήμα 2-1 υιοθετείται από το Workflow Management Coalition [WFMC99]. Παρέχει μια εννοιολογική επισκόπηση για το πώς οι όροι που συζητούνται κατωτέρω αφορούν την έννοια της ροής της δουλειάς.



Εικόνα 2 - Ορολογία Workflow

### 2.2.1 Επιχειρησιακή διαδικασία

Μια επιχειρησιακή διαδικασία απεικονίζει την ένωση των δραστηριοτήτων που, όταν εκτελείται με έναν συστηματικό τρόπο, τελικά επιτυγχάνει κάποιο σκοπό ή στόχο. Οι σχέσεις μεταξύ αυτών των δραστηριοτήτων και των στόχων που επιτυγχάνουν καθορίζονται χαρακτηριστικά χορηγώντας στις οργανωτικές πολιτικές και τις δομές.

Οι απαιτητικές ανάγκες των επιχειρήσεων σήμερα απαιτούν ότι οι επιχειρησιακές διαδικασίες καθαρίζονται για να παραγάγουν τη μέγιστη αποδοτικότητα, έτσι ώστε είναι δυνατό για μια οργάνωση να ανταγωνιστεί στη σφαιρική αγορά. Συνεπώς, πολλές οργανώσεις έχουν υποβληθεί σε BPR, όποιος περιλαμβάνει τη «αξιολόγηση, ανάλυση, διαμόρφωση, καθορισμό και επόμενη λειτουργική εφαρμογή των επιχειρησιακών διαδικασιών πυρήνων μιας οργάνωσης» [WFMC95]. Η εφαρμογή της τεχνολογίας ροής της δουλειάς είναι συχνά μια ιδανική λύση, αν και αυτό δεν ισχύει για όλες τις περιπτώσεις. Είναι επίσης σημαντικό να σημειωθεί ότι οι επιχειρησιακές διαδικασίες δεν χρειάζονται να υποβληθούν BPR που εφαρμόζεται στις ροές εργασίας.

### 2.2.2 Διαδικασίες

Ο διοικητικός συνασπισμός ροής εργασίας [WFMC99] καθορίζει μια δραστηριότητα ως «περιγραφή ενός κομματιού της εργασίας που διαμορφώνει ένα λογικό βήμα μέσα σε μια διαδικασία». Οι δραστηριότητες αντιπροσωπεύουν γενικά τη μικρότερη μονάδα της εργασίας στη ροή εργασίας. Οι δραστηριότητες είναι ταξινομημένες ως χειρωνακτικές ή αυτοματοποιημένες. Και οι δύο τύποι αντιπροσωπεύονται στον καθορισμό διαδικασίας.

Οι χειρωνακτικές δραστηριότητες είναι εκείνες οι δραστηριότητες που δεν είναι, ή δεν μπορούν να αυτοματοποιηθούν. Είναι στόχοι που εκτελούνται από τους ανθρώπους ή αποτελούν τη χειρωνακτική εργασία. Συνεπώς, τέτοιες δραστηριότητες δεν αποτελούν μέρος της ροής της δουλειάς, ακόμα κι αν συμπεριλαμβάνονται στον καθορισμό διαδικασίας, δεδομένου ότι δεν είναι ικανοί της αυτοματοποίησης υπολογιστών. Το παράδειγμα μιας συναλλαγής επεξεργασίας επιταγών χρησιμοποιείται για να επεξηγήσει αυτό. Αυτή η ροή της δουλειάς μπορεί να αποτελέσει τις δραστηριότητες της κατάθεσης μιας επιταγής, η επιταγή που εξαργυρώνεται και που χρεώνεται από τον απολογισμό του συρταριού και τον απολογισμό του καταθέτη που πιστώνονται. Σε αυτή την περίπτωση, μια χειρωνακτική δραστηριότητα μπορεί να είναι ο φυσικός χειρισμός της επιταγής από τον πελάτη στην τράπεζα.

Οι αυτοματοποιημένες δραστηριότητες μπορούν να αυτοματοποιηθούν και να ρυθμιστούν από ένα WFMS. Οι αυτοματοποιημένες δραστηριότητες αναφέρονται συνήθως ως «δραστηριότητες ροής της δουλειάς». Οι όροι «δραστηριότητες» και «στόχοι» θα χρησιμοποιούνται εναλλακτικά σε όλο το υπόλοιπο αυτής της

εργασίας για την αναφορά στις αυτοματοποιημένες δραστηριότητες. Στο ανωτέρω παράδειγμα, οι αυτοματοποιημένες δραστηριότητες θα περιελάμβαναν την επιταγή που εξαργυρώνεται και που χρεώνεται από τον απολογισμό του συρταριού και τον απολογισμό του καταθέτη που πιστώνονται.

Η εκτέλεση των δραστηριοτήτων σε μια διαδικασία ροής της δουλειάς συμβάλλει στην επίτευξη κάποιου στόχου ή επιχειρήσεων. Η διαταγή στην οποία οι δραστηριότητες εκτελούνται είναι σημαντική. Είναι συνήθως η περίπτωση ότι το αποτέλεσμα μιας εκτέλεσης στόχου εντυπωσιακά έχει επιπτώσεις σε μια επόμενη δραστηριότητα, και σε σοβαρότερες περιπτώσεις, η αποτυχία να ολοκληρωθεί μια υποχρέωση μπορεί να αποτρέψει την πιθανή ή επιτυχή εκτέλεση άλλη. Τέτοιοι όροι που επιβάλλονται σε αυτές τις δραστηριότητες καλούνται ως «εξαρτήσεις στόχου». Υπάρχουν πρώτιστα δύο τύποι εξαρτήσεων στόχου: στατικοί ή δυναμικοί.

Οι στατικές εξαρτήσεις στόχου διευκρινίζονται εκ των προτέρων και δεν αλλάζουν κατά τη διάρκεια της εκτέλεσης της ροής της δουλειάς. Παραδείγματος χάριν, ο απολογισμός του καταθέτη δεν μπορεί να πιστωθεί έως ότου έχει εξαργυρωθεί η επιταγή.

Δυναμικές εξαρτήσεις, εντούτοις, αλλαγή σε όλη την εκτέλεση ροής της δουλειάς [AH96a]. Παραδείγματος χάριν, το ποσό που αναγνωρίζεται στον καταθέτη μπορεί να ποικίλει σύμφωνα με τους παράγοντες όπως οι δαπάνες τραπεζών και τα τρέχοντα επιτόκια.

### **2.2.3 Οντότητες Διαδικασιών**

Κάθε εκτέλεση ενός συστήματος ροής της δουλειάς δημιουργεί μια περίπτωση ροής εργασίας. Μια περίπτωση ροής της δουλειάς αντιπροσωπεύει μια ορισμένη περίπτωση της διαδικασίας ροής εργασίας [ORG98]. Οι δυναμικές εξαρτήσεις καθορίζονται συχνά από μια ιδιαίτερη περίπτωση ροής της δουλειάς, όποιος μπορεί μόνο να καθοριστεί κατά τη διάρκεια της εκτέλεσης διαδικασίας. Στο παράδειγμα, μια περίπτωση διαδικασίας θα περιελάμβανε την επεξεργασία μιας ενιαίας επιταγής για έναν ενιαίο πελάτη.

### **2.2.4 Οντότητες Δραστηριοτήτων**

Κάθε διαδικασία μπορεί να περιέχει τις πολλαπλάσιες «περιπτώσεις» μιας δραστηριότητας, οι οποίες καλούνται «περιπτώσεις δραστηριότητας». Για την απλότητα, υποθέστε την απουσία πολλαπλάσιων περιπτώσεων δραστηριότητας μέσα σε μια περίπτωση διαδικασίας. Στο παράδειγμά μας, μια περίπτωση δραστηριότητας θα μπορούσε να περιλάβει την επεξεργασία των πολλαπλάσιων επιταγών για έναν ενιαίο πελάτη.

Τώρα που έχουν συζητηθεί οι βασικές έννοιες της ροής εργασίας, η προσοχή θα δοθεί τώρα στο ρόλο ενός συστήματος διαχείρισης ροής εργασίας που διαχειρίζεται τη ροή εργασίας.

## **2.3 Workflow Management System (WFMS)**

Ο διοικητικός συνασπισμός ροής της δουλειάς [WFMC99] καθορίζει ένα WFMS ως «σύστημα που καθορίζει, δημιουργεί και διαχειρίζεται την εκτέλεση των ροών της δουλειάς μέσω της χρήσης του λογισμικού, τρέχοντας σε μια ή περισσότερες μηχανές ροής εργασίας, όποιος είναι σε θέση να ερμηνεύσει τον καθορισμό διαδικασίας, αλληλεπιδράσει με τους συμμετέχοντες ροής εργασίας και, όπου απαιτείται, επικαλείται τη χρήση των εργαλείων και των εφαρμογών IT». Εν ολίγοις, ένα WFMS είναι αρμόδιο για την αυτοματοποίηση και την εκτέλεση των επιχειρησιακών διαδικασιών ως ροές της δουλειάς. Ακόμα και εντούτοις πολλά συστήματα ροής της δουλειάς χρησιμοποιούν τις διαφορετικές τεχνικές και τις τεχνολογίες εφαρμογής, και λειτουργήστε μέσα στα αντιπαραβαλλόμενα περιβάλλοντα, κάθε WFMS επιδεικνύει τα ακόλουθα κοινά χαρακτηριστικά:

- Built-time Λειτουργίες
- Λειτουργίες ελέγχου run-time
- Αλληλεπιδράσεις σε διαδικασίες run-time

### **2.3.1 Built-time Λειτουργίες**

Αυτές οι λειτουργίες είναι αρμόδιες για τη διαμόρφωση της επιχειρησιακής διαδικασίας σε έναν αυτοματοποιημένο καθορισμό. Αυτό επιτυγχάνεται μέσω της χρήσης της ανάλυσης επιχειρησιακής διαδικασίας, εργαλεία διαμόρφωσης και καθορισμού. Ο προκύπτουν καθορισμός συνηθέστερα αναφέρεται ως καθορισμός διαδικασίας.

#### **2.3.1.1 Καθορισμός διαδικασίας**

Ο καθορισμός διαδικασίας διευκρινίζει τη διαδικασία ροής της δουλειάς με έναν τρόπο που το καθιστά πιθανό να ρυθμιστεί από ένα WFMS. Απεικονίζει τις ενώσεις μεταξύ των δραστηριοτήτων στη ροή της δουλειάς καθώς επίσης και των σχέσεων που τις κυβερνούν. Είναι σημαντικό να σημειωθεί ότι ο καθορισμός διαδικασίας περιλαμβάνει τον καθορισμό ροής της δουλειάς καθώς επίσης και το χειρωνακτικό καθορισμό εκείνων των δραστηριοτήτων που δεν μπορούν να αυτοματοποιηθούν [WFMC95].

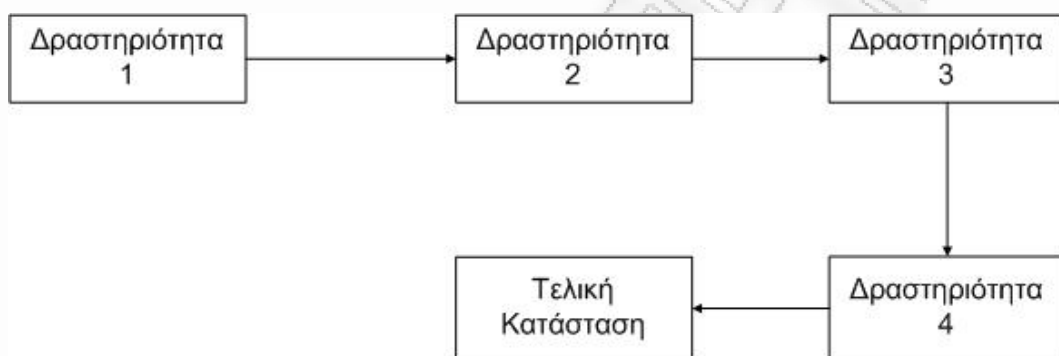
Η σημείωση που χρησιμοποιείται για να απεικονίσει έναν καθορισμό διαδικασίας μπορεί να ποικίλει μεταξύ του κειμένου, προσεγγίσεις γραφικής ή επίσημης γλώσσας. Αυτή η διατριβή θα χρησιμοποιήσει μια παραλλαγή προσβολών της



σημείωσης που υιοθετείται από Olivier και λοιποί. [ORG98], όποιος είναι βασισμένος σε μια έκδοση Petri των δικτύων.

Το σχήμα 3 είναι ένα παράδειγμα ενός καθορισμού διαδικασίας ροής της δουλειάς όπου:

- Τα ορθογώνια αναπαριστούν τις διεργασίες και
- Τα βέλη αντιπροσωπεύουν τις μεταβάσεις που συνδέουν κάθε δραστηριότητα. Οι ροές της δουλειάς μπορούν να αρχίσουν από τις μεταβάσεις, για τις περιπτώσεις όταν μια εξωτερική όψη δραστηριότητας ή διαδικασία ροής της δουλειάς αρχίζει τη ροή της δουλειάς. Μια τελική κρατική μετάβαση εμφανίζεται όταν ολοκληρώνει η προηγούμενη δραστηριότητα.



Εικόνα 3 - Σημείωση καθορισμού διαδικασίας

### 2.3.2 Λειτουργίες ελέγχου run-time

Αυτές οι λειτουργίες είναι αρμόδιες για την ερμηνεία του καθορισμού διαδικασίας και να απαιτήσουν επάνω στους κατάλληλους πόρους συστημάτων τεχνολογίας πληροφοριών (ΤΠ) απαραίτητους για την εκτέλεση διαδικασίας. Επιπλέον, δραστηριότητες ακολουθιών αυτού του λογισμικού σε κάθε μια περίπτωση διαδικασίας που δημιουργεί. Ο όρος «μηχανή» χρησιμοποιείται για να περιγράψει το λογισμικό ελέγχου διεργασίας χρόνου εκτέλεσης που δημιουργεί, διαγράφει και διαχειρίζεται τις διαδικασίες στο περιβάλλον χρόνου εκτέλεσης [WFMC95].

#### 2.3.2.1 Υπηρεσία θέσπισης Ροών Εργασίας

Η υπηρεσία θέσπισης ροής της δουλειάς είναι αρμόδια για τον έλεγχο των περιπτώσεων διαδικασίας με τη χρησιμοποίηση διάφορων μηχανών ροής εργασίας που ερμηνεύουν τον κατάλληλο καθορισμό διαδικασίας ροής εργασίας. Διατηρεί τα εσωτερικά κρατικά στοιχεία που συνδέονται με τις δραστηριότητες και τις διαδικασίες που υποβάλλονται στην εκτέλεση, όπως και το σημείο ελέγχου και τις πληροφορίες αποκατάστασης σε περίπτωση όρου αποτυχίας. Είναι επίσης αρμόδιο για τη διασύνδεση μεταξύ των διανεμημένων χρηστών και των εφαρμογών [WFMC95].

### **2.3.3 Αλληλεπιδράσεις σε διαδικασίες run-time**

Η επιτυχής ολοκλήρωση των δραστηριοτήτων στηρίζεται συνήθως στην αλληλεπίδραση μεταξύ των ανθρώπων και των πόρων ΤΠ που απαιτούνται για έναν ιδιαίτερο στόχο. Οι λειτουργίες αλληλεπίδρασης δραστηριότητας χρόνου εκτέλεσης επιτρέπουν αυτόν τον τύπο επικοινωνίας μεταξύ των ανθρώπινων συμμετεχόντων και του λογισμικού ελέγχου διεργασίας. Αυτό είναι σημαντικό, δεδομένου ότι διευκολύνει τη μεταφορά του ελέγχου μεταξύ των δραστηριοτήτων και επιτρέπει στους ανθρώπους για να αξιολογήσει τη θέση των εκτελέσεων διαδικασίας και δραστηριότητας [WFMC95].

#### **2.3.3.1 Λίστες εργασιών και Διαχειριστής Λίστας εργασιών**

Η λίστα εργασιών περιέχει τα στοιχεία, συνδεδεμένα με έναν συμμετέχοντα ροής εργασίας, ότι οι μηχανές ροής εργασίας κρίνουν απαραίτητα να παρουσιαστούν στην προσοχή ενός χειριστή worklist. Ρυθμίζονται από το σύστημα ροής εργασίας και μπορούν ή δεν να είναι ορατοί στον ανθρώπινο χρήστη, ανάλογα με τον επιθυμητό βαθμό ανθρώπινης επέμβασης. Ο χειριστής λίστας εργασιών είναι ένα κομμάτι του λογισμικού που χρησιμοποιείται για να διαχειριστεί τις αλληλεπιδράσεις δραστηριότητας χρόνου εκτέλεσης μεταξύ της λίστας εργασίας και των χρηστών. Αναφέρεται επίσης και ως διευθυντής στόχου είναι σε θέση με διάφορα συστήματα ροής της δουλειάς [WFMC95].

#### **2.3.3.2 Στοιχεία εφαρμογής ροής εργασίας, σχετικά στοιχεία και στοιχεία ελέγχου**

«Το στοιχείο εφαρμογής ροής εργασίας» είναι ο όρος που δίνεται στο στοιχείο που χειρίζεται άμεσα από τις εφαρμογές ροής εργασίας και δεν προσεγγίζεται από το WFMS. Το σχετικό στοιχείο ροής εργασίας είναι το στοιχείο εφαρμογής ροής εργασίας διαθέσιμο στις μηχανές ροής εργασίας και τις εφαρμογές αυτής. Αυτός ο τύπος ενισχύσεων στοιχείων στις αποφάσεις λειτουργίας ελέγχου επειδή το κράτος του μπορεί άμεσα να έχει επιπτώσεις στην επιλογή των επόμενων δραστηριοτήτων. Το σχετικό στοιχείο ροής εργασίας ενημερώνεται από τις εφαρμογές που επικαλούνται από τις διοικητικές μηχανές ροής εργασίας ή το χειριστή worklist. Αναφέρεται επίσης ως «στοιχεία περίπτωσης» [WFMC95]. Το στοιχείο ελέγχου ροής της εργασίας περιέχει τις πληροφορίες σχετικά με τις καταστάσεις των περιπτώσεων δραστηριότητας και διαδικασίας. Αυτό το στοιχείο δεν είναι προσιτό στις εφαρμογές ροής εργασίας και ρυθμίζεται από τις μηχανές ροής εργασίας και το WFMS [WFMC99].

## **2.4 Τύποι Ροών Εργασίας**

Ο τύπος της ροής εργασίας που θα επιλέξει ένας οργανισμός, εξαρτάται από το επιθυμητό αποτέλεσμα που θέλει να έχει ο οργανισμός αυτός. Ο Allen [All01, pp. 15-38] προτείνει τέσσερα είδη ροών εργασίας: τα παραγωγικά, τα διοικητικά, τα συνεργατικά και τα ειδικά. Πολλά διαφορετικά είδη ροών εργασίας συνήθως χρησιμοποιούνται μέσα σε μεγάλους οργανισμούς.

### **2.4.1 Παραγωγικές Ροές εργασίας**

Οι παραγωγικές ροές εργασίας έχουν ως στόχο την αύξηση της παραγωγικότητας, καθώς μπορούν να αναλαμβάνουν ταυτόχρονα όμοιες διεργασίες. Οι περισσότερες από αυτές τις διεργασίες είναι αυτοματοποιημένες και ο ανθρώπινος προάγοντας δεν μετέχει καθόλου. Αυξημένη ποιότητα και μεγάλη αποτελεσματικότητα επιτυγχάνεται, καθώς συνεχώς επαναλαμβάνονται όμοιες εκτελέσεις. Οι παραγωγικές ροές εργασίας μπορούν εύκολα να εισαχθούν στα υπάρχοντα συστήματα. Υπάρχουν δύο τύποι παραγωγικών ροών εργασίας, τα αυτόνομα και τα ενσωματωμένα.

Τα αυτόνομα συστήματα ροών εργασίας μπορούν να λειτουργήσουν χωρίς την προσθήκη επιπλέον λογισμικού. Τα ενσωματωμένα συστήματα ροών εργασίας λειτουργούν κάτω από ένα άλλο λογισμικό σύστημα.

### **2.4.2 Διοικητικές Ροές εργασίας**

Στοχεύουν περισσότερο στην ευελιξία παρά στην παραγωγικότητα. Συνάγουν στην απλοποίηση της διαδικασίας.

### **2.4.3 Συνεργατικές Ροές εργασίας**

Οι συνεργατικές ροές εργασίας περιλαμβάνουν τη χρήση συνεργαζόμενων ομάδων για την επίτευξη των κοινών στόχων. Οι ορισμοί των διαδικασιών είναι ευέλικτοι και το άμεσο αποτέλεσμα δεν θεωρείται κύριος στόχος.

### **2.4.4 Ειδικές Ροές εργασίας**

Οι ειδικές ροές εργασίας σχεδιάζονται για να μεγιστοποιήσουν την ευελιξία με την άδεια των χρηστών για να αλλαχτούν οι ορισμοί διαδικασίας τόσο πολύ όπως απαιτούνται. Κατά συνέπεια, η ασφάλεια δεν είναι μια προτεραιότητα.

Μια δυναμική προσέγγιση στην τεχνολογία ροής εργασίας έχει επιτρέψει τη μεγαλύτερη ολοκλήρωση μεταξύ αυτών των τύπων ροών της δουλειάς και των

σχετικών προϊόντων τους. Αυτή η προσέγγιση επιτρέπει τη δυναμική αλλαγή των περιπτώσεων διαδικασίας [All01, Σ. 15-38].

Αυτό το κεφάλαιο έχει καθορίσει τις βασικές έννοιες σχετικά με τα συστήματα ροής εργασίας και διαχείρισης ροής εργασίας καθώς επίσης και τους διάφορους τύπους ροών εργασίας. Διάφορες αναφορές έχουν γίνει στα διάφορα έγγραφα που δημοσιεύονται από το διοικητικό συνασπισμό ροής εργασίας (WfMC). Το επόμενο τμήμα θα διαμορφώσει στην προέλευση αυτής της οργάνωσης, ο οποίος έχει πολλές σημαντικές συνεισφορές στον τομέα της ροής εργασίας, με την πρόθεση της καθιέρωσης της αρχής αυτών των συνεισφορών.

## **2.5 Διοικητικός συνασπισμός Ροής εργασίας**

Τα προϊόντα λογισμικού ροής εργασίας έχουν προκύψει από πολλές διαφορετικές πηγές, κάθε ένας με τα πλεονεκτήματά τους, μειονεκτήματα, ορολογία και διεπαφές. Με την πρόοδο της τεχνολογίας ροής εργασίας τα τελευταία χρόνια, η ανάγκη προέκυψε για τα κοινά πρότυπα που αναπτύσσονται, έτσι ώστε οι δυνάμεις κάθε μια που προσφέρει μπορούν να προσαρμοστούν. Ο διοικητικός συνασπισμός ροής εργασίας (WfMC) διαμορφώθηκε τον Αύγουστο του 1993 ως μη κερδοσκοπική οργάνωση με σκοπό να ικανοποιήσει αυτήν την ανάγκη και πολλές άλλες απαιτήσεις των χρηστών. Ο συνασπισμός έχει πάνω από 285 μέλη που διαιρούνται σε τρεις κύριες επιτροπές που συνεδριάζουν περίπου τρεις φορές ετησίως. Αυτές οι επιτροπές είναι οι τεχνικές, Εξωτερικές σχέσεις, και οργανωτικές επιτροπές. Είναι αρμόδιοι για την ανάπτυξη των προτύπων ροής της δουλειάς για την ορολογία, διαλειτουργικότητα, απαιτήσεις συνδετικότητας και προσαρμογής [WFMC01]. Η ανάπτυξη των προτύπων είναι σημαντική δεδομένου ότι βεβαιώνει τους χρήστες ότι τα συστήματα ροής της δουλειάς ικανοποιούν οι θεμελιώδεις απαιτήσεις και επιτυγχάνουν ένα ορισμένο επίπεδο απόδοσης. Αυτό ισχύει επίσης για τις περιπτώσεις όπου η inter-organizational επιχείρηση διευθύνεται και ανάγκης των χρηστών επιβεβαίωση της ικανότητας των εξωτερικών συστημάτων ροής της δουλειάς. Το WfMC στοχεύει να ικανοποιήσει και να εκτοπίσει τις προσδοκίες πελατών μέσω της χρήσης της τεχνολογίας ροής της δουλειάς. Με αυτό τον τρόπο, ελπίζει να αυξήσει τη συνειδητοποίηση της ροής της δουλειάς. Τέλος, στοχεύει να μειώσει τον κίνδυνο που περιλαμβάνεται στη χρήση των προϊόντων ροής της δουλειάς [WFMC01].

Για να επιτύχουν αυτούς τους στόχους, έχει καθορίσει τις προδιαγραφές για την εφαρμογή προϊόντων ροής εργασίας που επιτρέπουν ένα μεγαλύτερο επίπεδο διαλειτουργικότητας μεταξύ των προϊόντων αυτής. Αυτό, στη συνέχεια, θα βελτιώσει την ολοκλήρωση των προϊόντων ροής εργασίας με τα συστήματα και τις υποδομές ΤΠ. Συνεπώς, αυτή η τεχνολογία ροής εργασίας έχει μια πολύτιμη συμβολή στις υπάρχουσες υπηρεσίες ΤΠ, με να τις καταστήσει αποδοτικότερες και αποτελεσματικές [WFMC98].

Η σημαντικότερη συμβολή που γίνεται από το WfMC είναι η ανάπτυξη του προτύπου αναφοράς ροής της εργασίας, το οποίο καθορίζει ένα πλαίσιο για τα πρότυπά του.

## **2.6 Συμπεράσματα**

Αυτό το κεφάλαιο εισήγαγε τις έννοιες που είναι θεμελιώδεις σε μια κατανόηση των συστημάτων ροής της δουλειάς. Το περιεχόμενο βασίστηκε κατά ένα μεγάλο μέρος στους ορισμούς και τα χαρακτηριστικά που αναπτύχθηκαν από το WfMC σε αρκετά από τα δημοσιευμένα έγγραφα του. Είναι σαφές ότι η τεχνολογία ροής εργασίας ασκεί σημαντική επίδραση στις διαδικασίες των επιχειρησιακών διαδικασιών και της επιχείρησης γενικά. Κατά συνέπεια, είναι σημαντικό ότι αυτή η τεχνολογία βελτιστοποιείται στην εφαρμογή της.

Η εξασφάλιση των πληροφοριών και των διαδικασιών που συνδέονται με τις ροές της εργασίας είναι ένας τομέας αυξανόμενης ανησυχίας. Μπορεί σχεδόν να ειπωθεί ότι η επιτυχία μιας εφαρμογής ροής εργασίας εξαρτάται κατά ένα μεγάλο μέρος από την ασφαλή εκτέλεσή της. Αυτό ισχύει ιδιαίτερα όταν εξετάζει το ένα την ανάγκη να διαχειριστεί τα δικαιώματα έγκρισης μελετώντας τα ευαίσθητα στοιχεία, πληροφορίες ή διαδικασίες.

## Κεφάλαιο 3 - Επισκόπηση των Εννοιών Ασφάλειας

### 3.1 Εισαγωγή

Η τελευταία δεκαετία ανακοίνωσε το μετασχηματισμό της κοινωνίας μας σε αυτό που καλείται εποχή πληροφοριών, και συνεπώς, μια δραματική αλλαγή στη στάση και τις προσεγγίσεις απέναντι στη διεύθυνση της επιχείρησης έχει προκύψει. Κατά τη διάρκεια αυτής της μετάβασης, η σημασία των πληροφοριών έχει αυξηθεί σε ένα σημείο όπου η κατοχή της μπορεί απλώς να καθορίσει την ανταγωνιστικότητα μιας επιχείρησης πέρα από τους ανταγωνιστές της. Έχει αναγνωριστεί ότι η διαχείριση αυτών των πληροφοριών διαδραματίζει έναν σημαντικό ρόλο στη λειτουργία και τη γενική διαχείριση μιας επιχείρησης με την εξασφάλιση ότι οι εξακριβωμένες πληροφορίες που είναι σχετικές συγκεντρώνονται, εφαρμόζονται και διατηρούνται για να ενισχύσουν την υπηρεσία της επιχείρησης στους πελάτες της. Για λόγους όπως αυτοί, η προστασία αυτού του πολύτιμου προτερήματος είναι ύψιστης σημασίας στις περισσότερες οργανώσεις σήμερα.

Η εξασφάλιση των πληροφοριών αποτελείται από τη διατήρηση έξι χαρακτηριστικών ασφάλειας: εμπιστευτικότητα, ακεραιότητα, διαθεσιμότητα, χρησιμότητα, αυθεντικότητα και κατοχή [Par95].

1. Η προστασία της εμπιστευτικότητας των πληροφοριών συνεπάγεται ότι οι πληροφορίες δεν ανακτώνται παράνομα από τους αναρμόδιους ανθρώπους.
2. Η ακεραιότητα των πληροφοριών πρέπει να συντηρηθεί με την εξασφάλιση ότι τις πληροφορίες ότι με κάθε τρόπο από ένα εισβολέα δεν αλλάζουν στο σύστημα.
3. Το σύστημα πρέπει αποτελεσματικά να παρέχει τις πληροφορίες στα εξουσιοδοτημένα πρόσωπα όταν απαιτούνται ώστε να καταστήσουν τις πληροφορίες διαθέσιμες.
4. Πρέπει επίσης να εξασφαλιστεί ότι οι πληροφορίες που περιλαμβάνονται μέσα στο σύστημα ισχύουν χρήσιμες και στη λειτουργία της επιχείρησης έτσι ώστε η ανώτερη διαχείριση μπορεί να πάρει τις σωστές αποφάσεις βασισμένες σε αυτές.
5. Επιπλέον, ένα ασφαλές σύστημα πρέπει να είναι αυθεντικό, ποια μέσα με τα οποία οι πληροφορίες αυτό αποθηκεύουν τόσο καλά όσο τις λειτουργίες που παρέχουν πρέπει να αντιστοιχούν και να ισχύσουν στο τρέχον εξωτερικό επιχειρησιακό περιβάλλον.
6. Η επιβολή της ασφάλειας πληροφοριών σημαίνει επίσης την κατοχή των πληροφοριών. Η απώλεια πληροφοριών μπορεί να έχει μια καταστρεπτική επίδραση στην απόδοση της επιχείρησης, ειδικά εάν περιέρχεται στα χέρια ενός κακοποιού ή ενός ανταγωνιστή.

Η εξασφάλιση της συντήρησης των προαναφερθέντων χαρακτηριστικών ασφάλειας απαιτεί την καθιέρωση μιας διοικητικής υποδομής ασφάλειας

πληροφοριών, η οποία μπορεί να εξασφαλίσει ότι οι πληροφορίες κρατιούνται διαθέσιμες, ακριβείς και σχετικές με την επιχείρηση και εκείνους που τις χρειάζονται για να κάνουν ενημερώσεις και να λάβουν ακριβείς αποφάσεις.

### **3.2 Οι 5 στυλοβάτες της ασφάλειας πληροφοριών**

Η ασφάλεια πληροφοριών αποτελεί τη συντήρηση πέντε θεμελιωδών υπηρεσιών ασφάλειας, η οποία οδηγεί στην καθιέρωση των αξιόπιστων συστημάτων που ικανοποιούν τις ανάγκες μιας οργάνωσης. Αυτές οι υπηρεσίες, όπως καθορίζονται [ISO89], είναι προσδιορισμός και επικύρωση, έγκριση, εμπιστευτικότητα, ακεραιότητα και μη αποκλήρυξη. Για να ικανοποιήσει αυτές τις υπηρεσίες εξασφαλίζει ότι η διαχείριση μπορεί να λάβει τις ενημερωμένες αποφάσεις που είναι βασισμένες στις εξακριβωμένες πληροφορίες. Αυτό τους επιτρέπει να ενεργήσουν σύμφωνα με τις προθέσεις της επιχείρησης, ώστε να κερδίσουν μια ανταγωνιστικότητα στην εξωτερική αγορά.

#### **3.2.1 Προσδιορισμός και Επικύρωση**

Αρχικά, τα συστήματα πρέπει να προωθήσουν τις αρχές του προσδιορισμού και της επικύρωσης. Αυτό περιλαμβάνει τον προσδιορισμό των νομικών χρηστών στο σύστημα καθώς επίσης και την εξασφάλιση ότι είναι αυτοί ακριβώς που υποθέτουν πώς είναι. Το WfMC [WfMC98] καθορίζει την επικύρωση ως «διαδικασία από την οποία ένα συγκρότημα ηλεκτρονικών υπολογιστών ή ένας (ανθρώπινος) χρήστης συστημάτων προσδιορίζεται σαφώς σε ένα άλλο συγκρότημα ηλεκτρονικών υπολογιστών, κανονικά στα πλαίσια του κέρδους της πρόσβασης στις διάφορες υπηρεσίες που το επικυρωμένο συμβαλλόμενο μέρος εξουσιοδοτείται για να χρησιμοποιήσει σε εκείνο το συγκρότημα ηλεκτρονικών υπολογιστών». Αυτή η υπηρεσία παρέχεται με τον καθορισμό των προσδιορισμών χρηστών καθώς επίσης και την εφαρμογή των μηχανισμών επικύρωσης. Οι παράμετροι επικύρωσης έρχονται χαρακτηριστικά με τρεις μορφές: κάτι που ο χρήστης ξέρει (PIN ή κωδικός πρόσβασης), κάτι που ο χρήστης κατέχει (κάρτα σημείου ή μνήμης) και κάτι γενετικά μοναδικό στο χρήστη (ανίχνευση δακτυλικών αποτυπωμάτων ή αμφιβληστροειδών) [ISO89].

Σε ένα πλαίσιο ροής εργασίας, ένας χρήστης προσδιορίζεται χαρακτηριστικά και επικυρώνεται κατά τη διάρκεια ενός τύπου δραστηριότητας σύνδεσης που εμφανίζεται προτού να οριστούν οι εγκρίσεις για τους στόχους ροής εργασίας [WfMC98].

#### **3.2.2 Έλεγχος πρόσβασης (έγκριση)**

Η δεύτερη υπηρεσία ασφάλειας που πρέπει να εφαρμοστεί είναι η έγκριση ή ο λογικός έλεγχος πρόσβασης. Αυτή η υπηρεσία αναμιγνύεται στον έλεγχο των δικαιωμάτων πρόσβασης των νομικών χρηστών στους διάφορους πόρους όπως οι συναλλαγές, βάσεις δεδομένων, αρχεία και κεντρικοί υπολογιστές. Schneier

[Sch00, σελ. 123] προσδιορίζει δύο θεμελιώδεις προσεγγίσεις στον έλεγχο πρόσβασης. Ο πρώτος περιλαμβάνει την τοποθέτηση των περιορισμών σε ποιους χρήστες μπορούν να κάνουν τι, ενώ άλλος περιλαμβάνει ποιες διαδικασίες μπορούν να εκτελεστούν σε ορισμένα αντικείμενα. Το WfMC [WFMC98] κάνει μια διάκριση μεταξύ των όρων «έγκριση» και «έλεγχος πρόσβασης». Η έγκριση ορίζεται ως «η διαδικασία στο συγκρότημα ηλεκτρονικών υπολογιστών τις διάφορες λειτουργίες που ένας χρήστης (άνθρωπος και ενδεχομένως ένα συγκρότημα ηλεκτρονικών υπολογιστών) μπορεί να αναλάβει». Ο έλεγχος πρόσβασης ορίζεται ως «ο μηχανισμός από τον οποίο οι χρήστες επιτρέπεται να έχουν πρόσβαση στις διάφορες διαδικασίες ή τα στοιχεία μέσα σε ένα συγκρότημα ηλεκτρονικών υπολογιστών, την ταυτότητά τους (που καθιερώνεται σύμφωνα με την επικύρωση) και τα σχετικά προνόμια (που καθιερώνονται από την έγκριση). Εντούτοις, οι όροι «έλεγχος πρόσβασης» και «έγκριση» θα χρησιμοποιηθούν εναλλακτικά σε όλη αυτήν την εργασία.

Η υπηρεσία ασφάλειας της έγκρισης, ή έλεγχος πρόσβασης, είναι η πιο σχετική με αυτήν την εργασία. Σε ένα πλαίσιο ροής εργασίας, οι μηχανισμοί ελέγχου πρόσβασης περιορίζουν την ανάθεση των εγκρίσεων στις συγκεκριμένες περιπτώσεις διαδικασίας ή δραστηριότητας. Μερικοί μηχανισμοί έγκρισης περιλαμβάνουν τις μήτρες ελέγχου πρόσβασης και τους καταλόγους καταλόγου.

### 3.2.3 Εμπιστευτικότητα

Η τρίτη υπηρεσία ασφάλειας της εμπιστευτικότητας ενδιαφέρεται για την προστασία των πληροφοριών από παράνομα να ανακτηθεί από τους αναρμόδιους ανθρώπους. Η εμπιστευτικότητα εφαρμόζεται χρησιμοποιώντας τους μηχανισμούς κρυπτογράφησης που εκτελούν τους μαθηματικούς αλγόριθμους στα στοιχεία κοινού κειμένου, με αυτόν τον τρόπο δίνοντας το ως ακατανόητο κρυπτογράφημα στους παράνομους χρήστες [Pf197]. Υπάρχουν πρώτιστα δύο τύποι κρυπτογραφήσεων: συμμετρική κρυπτογράφηση, χρησιμοποιείται ο ίδιος αλγόριθμος για να κρυπτογραφήσει και να αποκρυπτογραφήσει τα στοιχεία, και ασύμμετρη κρυπτογράφηση, ποιες χρήσεις διαφορετικές, αλλά από μαθηματική άποψη σχετικός, κλειδιά στην κρυπτογράφηση και διαδικασία αποκρυπτογράφησης.

Σε ένα περιβάλλον ροής εργασίας, είναι σημαντικό να εξασφαλιστεί ότι η εμπιστευτικότητα των πληροφοριών είναι συνεχής. Εάν δεν είναι, θα σήμαινε ότι οι μηχανισμοί ελέγχου πρόσβασης έχουν υπονομευθεί.

### 3.2.4 Ακεραιότητα

Η ακεραιότητα πληροφοριών είναι η τέταρτη υπηρεσία που πρέπει να επιβληθεί. Αυτό περιλαμβάνει την εξασφάλιση αυτού που προσδιορίζεται μόνο επιτυχώς, τα επικυρωμένα και εξουσιοδοτημένα πρόσωπα μπορούν να αλλάξουν το στοιχείο που περιλαμβάνεται μέσα στο σύστημα ή ταξίδι μέσω ένα δίκτυο. Οι μηχανισμοί που χρησιμοποιούνται για να επιβάλουν την ακεραιότητα περιλαμβάνουν τις



μεθόδους κρυπτογράφησης όπως αναφέρονται ανωτέρω, όπως και τους κώδικες επικύρωσης μηνυμάτων (MACs), checksums και αφομοιώσεις μηνυμάτων. Τα MACs είναι σημεία (καλούμενα τα authenticators) που υπολογίζονται σε ένα μήνυμα και στέλνονται έπειτα με την κρυπτογραφημένη έκδοση του μηνύματος. Η ακεραιότητα μπορεί να επιβληθεί με τον έλεγχο ότι η αποκρυπτογραφημένη έκδοση του μηνύματος αντιστοιχεί με το authenticator της. Εάν όχι, το μήνυμα με κάποιο τρόπο από ένας εισβολέας έχει αλλάξει στη συναλλαγή. Checksums χρησιμοποιούν τα μυστικά κλειδιά για να επιλύσουν τις τιμές που αφορούν τα στοιχεία που στέλνονται. Κατά συνέπεια, είναι παρόμοιο με την κρυπτογράφηση αν και η διαδικασία δεν μπορεί να αντιστραφεί. Οι αφομοιώσεις μηνυμάτων δημιουργούνται χρησιμοποιώντας τους αλγόριθμους που υπολογίζουν τα δακτυλικά αποτυπώματα ενός μηνύματος. Η MAC που προκύπτει από αυτήν την λειτουργία χρησιμοποιείται έπειτα στο μήνυμα εισαγωγής σε έναν hash αλγόριθμο.

Στα πλαίσια μιας ροής εργασίας, η συντήρηση της ακεραιότητας αντικειμένου είναι σημαντική. Ο έλεγχος πρόσβασης συμβάλλει στην επιβολή της ακεραιότητας αντικειμένου με τον περιορισμό της πρόσβασης χρηστών σε ορισμένα αντικείμενα. Εάν, εντούτοις, η ακεραιότητα συμβιβάζεται μέσω της αναρμόδιας παρεμπόδισης των πληροφοριών, η ακεραιότητα της ολόκληρης ροής εργασίας θα συμβιβαστεί.

### 3.2.5 Non-repudiation (μη-άρνηση)

Αυτό μας φέρνει στην πέμπτη υπηρεσία ασφάλειας, το non-repudiation. Η υποστήριξη αυτής της αρχής αποτρέπει οποιοδήποτε συμβαλλόμενο μέρος που συμμετέχει σε μια συναλλαγή από την πιο πρόσφατη συμμετοχή άρνησης. Η επιβολή της μη-άρνησης είναι ιδιαίτερα σημαντική έτσι σε ένα περιβάλλον ηλεκτρονικού εμπορίου, παραδείγματος χάριν, σε περίπτωση που ένας πελάτης αρνείται μια ηλεκτρονική αγορά ή μια επένδυση. Οι non-repudiation μηχανισμοί απαιτούν κάθε συμβαλλόμενο μέρος που συμμετέχει σε οποιαδήποτε συναλλαγή να έχει στην κατοχή του μια μυστική ψηφιακή υπογραφή που το προσδιορίζει μεμονωμένα, με αυτόν τον τρόπο χρησιμεύοντας ως το αντίτιμο μιας αναλογικής υπογραφής. Τα «ψηφιακά πιστοποιητικά ελέγχουν την ταυτότητα του αποστολέα, τοποθετήστε μια πλαστογράφηση-ανθεκτική σφραγίδα σε ένα μήνυμα, και χορηγήστε την απόδειξη ότι μια συναλλαγή έχει εμφανιστεί. Τα ψηφιακά πιστοποιητικά δίνουν στο Διαδίκτυο ένα υψηλό επίπεδο της βεβαιότητας, όπως ένα διαβατήριό ή ενός οδηγού η άδεια ελέγχει την ταυτότητα ενός προσώπου» [Dav98]. Αυτές οι ψηφιακές υπογραφές ενσωματώνονται με την ασύμμετρη (δημόσιο κλειδί) κρυπτογράφηση για να επιβάλουν την ασφάλεια μη-άρνησης.

Σε ένα πλαίσιο ροής εργασίας, η επιβολή της μη-άρνησης είναι ιδιαίτερα σημαντική όταν έχουν οι δραστηριότητες ροής εργασίας οικονομικές επιπτώσεις.

Το WfMC [WFMC98] προτείνει την προσθήκη των υπηρεσιών για την ιδιωτικότητα ελέγχου και στοιχείων. Τα στοιχεία λογιστικού ελέγχου μπορούν να αναλυθούν για τα στοιχεία των πιθανών ή υπαρχουσών παραβιάσεων ασφάλειας.

Η ιδιωτικότητα στοιχείων ενδιαφέρεται για την επιβολή της εμπιστευτικότητας ταυτόχρονα διατηρώντας την ανωνυμία.

### **3.3 Προσεγγίσεις ελέγχου πρόσβασης για τις πολιτικές ασφάλειας**

Μια πολιτική ασφάλειας καθορίζει τα αναμενόμενα πρότυπα της επιβολής ασφάλειας μέσα σε μια οργάνωση. Πρώτιστα, διευθύνσεις μιας ασφάλειας πολιτικής που έχει πρόσβαση σε ποιους πόρους, όπως και το πώς αυτή η πρόσβαση πρόκειται να ρυθμιστεί και να διαχειριστεί. Ο Pfleeger [Pfl97, σελ. 476] καθορίζει μια αποτελεσματική πολιτική ασφάλειας ως μια που είναι ανθεκτική έτσι ώστε να μπορεί να προσαρμοστεί καλά στις μεταβαλλόμενες περιστάσεις, χρήσιμο δεδομένου ότι μπορεί να γίνει κατανοητό και να ακολουθηθεί από τον καθέναν για τον οποίο ισχύει και ρεαλιστικός από άποψη η προδιαγραφή του είναι για εφικτούς στόχους ασφάλειας. Ένα πολιτικό έγγραφο ασφάλειας είναι σημαντικό δεδομένου ότι καθιστά τους υπαλλήλους ενήμερους για τη θέση της οργάνωσης, με αυτόν τον τρόπο προστατεύοντας την οργάνωση από τις τυχαίες παραβιάσεις ασφάλειας εργαζομένων. Επιπλέον, καθορίζει πρότυπα σύμφωνα με τα οποία η τιμωρία των σκόπιμων παραβάσεων μπορεί να αξιολογηθεί.

Οι ακόλουθες προσεγγίσεις μπορούν να χρησιμοποιηθούν για να καθορίσουν τον έλεγχο πρόσβασης σε μια οργανωτική πολιτική ασφάλειας. Είναι, διακριτικός έλεγχος πρόσβασης (DAC), υποχρεωτικός έλεγχος πρόσβασης (MAC), και ρολοκεντρικός έλεγχος πρόσβασης (RBAC).

#### **3.3.1 Διακριτικός Έλεγχος Πρόσβασης (DAC)**

Η προσέγγιση DAC επιτρέπει στο δημιουργό ενός αντικειμένου, ή οποιοσδήποτε που εξουσιοδοτείται για να τον ελέγξει, για να λάβουν τις αποφάσεις ελέγχου πρόσβασης. Αυτά τα δικαιώματα αλλάζουν δυναμικά κατά την κρίση του ιδιοκτήτη ενός αντικειμένου [Pfl97, σελ. 290]. Οι μηχανισμοί ελέγχου πρόσβασης που υποστηρίζουν μια πολιτική DAC περιλαμβάνουν τους καταλόγους καταλόγου, κατάλογοι ελέγχου πρόσβασης και μήτρες ελέγχου πρόσβασης. Καθένας θα αξιολογηθεί τώρα.

Οι κατάλογοι καταλόγου διατηρούν έναν κατάλογο για κάθε χρήστη (θέμα) στο σύστημα. Κάθε είσοδος καταλόγου περιέχει τους τομείς του αντικειμένου και των σχετικών δικαιωμάτων πρόσβασης που έχει ένας χρήστης. Ένας δείκτης σε κάθε αντικείμενο συμπεριλαμβάνεται επίσης. Ένα μειονέκτημα αυτής της προσέγγισης είναι ότι οι διοικητικές περιπλοκές προκύπτουν σε περίπτωση που ένα αντικείμενο αφαιρείται ή μετονομάζεται. Η συντήρηση είναι δύσκολη καθώς αναπροσαρμογές απαιτούνται στον κατάλογο πολλών χρηστών [Pfl97, σελ. 243].

Οι κατάλογοι ελέγχου πρόσβασης διατηρούν τους καταλόγους πρόσβασης για κάθε αντικείμενο στο σύστημα. Ομοίως, οι περιπλοκές διοίκησης προκύπτουν εάν

ένας χρήστης παραιτείται, δεδομένου ότι το σύστημα πρέπει να ψάξει για και να διαγράψει κάθε υπαγόμενο λήμμα από πολλούς καταλόγους αντικειμένου [Pfl97, Σ. 244-246]. Και οι δύο από τις ανωτέρω προσεγγίσεις μπορούν να είναι κατάλληλες για μικρές, σταθερές επιχειρήσεις.

Οι μήτρες ελέγχου πρόσβασης είναι δισδιάστατες χαρτογραφήσεις των θεμάτων εναντίον αντικειμένων, οι διατομές των οποίων αντιπροσωπεύουν το αντίστοιχο δικαίωμα πρόσβασης. Αυτή η προσέγγιση δεν είναι απαραίτητως πρακτική δεδομένου ότι μια μήτρα ελέγχου πρόσβασης περιέχει χαρακτηριστικά πολλές κενές καταχωρήσεις [Pfl97, σελ. 246].

Από τις προσεγγίσεις DAC που συζητούνται, η μήτρα ελέγχου πρόσβασης είναι μοναδική με τη δυνατότητα για την εφαρμογή του δυναμικού ελέγχου πρόσβασης σε ένα περιβάλλον ροής εργασίας.

### 3.3.2 Υποχρεωτικός Έλεγχος Πρόσβασης (MAC)

Μια προσέγγιση της MAC απαιτεί ότι οι αποφάσεις ελέγχου πρόσβασης λαμβάνονται πέρα από τον έλεγχο του ιδιοκτήτη ενός αντικειμένου. Αυτοί έχουν πρόσβαση στις αποφάσεις ελέγχου καθορίζονται από μια κεντρική αρχή, και οι χρήστες δεν έχουν καμία εξουσία για να τους αλλάξουν [Pfl97, σελ. 290]. Το στρατιωτικό πρότυπο ασφάλειας και το πρότυπο Bell - Lapadula υποστηρίζουν μια πολιτική της MAC. Αναφέρονται ως πολλαπλής στάθμης πρότυπα επειδή επιβάλλουν τον έλεγχο πρόσβασης πέρα από μια σειρά των θεμάτων και των αντικειμένων στην οργάνωση, με βάση τα επίπεδα εκκαθάρισης των θεμάτων και τα επίπεδα ευαισθησίας των αντικειμένων.

#### 3.3.2.1 Στρατιωτικό Πρότυπο Ασφάλειας

Το στρατιωτικό πρότυπο ασφάλειας χρησιμοποιεί μια ιεραρχία ταξινόμησης ασφάλειας για να ταξινομήσει τις πληροφορίες και τα θέματα. Χαρακτηριστικά, αυτά τα επίπεδα ασφάλειας είναι κορυφαίο μυστικό, μυστικό, εμπιστευτικός και δημόσιος. Οι πληροφορίες και τα θέματα συνδέονται με τα διαμερίσματα που αντιπροσωπεύουν τα προγράμματα ή τα τμήματα για τα οποία ισχύουν. Οι πληροφορίες και τα διαμερίσματα συνδέονται έπειτα σε μια need-to-know βάση έτσι ώστε λιγότερες αρχές προνομίου επιβάλλονται. Αυτό επιτυγχάνεται με μόνο να ορίσει τα δικαιώματα πρόσβασης σε ένα θέμα σε ένα σχετικό διαμέρισμα που έχει την απαραίτητη ταξινόμηση ασφάλειας [Pfl97, Σ. 271-273]. Τυπικά, οι εγκρίσεις ορίζονται σύμφωνα με τον ακόλουθο κανόνα:

- Ένα θέμα μπορεί να αποκτήσει πρόσβαση σε ένα αντικείμενο εάν και μόνο εάν:  
 $Security\ Classification(subject) \_ Security\ Classification(object) \text{ AND}$   
 $Compartment(object) \_ Compartment(subject).$

Μπορεί να δει από αυτόν τον κανόνα ότι οι πληροφορίες ρέουν μόνο από τα χαμηλότερα επίπεδα ασφάλειας προς τα πάνω. Αν και αυτό το πρότυπο παρέχει

μια πολύ ασφαλή εφαρμογή ελέγχου πρόσβασης, είναι πολύ άκαμπτο και η ταξινόμηση κάθε κομματιού των πληροφοριών στην οργάνωση είναι δύσκολη. Είναι, εντούτοις, ένα κατάλληλο πρότυπο για τις καταστάσεις όπου ο αυστηρός έλεγχος πρόσβασης μπορεί να ρυθμιστεί από μια κεντρική αρχή.

### 3.3.2.2 Πρότυπο Bell – Lapadula

Το μοντέλο Bell LaPadula (BLM) [M85], που επίσης αποκαλείται το «πολύ-επίπεδο» μοντέλο, είχε προταθεί από τους Bell και LaPadula για να επιβάλει έλεγχο πρόσβασης σε αντικείμενα για στρατιωτικές και κυβερνητικές εφαρμογές. Σε αυτές τις εφαρμογές οι χρήστες και τα αντικείμενα είναι διαχωρισμένα σε διαφορετικά επίπεδα ασφαλείας. Ένας χρήστης μπορεί να αποκτήσει πρόσβαση μόνο σε ορισμένα αντικείμενα που βρίσκονται σε κάποια επίπεδα ασφαλείας που συσχετίζονται με το δικό του επίπεδο ασφαλείας. Για παράδειγμα, οι δυο επόμενες προτάσεις είναι κάποιες τυπικές προτάσεις πρόσβασης που μπορούν να διατυπωθούν μέσα από το μοντέλο αυτό:

- Μη ταξινομημένο προσωπικό δεν μπορεί να διαβάσει δεδομένα που βρίσκονται σε εμπιστευτικά επίπεδα.
- Άκρως απόρρητα δεδομένα δεν μπορούν να γραφτούν σε αρχεία που βρίσκονται σε μη εμπιστευτικά επίπεδα.

Ένα από τα σημαντικά στοιχεία του μοντέλου Bell LaPadula που το διαφοροποιεί από τα άλλα μοντέλα είναι ότι εστιάζει και στον έλεγχο της ροής της πληροφορίας. Αυτό το μοντέλο ασφαλείας αποτελείται από τα εξής στοιχεία:

- Ένα σύνολο από χρήστες (πρόσωπα), ένα σύνολο από αντικείμενα και έναν πίνακα ελέγχου πρόσβασης (access control matrix).
- Μερικά διατεταγμένα επίπεδα ασφαλείας. Κάθε χρήστης (πρόσωπο) έχει άδεια να διαχειρίζεται αντικείμενα κάποιου επιπέδου και κάθε αντικείμενο είναι ταξινομημένο έτσι ώστε να ανήκει σε κάποιο επίπεδο ασφαλείας. Ακόμη, ο κάθε χρήστης ορίζει ένα τρέχων επίπεδο εξουσιοδότησης το οποίο δεν ξεπερνά το επίπεδο εξουσιοδότησης που έχει οριστεί για τον χρήστη.

Το σύνολο από τα δικαιώματα πρόσβασης είναι το παρακάτω:

- **Read-Only:** Ο χρήστης μπορεί μόνο να διαβάσει το αντικείμενο.
- **Append:** Ο χρήστης μπορεί να προσαρτήσει πληροφορία στο αντικείμενο αλλά δεν μπορεί να το διαβάσει.
- **Execute:** Ο χρήστης μπορεί να εκτελέσει το αντικείμενο αλλά δεν μπορεί να το διαβάσει ή να το γράψει.
- **Read-Write:** Ο χρήστης έχει δικαίωμα να γράψει και να διαβάσει το αντικείμενο.

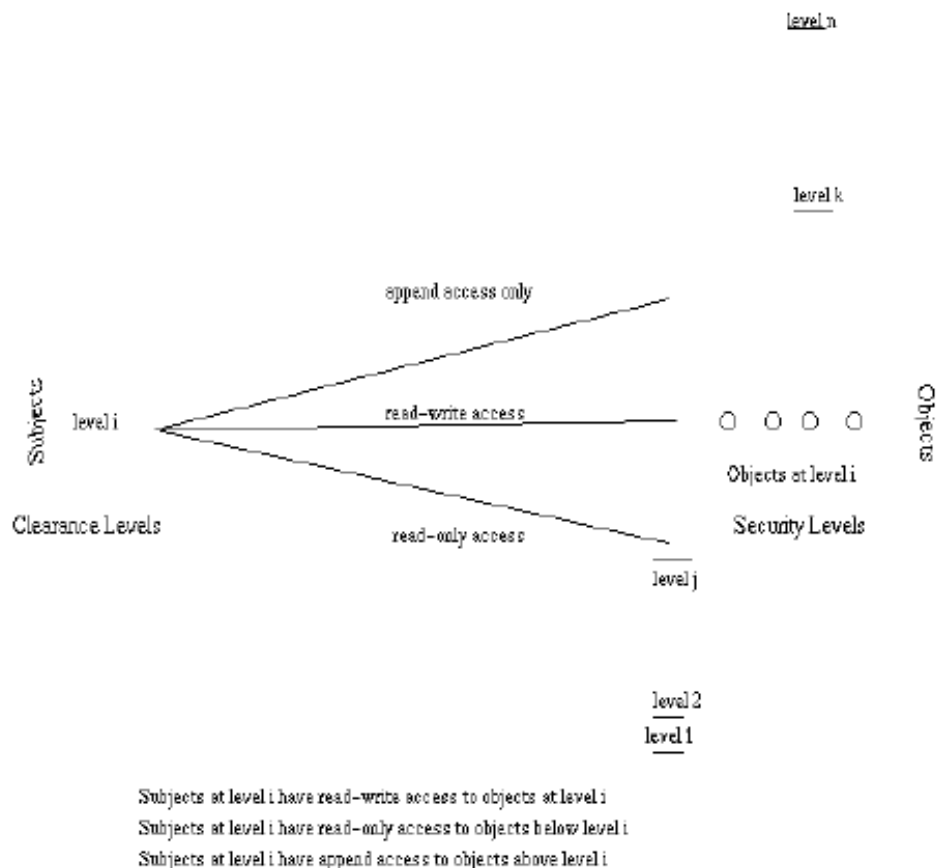
**Χαρακτηριστικό έλεγχου:** Το χαρακτηριστικό αυτό δίνεται στον χρήστη ο οποίος δημιουργεί ένα αντικείμενο. Λόγω αυτού, ο δημιουργός ενός αντικείμενου μπορεί να δώσει οποιοδήποτε (από τα παραπάνω) δικαίωμα πρόσβασης σε οποιονδήποτε χρήστη, εκτός βέβαια από το ίδιο χαρακτηριστικό έλεγχου. Ο χρήστης δημιουργός θα ονομάζεται και «χειριστής» του αντίστοιχου αντικειμένου.

Οι παρακάτω είναι οι περιορισμοί που επιβάλλονται από το μοντέλο:

- **reading down:** Ένας χρήστης έχει άδεια για να διαβάσει τα αντικείμενα των οποίων το επίπεδο ασφαλείας είναι χαμηλότερο από το τρέχων επίπεδο αδειας του χρήστη. Αυτό εμποδίζει ένα χρήστη να αποκτήσει πρόσβαση σε πληροφορίες που βρίσκονται σε επίπεδα ασφαλείας υψηλότερα από το τρέχων επίπεδο αδειας του.
- **writing up:** Ένας χρήστης έχει την άδεια «Append» να γράφει και να μην διαβάζει σε αντικείμενα των οποίων το επίπεδο ασφαλείας είναι μεγαλύτερο από το τρέχων επίπεδο αδειας. Αυτό εμποδίζει ένα χρήστη να περάσει πληροφορία σε επίπεδα χαμηλότερα από το τωρινό επίπεδο.

Για παράδειγμα, έστω ότι υπάρχουν τα επόμενα επίπεδα ασφαλείας: Μη ευαίσθητο < Ευαίσθητο < Απόρρητο < Άκρως απόρρητο. Έστω ότι ένας χρήστης έχει επίπεδο άδειας «απόρρητο». Τότε μπορεί μόνο να προσαρτήσει πληροφορία στα αντικείμενα που έχουν ετικέτα «άκρως απόρρητο», μπορεί να αναγνώσει και να γράψει τα αντικείμενα που βρίσκονται στο επίπεδο «απόρρητο», και τέλος μπορεί μόνο να διαβάσει τα αντικείμενα που έχουν ετικέτα: «ευαίσθητο» και «μη ευαίσθητο». Αυτός ο μηχανισμός φαίνεται στην Εικόνα 4.

Το μοντέλο Bell LaPadula συμπληρώνει το πίνακα πρόσβασης (access matrix) με τους παραπάνω περιορισμούς για να ελέγξει την πρόσβαση και την ροή δεδομένων. Για παράδειγμα, ένας χρήστης μπορεί να έχει την άδεια για ανάγνωση ενός αντικείμενου (read access) μέσω του πίνακα ελέγχου πρόσβασης, αλλά να μην μπορεί να ασκήσει αυτό το δικαίωμα αν το αντικείμενο είναι σε υψηλότερο επίπεδο ασφαλείας από το επίπεδο αδειας του χρήστη.



Εικόνα 4 - Τρόπος Λειτουργίας Μοντέλου Bell Lapadula

Το Bell and LaPadula έχει μοντελοποιήσει την συμπεριφορά ενός συστήματος ασφαλείας ως μια μηχανή πεπερασμένων καταστάσεων (finite state machine) και έχει ορίσει ένα σύνολο από μεταβάσεις όπου δεν παραβαίνουν την ασφάλεια του συστήματος. Οι παρακάτω πράξεις εγγυούνται ένα ασφαλές σύστημα:

- **get access**: Χρησιμοποιείται από ένα χρήστη για να ξεκινήσει την πρόσβαση σε ένα αντικείμενο. (read, append, execute...)
- **release access**: Χρησιμοποιείται από ένα χρήστη για να παραδώσει μια πρόσβαση που έχει αρχικοποιηθεί.
- **give access**: Ο χειριστής ενός αντικειμένου (δημιούργησε το αντικείμενο) μπορεί να δώσει μία συγκεκριμένη πρόσβαση (στο αντικείμενο αυτό) σε κάποιον χρήστη.
- **rescind access**: Ο χειριστής ενός αντικειμένου μπορεί να ανακαλέσει μια συγκεκριμένη πρόσβαση (στο αντικείμενο αυτό) από ένα χρήστη.
- **create object**: Αφήνει ένα χρήστη να δημιουργήσει ένα αντικείμενο.
- **delete object**: Αφήνει ένα χρήστη να σβήσει ένα αντικείμενο.
- **change security level**: Αφήνει ένα χρήστη να αλλάξει το τρέχων του επίπεδο άδειας (χαμηλότερο από το επίπεδο άδειας που του έχει ανατεθεί)

Όμως υπάρχουν κάποιες συνθήκες που πρέπει να ισχύουν για να μπορούν να εκτελεστούν οι παραπάνω πράξεις. Για παράδειγμα, ένας χρήστης πρέπει να

εκχωρεί και ανακαλεί δικαιώματα σε ένα αντικείμενο αν και μόνο αν έχει τα χαρακτηριστικά ελέγχου του αντικειμένου αυτού.

Το Bell LaPadula είναι ένα απλό γραμμικό μοντέλο που ασκεί έλεγχο στην πρόσβαση και ροή δεδομένων μέσω των παραπάνω ιδιοτήτων και πράξεων. Βλέπουμε ακόμη ότι ενώ άρχισε ως ένα μοντέλο υποχρεωτικού ελέγχου πρόσβασης, η ύπαρξη ενός πίνακα πρόσβασης (access matrix) του προσδίδει κάποια στοιχεία από τα μοντέλα προαιρετικού ελέγχου πρόσβασης, τα οποία είναι αρκετά περιορισμένα έτσι ώστε να μην υπάρχουν τα προβλήματα του προαιρετικού μοντέλου πρόσβασης στο μοντέλο αυτό. Όμως ένα από τα κύρια μειονεκτήματα του είναι ότι τα επίπεδα ασφαλείας είναι στατικά. Οι ιδιότητες αυτού του μοντέλου μπορεί να γίνουν πολύ περιοριστικές σε περιπτώσεις που συγκεκριμένες λειτουργίες είναι εκτός του πλαισίου του συστήματος ασφαλείας.

### 3.3.3 Ρολοκεντρικός Έλεγχος Πρόσβασης (RBAC)

Το RBAC μπορεί να θεωρηθεί ως υβρίδιο των πολιτικών MAC και DAC. Στις περισσότερες οργανώσεις, μια πολιτική ασφάλειας πρέπει να εφαρμοστεί στις εκατοντάδες, εάν όχι χιλιάδες, από τους υπαλλήλους. Για να απλοποιήσει τη διοίκηση ασφάλειας, πολλές οργανώσεις καθορίζουν τους ρόλους με τους οποίους τα πολλαπλάσια άτομα μπορούν να συνδεθούν. Η πολιτική ασφάλειας της οργάνωσης καθορίζει έπειτα πώς οι άδειες πρόκειται να συνδεθούν με αυτούς τους ρόλους. Η προσέγγιση RBAC στον έλεγχο πρόσβασης είναι ιδιαίτερα αποτελεσματική σε αυτήν όταν γίνονται οι αλλαγές στην οργανωτική πολιτική ασφάλειας, χρειάζονται μόνο να γίνουν στις αναθέσεις ρόλου, οι όποιοι είναι σημαντικά λιγότεροι από τις μεμονωμένες αναθέσεις.

## 3.4 Ασφάλεια στις Ροές Εργασίας

Αναφέρθηκε ότι η ασφάλεια στα περιβάλλοντα ροής εργασίας ενδιαφέρεται πρώτιστα για την επιβολή του ελέγχου πρόσβασης. Σε ένα πλαίσιο ροής εργασίας, οι μηχανισμοί ασφάλειας ελέγχου πρόσβασης πρέπει να εξασφαλίσουν ότι οι εξαρτήσεις στόχου συντονίζονται και ότι οι στόχοι εκτελούνται από τα εξουσιοδοτημένα θέματα μόνο. Η συντήρηση της ακεραιότητας αντικειμένου είναι η πιο σχετική, ιδιαίτερα σημασιολογική ακεραιότητα, η οποία αφορά τη συνέπεια των πληροφοριών με τους επιχειρησιακούς κανόνες [LR00]. Λόγω των εξειδικευμένων αναγκών ασφάλειας των συστημάτων ροής της δουλειάς, τα πρότυπα ελέγχου πρόσβασης πρέπει να ικανοποιήσουν τις νέες απαιτήσεις με σκοπό την επιβολή της ασφάλειας σε τους. Μερικοί από εκείνους προσδιόρισαν στη λογοτεχνία [AH96b, BE01, CBE00, KB98, Mof98], θα διαμορφωθεί τώρα. Ευρέως θεωρείται ότι ένα αποτελεσματικό πρότυπο έγκρισης ροής της δουλειάς πρέπει:

1. Επιβάλλεται ο χωρισμός των περιορισμών καθηκόντων που απεικονίζουν τους επιχειρησιακούς κανόνες της οργάνωσης. Αυτό θα εξασφαλίσει ότι η σημασιολογική ακεραιότητα των πληροφοριών είναι συνεχής σε όλη τη

διαδικασία ροής της δουλειάς [BE01]. Τυπικά, ο χωρισμός των καθηκόντων εφαρμόζεται μέσω της προδιαγραφής των περιορισμών. Αυτή η αρχή είναι σχετική για τις περιπτώσεις όπου η εμπειρία και η γνώση που αποκτιούνται σε μια προηγούμενη εκτέλεση στόχου απαιτούνται για να εκτελέσουν έναν επόμενο στόχο.

2. Συγχρονίστε τη χορήγηση και την ανάκληση των εγκρίσεων με τις ενάρξεις και τις ολοκληρώσεις στόχου, αντίστοιχα, π.χ. η ροή εργασίας με τη ροή έγκρισης. Αυτό θα εξασφαλίσει ότι οι άδειες απονέμονται μόνο κατά τη διάρκεια των στόχων και θα συμβάλει στην επιβολή του ακριβούς λιγότερη έννοια προνομίου. Αυτό είναι επειδή θα αποτρέψει τους χρήστες από την απόκτηση των προνομίων για τα πιο μακροχρόνια χρονικά διαστήματα από τα απαιτημένα. Η εφαρμογή ακριβής λιγότερες αρχές προνομίου περιλαμβάνει ουσιαστικά σε έναν χρήστη των λιγότερων πιθανών αδειών που απαιτούνται τη χορήγηση για να εκτελέσουν έναν στόχο [CBE00]. Συνδεδεμένος με το συγχρονισμό ροής εργασίας και ροής έγκρισης, ο κίνδυνος πριν από τον οποίο η ακεραιότητα πληροφοριών μπορεί να συμβιβαστεί - ή μετά από τις εκτελέσεις στόχου - μειώνεται πολύ [AH96b].
3. Χρονικοί περιορισμοί λαβών έτσι ώστε ο ανωτέρω συγχρονισμός της ροής έγκρισης με τη ροή της δουλειάς μπορεί να διατηρηθεί. Το Kuechler και Burg [KB98] υιοθετούν ένα πρότυπο ώθησης για να επιτύχουν αυτό. Το μοντέλο αυτό ροής εργασίας συγχρονίζει την ισχύ των εγκρίσεων με τα χρονικά διαστήματα που διευκρινίζονται για τις εκτελέσεις στόχου.
4. Εφαρμόστε τις γεγονός-βασισμένες στο εγκρίσεις. Οι στόχοι σε μια διαδικασία ροής εργασίας εξαρτώνται χαρακτηριστικά από την έκβαση των προηγούμενων στόχων. Κατά συνέπεια, η χορήγηση των εγκρίσεων εξαρτάται επίσης από τις εκβάσεις στόχου και πρέπει να είναι σύμφωνες με τις [AH96b, LR00].
5. Εφαρμόστε κάποιο είδος της ρολοκεντικής δομής ελέγχου πρόσβασης που θα απλοποιήσει την έκφραση των επιχειρησιακών κανόνων που διευκρινίζονται ως περιορισμοί [AH96b].
6. Μεταβίβαση υποστήριξης της αρχής που θα επιτρέψει στα delegators για να μεταφέρει και να ανακαλέσει τα προνόμια σε άλλα άτομα [Mof98].
7. Εφαρμόστε μια δομή υποβολής εκθέσεων για να διευκολύνετε τη επίβλεψη και την αναθεώρηση. Αυτή η δομή υποβολής εκθέσεων συνήθως σιωπηρά καθορίζεται στην οργανωτική ιεραρχία, όπου οι υφιστάμενοι υποβάλλουν έκθεση στους ανωτέρους. Κατά μια πρότυπες επίβλεψη εφαρμογή της αντιπροσωπείας και την αναθεώρηση γίνεται ιδιαίτερα σχετική, δεδομένου ότι τα delegators πρέπει να εξασφαλίσουν ότι τα delegates έχουν εκτελέσει επιτυχώς τους στόχους που ορίζονται σε τα [Mof98]. Μπορεί να συμβεί ότι οι υφιστάμενοι δεν χρειάζονται απαραίτητα την έκθεση στους άμεσους ανωτέρους τους, αλλά αυτό δεν είναι η περίπτωση σε μια οργανωτική ιεραρχία.

Η προδιαγραφή των περιορισμών διαδραματίζει έναν σημαντικό ρόλο στην επιβολή των ανωτέρω αρχών. Υπάρχουν πρώτιστα δύο τύποι περιορισμών, π.χ. στατικός ή δυναμικός [BFA99]. Οι στατικοί περιορισμοί παραμένουν συνεπείς σε όλη την εκτέλεση διαδικασίας και μπορεί έτσι να αναφερθεί ελλείψει μιας περίπτωσης διαδικασίας. Σε αντίθεση, οι δυναμικοί περιορισμοί καθορίζονται



σύμφωνα με την ιστορία της εκτέλεσης διαδικασίας, και επομένως μπορεί μόνο να ελεγχθεί στα πλαίσια μιας διαδικασίας εκτέλεσης. Ο χωρισμός των καθηκόντων και η δέσμευση των καθηκόντων είναι παραδείγματα των δυναμικών περιορισμών. Κατά συνέπεια, η δυναμική προδιαγραφή περιορισμού είναι σημαντική για την επιβολή της ασφάλειας στις ροές της δουλειάς.

### 3.4.1 Τρία επίπεδα ασφαλείας WfMS

Οι Olivier et al [ORG98] δηλώστε ότι ένα σύστημα ροής της δουλειάς πρέπει να εξεταστεί σε τρία επίπεδα από άποψη τα συστατικά του. Η εξασφάλιση μιας ροής της δουλειάς περιλαμβάνει την επιβολή των αρχών ασφαλείας καθόλου τρία επίπεδα.

Το επίπεδο 1 αποτελεί τα ελλοχεύοντα αντικείμενα στοιχείων που παραπέμπονται στους στόχους ροής της δουλειάς. Αυτό το στοιχείο πρέπει να προστατευθεί από την αναρμόδια πρόσβαση χρησιμοποιώντας τις τεχνικές όπως οι μηχανισμοί κρυπτογράφησης και ελέγχου πρόσβασης. Εάν αυτά τα αντικείμενα δεν εξασφαλίζονται, η ασφάλεια που εφαρμόζεται σε οποιαδήποτε πιο υψηλά επίπεδα υπονομεύεται.

Το επίπεδο 2 μηχανισμοί ασφαλείας εξασφαλίζουν ότι τα δικαιώματα πρόσβασης στις δραστηριότητες που χρησιμοποιούν αυτά τα αντικείμενα είναι ρυθμισμένα. Επιπλέον, αυτή η πρόσβαση πρέπει να περιοριστεί στο χρονικό διάστημα μέσα στο οποίο ένας στόχος εκτελεί.

Το επίπεδο 3 μηχανισμοί ασφαλείας εφαρμόζει τις οριζόμενες από εφαρμογή απαιτήσεις ασφαλείας. Κατά συνέπεια, αφορούν τη διευκρίνιση και την εφαρμογή των πολιτικών ασφαλείας για τα συστήματα ροής εργασίας. Στο επίπεδο 3 απαιτήσεις ασφαλείας μπορεί να αλλάξει δυναμικά για τις διάφορες περιπτώσεις ροής εργασίας [ORG98].

### 3.4.2 Διευκρίνιση επιπέδων εφαρμογών πολιτικών ασφαλείας

Αυτό το τμήμα θα εστιάσει στην εξέταση των δυναμικών περιορισμών όσον αφορά τη διευκρίνιση των πολιτικών ασφαλείας επιπέδων εφαρμογής. Οι προσεγγίσεις που συζητήθηκαν στην παράγραφο 3.3 εξέτασαν την προδιαγραφή των στατικών περιορισμών. Οι Olivier et al [ORG98] παρουσίασαν έναν αριθμό μεθόδων για το επίπεδο 3 πολιτικών ασφαλείας. Αυτές οι μέθοδοι μπορούν μόνο να εφαρμοστούν μόλις είναι μια υποδομή που υποστηρίζει το επίπεδο 1 και το επίπεδο 2 ασφαλεία σε ισχύ. Τρεις προσεγγίσεις συζητούνται: μια δυναμική μήτρα ελέγχου πρόσβασης (DACM), μια απαράδεκτη προσέγγιση ιστορίας και μια δυναμική προσέγγιση απαγόρευσης.

### 3.4.2.1 Δυναμική μήτρα ελέγχου πρόσβασης (DACM)

Η δυναμική μήτρα ελέγχου πρόσβασης διαφέρει από τη μήτρα ελέγχου πρόσβασης, δεδομένου ότι μια είσοδος στη μήτρα δείχνει ότι σε ένα θέμα μπορεί να χορηγηθεί η πρόσβαση σε ένα αντικείμενο. Κατά συνέπεια, αυτή η είσοδος δεν υπονοεί ότι το θέμα μπορεί σίγουρα να έχει πρόσβαση σε αυτό το αντικείμενο. Κάθε είσοδος είναι ένα τριπλάσιο της μορφής (s, π, α), όπου το s αντιπροσωπεύει το θέμα που μπορεί να χορηγηθεί η πρόσβαση στη δραστηριότητα α μέσα στη διαδικασία π [ORG98]. Η πρόσβαση μπορεί μόνο να χορηγηθεί σε εκείνα τα θέματα που απαριθμούνται στο DACM που διευκρινίζονται σε γενική πολιτική εφαρμόσιμη σε όλες τις περιπτώσεις διαδικασίας. Μια συγκεκριμένη πολιτική θα προκύψει σε περίπτωση συγκεκριμένης περίπτωσης διαδικασίας, όπως ένας χωρισμός του περιορισμού καθηκόντων. Αυτή η πολιτική θα οδηγήσει σε ορισμένα θέματα που αφαιρούνται από το σύνολο επιλέξιμων θεμάτων όπως διευκρινίζεται στο DACM. Το μεμονωμένο θέμα που είναι να εκτελεσθεί ο στόχος επιλέγεται έπειτα από το καθαρισμένο σύνολο όπως καθορίζεται στο DACM. Ακόμα κι αν η προσέγγιση DACM είναι πολύ διαισθητική, προκαλεί τα υπερβολικά γενικά έξοδα χρόνου εκτέλεσης, ο οποίος το κάνει μια λιγότερο επιθυμητή επιλογή από τις προσεγγίσεις κατωτέρω [ORG98].

### 3.4.2.2 Μη Αποδεκτή Ιστορική Προσέγγιση

Αυτή η προσέγγιση ενημερώνει ένα αρχείο των προηγούμενων γεγονότων που εμφανίζονται στο σύστημα ροής εργασίας σε έναν κατάλογο ιστορίας. Αυτός ο κατάλογος ιστορίας διατηρεί το αρχείο των προηγούμενων εγκρίσεων, και οι πολιτικές είναι βασισμένες στις πληροφορίες που κρατιούνται σε αυτό. Κάθε είσοδος στον κατάλογο ιστορίας είναι της μορφής (s, π, α), δείχνοντας ότι στο θέμα s χορηγήθηκε η έγκριση για να εκτελέσει τη δραστηριότητα α στην περίπτωση π διαδικασίας [ORG98]. Οι καταχωρήσεις σε αυτόν τον κατάλογο ιστορίας μπορούν έπειτα να χρησιμοποιηθούν για να διευκρινίσουν τους δυναμικούς περιορισμούς και να απαγορεύσουν έτσι άλλες εγκρίσεις. Αυτό γίνεται με την απαγόρευση του περιστατικού ορισμένων συνδυασμών καταχωρήσεων. Αν και αυτή η προσέγγιση είναι πολύ ισχυρή και συνοπτική, οι συγκρούσεις περιορισμού ανιχνεύονται μόνο μόλις είναι στον κατάλογο ιστορίας και έτσι έχουν εμφανιστεί ήδη. Για αυτόν τον λόγο, η προδιαγραφή περιορισμού που χρησιμοποιεί την μη αποδεκτή προσέγγιση ιστορίας μπορεί να είναι πολύ περίπλοκη [ORG98].

### 3.4.2.3 Δυναμική προσέγγιση απαγορεύσεων

Η δυναμική προσέγγιση απαγορεύσεων χρησιμοποιεί ένα στατικό θετικό καθορισμένο που περιέχει τις εγκεκριμένες εγκρίσεις. Αυτό το θετικό σύνολο μπορεί να μοιάσει με τις καταχωρήσεις DACM. Ένα αρνητικό σύνολο περιέχει όλες τις απαγορεύσεις σχετικά με τις συγκεκριμένες περιπτώσεις διαδικασίας και

τροποποιείται δυναμικά αναλόγως. Οι εγκρίσεις καθορίζονται έπειτα από τη διαφορά μεταξύ των δύο συνόλων. Οι Olivier et all [ORG98] διευκρίνισαν αυτά τα σύνολα χρησιμοποιώντας την ακόλουθη σημείωση:

Θετική:  $P(\text{SUBJ} \times \text{ACT})$

Αρνητική:  $P(\text{SUBJ} \times \text{PROC} \times \text{ACT})$

Πιθανή:  $= \{8p : \text{PROC}; s : \text{SUBJ}; a : \text{ACT} \mid (s, a) \in 2 \text{ positive} \bullet (s, p, a)\} \setminus \text{negative}$

Από τους ανωτέρω ορισμούς, διάφορες παρατηρήσεις μπορούν να γίνουν. Αρχικά, το θετικό περιέχει γενικές πολιτικές που εφαρμόζονται σε όλες τις διαδικασίες και παραμένει στατικό σε όλες τις εκτελέσεις διαδικασίας. Ο αρνητικός καθορίζει τις απαγορεύσεις για ορισμένες δραστηριότητες σε μια διαδικασία και αγνοεί οποιοσδήποτε προηγούμενος καθορισμένες θετικές εγκρίσεις για να διαμορφωθεί το σύνολο πιθανών εγκρίσεων.

Η δυναμική προσέγγιση απαγορεύσεων πάσχει από το μειονέκτημα ότι είναι διαδικαστικής φύσης. Εντούτοις, έχει χαμηλότερα γενικά έξοδα χρόνου εκτέλεσης από τη μήτρα DACM επειδή το συσσωρευτικό διάστημα που απαιτείται για όλα τα σύνολα είναι πολύς λιγότερο από αυτό που απαιτείται για το DACM.

Οι Rabitti et all [RBKW91] συζήτησαν τη δυνατότητα που οι αρνητικές εγκρίσεις έχουν σε ένα πλαίσιο ασφάλειας, ιδιαίτερα ασφάλεια βάσεων δεδομένων. Για τους σκοπούς αυτής της εργασίας, εντούτοις, μια βασική γνώση των αρνητικών εγκρίσεων είναι ικανοποιητική.

Οι Olivier et all [ORG98] πρότειναν ότι το επίπεδο που 3 πολιτικές ασφάλειας πρέπει να είναι προσδιορισμένο με τη χρησιμοποίηση της απαράδεκτης προσέγγισης ιστορίας, και έπειτα τροποποιημένο στις δυναμικές απαγορεύσεις. Αυτό θα επιτρέψει στην πολιτική για να διευκρινιστεί με έναν συνοπτικό τρόπο, ταυτόχρονα διατηρώντας τα χαμηλά γενικά έξοδα. Ένα περιεκτικό πλαίσιο για την ασφάλεια και στα τρία επίπεδα συζητείται από τους Gudes et all [GOv99].

### 3.5 Συμπέρασμα

Αυτό το κεφάλαιο έδωσε μια επισκόπηση των εννοιών ασφάλειας σχετικών με τα συστήματα ροής εργασίας. Η υπηρεσία ασφάλειας του ελέγχου πρόσβασης εισήχθη μαζί με τέσσερις άλλους (προσδιορισμός και επικύρωση, εμπιστευτικότητα, ακεραιότητα, non-repudiation) για να εξηγήσει το ρόλο που διαδραματίζει στη συνολική προστασία συστημάτων.

Τρεις προσεγγίσεις ελέγχου πρόσβασης στη διευκρίνιση των πολιτικών ασφάλειας συζητήθηκαν, δηλαδή, υποχρεωτικός έλεγχος πρόσβασης, διακριτικός έλεγχος πρόσβασης, και ρόλοκεντρικός έλεγχος πρόσβασης. Ευρέως γίνεται αποδεκτό ότι τα πρότυπα RBAC είναι τα αποτελεσματικότερα στην επιβολή της

ασφάλειας ροής της δουλειάς. Συνεπώς, τα περισσότερα υπάρχοντα πρότυπα είναι βασισμένα στις αρχές του [AH96, HA99, SCFY96, BFA98].

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΡΡΑΙΑ

## Κεφάλαιο 4 – Τρέχων Παράδειγμα

### Εισαγωγή

#### 4.1 Περιγραφή Συστήματος

Το σύστημα στο οποίο αναφερόμαστε, είναι βασισμένο και σχεδιασμένο να λειτουργεί μέσα σε ροές εργασίας. Πρέπει να σημειωθεί πώς λειτουργεί με μεγάλη επιτυχία στη διαχείριση λογαριασμών οφειλετών τραπεζών και για το λόγο αυτό η αναφορά στον κεντρικό κορμό του συστήματος θα αποφευχθεί. Θα αναφερθούμε στη βασική δομή εισαγωγής μιας νέας συνεργασίας στο σύστημα, με υποθετικούς λογαριασμούς και αναθέσεις στοιχείων.

Είναι ένα σύστημα που στηρίζεται σε ροές δεδομένων και η πρόσβαση σε αυτό γίνεται μέσω του διαδικτύου. Οι βασικές προϋποθέσεις για τη λειτουργία του συστήματος είναι η ύπαρξη ενός διακομιστή για την εγκατάσταση της βάσης δεδομένων, ένας application server, η ύπαρξη δικτύου καθώς και υπολογιστές που θα έχουν πρόσβαση στην εφαρμογή.

#### 4.2 Βασική δομή Συστήματος

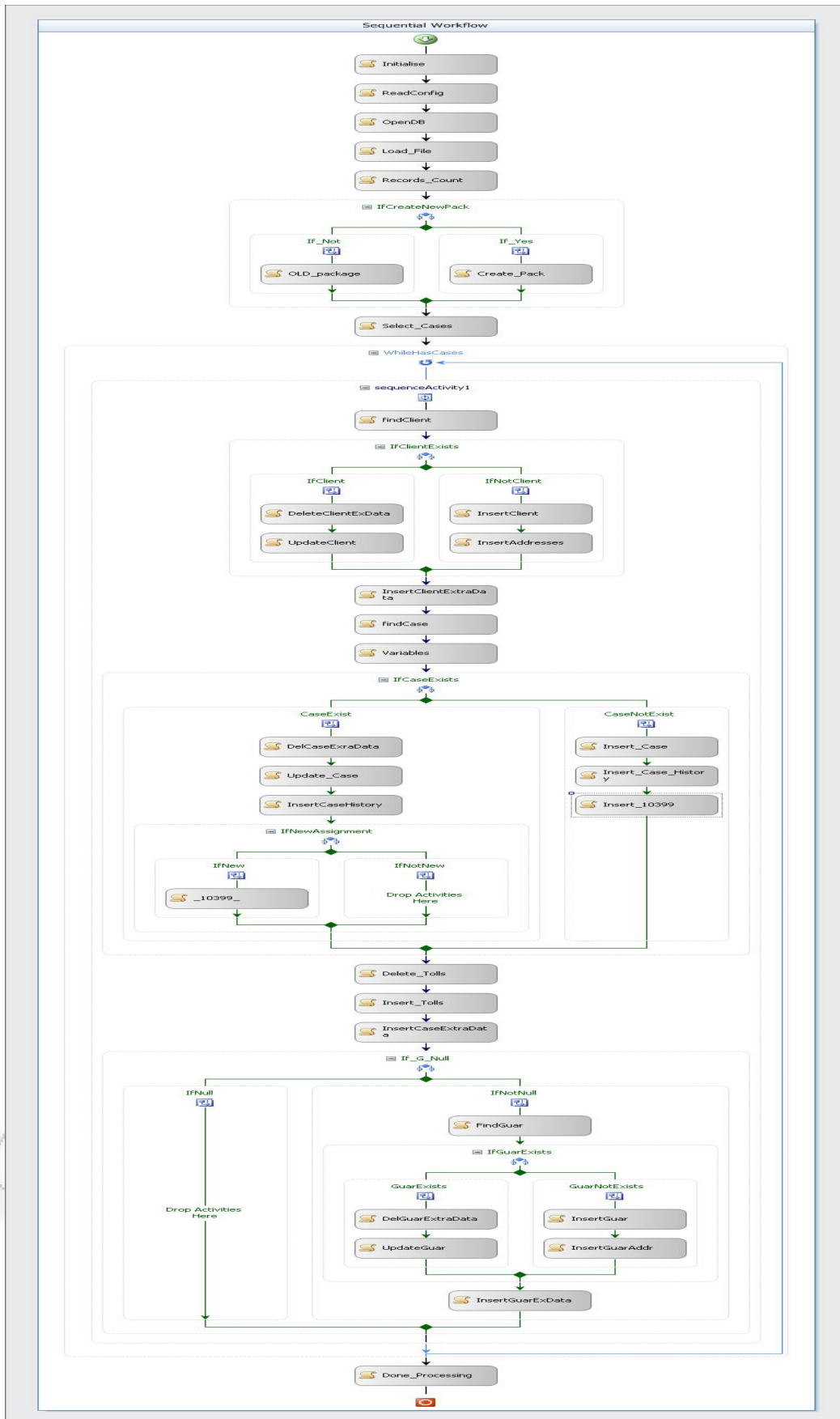
Το σύστημα χωρίζεται σε 5 βασικές δομές. Στον βασικό κορμό που υποστηρίζει την αλληλεπίδραση των υπολοίπων 4 δομών, στο σύστημα εισαγωγής στοιχείων, στο σύστημα εξαγωγής αναφορών που διαφοροποιούνται ανάλογα με τη συνεργασία, στο σύστημα αυτόματης διαχείρισης κινήσεων (production) και στο σύστημα διαχείρισης κινήσεων λογαριασμών (StateMachine).

Για την καλύτερη κατανόηση, θα αναλυθεί η κάθε μία δομή αναλυτικά.

##### 4.2.1 Εισαγωγή Στοιχείων – Loading Workflow

Η εισαγωγή στοιχείων στο σύστημα γίνεται μέσω ενός συστήματος ροής. Αναγκαίο είναι το αρχείο ανάθεσης, στην συγκεκριμένη περίπτωση το θεωρούμε μορφής text.

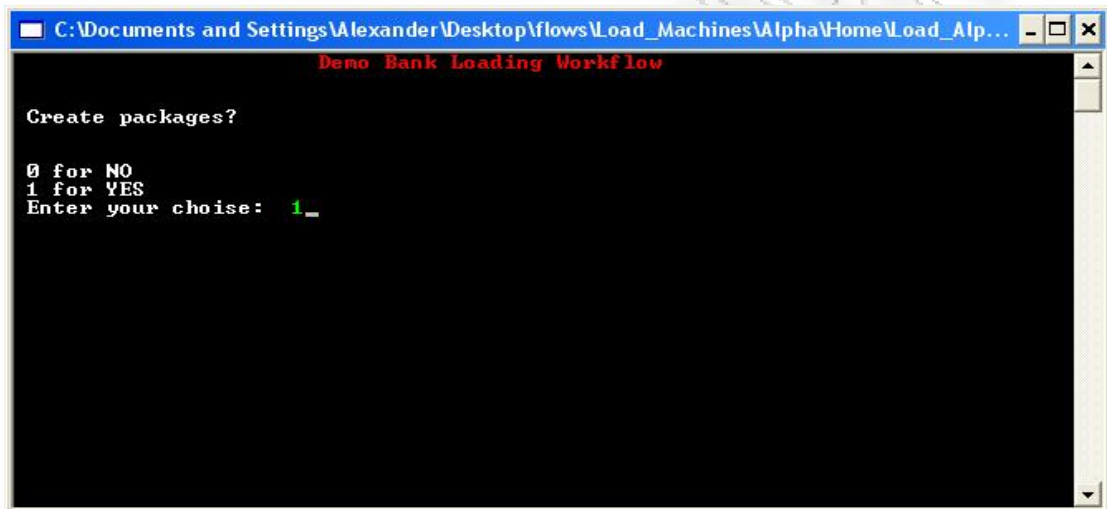
Όπως φαίνεται και ην εικόνα 5, τα βασικά στοιχεία του συστήματος είναι τα εξής. Τα βασικά βήματα σύνδεσης στη βάση, το φόρτωμα των αρχείων σε έναν temporary πίνακα, η μορφή του οποίου τροποποιείται ανάλογα με το αρχείο ανάθεσης, η εισαγωγή της κάθε γραμμής του αρχείου στο σύστημα και τέλος αποσύνδεση με τη βάση.



Εικόνα 5 - Loading Workflow

Στα πρώτα βήματα, γίνεται η σύνδεση με τη βάση. Εν συνεχεία φορτώνονται τα δεδομένα σε έναν temporary πίνακα. Εκεί δημιουργούνται και κάποια αρχεία που κρατούν πληροφορία για το αν όλα τα δεδομένα φορτώθηκαν επιτυχώς στον βοηθητικό πίνακα ή υπήρχε κάποιο πρόβλημα. Κατά την έναρξη της διαδικασίας, ο χειριστής επιλέγει εάν θα δημιουργηθεί νέο πακέτο δεδομένων, ή αν θα χρησιμοποιηθεί ένα ήδη υπάρχον.

Εν συνεχεία, προσμετράται το σύνολο των γραμμών του βοηθητικού πίνακα και ανάλογα με την επιλογή του χειριστή για τα πακέτα, είτε δημιουργείται νέο είτε χρησιμοποιείται κάποιο υπάρχον (εικόνα 6).



Εικόνα 6 - Επιλογή Δημιουργίας Πακέτων

Το επόμενο βήμα είναι να επιλεγθούν όλα τα δεδομένα από τον βοηθητικό πίνακα, ώστε να φορτωθούν σε ένα αναγνώστη (reader).

```
Private Sub Select_Cases_ExecuteCode(ByVal sender As System.Object, ByVal e As System.EventArgs)
    Show_Status("Selecting Cases... ")
    Dim Sql As String = " SELECT * FROM MCS_LOAD.Load_Demo_Bank_CASES "
    Dim Com As OracleCommand = New OracleCommand(Sql.ToUpper, ConLoad)
    oReader = Com.ExecuteReader()
End Sub
```

Για κάθε μία γραμμή που διαβάζεται από τον βοηθητικό πίνακα, τα δεδομένα της φορτώνονται στον αναγνώστη. Το πρώτο βήμα που γίνεται είναι να γίνει έλεγχος αν ο οφειλέτης υπάρχει ήδη στη βάση, ώστε να γίνει ενημέρωση των στοιχείων του. Αν δεν υπάρχει, τότε τα στοιχεία του προστίθενται στους ανάλογους πίνακες.

Στη συνέχεια, ελέγχεται αν η ίδια υπόθεση υπάρχει στο σύστημα. Αν υπάρχει ενημερώνονται τα στοιχεία της, διαφορετικά δημιουργείται μία νέα εγγραφή στον ανάλογο πίνακα.

Η ίδια διαδικασία ακολουθείται, μέχρι και την τελευταία εγγραφή στο βοηθητικό πίνακα. Όταν πλέον όλα τα δεδομένα έχουν περαστεί στους κυρίως πίνακες του συστήματος, η όλη διαδικασία εισαγωγής τερματίζεται.

Στο παράρτημα θα μπορείτε να βρείτε και τον κώδικα της όλης διαδικασίας, καθώς και λεπτομέρειες για τους κυρίως πίνακες δεδομένων της εφαρμογής.

#### 4.2.2 Σύστημα Αυτόματης Διαχείρισης Κινήσεων – Production Workflow

Οι χειριστές, κατά την διαχείριση κάθε λογαριασμού, εισάγουν κάποιες κινήσεις (actions). Ανάλογα με τις κινήσεις αυτές, κατά την εισαγωγή τους, τρέχει ένα σύστημα ροής εργασίας και τοποθετεί τους λογαριασμούς σε διάφορους κουβάδες (queues). Η διαχείριση αυτή ορίζεται δυναμικά από συνεργασία σε συνεργασία, ανάλογα με τον τρόπο που ζητείτε από τον κάθε πελάτη.

Παρατίθεται στο παράρτημα ένα σκαρίφιμα της διαχείρισης για την συνεργασία που παρουσιάζεται στην εργασία.

Στην εικόνα 7 παρατηρείτε το Σύστημα αυτόματης Διαχείρισης Κινήσεων. Χωρίζεται στα εξής βασικά βήματα. Είναι ένα sequential workflow library. Η χρησιμότητά του είναι ώστε κατά την διαχείριση υποθέσεων σε ένα CRM τραπεζικό σύστημα, να μετακινεί τις διαχειρίσιμες υποθέσεις στα ανάλογα πακέτα – queues.

Τα βήματα που ακολουθούνται έχουν ως εξής.

Ελέγχεται αν ο χρήστης κάνει κίνηση διαγραφής δεδομένων, οπότε και ενημερώνεται κατάλληλα η υποστηρίζουσα βάση. (CodeBoxes **DeleteContact** και **SetOkDeleteStatus**).

Εν συνεχεία, εξετάζεται αυτόματα, αν οι μέχρι τώρα λειτουργίες είναι εντάξει ή εάν έχει δημιουργηθεί κανένα σφάλμα. Εάν υπάρχει σφάλμα, τερματίζεται η σύνδεση με τη βάση και σταματά η διαχείριση.

Στην περίπτωση που δεν υπάρχει σφάλμα, ο χειριστής έχει τη δυνατότητα να συνεχίσει τη διαχείριση. Μπορεί να προσθέσει, να αφαιρέσει και να τροποποιήσει δεδομένα.

Μετά από αυτές τις κινήσεις διαχείρισης, γίνεται μια ενημέρωση στη βάση, σχετικά με τον τελευταίο ιεραρχικά action code (κίνηση), καθώς και τον τελευταίο χειριστή της υπόθεσης.

Ένα πολύ σημαντικό σημείο του συνολικού workflow βρίσκεται στο Location Rules.

Εκεί υπάρχουν διάφορα Code Boxes, το καθένα από το οποίο οδηγεί την διεχειρίζουσα υπόθεση στον ανάλογο κουβά – queue, ανάλογα με την κίνηση που έχει γίνει.





Πιο συγκεκριμένα, το codeBox **\_NonWorking2** περιέχει τον παρακάτω κώδικα:

```
Private Sub _NonWorking2_ExecuteCode(ByVal sender As System.Object, ByVal e As
System.EventArgs)
    Dim InString As String = "10401,77777,60400"
    If LOCATION_ID = 0 Then
        If InStr(InString, LAST_ACTION_CODE.ToString) > 0 Then
            LOCATION_ID = 180
            LOCATION_DESC = "NON WORKING 2 QUEUE"
        End If
    End If
End Sub
```

Εάν η κίνηση του χειριστή ανήκει στα συγκεκριμένα action\_id, π.χ. 10401, τότε η διαχειριζόμενη υπόθεση μεταφέρεται στο queue “NON WORKING 2 QUEUE”, το οποίο έχει Location\_id 180, στον πίνακα location της υποστηρίζουσας βάσης του συστήματος.

Αξίζει να σημειωθεί πως είναι αναγκαίο να οριστεί ένα αρχικό Location\_id, πριν την έναρξη της διαχείρισης των όποιων υποθέσεων. Επίσης, τελευταίο από όλα τα codeboxes, παραμένει πάντα το CheckWorking, που σύμφωνα με το σχεδιασμό του οικείου συστήματος, όποιος χειρισμός δεν υθάγεται στους αναλυόμενους στο workflow, οδηγείται στο Working queue.

Τέλος, μετά το πέρας της διαχείρισης των υποθέσεων που θέλουμε να διαχειριστούμε, αναγκαία είναι και η ενημέρωση της βάσης.

```
Private Sub UpdateCase_ExecuteCode(ByVal sender As System.Object, ByVal e As
System.EventArgs)

    Dim Sql As String = " UPDATE CASES "
    Sql = Sql & " SET LAST_CONTACT_ID = " & LAST_CONTACT_ID & ", "
    Sql = Sql & " LAST_STRONG_CONTACT_ID = " & LAST_STRONG_CONTACT_ID &
    ", "
    Sql = Sql & " LAST_ACTION_CODE = " & LAST_ACTION_CODE & ", "
    Sql = Sql & " LAST_STRONG_ACTION_CODE = " & LAST_STRONG_ACTION_CODE
    & ", "
    Sql = Sql & " CASE_BEHAVIOR_SCORE = " & CASE_BEHAVIOR_SCORE & ", "
    Sql = Sql & " LOCATION_DESC = '" & LOCATION_DESC & "', "
    Sql = Sql & " LOCATION_ID = " & LOCATION_ID & ", "
    Sql = Sql & " LAST_CONTACT_DATE = SYSDATE "
    Sql = Sql & " WHERE CASE_ID = " & _Case_ID & " AND CUSTOMER_ID = " &
    _Customer_ID

    Dim Com As OracleCommand = New OracleCommand(Sql.ToUpper, Con)
    Com.ExecuteNonQuery()
End Sub
```

Εκεί ενημερώνουμε την βάση για την τελευταία κίνηση κάθε λογαριασμού, το τελευταίο queue στο οποίο έχει μεταφερθεί, ώστε αν χρειαστεί να επαναδιαχειριστεί, καθώς και με τις τελευταίες ισχυρές κινήσεις, σύμφωνα με την ορισμένη από εμάς ιεράρχιση κινήσεων.

### 4.2.3 Σύστημα Διαχείρισης Κινήσεων Λογαριασμών (State Machine)

Αυτό το σύστημα ροής εργασίας είναι προγραμματισμένο να λειτουργεί σε μη παραγωγική ώρα. Η ουσιαστική λειτουργικότητά του είναι να ελέγχει και να μεταφέρει τις υποθέσεις στο σωστό queue, σύμφωνα με τις προσυμφωνημένες συνθήκες.

Παρατίθεται στο παράρτημα ένα σκαρίφιμα της διαχείρισης για την συνεργασία που παρουσιάζεται στην εργασία.

Στην εικόνα 8 προβάλλετε το Σύστημα Διαχείρισης Κινήσεων. Χωρίζεται στα εξής βασικά βήματα. Είναι ένα sequential workflow library.

Η χρησιμότητά του είναι ώστε μετά την διαχείριση υποθέσεων σε ένα CRM τραπεζικό σύστημα, να μετακινεί τις διαχειρίσιμες υποθέσεις στα ανάλογα πακέτα – queues, ανάλογα με τις συνθήκες που έχουν οριστεί.

Παρά το ότι το Σύστημα Αυτόματης Διαχείρισης Κινήσεως τοποθετεί τους λογαριασμούς σε διάφορα queue, αυτό το σύστημα διαχείρισης λαμβάνει υπόψην του και παράγοντες όπως την ημερομηνία επόμενης επαφής και το ποσό που έχει καταβληθεί από τον οφειλέτη, ώστε να υπάρχει μια πιο σαφής ενημέρωση με τη διαχείριση.

Θα αναληθούν δύο παραδείγματα για την καλύτερη κατανόηση του θέματος αυτού.

Σύμφωνα με το κώδικα που υπάρχει πιο κάτω, με την εισαγωγή κίνησης 10500 ή 10501, γίνεται ένας έλεγχος για το αν η ημερομηνία επαφής είναι μεγαλύτερη από τη σημερινή. Αν ισχύει αυτό τότε ο λογαριασμός τοποθετείται στο Rest Queue ενώ διαφορετικά θα τοποθετούνταν στο No Contact Queue.

```
Private Sub CheckNoContact_ExecuteCode(ByVal sender As System.Object, ByVal e As System.EventArgs)
    Dim InString As String = "10500,10501"
    If LOCATION_ID = 0 Then
        If InStr(InString, LAST_ACTION_CODE.ToString) > 0 Then
            If NEXT_CONTACT_DATE <= Now.Date Then
                LOCATION_ID = 178
                LOCATION_DESC = "NO CONTACT QUEUE"
            Else
                LOCATION_ID = 174
                LOCATION_DESC = "REST QUEUE"
            End If
        End If
    End If
End Sub
```

Εν συνεχεία, έστω ότι ο χειριστής έχει εισάγει κίνηση υπόσχεσης στο σύστημα (10105). Όταν λειτουργεί το σύστημα ροής εργασίας, γίνεται έλεγχος αν η ημερομηνία που έχει εισαχθεί σαν εκπλήρωση της υπόσχεσης έχει παρέλθει. Αν έχει παρέλθει τότε ο λογαριασμός τοποθετείται στο Broken Promise Queue, διαφορετικά παραμένει στο Promise Queue.

```
Private Sub CheckPromise_ExecuteCode(ByVal sender As System.Object, ByVal e As
System.EventArgs)
    Dim InString As String = "10105"
    If LOCATION_ID = 0 Then
        If InStr(InString, LAST_ACTION_CODE.ToString) > 0 Then
            If Now.Date >= PAY_DATE.AddDays(+1) Then
                LOCATION_ID = 179
                LOCATION_DESC = "BROKEN PROMISE"
                CASE_BEHAVIOR_SCORE = 3
            Else
                LOCATION_ID = 176
                LOCATION_DESC = "PROMISE QUEUE"
            End If
        End If
    End If
End Sub
```



#### 4.2.4 Σύστημα Εξαγωγής Αναφορών

Ανάλογα με την κάθε συνεργασία, οι αναφορές που μπορούν να εξαχθούν από τα στοιχεία του συστήματος διαφοροποιούνται. Στην συγκεκριμένη συνεργασία, υπήρχε ζήτηση δημιουργίας καθημερινής αναφοράς κινήσεων που εισάγουν οι χειριστές. Μάλιστα, υπήρχε απαίτηση η αναφορά αυτή να εξαγεται με περιορισμό στις ημέρες καθυστέρησης που έχει ο κάθε λογαριασμός την ημέρα της ανάθεσης του λογαριασμού από τον χρηματοπιστωτικό οργανισμό στην εισπρακτική.

Οι διαδικασίες που ακολουθούνται στην συγκεκριμένη περίπτωση είναι οι εξής. Δημιουργείται μια σύνδεση στη βάση, από όπου θα αντληθούν τα δεδομένα για τη δημιουργία της αναφοράς. Εν συνεχεία, γίνεται καταμέτρηση των κινήσεων που έχουν εισαχθεί την ημέρα, για την οποία ζητείται η αναφορά. Στο σημείο αυτό θα πρέπει να αναφερθεί, πως η ημέρα δίνεται από το ανάλογο χειριστή που πάει να λειτουργήσει το σύστημα αυτό (εικόνα 9).

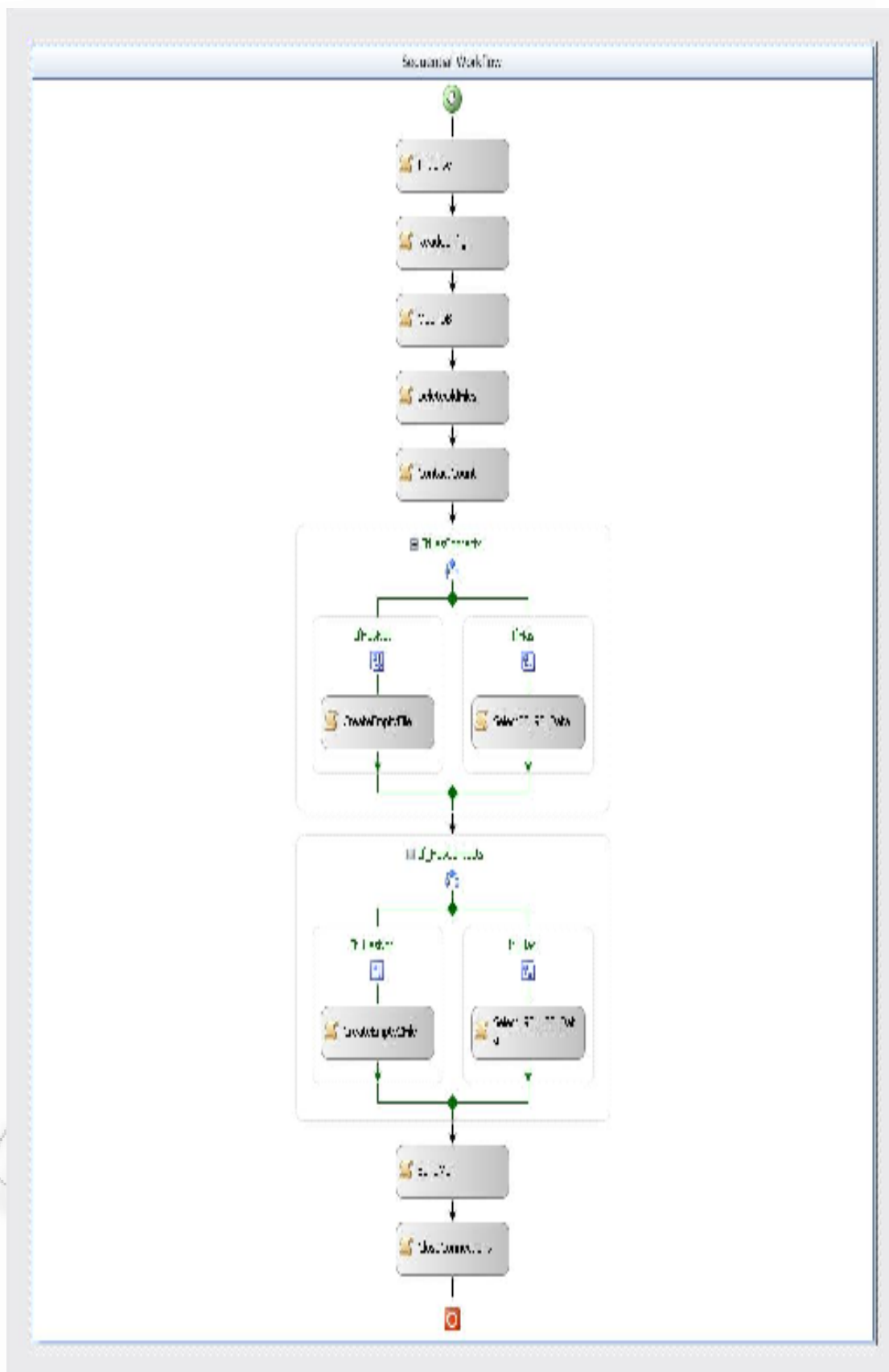


Εικόνα 9 Επιλογή ημερημνίας

Εν συνεχεία δημιουργούνται τα 2 αρχεία τύπου csv για τις δύο ζητούμενες περιοχές καθυστέρησης καταβολής. Στην περίπτωση που δεν υπάρχει κάποιο δημιουργείται κενό αρχείο.

Όταν ολοκληρωθεί η δημιουργία των αρχείων, αυτόματα αποστέλονται με ηλεκτρονικό ταχυδρομείο στον ανάλογο συνεργάτη και κλείνει η σύνδεση με τη βάση.

Παρατίθετε πιο κάτω το Σύστημα Ροής Εργασίας δημιουργίας Αναφοράς



Εικόνα 10 - Workflow Αναφορών



Το αποστέλλόμενο αρχείο αναφοράς είναι σε μορφή csv. Στην εικόνα 11 παρουσιάζεται και η μορφή του

|    | A                                | B | C                                    | D    |
|----|----------------------------------|---|--------------------------------------|------|
| 2  | ΚΑΘΗΜΕΡΕ ΤΡΕΤΩΝ-                 |   | ΑΓΙΟΥΣ ΕΒΔΟΜΗ ΠΡΟΣ ΤΟ-ΡΟΛΗΣ-ΥΠΟΣΦΕ-Σ |      |
| 3  | ΥΠΕΡ ΤΡΟΣ ΕΠΟΤΕΣ-ΣΟΤΙΣ ΕΠΙΧΟΝΟΙΑ |   | 1                                    | 10,0 |
| 4  | ΥΠΕΡ ΤΡΟΣ ΕΠΟΤΕΣ-ΣΟΤΙΣ ΕΠΙΧΟΝΟΙΑ |   | 1                                    |      |
| 5  | ΥΠΕΡ ΤΡΟΣ ΕΠΟΤΕΣ-ΣΟΤΙΣ ΕΠΙΧΟΝΟΙΑ |   | 1                                    |      |
| 6  | ΥΠΕΡ ΤΡΟΣ ΕΠΟΤΕΣ-ΣΟΤΙΣ ΕΠΙΧΟΝΟΙΑ |   | 1                                    |      |
| 7  | ΥΠΕΡ ΤΡΟΣ ΕΠΟΤΕΣ-ΣΟΤΙΣ ΕΠΙΧΟΝΟΙΑ |   | 1                                    |      |
| 8  | ΥΠΕΡ ΤΡΟΣ ΕΠΟΤΕΣ-ΣΟΤΙΣ ΕΠΙΧΟΝΟΙΑ |   | 1                                    |      |
| 9  | ΥΠΕΡ ΤΡΟΣ ΕΠΟΤΕΣ-ΣΟΤΙΣ ΕΠΙΧΟΝΟΙΑ |   | 1                                    |      |
| 10 | ΥΠΕΡ ΤΡΟΣ ΕΠΟΤΕΣ-ΣΟΤΙΣ ΕΠΙΧΟΝΟΙΑ |   | 1                                    |      |
| 11 | ΥΠΕΡ ΤΡΟΣ ΕΠΟΤΕΣ-ΣΟΤΙΣ ΕΠΙΧΟΝΟΙΑ |   | 1                                    |      |
| 12 | ΥΠΕΡ ΤΡΟΣ ΕΠΟΤΕΣ-ΣΟΤΙΣ ΕΠΙΧΟΝΟΙΑ |   | 1                                    |      |
| 13 | ΥΠΕΡ ΤΡΟΣ ΕΠΟΤΕΣ-ΣΟΤΙΣ ΕΠΙΧΟΝΟΙΑ |   | 1                                    |      |
| 14 | ΥΠΕΡ ΤΡΟΣ ΕΠΟΤΕΣ-ΣΟΤΙΣ ΕΠΙΧΟΝΟΙΑ |   | 1                                    |      |
| 15 | ΥΠΕΡ ΤΡΟΣ ΕΠΟΤΕΣ-ΣΟΤΙΣ ΕΠΙΧΟΝΟΙΑ |   | 1                                    |      |
| 16 | ΥΠΕΡ ΤΡΟΣ ΕΠΟΤΕΣ-ΣΟΤΙΣ ΕΠΙΧΟΝΟΙΑ |   | 1                                    |      |
| 17 | ΥΠΕΡ ΤΡΟΣ ΕΠΟΤΕΣ-ΣΟΤΙΣ ΕΠΙΧΟΝΟΙΑ |   | 1                                    |      |
| 18 | ΥΠΕΡ ΤΡΟΣ ΕΠΟΤΕΣ-ΣΟΤΙΣ ΕΠΙΧΟΝΟΙΑ |   | 1                                    |      |
| 19 | ΥΠΕΡ ΤΡΟΣ ΕΠΟΤΕΣ-ΣΟΤΙΣ ΕΠΙΧΟΝΟΙΑ |   | 1                                    |      |
| 20 | ΥΠΕΡ ΤΡΟΣ ΕΠΟΤΕΣ-ΣΟΤΙΣ ΕΠΙΧΟΝΟΙΑ |   | 1                                    |      |
| 21 | ΥΠΕΡ ΤΡΟΣ ΕΠΟΤΕΣ-ΣΟΤΙΣ ΕΠΙΧΟΝΟΙΑ |   | 1                                    |      |
| 22 | ΥΠΕΡ ΤΡΟΣ ΕΠΟΤΕΣ-ΣΟΤΙΣ ΕΠΙΧΟΝΟΙΑ |   | 1                                    |      |
| 23 | ΥΠΕΡ ΤΡΟΣ ΕΠΟΤΕΣ-ΣΟΤΙΣ ΕΠΙΧΟΝΟΙΑ |   | 1                                    |      |
| 24 | ΥΠΕΡ ΤΡΟΣ ΕΠΟΤΕΣ-ΣΟΤΙΣ ΕΠΙΧΟΝΟΙΑ |   | 1                                    |      |
| 25 | ΥΠΕΡ ΤΡΟΣ ΕΠΟΤΕΣ-ΣΟΤΙΣ ΕΠΙΧΟΝΟΙΑ |   | 1                                    |      |

Εικόνα 11 Μορφή Αρχείου Αναφοράς







### 4.3 Υποστηρίζουσα Βάση Συστήματος

Λόγω του μεγάλου όγκου δεδομένων που έχει πρόσβαση του σύστημα και για το λόγο ότι πρέπει να είναι ιδιαίτερα σταθερό, προτιμήθηκε η βάση που θα υποστηρίζει να είναι η Oracle Database 10g Enterprise Edition (Oracle 10.2.0.1.9P). Οι κύριοι πίνακες που υποστηρίζουν το όλο σύστημα είναι 18. Αναλύονται παρακάτω.

#### 4.3.1 Πίνακας Customers

Ο συγκεκριμένος πίνακας κρατάει την πληροφορία για την ονομασία των πελατών – συνεργασιών που υπάρχουν μέσα στο σύστημα. Ο κάθε πελάτης έχει το δικό του ID.












| MCS.CUSTOMERS   |                   |                |  |
|---|-------------------|----------------|--|
|  | CUSTOMER_ID       | NUMBER         |  IDX_1 |
|   | INST_ID           | NUMBER         |        |
|   | CUST_TYPE_ID      | NUMBER         |        |
|   | CUSTOMER_NAME     | NVARCHAR2 (50) |        |
|   | START_DATE        | DATE           |  |
|   | ORDER_BY_PRIORITY | NUMBER (1)     |        |

Εικόνα 12 - Πίνακας Customers

### 4.3.2 Πίνακας Packs










Ο πίνακας αυτός κρατάει πληροφορία για τα πακέτα που δημιουργούνται για κάθε συνεργασία. Έχει δεδομένα για το είδος των λογαριασμών που περιέχονται στο πακέτο, για το περιεχόμενο προϊόν καθώς και τον πελάτη για τον οποίο είναι το πακέτο.

| MCS.PACKS   |                   |                |  |
|---|-------------------|----------------|--|
|  | PACK_ID           | NUMBER         |  IDX_3       |
|   | INST_ID           | NUMBER         |              |
|  | CUSTOMER_ID       | NUMBER         |  IDX_4,IDX_3 |
|   | PACK_DESC         | NVARCHAR2 (50) |              |
|   | PACK_TYPE_ID      | NUMBER         |  IDX_4       |
|   | PRODUCT_TYPE_ID   | NUMBER         |  IDX_4       |
|   | ACTIVE_CASES      | NUMBER         |  |
|   | ACTIVE_AMOUNT     | NUMBER         |  |
|   | IS_ACTIVE         | NUMBER         |  IDX_1,IDX_4 |
|   | BILLING_DATE      | DATE           |  |
|   | BILLING_DESC      | NVARCHAR2 (50) |  |
|   | START_DATE        | DATE           | IDX_2  |
|   | END_DATE          | DATE           |  |
|   | WORKFLOW_ID       | NUMBER         |  |
|   | DEACTIVATION_DATE | DATE           |  |

Εικόνα 13 - Πίνακας Packs

### 4.3.3 Πίνακας Client

Ο πίνακας Client περιέχει πληροφορίες για τους οφειλέτες. Κρατάει στοιχεία δημογραφικά.










| MCS.CLIENT  |                     |                 |   |
|---|---------------------|-----------------|---|
|  | CLIENT_ID           | NUMBER          |  IDX_1       |
|  | CUSTOMER_ID         | NUMBER          |  IDX_1,IDX_7 |
|  | INST_ID             | NUMBER          |  IDX_2       |
|   | CLIENT_SEC_ID       | NVARCHAR2 (50)  |  IDX_5       |
|   | SURNAME             | NVARCHAR2 (100) |  IDX_6       |
|   | FIRSTNAME           | NVARCHAR2 (50)  | IDX_6   |
|   | FATHERNAME          | NVARCHAR2 (50)  | IDX_6   |
|   | IDENTITYNO          | NVARCHAR2 (30)  | IDX_4   |
|   | TAX_NO              | NVARCHAR2 (30)  | IDX_3   |
|   | BEHAVIOR_SCORE      | NVARCHAR2 (10)  |   |
|   | CLIENT_EXTRA_DATA   | NUMBER (1)      |              |
|   | ID_ISSUE_DATE       | NVARCHAR2 (40)  |   |
|   | ID_ISSUE_REGON      | NVARCHAR2 (50)  |   |
|   | TIME_STAMP          | DATE            |   |
|   | IS_ORGANIZATION     | NVARCHAR2 (2)   |   |
|   | MIGRATION_CLIENT_ID | NVARCHAR2 (50)  | IDX_8   |

Εικόνα 14 - Πίνακας Client

#### 4.3.3.1 Πίνακες Client\_Addresses, Client\_Phones και Client\_Ex\_Data










Τον πίνακα Client ολοκληρώνουν σαν όγκο πληροφορίας οι πίνακες Client\_Addresses, Client\_Phones και Client\_Ex\_Data.

Ο πίνακας Client\_Addresses έχει πληροφορίες για τη διεύθυνση του οφειλέτη.

| MCS.CLIENT_ADDRESSES  |                 |                 |  |
|---|-----------------|-----------------|--|
|  | ADDRESS_ID      | NUMBER          |  IDX_8       |
|  | CUSTOMER_ID     | NUMBER          |  IDX_8,IDX_7 |
|  | CLIENT_ID       | NUMBER          |  IDX_6       |
|   | ADDRESS_TYPE_ID | NUMBER          |  IDX_5       |
|   | ADDRESS         | NVARCHAR2 (100) |  IDX_1       |
|   | TOWN            | NVARCHAR2 (60)  | IDX_2  |
|   | ZIP             | NVARCHAR2 (30)  | IDX_3  |
|   | MALLING_ADDRESS | NUMBER (1)      |  IDX_4       |
|   | TIME_STAMP      | DATE            |  |
|   | EXTRA           | NVARCHAR2 (100) |  |









Εικόνα 15 - Πίνακας Client\_Addresses

Ο πίνακας Client\_Phones έχει πληροφορίες για τα τηλέφωνα του οφειλέτη.

| MCS.CLIENT_PHONES   |               |                 |   |
|---|---------------|-----------------|---|
|  | PHONE_ID      | NUMBER          |  IDX_5             |
|  | CUSTOMER_ID   | NUMBER          |  IDX_5,IDX_2,IDX_1 |
|  | PHONE_TYPE_ID | NUMBER          |  IDX_4             |
|  | CLIENT_ID     | NUMBER          |  IDX_1             |
|   | PHONE         | NVARCHAR2 (40)  |  IDX_3             |
|   | TIME_STAMP    | DATE            |   |
|   | EXTRA         | NVARCHAR2 (100) |   |

Εικόνα 16 - Πίνακας Client\_Phones

Τέλος ο πίνακας Client\_EX\_Data περιέχει στοιχεία για οντν οφειλέτη που δεν μπορούν αν εισαχθούν σε κανένα άλλο πίνακα. Για παράδειγμα την αρχή έκδοσης της ταυτότητάς του.











| MCS.CLIENT_EX_DATA  |              |                 |   |
|---|--------------|-----------------|---|
|  | EX_DATA_ID   | NUMBER          |  IDX_3             |
|  | CUSTOMER_ID  | NUMBER          |  IDX_2,IDX_1,IDX_3 |
|  | CLIENT_ID    | NUMBER          |  IDX_1             |
|   | EX_DATA_DESC | NVARCHAR2 (100) |  IDX_4             |
|   | EX_DATA      | NVARCHAR2 (100) |                    |
|   | NAVURL       | NVARCHAR2 (100) |   |
|   | TIME_STAMP   | DATE            |   |

Εικόνα 17 - Πίνακας Client\_EX\_Data

#### 4.3.3.2 Πίνακες Δευτερευόντων Προσώπων και Στοιχείων τους










Υπάρχουν 4 πίνακες όμοιοι με τους πίνακες που κρατάνε τα στοιχεία των οφειλετών, που περιέχουν στοιχεία για τα δευτερεύοντα άτομα. Δηλαδή εγγυητές και συνοφειλέτες.

### Πίνακας Sec\_Person

| MCS.SEC_PERSON  |                       |                 |   |
|---|-----------------------|-----------------|---|
|  | SEC_PERSON_ID         | NUMBER          |  IDX_6             |
|  | CUSTOMER_ID           | NUMBER          |  IDX_7,IDX_6,IDX_5 |
|   | INST_ID               | NUMBER          |  IDX_1             |
|  | CASE_ID               | NUMBER          |  IDX_7             |
|   | SEC_PERSON_TYPE_ID    | NUMBER          |                    |
|   | SURNAME               | NVARCHAR2 (100) |  IDX_4             |
|   | FIRSTNAME             | NVARCHAR2 (50)  | IDX_4   |
|   | FATHERNAME            | NVARCHAR2 (50)  | IDX_4   |
|   | IDENTITYNO            | NVARCHAR2 (30)  | IDX_3   |
|   | TAX_NO                | NVARCHAR2 (30)  | IDX_2   |
|   | SEC_PERSON_EXTRA_DATA | NUMBER (1)      |                    |
|   | ID_ISSUE_DATE         | NVARCHAR2 (40)  |   |
|   | SEC_PERSON_SEC_ID     | NVARCHAR2 (200) |   |

Εικόνα 18 - Πίνακας Sec\_Person








### Πίνακας Sec\_Person\_Addresses

| MCS.SEC_PERSON_ADDRESSES  |                 |                 |  |
|---|-----------------|-----------------|--|
|  | ADDRESS_ID      | NUMBER          |  IDX_4             |
|  | CUSTOMER_ID     | NUMBER          |  IDX_4,IDX_3,IDX_2 |
|  | SEC_PERSON_ID   | NUMBER          |  IDX_2             |
|   | ADDRESS_TYPE_ID | NUMBER          |  IDX_1             |
|   | ADDRESS         | NVARCHAR2 (100) |                    |
|   | TOWN            | NVARCHAR2 (60)  |  |
|   | ZIP             | NVARCHAR2 (30)  |  |
|   | MALLING_ADDRESS | NUMBER (1)      |                    |
|   | EXTRA           | NVARCHAR2 (100) |  |
|   | TIME_STAMP      | DATE            |  |

Εικόνα 19 - Πίνακας Sec\_Person\_Addresses











## Πίνακας Sec\_Person\_Phones

| MCS.SEC_PERSON_PHONES   |               |                |   |
|---|---------------|----------------|---|
|  | PHONE_ID      | NUMBER         |  IDX_1             |
|  | CUSTOMER_ID   | NUMBER         |  IDX_3,IDX_2,IDX_1 |
|   | PHONE_TYPE_ID | NUMBER         |  IDX_5             |
|  | SEC_PERSON_ID | NUMBER         |  IDX_2             |
|   | PHONE         | NVARCHAR2 (40) | IDX_4   |
|   | EXTRA         | VARCHAR2 (100) |   |
|   | TIME_STAMP    | DATE           |   |

Εικόνα 20 - Πίνακας Sec\_Person\_Phones

## Πίνακας Sec\_Person\_Ex\_Data

| MCS.SEC_PERSON_EX_DATA  |               |                 |   |
|---|---------------|-----------------|---|
|  | EX_DATA_ID    | NUMBER          |  IDX_1             |
|  | CUSTOMER_ID   | NUMBER          |  IDX_1,IDX_3,IDX_2 |
|  | SEC_PERSON_ID | NUMBER          |  IDX_2             |
|   | EX_DATA_DESC  | NVARCHAR2 (100) |  IDX_4             |
|   | EX_DATA       | NVARCHAR2 (100) |                    |

Εικόνα 21 - Πίνακας Sec\_Person\_Ex\_Data

### 4.3.4 Πίνακας Cases









Ο πίνακας αυτός είναι ένας από τους κυριότερους του συστήματος, καθώς περιέχει τις πιο χρησιμοποιούμενες πληροφορίες. Κύρια στοιχεία του είναι ο αριθμός λογαριασμού (case\_sec\_id), το ελάχιστο ποσό καταβολής και το συνολικό οφειλόμενο ποσό του οφειλέτη προς τον χρηματοδότη οργανισμό.

| MCS.CASES |                         |                |                 |
|-----------|-------------------------|----------------|-----------------|
| KEY       | CASE_ID                 | NUMBER         | ⊗ IDX_30        |
| KEY       | CLIENT_ID               | NUMBER         | ⊗ IDX_28        |
| KEY       | CUSTOMER_ID             | NUMBER         | ⊗ IDX_29,IDX_30 |
|           | INST_ID                 | NUMBER         | ⊗ IDX_27        |
|           | CASE_SEC_ID             | NVARCHAR2 (50) | ⊗ IDX_26        |
| KEY       | PACK_ID                 | NUMBER         | ⊗ IDX_24        |
|           | LOCATION_DESC           | NVARCHAR2 (50) | ⊗ IDX_16        |
|           | INIT_LENT_CAPITAL       | NUMBER         | ⊗               |
|           | INIT_LENT_INTEREST      | NUMBER         | ⊗               |
|           | INIT_LENT_EXPENSES      | NUMBER         | ⊗               |
|           | LENT_CAPITAL            | NUMBER         | ⊗               |
|           | LENT_INTEREST           | NUMBER         | ⊗               |
|           | LENT_EXPENSES           | NUMBER         | ⊗               |
|           | TOTAL_LENT_AMOUNT       | NUMBER         | ⊗ IDX_3         |
|           | TOTAL_INIT_LENT_AMOUNT  | NUMBER         | ⊗ IDX_4         |
|           | INIT_MIN_PAY_AMOUNT     | NUMBER         | ⊗ IDX_5         |
|           | MIN_PAY_AMOUNT          | NUMBER         | ⊗ IDX_6         |
|           | CURRENCY_CODE           | NVARCHAR2 (3)  | ⊗               |
|           | LAST_PAY_DATE           | DATE           | IDX_7           |
|           | LAST_CONTACT_ID         | NUMBER         |                 |
|           | LAST_STRONG_CONTACT_ID  | NUMBER         |                 |
|           | LAST_ACTION_CODE        | NUMBER         | IDX_8           |
|           | LAST_UPDATE_DATE        | DATE           | IDX_9           |
|           | LAST_CONTACT_DATE       | DATE           | IDX_10          |
|           | IS_ACTIVE               | NUMBER (1)     | ⊗ IDX_2,IDX_21  |
|           | USER_ID                 | NUMBER         | IDX_2,IDX_22    |
|           | OLD_USER_ID             | NUMBER         |                 |
|           | INIT_BUCKET_ID          | NUMBER         | IDX_11          |
|           | BUCKET_ID               | NUMBER         | IDX_12          |
|           | PRODUCT_TYPE_ID         | NUMBER         | IDX_32          |
|           | HAS_TOLLS               | NUMBER (1)     | ⊗               |
|           | INVOICE_PERCENT         | NUMBER         |                 |
|           | CASE_EXT_DATA           | NUMBER (1)     | ⊗               |
|           | WORKFLOW_ID             | NUMBER         | ⊗               |
|           | BILLING_DATE            | DATE           | IDX_13          |
|           | BILLING_DESC            | NVARCHAR2 (50) | IDX_33          |
| KEY       | LOCATION_ID             | NUMBER         | ⊗ IDX_23        |
|           | PRIORITY                | NUMBER         | ⊗ IDX_25        |
|           | IS_VISABLE              | NUMBER         | ⊗ IDX_20        |
|           | CUSTOMER_NAME           | NVARCHAR2 (50) | ⊗ IDX_31        |
|           | PACK_DESC               | NVARCHAR2 (50) | ⊗ IDX_17        |
|           | WORKFLOW_NAME           | NVARCHAR2 (50) | ⊗               |
|           | CREATE_DATE             | DATE           | IDX_14          |
|           | HAS_SEC_PERSON          | NUMBER (1)     | ⊗               |
|           | DEACTIVATION_DATE       | DATE           |                 |
|           | LAST_PAY_AMOUNT         | NUMBER         | ⊗ IDX_18        |
|           | IS_LEGAL_CASE           | NUMBER (1)     | ⊗               |
|           | DELEQUANCY_ID           | NUMBER         | ⊗               |
|           | PACK_TYPE_ID            | NUMBER         | ⊗ IDX_19        |
|           | LAST_STRONG_ACTION_CODE | NUMBER         | IDX_15          |
|           | ASSIGN_DATE             | DATE           |                 |
|           | RETURN_DATE             | DATE           |                 |
|           | WORKED                  | NUMBER (1)     |                 |
|           | LENT_DATE               | DATE           |                 |
|           | APPLICATION_NO          | VARCHAR2 (50)  |                 |
|           | CASE_SEC_ID_2           | VARCHAR2 (100) |                 |
|           | CASE_SEC_ID_3           | VARCHAR2 (50)  | IDX_1           |
|           | TOTAL_PAY_AMOUNT        | NUMBER         | ⊗               |
|           | CASE_BEHAVIOR_SCORE     | NUMBER         | ⊗               |

Εικόνα 22 - Πίνακας Cases

#### 4.3.4.1 Πίνακας Case\_EX\_Data





Ο πίνακας αυτός περιέχει πληροφορίες που δεν εμπεριέχονται στον κύριο πίνακα Cases. Τέτοιες πληροφορίες μπορεί να είναι το υποκατάστημα που χορηγήθηκε το ποσό, και το πιστωτικό όριο του πελάτη.

| MCS.CASE_EX_DATA  |              |                 |   |
|---|--------------|-----------------|---|
|  | EX_DATA_ID   | NUMBER          |  IDX_1             |
|  | CUSTOMER_ID  | NUMBER          |  IDX_1,IDX_4,IDX_2 |
|  | CASE_ID      | NUMBER          |  IDX_2             |
|   | EX_DATA_DESC | NVARCHAR2 (100) |  IDX_3             |
|   | EX_DATA      | NVARCHAR2 (100) |                    |
|   | NAVURL       | NVARCHAR2 (100) |   |

Εικόνα 23 - Πίνακας Case\_Ex\_Data

#### 4.3.5 Πίνακας Locations















Ο πίνακας αυτός έχει τα queues που χρειάζεται να δημιουργηθούν , ώστε να οδηγούνται οι λογαριασμοί ανάλογα με τον τρόπο διαχείρισης που ορίζεται κάθε φορά.

| MCS.LOCATIONS   |                  |                  |  |
|---|------------------|------------------|--|
|  | LOCATION_ID      | NUMBER           |  IDX_1       |
|  | CUSTOMER_ID      | NUMBER           |  IDX_3,IDX_1 |
|   | LOCATION_DESC    | NVARCHAR2 (50)   |  IDX_2       |
|   | LOCATION_TYPE_ID | NUMBER           |  IDX_4       |
|   | VIEW_ROLES       | NVARCHAR2 (1000) |  |

Εικόνα 24 - Πίνακας Locations

#### 4.3.6 Πίνακας Contact

Στην πίνακα αυτό καταχωρούνται εγγραφές κάθε φορά που ένας χειριστής κάνει μια κίνηση σε κάθε καρτέλα οφειλέτη. Περιέχει την κίνηση, την ώρα που έγινε, τον χειριστή που έκανε την κίνηση, αν έχει υπόσχεση το ποσό που θα καταβάλλει ο οφειλέτης και την ημερομηνία και αν έχει ημερομηνία επόμενης επαφής καταγράφεται και αυτή.













| MCS.CONTACT   |                              |                 |   |
|---|------------------------------|-----------------|---|
|  | CONTACT_ID                   | NUMBER          |  IDX_5             |
|   | ACTION_CODE                  | NUMBER          |  IDX_12            |
|  | CASE_ID                      | NUMBER          |  IDX_3             |
|  | CUSTOMER_ID                  | NUMBER          |  IDX_5,IDX_4,IDX_3 |
|   | INST_ID                      | NUMBER          | IDX_10  |
|   | USER_ID                      | NUMBER          |  IDX_9             |
|   | PAY_DATE                     | DATE            | IDX_1   |
|   | PAY_AMOUNT                   | NUMBER          | IDX_2   |
|   | NEXT_CONTACT_DATE            | DATE            | IDX_8,IDX_6   |
|   | COMMENTS                     | NVARCHAR2 (800) |   |
|   | LAST_CONTACT                 | NUMBER (1)      |  IDX_6             |
|   | LAST_STRONG_CONTACT          | NUMBER (1)      |  IDX_7             |
|   | PREVIOUS_CONTACT_ID          | NUMBER          |   |
|   | PREVIOUS_STRONG_CONTACT_ID   | NUMBER          |   |
|   | LAST_PROMISE_ID              | NUMBER          |   |
|   | IS_REMINDER                  | NUMBER (1)      |  IDX_6             |
|   | DATA1                        | NVARCHAR2 (300) |   |
|   | DATA2                        | NVARCHAR2 (50)  |   |
|   | DATA3                        | NVARCHAR2 (30)  |   |
|   | DATA4                        | NVARCHAR2 (30)  |   |
|   | CONTACT_DATE                 | DATE            |  IDX_13           |
|   | ACTION_TYPE_ID               | NUMBER          |   |
|   | PROMISE_KEPT_BROKEN          | NVARCHAR2 (20)  |                  |
|   | NEXT_CONTACT_TIME            | NVARCHAR2 (5)   | IDX_8   |
|   | PREVIOUS_CONTACT_CODE        | NUMBER          |   |
|   | PREVIOUS_STRONG_CONTACT_CODE | NUMBER          |   |
|   | LEGAL_LAND_REGISTRY          | NVARCHAR2 (50)  |   |
|   | LEGAL_ORDER_DATE             | DATE            |   |
|   | LEGAL_PREFECT                | NVARCHAR2 (50)  |   |
|   | LEGAL_DATE_COMPLETED         | DATE            |   |
|   | LEGAL_PROPERTY_TYPE          | NUMBER          |   |
|   | LEGAL_PAY_AMOUNT             | NUMBER          |   |
|   | LEGAL_CITY_COURT             | NVARCHAR2 (50)  |   |
|   | LEGAL_RESULTS                | NUMBER          |   |
|   | LEGAL_TEXT1                  | NVARCHAR2 (50)  |   |
|   | LEGAL_TEXT2                  | NVARCHAR2 (50)  |   |
|   | LEGAL_DATE_SENT              | DATE            |   |
|   | LEGAL_DATE_RETURNED          | DATE            |   |
|   | LEGAL_LAWER                  | NVARCHAR2 (50)  |   |
|   | LEGAL_ACTION_DATE            | DATE            |   |
|   | LEGAL_COURT                  | NVARCHAR2 (50)  |   |
|   | LEGAL_REFERS_TO              | NUMBER          |   |
|   | LEGAL_TEXT3                  | NVARCHAR2 (50)  |   |
|   | WORKED                       | NUMBER (1)      | IDX_11  |
|   | CONTACT_GUID                 | VARCHAR2 (32)   |   |
|   | REJECTED                     | VARCHAR2 (1)    |   |
|   | PROMISE_NUMBER               | NUMBER          |   |
|   | PACK_ID                      | NUMBER          |                  |

Εικόνα 25 - Πίνακας Contact



### 4.3.7 Πίνακας Payments







Κάθε φορά που στο σύστημα εισάγεται μια πληρωμή, τότε καταχωρείται και μία εγγραφή στον πίνακα αυτό. Καταχωρείται το ποσό, η ημερομηνία πληρωμής, ο λογαριασμός για τον οποίο έγινε η πληρωμή.

| MCS.PAYMENTS  |                             |                |   |
|---|-----------------------------|----------------|---|
|  | PAY_ID                      | NUMBER         |  IDX_3,IDX_1       |
|  | CASE_ID                     | NUMBER         |  IDX_2             |
|  | CUSTOMER_ID                 | NUMBER         |  IDX_4,IDX_3,IDX_2 |
|   | PACK_ID                     | NUMBER         | IDX_5   |
|   | INST_ID                     | NUMBER         |  IDX_6             |
|   | PAY_DATE                    | DATE           |                    |
|   | PAY_AMOUNT                  | NUMBER         |                    |
|   | PAY_INTEREST                | NUMBER         |   |
|   | PAY_EXPENSES                | NUMBER         |   |
|   | TOTAL_PAY_AMOUNT            | NUMBER         |  IDX_7             |
|   | ACTION_CODE                 | NUMBER         |                    |
|   | COMMENTS                    | NVARCHAR2 (50) |   |
|   | CURRENCY_CODE               | NVARCHAR2 (3)  |                   |
|   | TOLLS_ID                    | NUMBER         | IDX_9   |
|   | PREVIOUS_LAST_ACTION_ID     | NUMBER         |   |
|   | PREVIOUS_LAST_CONTACT_ID    | NUMBER         |   |
|   | PREVIOUS_STRONG_ACTION_ID   | NUMBER         |   |
|   | PREVIOUS_STRONG_CONTACT_ID  | NUMBER         |   |
|   | PREVIOUS_PROMISE_CONTACT_ID | NUMBER         |   |
|   | PROMISE_KEPT_BROKEN         | NVARCHAR2 (20) | IDX_8   |
|   | TIME_STAMP                  | DATE           |   |
|   | PREVIOUS_ACTION_USER_ID     | NUMBER         |   |
|   | BANK_PAY_ID                 | VARCHAR2 (50)  |   |

Εικόνα 26 - Πίνακας Payments







#### Πίνακες Costs, Installments και Tolls

Οι πίνακες Installments χρησιμοποιείται αν υπάρχει κάποιος συμβιβασμός ώστε να υπάρχουν καταγγραμμένες στο σύστημα οι δόσεις που θα πληρώσει ο οφειλέτης.














| MCS.INSTALLMENTS  |                    |                |   |
|---|--------------------|----------------|---|
|  | INSTALLMENT_ID     | NUMBER         |  IDX_1             |
|  | CASE_ID            | NUMBER         |  IDX_6,IDX_2,IDX_1 |
|  | CUSTOMER_ID        | NUMBER         |  IDX_6,IDX_1       |
|   | INSTALLMENT_DATE   | DATE           | IDX_3   |
|   | INSTALLMENT_AMOUNT | NUMBER         | IDX_4   |
|   | TIMESTAMP          | DATE           |   |
|   | PAYED              | NVARCHAR2 (20) | IDX_5   |

Εικόνα 27 - Πίνακας Installments

Οι πίνακες Costs και Tolls χρησιμοποιούνται βοηθητικά για την τιμολόγηση του χρηματοπιστωτικού οργανισμού με τους πελάτες του.

| MCS.COSTS   |             |                 |   |
|---|-------------|-----------------|---|
|    | COST_ID     | NUMBER          |  IDX_2 |
|    | CUSTOMER_ID | NUMBER          |  IDX_2 |
|    | CASE_ID     | NUMBER          | IDX_3   |
|  | TYPE_ID     | NUMBER          | IDX_1   |
|   | DATA1       | NVARCHAR2 (50)  |   |
|   | DATA2       | NVARCHAR2 (50)  |   |
|   | DATA3       | NVARCHAR2 (50)  |   |
|   | DATE1       | DATE            |   |
|   | DATE2       | DATE            |   |
|   | NUMBER1     | NUMBER          |   |
|   | NUMBER2     | NUMBER          |   |
|   | MEMO        | NVARCHAR2 (800) |   |
|   | NUMBER3     | NUMBER          |   |
|   | DATA4       | NVARCHAR2 (50)  |   |
|   | TIME_STAMP  | DATE            |   |

Εικόνα 28 - Πίνακας Costs

| MCS.TOLLS   |                |                 |   |
|---|----------------|-----------------|---|
|  | TOLLS_ID       | NUMBER          |  IDX_6       |
|  | CASE_ID        | NUMBER          |  IDX_7       |
|  | CUSTOMER_ID    | NUMBER          |  IDX_1,IDX_6 |
|  | TOLL_STATUS_ID | NUMBER          |  IDX_3       |
|   | TOLLDATE       | DATE            |  IDX_2       |
|   | TOLLCAPTIAL    | NUMBER          |              |
|   | TOLLINTEREST   | NUMBER          |   |
|   | TOLLEXPENSES   | NUMBER          |   |
|   | TOLLTOTAL      | NUMBER          |              |
|   | DATA1          | NVARCHAR2 (255) |   |
|   | DATA2          | NVARCHAR2 (255) |   |
|   | DATA3          | NVARCHAR2 (255) |   |
|   | DATA4          | NVARCHAR2 (255) |   |
|   | TMP            | NUMBER          |   |
|   | INST_ID        | NUMBER          |  IDX_4       |
|   | PACK_ID        | NUMBER          |  IDX_5      |

Εικόνα 29 - Πίνακας Tolls

Στο παράρτημα υπάρχει μορφοποιημένο το Σχεσιακό Σχήμα της Βάσης.

## **Κεφάλαιο 5 – Role-based Access Control (RBAC)**

### **5.1 Εισαγωγή**

Στους περισσότερους οργανισμούς, υπάρχουν διάφορες χιλιάδες υπάλληλοι, στους οποίους οι αρχές διαχείρισης ασφάλειας πρέπει να εφαρμοστούν. Για να απλοποιήσει τη διοίκηση ασφάλειας και ροής εργασίας, πολλές οργανώσεις καθορίζουν τους ρόλους με τους οποίους τα πολλαπλάσια άτομα μπορούν να συνδεθούν. Οι άδειες συνδέονται έπειτα με αυτούς τους ρόλους σύμφωνα με μια πολιτική ελέγχου πρόσβασης όπως καθορίζονται από την οργάνωση. Υπάρχουν άλλα πρότυπα που ενσωματώνουν το RBAC [KFC95, San97], αλλά η εξήγηση του ερευνητή θα βασιστεί πρώτιστα στα πρότυπα που προτάθηκαν και τυποποιήθηκαν από τους Sandhu et al[SCFY96].

### **5.2 Ορισμός Ρόλων**

Ένας ρόλος είναι ένα σύνολο χρηστών ή ατόμων με τις σχετικές άδειές τους και ευθύνες όσον αφορά έναν στόχο ή μια δραστηριότητα. Σε σχέση με το τρέχον παράδειγμα, ο πίνακας επιλογής θα αντιπροσώπευε έναν ρόλο με την ευθύνη έναν υποψήφιο για την απασχόληση, συμπεριλαμβανομένων όλων των αδειών που απαιτούνται για να εκπληρώσουν αυτήν την ευθύνη. Οι ρόλοι σε μια οργάνωση συνήθως καθορίζονται σύμφωνα με τις λειτουργίες εργασίας υπαλλήλων, την ικανότητα και τα προσόντα. Στις περισσότερες περιπτώσεις, το RBAC υποθέτει ότι ένα άτομο μπορεί να εγκριθεί για να εκτελέσει τις πολλαπλάσιες λειτουργίες ρόλου και ότι ένας ρόλος μπορεί να υποτεθεί από πολλά άτομα.

Στις περισσότερες εκδόσεις RBAC, οι σχέσεις διαμορφώνονται μεταξύ των ρόλων, όπως και μεταξύ των ρόλων και των αδειών. Αυτές οι σχέσεις μπορούν να έχουν ως σκοπό να επιβάλουν τις θεμελιώδεις αρχές στις περισσότερες οργανωτικές πολιτικές ασφάλειας, π.χ. το χωρισμό των καθηκόντων και της αντιπροσωπείας αρχής. Η διάκριση μεταξύ των ρόλων και των ομάδων χρηστών είναι ότι οι ομάδες δεν έχουν καμία άμεση ένωση με τις άδειες. Οι ρόλοι δεν είναι συγκεκριμένοι για μια ιδιαίτερη πολιτική ασφάλειας, αντίθετα από τα διαμερίσματα, τα οποία χρησιμοποιούνται για να επιβάλουν τις μονόδρομες πληροφορίες στα συγκεκριμένα πρότυπα τέτοια όπως το στρατιωτικό πρότυπο ασφάλειας. Αυτό σημαίνει ότι το RBAC είναι ουσιαστικά ανεξάρτητο από τα πρότυπα MAC ή DAC, αλλά μπορεί να χρησιμοποιηθεί από κοινού[SCFY96].

### **5.3 Σύνοψη του RBAC**

Αυτό το τμήμα θα παράσχει μια συνοπτική επισκόπηση των θεμελιωδών εννοιών RBAC. Στην ουσία RBAC χρησιμοποιεί τις έννοιες των οντοτήτων, σχέσεων και περιορισμών:

Οι οντότητες περιλαμβάνουν τα εξής:

- Ο χρήστης (U) δείχνει το άτομο.
- Το προνόμιο (P) δείχνει το δικαίωμα που ένας χρήστης πρέπει να εκτελέσει έναν στόχο στη ροή εργασίας ή να χειριστεί ένα αντικείμενο. Τα προνόμια συχνότερα αναφέρονται ως άδειες.
- Ο ρόλος (R) καθορίζει ένα σύνολο χρηστών στο οποίο οι ομάδες προνομίων μπορούν να διοριστούν. Οι χρήστες και οι άδειες συνδέονται σιωπηρά μέσω των ρόλων, ποτέ άμεσα με τον έναν. Οι ρόλοι, στη συνέχεια, συσχετίζονται ο ένας με τον άλλον σε μια μερική διαταγή π.χ. ιεραρχία ρόλου. Αυτό σημαίνει ότι οι ανώτεροι ρόλοι (και οι χρήστες που τους σύνδεσαν) κληρονομούν τις άδειες των υφισταμένων τους. Οι Sandhu et al [SCFY96] προτείνουν τη χρήση των ιδιωτικών ρόλων για να εξασφαλίζεται ότι η μεταβατική φύση της ιεραρχίας ρόλου δεν επιτρέπει την ακατάλληλη κληρονομιά των ισχυρών προνομίων.

Οι σχέσεις μεταξύ των οντοτήτων καθορίζονται έτσι ώστε μπορούν να βοηθήσουν να καθορίσουν τα δικαιώματα ενός χρήστη σε μια ιδιαίτερη στιγμή εγκαίρως.

Οι περιορισμοί αναφέρονται στους περιορισμούς που τοποθετούνται πέρα από τις σχέσεις που διαμορφώνονται μεταξύ των οντοτήτων.

## **5.4 Τέσσερα εννοιολογικά πρότυπα RBAC**

Αυτό το τμήμα αναφέρεται άμεσα στην οικογένεια τεσσάρων εννοιολογικών προτύπων RBAC όπως καθορίζεται από τους Sandhu et al [SCFY96]. Αυτά τα πρότυπα παρέχουν μια σαφή και συστηματική εξήγηση των σημαντικών στοιχείων RBAC.

### **5.4.1 Βασικό μοντέλο**

Το πρότυπο βάσεων ή RBAC<sub>0</sub>, ουσιαστικά καθορίζει τις ελάχιστες απαιτήσεις συστημάτων για την υποστήριξη RBAC. Εισάγει τις έννοιες τριών οντοτήτων: χρήστες (U), ρόλοι (R) και άδειες (P). Οι άδειες αντιπροσωπεύουν τα προνόμια και υποτίθεται ότι ήταν πάντα θετικές. Επιπλέον, μπορούν να ισχύσουν για τα ενιαία ή πολλαπλάσια αντικείμενα. Μια σύνοδος (S) καθορίζεται για να προσδιορίσει μια ιδιαίτερη περίπτωση ή μια περίπτωση, όπως μια περίπτωση στόχου σε μια ροή εργασίας. Κάθε σύνοδος συσχετίζεται με έναν χρήστη, ρόλο και άδεια με κάποιο τρόπο. Οι σχέσεις που καθορίζονται από το πρότυπο RBAC<sub>0</sub> περιλαμβάνουν:

- Ανάθεση χρηστών (UA): αφορά τους χρήστες στους ρόλους κατά τρόπο πολλαπλό.
- Ανάθεση άδειας (PA): αφορά τις άδειες στους ρόλους κατά τρόπο πολλαπλό.

Οι σύνοδοι αντιστοιχούν τους χρήστες σε έναν ή πολλούς ρόλους. Όταν ένας ρόλος ενεργοποιείται σε μια σύνοδο, ο χρήστης υποθέτει όλες τις άδειες που συνδέονται με αυτό, μέσω της σχέσης PA. Αυτές οι άδειες ορίζονται στον εν λόγω χρήστη κατά την ολόκληρη διάρκεια της συνόδου, η οποία μπορεί να ολοκληρωθεί κατά την κρίση ενός χρήστη. Κάθε σύνοδος μπορεί να θεωρηθεί ως «μονάδα του ελέγχου πρόσβασης» [SCFY96]. Το RBAC<sub>0</sub> επίσης ικανοποιεί τη δυναμική ενεργοποίηση και την απενεργοποίηση των ρόλων κατά τη διάρκεια της διάρκειας ζωής μιας συνόδου.

Ο ακόλουθος επίσημος καθορισμός του προτύπου βάσεων RBAC<sub>0</sub> αναφέρεται άμεσα από τους Sandhu et al [SCFY96].

**Ορισμός:** Το μοντέλο RBAC<sub>0</sub> έχει τα εξής στοιχεία

- U, R, P και S (χρήστες, ρόλοι, άδειες και σύνοδοι αντίστοιχα),
- PA  $_ P \times R$ , μια πολλαπλή άδεια στη σχέση ανάθεσης ρόλου,
- UA  $_ U \times R$ , ένας πολλαπλός χρήστης στη σχέση ανάθεσης ρόλου,
- χρήστης:  $S \rightarrow U$ , μια λειτουργία που χαρτογραφεί κάθε  $s_i$  συνόδου στον ενιαίο χρήστη ( $s_i$ ) (σταθερό για τη διάρκεια ζωής της συνόδου), και
- ρόλοι:  $S \rightarrow 2^R$ , μια λειτουργία που χαρτογραφεί κάθε  $S_i$  συνόδου σε ένα σύνολο ρόλων ( $s_i$ ) υποσύνολο  $\{r \mid (\text{user}(s_i), r) \text{ ανήκει UA}\}$  και η σύνοδος  $s_i$  έχει τις άδειες  $U_{\text{ανήκειroles}(s_i)} \{p \mid (p, r) \text{ ανήκει PA}\}$

#### 5.4.2 Ιεραρχίες ρόλων

Το πρότυπο RBAC<sub>1</sub> στηρίζεται στο πρότυπο βάσεων με την εισαγωγή της έννοιας των ιεραρχιών ρόλου (RH). Αυτές οι ιεραρχίες διαμορφώνουν τις σχέσεις μεταξύ των ρόλων μέσα στην οργανωτική δομή. Οι σχέσεις είναι ενδεικτικές των καναλιών της αρχής και αρμοδιότητας για την οργάνωση. Οι ιεραρχίες ρόλου σε αυτό το πρότυπο είναι δομημένες με τον κυρίαρχο, ή περισσότερος πρεσβύτερος, ρόλοι προς την κορυφή των διαγραμμάτων, και οι κατώτεροι ρόλοι στο κατώτατο σημείο. Στις ιεραρχίες ρόλου RBAC<sub>1</sub>, οι ρόλοι κληρονομούν τις άδειες κατά τρόπο μεταβατικό. Κατά συνέπεια, ένας ανώτερος ρόλος θα κληρονομήσει τις άδειες από έναν κατώτερο ρόλο. Ο ακόλουθος επίσημος καθορισμός του προτύπου RBAC<sub>1</sub> αναφέρεται άμεσα από τους Sandhu et al [SCFY96].

**Ορισμός:** Το μοντέλο RBAC<sub>1</sub> έχει τα εξής στοιχεία:

- Τα U, R, P, PA, UA και ο χρήστης είναι αμετάβλητα από το RBAC<sub>0</sub>,
- RH υποσύνολο  $R \times R$ , είναι μια μερική διαταγή σχετικά με το R αποκαλούμενο την ιεραρχία ρόλου ή σχέση κυριαρχίας ρόλου, επίσης γραπτός ως  $\geq$ , και,

- Ρόλοι:  $S \rightarrow 2^R$  τροποποιείται από το RBAC<sub>0</sub> ώστε  $\text{roles}(s_i)$  υποσύνολο  $\{r \mid (\text{υπάρχει } r \geq r)[(\text{user}(s_i), r') \text{ ανήκει } UA]\}$  και μια σύνοδος  $s_i$  έχει τις άδειες  $U_{r2\text{roles}(s_i)}\{p \mid (\text{υπάρχει } r' \leq r)[(p, r') \text{ ανήκει } PA]\}$

Είναι προφανές ότι ο μεταβατικός τρόπος με τον οποίο οι άδειες κληρονομούνται μπορεί να παρέχει έναν κίνδυνο ασφάλειας για τις περιπτώσεις όπου οι ισχυρές άδειες ρόλου πρέπει να κρατηθούν ιδιωτικές σε έναν ιδιαίτερο ρόλο. Οι Sandhu et al [SCFY96] προτείνουν την προσθήκη των ιδιωτικών ρόλων για τέτοιες περιπτώσεις και περιέχουν τις άδειες που κρατιούνται ιδιωτικές στον εν λόγω ρόλο. Αυτοί οι ιδιωτικοί ρόλοι μπορούν να διαμορφώσουν μια χωριστή υποιεραρχία των ρόλων για τις περιπτώσεις όπου το πεδίο της κληρονομιάς πρέπει να περιοριστεί. Ο Moffet [Mof98] συζητά τη χρήση μιας εναλλακτικής λύσης διατάζοντας για να χρησιμοποιηθεί στο σχέδιο μιας ιεραρχίας ρόλου που διαφέρει από παραδοσιακά χρησιμοποιημένη οργανωτική.

### 5.4.3 Περιορισμοί

Το πρότυπο RBAC<sub>0</sub> παρέχει ένα πλαίσιο για τους στατικούς περιορισμούς που μπορούν να απεικονίσουν τη γενική πολιτική ασφάλειας μιας οργάνωσης. Οι ορισμοί σχέσης βοηθούν στον καθορισμό μιας ευρείας βάσης των περιορισμών ελέγχου πρόσβασης αλλά υπολείπονται στην επιβολή των δυναμικών περιορισμών, ο σημαντικότερος τύπος που είναι χωρισμός των καθηκόντων. Η άδεια της δυνατότητας ότι ορισμένες περιπτώσεις διαδικασίας ροής εργασίας να συμβιβαστεί η ασφάλεια είναι πολύ σημαντική στις πραγματικές καταστάσεις. Κατά συνέπεια, είναι απαραίτητο να επιβληθούν οι περιορισμοί στις σχέσεις μεταξύ των ρόλων, όπως και μεταξύ των ρόλων και των αδειών που καθορίζονται στο πρότυπο ροής εργασίας.

Το RBAC<sub>2</sub> έχει μια σημαντική συμβολή στο αρχικό πρότυπο RBAC<sub>0</sub> με την εισαγωγή της έννοιας των περιορισμών. Στο RBAC<sub>2</sub> υπάρχουν οι περιορισμοί θέσεων στις σχέσεις UA και PA και το χρήστη και τις λειτουργίες ρόλων για κάθε σύνοδο. Αυτοί οι περιορισμοί απεικονίζουν χαρακτηριστικά τις οργανωτικές πολιτικές με τον καθορισμό των αποδεκτών τιμών που τα συστατικά του προτύπου RBAC<sub>2</sub> μπορούν να υιοθετήσουν, παραδείγματος χάριν, μέσω του ορισμού των αμοιβαία αποκλειστικών ρόλων [SCFY96].

Οι περιορισμοί που το RBAC<sub>2</sub> καθορίζει ενθαρρύνονται για να είναι αρκετά απλοί δεδομένου ότι αυτό θα διευκολύνει την εφαρμογή τους. Οι Sandhu et al [SCFY96] προτείνουν τους ακόλουθους ελάχιστους τύπους περιορισμών που πρέπει να εφαρμοστούν και στο UA και στις σχέσεις PA:

Αμοιβαίοι περιορισμοί αποκλεισμού: Χωρισμός υποστήριξης των καθηκόντων με την παρεμπόδιση του ίδιου χρήστη από να ανήκει σε περισσότερους από έναν ρόλους σε ένα αμοιβαία αποκλειστικό σύνολο. Ομοίως, μπορούν να απαιτήσουν ότι μια άδεια ανήκει ή όχι σε περισσότερους από έναν ρόλους σε ένα αμοιβαία αποκλειστικό σύνολο.



Περιορισμοί αριθμού στοιχείων συνόλου: Περιορίστε τον αριθμό ρόλων στον οποίο ένας χρήστης ή μια άδεια μπορεί να διοριστεί. Περιορίζουν επίσης τον αριθμό χρηστών ή αδειών που διορίζονται σε ένα σύνολο ρόλου. Αυτός ο περιορισμός περιορίζει πρώτιστα την ανάθεση των ισχυρών αδειών.

Προαπαιτούμενοι περιορισμοί ρόλου: Μόνο επιτρέψτε στους χρήστες ή τις άδειες για να διοριστούν σε έναν ρόλο εάν έχουν οριστεί ήδη σε έναν άλλο διευκρινισμένο ρόλο. Αυτός ο περιορισμός εξασφαλίζει ότι μόνο εκείνοι που είναι πεπειραμένοι ή ικανοί εκπληρώνουν ορισμένους ρόλους ή των οποίων ευθύνη συσχετίζεται άμεσα με τον προαπαιτούμενο ρόλο.

Περιορισμοί στις συνόδους: Περιορίστε τον αριθμό συνόδων ότι ένας χρήστης ή ένας ρόλος μπορεί να έχει ενεργό συγχρόνως.

Περιορισμοί ιεραρχίας ρόλου: Υπονοούμενος από το RH, καθορίστε την πορεία ότι οι άδειες μπορούν να υποτεθούν. Αυτοί οι περιορισμοί είναι, εντούτοις, καλύτερα καθορισμένοι σιωπηρά στο RH από ρητά μέσω των περιορισμών.

#### **5.4.4 Παγιωμένο πρότυπο**

Το πρότυπο RBAC<sub>3</sub> ενσωματώνει τις έννοιες των ιεραρχιών και των περιορισμών ρόλου από τα πρότυπα RBAC<sub>1</sub> και RBAC<sub>2</sub>, αντίστοιχα. Η αλληλεπίδραση μεταξύ αυτών των δύο εννοιών δίνει αφορμή για διάφορα ζητήματα. Οι αμοιβαίοι περιορισμοί αποκλεισμού και αριθμού στοιχείων συνόλου μπορούν να παραβιαστούν σε ορισμένες περιστάσεις από τους ρόλους στο RH. Η ασάφεια προκύπτει κατά την απόφαση εάν πρέπει να εφαρμοστούν αυτοί οι περιορισμοί στα κληρονομημένα μέλη των ρόλων για τους οποίους ισχύουν άμεσα. Οι ιδιωτικοί ρόλοι μπορούν να χρησιμοποιηθούν σε τέτοιες περιστάσεις για να μειώσουν τις συγκρούσεις. Οι περιορισμοί τοποθετούνται μόνο σε αυτούς τους ιδιωτικούς ρόλους χωρίς επιρροή της ιεραρχίας ρόλου και των αρχών κληρονομιάς που εφαρμόζονται σε αυτούς.

#### **5.5 Σύνοψη**

Μια ρόλοκεντρική προσέγγιση στον έλεγχο πρόσβασης έχει πολλά πλεονεκτήματα. Αρχικά, η ταξινόμηση των ατόμων σύμφωνα με τους ρόλους είναι κατάλληλη δεδομένου ότι ο αριθμός ρόλων θα είναι σημαντικά λιγότερος από τον αριθμό υπαλλήλων σε μια οργάνωση. Κατά συνέπεια, η επιβολή του ελέγχου πρόσβασης απλοποιείται αρκετά δεδομένου ότι ένας ενιαίος κανόνας μπορεί να ισχύσει για τα πολλαπλάσια άτομα, και όταν ένα άτομο αλλάζει τους ρόλους, οι κανόνες ελέγχου πρόσβασης που επιβάλλονται, μην χρειαστείτε να αλλάξετε [Mof98]. Επιπλέον, το RBAC είναι ιδιαίτερα κατάλληλο για τη χρήση στα συστήματα ροής της εργασίας όπως επιτρέπει περισσότερο παραγωγική διανομή εργασίας μεταξύ των πολλαπλάσιων ατόμων που εκπληρώνουν έναν ιδιαίτερο ρόλο [BFA98]. Αυτό επιτυγχάνεται μέσω της άδειας των πολλαπλάσιων κατόχων ενός ρόλου για να εκπληρωθούν οι ευθύνες ενός μη διαθέσιμου ατόμου.



Το RBAC εξετάζει αποτελεσματικά την επιβολή της ασφάλειας επειδή επιβάλλει τρία σημαντικά χαρακτηριστικά γνωρίσματα ασφάλειας, δηλαδή λιγότερο προνόμιο, αφαίρεση στοιχείων και χωρισμό των καθηκόντων. Λιγότερο προνόμιο επιβάλλεται με τη διαμόρφωση ώστε το RBAC για να επιτρέψει μόνο εκείνες τις άδειες που απαιτούνται για να εκτελέσουν έναν στόχο που ορίζεται στον εφαρμόσιμο ρόλο. Η αφαίρεση στοιχείων εξετάζεται μέσω του συνυπολογισμού περισσότερων αφηρημένων αδειών για τα αντικείμενα παρά απλά τις συμβατικότερες άδειες που χρησιμοποιούνται από τα λειτουργικά συστήματα [SCFY96]. Ένα προηγούμενο μειονέκτημα των πιο πρόωρων προσεγγίσεων RBAC ήταν ότι δεν ικανοποίησαν την επιβολή των δυναμικών περιορισμών, ιδιαίτερα, το χωρισμό των καθηκόντων. Η καθιέρωση των σχέσεων ρόλος σε ρόλο εξετάζει αυτό με τον καθορισμό των αμοιβαία αποκλειστικών ρόλων. Επιπλέον, σε περίπτωση που οι αλλαγές γίνονται στην οργανωτική πολιτική ασφάλειας, οι αλλαγές χρειάζονται μόνο να γίνουν σε αυτές τις αναθέσεις ρόλου και ενδεχομένως στην ιεραρχία ρόλου. Αυτό σημαίνει ότι είναι εύκολο να τηρηθεί ο έλεγχος πρόσβασης ασφαλή και ενήμερος για τις δυναμικές αλλαγές που γίνονται σε μια οργανωτική δομή.

Ένα αρχικό πλεονέκτημα RBAC είναι ότι διευκολύνει τη διαχείριση και τη διοίκηση των αδειών. Σήμερα, τα πρότυπα RBAC ενσωματώνονται στα περισσότερα εργαλεία διοίκησης ασφάλειας που εφαρμόζουν τα πρότυπα έγκρισης ροής εργασίας, όπως οι Lotus Notes. Ο καθορισμός των ιεραρχιών ρόλου συμβάλλει στο εννοιολογικό σχέδιο των προτύπων αντιπροσωπειών, επειδή είναι εύκολο να δουν τα κανάλια της αρχής μέσα στην οργανωτική δομή.

## Κεφάλαιο 6 - Πρότυπο Έγκρισης Ροής Εργασίας (WAM)

### 6.1 Εισαγωγή

Ένα πρότυπο έγκρισης ροής εργασίας (WAM) ήδη υπάρχει που επιβάλλει τις αρχές ασφάλειας σε μια διαδικασία ροής της εργασίας με την εξέταση της χορήγησης και της ανάκλησης των εγκρίσεων σε ένα WFMS. Αυτό το κεφάλαιο αναφέρεται στο WAM που προτείνεται από τους Atluri και Huang [AH96b], και το εκτεταμένο πρότυπό τους [HA99]. Αυτό το πρότυπο θα παράσχει ένα πλαίσιο ασφάλειας μέσα στο οποίο το προτεινόμενο πρότυπο αντιπροσωπειών θα λειτουργήσει. Το WAM επιτρέπει τη δυναμική χορήγηση των εγκρίσεων, ταυτόχρονα λαμβάνοντας υπόψη τα χρονικά διαστήματα των έγκυρων εγκρίσεων όσον αφορά το χρόνο κατά τη διάρκεια του οποίου η εκτέλεση στόχου εμφανίζεται. Αυτό θα εξασφαλίσει ότι οι εγκρίσεις στόχου ισχύουν μόνο για τις χρονικές περιόδους των διαρκειών στόχου. Επιπλέον, αυτό το WAM επιτρέπει επίσης τις δυναμικές εξαρτήσεις στόχου.

Το πρότυπο χρησιμοποιεί ένα πρότυπο έγκρισης (AT) που συνδέεται με κάθε στόχο στη διαδικασία ροής της εργασίας. Αυτό το πρότυπο εξασφαλίζει ότι οι κατάλληλες εγκρίσεις στόχου χορηγούνται και ανακαλούνται για το χρονικό διάστημα της εκτέλεσης στόχου μόνο, με αυτόν τον τρόπο επιτυγχάνεται συγχρονισμός επίτευξης μεταξύ της ροής έγκρισης και της ροής της δουλειάς. Μια βάση έγκρισης καθορίζεται για να περιέχει τις καταχωρήσεις όλων των εγκρίσεων. Η αρχική δύναμη του WAM είναι στο σχηματισμό ενός συνόλου επιλέξιμων θεμάτων που εγκρίνονται για να εκτελέσουν έναν στόχο. Ο σχηματισμός αυτού του συνόλου είναι σημαντικός δεδομένου ότι επιβάλλει το χωρισμό των περιορισμών καθηκόντων που έχουν καθιερωθεί ήδη ως σημαντική απαίτηση ασφάλειας ροής της εργασίας.

### 6.2 Το πρότυπο έγκρισης (AT)

Τα πρότυπα έγκρισης καθορίζουν τους στατικούς κανόνες για τις εγκρίσεις, οι οποίοι μπορεί να ορίζονται κατά τη διάρκεια της διαδικασίας σχεδίου ροής της εργασίας. Κάθε στόχος είναι συνδεδεμένος με μια AT η οποία χρησιμοποιείται για να παραγάγει τη σχετική έγκριση επάνω στην έναρξη στόχου. Αυτές οι εγκρίσεις καθορίζονται από τους Atluri και Huang [AH96b] ως εξής:

**Ορισμός 6.2.1:** Μια έγκριση είναι μια τετράδα  $A = (s, o, pr, [\tau_b, \tau_e])$ , όπου στο θέμα  $s$  χορηγείται η πρόσβαση στο αντικείμενο  $o$  με προνόμιο  $pr$  από τη στιγμή  $\tau_b$  έως τη στιγμή  $\tau_e$ .

Οι περισσότερες οργανώσεις βασίζουν τα συστήματα διαχείρισης ροής της εργασίας σε μια οργανωτική ιεραρχία. Αυτός ο τύπος δομής παραχωρεί σε μια

ρολοκεντρική προσέγγιση στον έλεγχο πρόσβασης όπου τα άτομα συνδέονται με τους οργανωτικούς ρόλους. Είναι σημαντικό να σημειωθεί ότι υπάρχει μια λειτουργία αντιστοιχίας των ρόλων και αντιτίθεται στους τύπους αντικειμένου στα πιο πρόσφατα WAMs. Κατά συνέπεια, για κάθε ρόλο, υπάρχει ένα σύνολο θεμάτων  $S_{r_i}$  που εξουσιοδοτούνται για να υποθέσουν το ρόλο  $r_i$ . Ομοίως, υπάρχει ένα σύνολο αντικειμένων  $O_{\gamma_i}$  που είναι ενός καθορισμένου τύπου αντικειμένου  $\gamma$ . Κάθε ένας περιέχει τις παραμέτρους σχετικά με τα θέματα και τα αντικείμενα που αναφέρονται, οι τύποι προνομίων που χορηγούνται, όπως και το χρονικό διάστημα κατά τη διάρκεια του οποίου ο στόχος πρέπει να εκτελεσθεί. Η προσθήκη της παραμέτρου χρονικού διαστήματος διακρίνει αυτήν την προσέγγιση από τα προηγούμενα πρότυπα ελέγχου πρόσβασης. Καθιερώνοντας αυτόν τον καθορισμό για μια έγκριση, οι Huang και Atluri [HA99] καθορίζουν ως εξής:

**Ορισμός 6.2.2:** Δεδομένου ενός στόχου  $tw_i$ , ένα πρότυπο έγκρισης  $AT(tw_i)$  ορίζεται ως η τετράδα  $AT(tw_i) = ((r_i, -), (\gamma_i, -), pr_i, [\tau_{li}, \tau_{ui}])$  όπου:

1.  $(r_i, -)$ , είναι μια υπαγόμενη τρύπα που μπορεί από ένα να γεμίσει  $S_i$  θεμάτων όπου  $G(s_i) = r_i$
2.  $(\gamma_i, -)$ , είναι μια τρύπα αντικειμένου που μπορεί από ένα αντικείμενο να γεμίσει  $o_i$  όπου  $F(o_i) = \gamma_i$
3.  $pr_i$  είναι το προνόμιο που δίνεται στο  $s_i$  για το αντικείμενο  $o_i$
4.  $[\tau_{li}, \tau_{ui}]$  είναι το χρονικό διάστημα που πρέπει να εκτελεσθεί η λειτουργία.

Στα ανωτέρω, η (1) θα εξασφαλίσει ότι μόνο τα θέματα που ανήκουν στο ρόλο  $r_i$  θα εγκριθούν για να εκτελέσουν το  $tw_i$  στόχου. Η (2) δηλώνει ότι μόνο αντικείμενα του τύπου  $\gamma_i$  μπορούν να χρησιμοποιηθούν στην εκτέλεση του  $tw_i$  στόχου. Η (3) δείχνει το προνόμιο που χορηγείται στο  $S_i$  θεμάτων για το αντικείμενο επεξεργασίας  $o_i$ . Η (4) ορίζει το χρονικό περιθώριο εκτέλεσης της  $tw_i$ .

Όταν ένας στόχος εκτελείτε σε μια ροή εργασίας, η συνδεδεμένη  $AT$  ερωτάται για να καθορίσει εάν μια νέα έγκριση μπορεί να χορηγηθεί. Το πολλαπλάσιο  $ATS$  μπορεί να συνδεθεί με έναν στόχο στην περίπτωση όπου περισσότερα από ένα θέματα απαιτούνται για να εκτελέσουν έναν στόχο, ή όταν πολλά αντικείμενα απαιτούνται για την εκτέλεση στόχου.

Το WAM διαμορφώνει έπειτα μια βάση έγκρισης ( $AB$ ) για να βοηθήσει στον προσδιορισμό των προηγούμενων εγκρίσεων που έχουν παραχθεί από το σύστημα. Τυπικά, το  $AB$  μπορεί να οριστεί ως ένα πεπερασμένο καθορισμένο  $AB = \{A1, A2, A3, \dots\}$  περιλαμβάνοντας όλες τις εγκρίσεις [AH96b] και ενημερώνεται δυναμικά. Η σημασία του  $AB$  θα προσδιοριστεί αργότερα στο επόμενο τμήμα όταν εξετάζεται ο δυναμικός χειρισμός περιορισμού.

### 6.3 Χωρισμός Περιορισμών Καθηκόντων επιβαλλόμενοι από το WAM

Ο χωρισμός των καθηκόντων περιλαμβάνει τη διατύπωση των περιορισμών σχετικά με τους οποίους τα θέματα εξουσιοδοτούνται για να εκτελέσουν ορισμένους στόχους στη διαδικασία ροής της δουλειάς. Το WAM εξετάζει αποτελεσματικά το χειρισμό αυτών των περιορισμών μέσω της προδιαγραφής των λογικών εκφράσεων. Ένα σύνολο πιθανών εγκρίσεων ( $PA_i$ ) δημιουργείται δυναμικά για να περιέχει όλες τις πιθανές εγκρίσεις για μια ιδιαίτερη περίπτωση στόχου. Αυτό το σύνολο προέρχεται από για το στόχο εν λόγω και εξαρτάται από την τρέχουσα κατάσταση του AB. Υπάρχουν πρώτιστα δύο τύποι περιορισμών που συνδέονται με την επιβολή του χωρισμού της αρχής καθηκόντων: κατηγορηματικοί και αποκλειστικοί περιορισμοί. Οι ακόλουθοι ορισμοί, όπως λαμβάνεται από τους Huang και Atluri [HA99], βοηθούν στη διάκριση κάθε μιας. Και στις δύο περιπτώσεις το χρονικό διάστημα που συνδέεται με τις συγκεκριμένες περιπτώσεις αφαιρείται για να καθορίσει τις πιθανές εγκρίσεις χωριστές από τις ιδιαίτερες περιπτώσεις.

**Ορισμός 6.3.1:** Λαμβάνοντας υπόψη ένα πρότυπο έγκρισης:  
 $AT(tw_i) = ((\tau_i, -), (\gamma_i, -), pr_i, [\tau_i, \tau_i])$ , ένα σύνολο πιθανών εγκρίσεων  $PA_i$  μπορεί να προέλθει από  $AT(tw_i)$ . Κάθε πιθανή έγκριση  $pa$  σε  $PA_i$  είναι μια τριάδα  $(s_i, o_i, pr_i)$ , όπως και  $si$

**Ορισμός 6.3.2:** Λαμβάνοντας υπόψη μια έγκριση  $A = (s, o, pr, [\tau_b, \tau_e])$  στο AB, καθορίζεται μια στατική προβολή  $\eta$   $A_{NT}$  του  $A$  ως  $A_{NT} = (s, o, pr)$ . Η στατική προβολή του AB είναι  $AB_{NT} = \{A_{NT1}, A_{NT2}, \dots\}$

Και οι δύο περιορισμοί εκφράζονται ως λογικές εκφράσεις της μορφής  $q \leftarrow p$ . Το  $q$  είναι μια λογική έκφραση που αποτελείται από το  $ANT$  και το  $p$  είναι μια μοναδική μεταβλητή όπου είναι  $pa$  ή  $\sim pa$  τέτοια ώστε  $pa \in PA_i$  κάποιου  $tw_i$ . Το  $S(p)$ , ή αλλιώς  $s(q)$ , ορίζει το σύνολο των αντικειμένων που ορίζονται στο  $pa \in PA_j$ .

Τυπικά, οι Huang και Atluri [HA99] καθορίζουν κάθε περιορισμό ως εξής:

**Αποκλειστικοί περιορισμοί:** το  $q$  είναι πάντα του τύπου  $\sim pa$  όπου  $pa$  ανήκει  $PA_j$  για το στόχο  $tw_j$ .

**Κατηγορηματικοί περιορισμοί:** το  $q$  είναι πάντα του τύπου  $pa$  όπου  $pa$  ανήκει  $PA_j$  για το στόχο  $tw_j$ .

Το AB αλλάζει δυναμικά και έτσι, έτσι κάνει το σύνολο επιλέξιμων χρηστών. Η τρέχουσα κατάσταση του AB επηρεάζεται από τη φύση των περιορισμών που επιβάλλονται στο στόχο, δηλαδή εάν είναι κατηγορηματικοί ή αποκλειστικοί. Μερικοί περιορισμοί, εντούτοις, δεν θα είναι σχετικοί με ορισμένες εγκρίσεις επειδή περιορίζουν τα θέματα που δεν περιλαμβάνονται σε αυτές τις εγκρίσεις.

Κατά συνέπεια, ένα σύνολο σχετικών περιορισμών,  $C_{twi}$ , διαμορφώνεται. Οι Huang και Atluri [HA99] το καθορίζουν:

**Ορισμός 6.3.3:** Όταν οι πολλαπλάσιοι περιορισμοί που επιβάλλονται σε έναν στόχο υπάρχουν, ένα σύνολο σχετικών περιορισμών καθορίζεται:  $C_{twi} = c_j \mid c_j$  ανήκει C που είναι της μορφής  $q_i \leftarrow p_j$  και  $q_i$  ανήκει  $PA_i$ .

Μόλις οι σχετικοί περιορισμοί έχουν προσδιοριστεί, ένα σύνολο επιλέξιμων θεμάτων καθορίζεται όσον αφορά το αντικείμενο  $o$  ως διαφορά μεταξύ εκείνων των θεμάτων ( $S_{pi}$ ) που επιτρέπονται για να υποθέσουν το σχετικό ρόλο και εκείνοι που έχουν αποκλειστεί ως αποτέλεσμα κάποιου περιορισμού:

**Ορισμός 6.3.4:** Λαμβάνοντας υπόψη ένα πρότυπο έγκρισης:  $AT(tw_i)((r_i, -), (\gamma_i, -), p_{fi}, [\tau_{li}, \tau_{ui}])$ , καθορίζουμε ένα σύνολο επιλέξιμων θεμάτων  $S_i^e(o)$  όπως:

1.  $S_i^e(o) = S_{pi}$ , εάν  $C_{twi} = \emptyset$
2.  $S_i^e(o) = S_1 \cap S_2 \cap S_3 \cap \dots \cap S_n$

Όπου κάθε

$S_k = S_{pi} - s(q_i)$ , εάν  $c_k: q_i \leftarrow p_j$  ανήκει  $c_{twi}$  είναι ένας αποκλειστικός περιορισμός και  $p_j$  είναι αληθές ως προς το  $AB_{NT}$  και

$S_k = s(q_i)$ , εάν  $c_k: q_i \leftarrow p_j$  ανήκει  $c_{twi}$  είναι ένας κατηγορηματικός περιορισμός και  $p_j$  είναι αληθές ως προς το  $AB_{NT}$ .

Ο καθορισμός δηλώνει ανωτέρω αυτόν στην περίπτωση των αποκλειστικών περιορισμών, το σύνολο επιλέξιμων θεμάτων λαμβάνεται από την αφαίρεση του απαγορευμένου θέματος από το σύνολο ρόλου θεμάτων που εγκρίνονται για να εκτελέσουν έναν στόχο. Στην περίπτωση των κατηγορηματικών περιορισμών, το σύνολο των αντικειμένων είναι αυτό που ορίζεται από το  $pa$ . Όπου κανένας περιορισμός δεν ισχύει για μια περίπτωση στόχου, το σύνολο επιλέξιμων θεμάτων είναι το ίδιο με το σύνολο θεμάτων που συνδέονται με το ρόλο εν λόγω. Κατά συνέπεια, ο χωρισμός των περιορισμών καθηκόντων επιβάλλεται με την παραγωγή ενός συνόλου επιλέξιμων θεμάτων με έναν τρόπο που λαμβάνει υπόψη τους δυναμικούς αποκλειστικούς περιορισμούς.

## 6.4 Χορήγηση και Ανάκληση εγκρίσεων

Οι εγκρίσεις χορηγούνται σύμφωνα με τον κανόνα επιχορήγησης. Αυτός ο κανόνας διευκρινίζει ότι μόνο εκείνα τα θέματα που περιλαμβάνονται μέσα στο σύνολο επιλέξιμων θεμάτων μπορούν να κερδίσουν την έγκριση. Οι αναφορές αυτής της διαδικασίας άμεσα αναφέρονται στο AT. Το χρονικό διάστημα για την ισχύ της έγκρισης ρυθμίζεται έπειτα σύμφωνα με το χρόνο της έναρξης στόχου και το χρόνο που απαιτείται για την ολοκλήρωση στόχου. Όταν μια έγκριση πρέπει να παραχθεί, το AT του στόχου πρέπει να ερωτηθεί στο χρόνο τρεξίματος. Το σύνολο επιλέξιμων θεμάτων πρέπει να ερωτηθεί για να ορίσει ένα θέμα στο

στόχο, αλλά μόνο όταν παραλαμβάνεται ένα αντικείμενο του τύπου που διευκρινίζεται από το στόχο. Οι εγκρίσεις ανακαλούνται σύμφωνα με το κανόνα ανάκλησης. Αρκετά απλά, αυτός ο κανόνας διευκρινίζει ότι οι εγκρίσεις ανακαλούνται με την ολοκλήρωση στόχου.

Οι ακόλουθοι ορισμοί εφαρμόζουν αυτούς τους κανόνες. Είναι σημαντικό να σημειωθεί ότι λαμβάνουν υπόψη το χωρισμό των περιορισμών καθηκόντων που συζητήθηκαν προηγουμένως. Ο ακόλουθος καθορισμός λαμβάνεται από Huang και Atluri [HA99]. Είναι μια επέκταση στο πρότυπο που πρότειναν [AH96b] δεδομένου ότι διευκολύνει το χωρισμό των καθηκόντων και μιας προσέγγισης RBAC.

Ορισμός 6.4.1: [Κανόνας παραγωγής έγκρισης για εκτεταμένο WAM]. Λαμβάνοντας υπόψη ένα πρότυπο έγκρισης  $AT(tw_i) = ((r_i, -), (\gamma_i, -), pr_i, [\tau_{li}, \tau_{ui}])$  μιας διαδικασίας  $tw_i$ , μια έγκριση  $A_i = (s, o, pr_i, [\tau_{bi}, \tau_{ei}])$  ορίζεται ως εξής:

**Κανόνας επιχορήγησης:** Έστω ότι το αντικείμενο  $x$  στέλνεται στο αντικείμενο  $y$  την χρονική στιγμή  $\tau_{ai}$  για να αρχίσει η  $tw_i$ .

Εάν  $x$  ανήκει  $O_{\gamma_i}$  και  $y$  ανήκει  $S_i^e(x)$  και  $\tau_{ui} \geq \tau_{ai}$ ,

$S_i \leftarrow y, o_i \leftarrow x, pr_i \leftarrow pr(AT)$

$\tau_{ei} \leftarrow \tau_{ui}$ ,

εάν  $\tau_{li} \geq \tau_{ai}$ ,

$\tau_{bi} \leftarrow \tau_{li}$ ,

διαφορετικά  $\tau_{bi} \leftarrow \tau_{ai}$ .

**Κανόνας Ανάκλησης:** Θεωρήστε ότι το  $w_i$  σταματάει στο  $\tau_{fi}$ , στο οποίο σημείο το  $o_i$  αποκολλάται από το  $tw_i$ .

Εάν  $\tau_{ui} \geq \tau_{fi}$

$\tau_{ei} \leftarrow \tau_{fi}$

Είναι ορατό πώς όχι μόνο αυτό το πρότυπο εξασφαλίζει ότι ο χωρισμός των περιορισμών καθηκόντων εφαρμόζεται, περιορίζει επίσης την ισχύ έγκρισης στο χρονικό διάστημα που απαιτείται για την εκτέλεση στόχου. Επιπλέον, διευκολύνει τη χρήση των ρόλων, όποιος είναι σημαντικός επίσης δεδομένου ότι απλοποιεί την προδιαγραφή των περιορισμών στη ροή της εργασίας.

## 6.5 Συμπέρασμα

Αυτό το κεφάλαιο αναφέρθηκε WAM που προτάθηκε από τους Atluri και Huang [AH96b] και το εκτεταμένο πρότυπό τους [HA99]. Συζητήθηκε πώς το WAM επιβάλλει τις αρχές ασφάλειας σε μια διαδικασία ροής εργασίας με την εξέταση της χορήγησης και της ανάκλησης των εγκρίσεων. Το WAM επιτρέπει τις δυναμικές εξαρτήσεις στόχου ταυτόχρονα εξασφαλίζοντας ότι αυτοί οι στόχοι εκτελούνται μέσα στους χρονικούς περιορισμούς που επιβάλλονται. Κατά συνέπεια, η ισχύς των εγκρίσεων είναι περιορισμένη στους χρονικούς περιορισμούς που καθορίζονται δυναμικά. Συνεπώς, το WAM χειρίζεται τα χρονικά προνόμια.

Το AT, συνδέεται με κάθε στόχο στη διαδικασία ροής της εργασίας που συζητήθηκε. Είναι ευθύνη αυτού να χορηγηθούν οι εγκρίσεις και να ανακληθούν στην ολοκλήρωση στόχου. Αυτό επιτυγχάνει το συγχρονισμό της ροής και της έγκρισης της ροής εργασίας, και είναι ένα σημαντικό πλεονέκτημα του προτύπου που οι καθαρές προσεγγίσεις του RBAC δεν έχουν. Το WAM είναι έτσι ένα κατάλληλο πρότυπο έγκρισης για τις ροές εργασίας όπου η πρόβλεψη των χρονικών διαστημάτων εκτέλεσης στόχου είναι πολύ δύσκολη. Από αυτό είναι εμφανές ότι το WAM προωθεί μια δυναμική προσέγγιση στη χορήγηση και την ανάκληση των εγκρίσεων.

Αυτό το κεφάλαιο εξήγησε περαιτέρω πώς το WAM διαμορφώνει ένα σύνολο επιλέξιμων θεμάτων με την εξέταση του χωρισμού των περιορισμών καθηκόντων. Τέλος, αυτό το πρότυπο διευκολύνει την ανάθεση των εγκρίσεων στους οργανωτικούς ρόλους, επιτρέποντας κατά συνέπεια στον ερευνητή για να το εφαρμόσει στο τρέχοντα παράδειγμα, το οποίο είναι βασισμένος σε μια οργανωτική ιεραρχία.

Το WAM ικανοποιεί τα περισσότερα κριτήρια που απαιτούνται για ένα βέλτιστο πρότυπο ελέγχου πρόσβασης, δηλαδή αυτά που είναι η επιβολή του χωρισμού των καθηκόντων και οι εγκρίσεις, ο χειρισμός των χρονικών περιορισμών, μια ρολοκεντρική εφαρμογή και ο συγχρονισμός της ροής της εργασίας με την έγκριση. Ο συγχρονισμός της ροής έγκρισης με τη ροή της εργασίας δεν μπορεί να συναντηθεί μέσω των καθαρών μηχανισμών RBAC. Μέχρι σήμερα, το WAM δεν διευκολύνει ρητά μια δομή υποβολής εκθέσεων και δεν ικανοποιεί την αντιπροσωπεία.

Η σημασία αυτού του κεφαλαίου είναι ότι παρουσίασε ένα παράδειγμα ενός πλαισίου ασφάλειας μέσα στο οποίο το προτεινόμενο πρότυπο αντιπροσωπειών μπορεί να λειτουργήσει. Επιπλέον, εισήγαγε τις αρχές που είναι θεμελιώδεις στην ασφαλή εφαρμογή του Προτύπου Έγκρισης Αντιπροσωπείας (DAM) που παρουσιάζεται.

## Παράρτημα

### 1. Κώδικας *Loading Workflow*

```

Imports System.Data.OracleClient
Imports System.IO
Imports System.Xml
Imports System.Xml.XPath
Imports System.DBNull

Public Class Workflow1
    Inherits SequentialWorkflowActivity
    Private ValidationFlag As Boolean
    Private Packs As String
    Private PacksFlag As Boolean
    Private LOCATION_DESC As String
    Private LOCATION_ID As Long
    Private Customer_ID As String
    Private Con As OracleConnection
    Private ConHistory As OracleConnection
    Private ConLoad As OracleConnection
    Private conString As String
    Private conStringHistory As String
    Private conStringLoad As String
    Private AppPath As String
    Private Record_Count As Long
    Private PACK_ID As Long
    Private PACK_DESC As String
    Private oReader As OracleDataReader
    Private CASE_ID As Long
    Private SEC_PERSON_ID As Long
    Private CLIENT_ID As Long
    Private CaseReader As OracleDataReader
    Private InactiveCaseReader As OracleDataReader
    Private ClientReader As OracleDataReader
    Private v_case_secID As String
    Private I As Integer
    Private v_client_sec_id As String
    Private v_client_name As String
    Private v_client_Address As String
    Private v_client_Town As String
    Private v_client_TK As String
    Private v_client_Afm As String
    Private v_client_Adt As String
    Private v_client_Birthdate As String
    Private v_client_Profession As String
    Private v_client_Phone As String
    Private v_client_ExAddress As String
    Private v_client_ExTown As String
    Private v_client_ExTK As String
    Private v_Guar_Name As String
    Private GuarReader As OracleDataReader
    Private v_Guar_Address As String
    Private v_Guar_Town As String
    Private v_Guar_TK As String

```



```

Private v_Guar_Afm As String
Private v_Guar_Adt As String
Private v_Guar_Birthdate As String
Private v_Guar_Profession As String
Private v_Guar_Phone As String
Private v_tollAmount1 As Double
Private v_tollAmount31 As Double
Private v_tollAmount91 As Double
Private v_tollAmount181 As Double
Private v_EnhmeroLentamount As Double
Private v_TollAmount As Double
Private v_NumberOfTolls_Left As Integer
Private v_NumberOfDelay_Tolls As Integer
Private v_LastOfeiliDays As Integer
Private v_LoanDuration As String
Private v_EndiksiPeriodouXaritos As String
Private v_LiksiPeriodouXaritos As String
Private v_lentamount As Double
Private v_min_pay_amount As Double
Private v_billing_date As Date
Private v_last_pay_amount As Double
Private v_last_pay_date As Date
Private v_branch_id As Integer
Private v_branch_name As String
Private v_poso_xorigisis As Double
Private v_date_xorigisis As String
Private v_bucket As Long
Private v_pack_type As Long
Private v_delequancy As Long
Private v_product_type As Long
Private v_flagAnathesis As Boolean
Private START_DATE As Date
Private PACK_ID_30 As Long
Private PACK_DESC_30 As String
Private PACK_ID_90 As Long
Private PACK_DESC_90 As String
Private v_flagItWasActive As Boolean

Private Sub Show_Status(ByVal Msg As String)
    Console.SetCursorPosition(0, 24)
    Console.ForegroundColor = ConsoleColor.Red
    Console.Write("
")
    Console.SetCursorPosition(0, 24)
    Console.Write(Msg)
    Console.ForegroundColor = ConsoleColor.White
End Sub

Private Sub Initialise_ExecuteCode(ByVal sender As
System.Object, ByVal e As System.EventArgs)
    Console.Clear()
    Console.SetCursorPosition(0, 0)
    Console.ForegroundColor = ConsoleColor.Red
    Console.Write("
Workflow      ")
    Console.ForegroundColor = ConsoleColor.White
    Customer_ID = "12"
    LOCATION_ID = 170
    LOCATION_DESC = "WORKING QUEUE"
    Demo Bank Loading

```

```

Do While ValidationFlag = False
    Console.SetCursorPosition(1, 3)
    Console.WriteLine("Create packages? ")
    Console.SetCursorPosition(1, 6)
    Console.WriteLine("0 for NO ")
    Console.SetCursorPosition(1, 7)
    Console.WriteLine("1 for YES ")
    Console.SetCursorPosition(1, 8)
    Console.WriteLine("Enter your choose: ")
    Console.ForegroundColor = ConsoleColor.Green
    Packs = Trim(Console.ReadLine())

    If Packs = "0" Then
        PacksFlag = False
        ValidationFlag = True
    ElseIf Packs = "1" Then
        ValidationFlag = True
        PacksFlag = True
    Else
        Console.SetCursorPosition(1, 18)
        Console.WriteLine("Your choose is not valid. Try to
run the program again! ")
        Console.Beep()
        Console.ReadLine()
    End If
Loop
End Sub

Private Sub ReadConfig_ExecuteCode(ByVal sender As
System.Object, ByVal e As System.EventArgs)
    AppPath =
System.IO.Path.GetDirectoryName(System.Reflection.Assembly.GetExec
utingAssembly().Location())
    Dim document As XPathDocument = New XPathDocument(AppPath
& "\wf_config.xml")
    Dim navigator As XPathNavigator =
document.CreateNavigator()
    Dim iterator As XPathNodeIterator =
navigator.Select("//Connection[@name='MainDB']")
    iterator.MoveNext()
    conString =
iterator.Current.GetAttribute("connectionString", "")
    iterator =
navigator.Select("//Connection[@name='HistoryDB']")
    iterator.MoveNext()
    conStringHistory =
iterator.Current.GetAttribute("connectionString", "")
    iterator =
navigator.Select("//Connection[@name='LoadDB']")
    iterator.MoveNext()
    conStringLoad =
iterator.Current.GetAttribute("connectionString", "")
End Sub

Private Sub OpenDB_ExecuteCode(ByVal sender As System.Object,
ByVal e As System.EventArgs)
    Con = New OracleConnection(conString) 'connect to
database
    Con.Open()

```

```

        ConHistory = New OracleConnection(conStringHistory)
'connect to database
        ConHistory.Open()
        ConLoad = New OracleConnection(conStringLoad) 'connect to
database
        ConLoad.Open()
    End Sub

    Private Sub Load_File_ExecuteCode(ByVal sender As
System.Object, ByVal e As System.EventArgs)
        Console.Clear()
        Show_Status("Loading CK... ")
        Dim cmd As String = "sqlldr.exe
USERID='MCS_LOAD/MCS_LOAD@MCS_LOAD ' "
        Dim cmd1 As String = "CONTROL='" & AppPath &
"\Load_Demo_Bank.CTL' "
        Dim cmd2 As String = "DATA='" & AppPath &
"\Load_Demo_Bank.txt' "
        Dim cmd3 As String = "BAD='" & AppPath &
"\Load_Demo_Bank.bad' "
        Dim cmd4 As String = "DISCARD='" & AppPath &
"\Load_Demo_Bank.dis' "
        Dim cmd5 As String = "DISCARDMAX= 5000 "
        Dim cmd6 As String = "ERRORS= 5000 "
        Dim cmdlast As String = "LOG='" & AppPath &
"\Load_Demo_Bank.log'" & vbCrLf
        Dim x As Integer = Shell(cmd & cmd1 & cmd2 & cmd3 & cmd4 &
cmd5 & cmd6 & cmdlast, vbNormalFocus, True)
    End Sub

    Private Sub Done_Processing_ExecuteCode_1(ByVal sender As
System.Object, ByVal e As System.EventArgs)
        Con.Close()
        ConHistory.Close()
        ConLoad.Close()
    End Sub

    Private Sub Records_Count_ExecuteCode(ByVal sender As
System.Object, ByVal e As System.EventArgs)
        Console.Clear()
        Console.SetCursorPosition(0, 0)
        Console.ForegroundColor = ConsoleColor.Red
        Console.Write("
File
")
        Console.ForegroundColor = ConsoleColor.White
        Show_Status("Getting Cases Count... ")
        Dim Sql As String = " SELECT COUNT(*) AS RECORD_COUNT FROM
Load_Demo_Bank_CASES "
        Dim Com As OracleCommand = New OracleCommand(Sql.ToUpper,
ConLoad)
        Dim lReader As OracleDataReader = Com.ExecuteReader()
        lReader.Read()
        Record_Count = CLng(lReader("RECORD_COUNT"))
        Console.SetCursorPosition(1, 3)
        Console.Write("Record Count = " & Record_Count)
    End Sub

    Private Sub Create_Pack_ExecuteCode(ByVal sender As
System.Object, ByVal e As System.EventArgs)

```

```

        Dim Sql As String = " SELECT SEQ_PACK.NEXTVAL AS
NEW_PACK_ID FROM DUAL "
        Show_Status("Creating New Pack for Loans... ")
        Dim Com As OracleCommand = New OracleCommand(Sql.ToUpper,
Con)

        Dim lReader As OracleDataReader = Com.ExecuteReader()
        lReader.Read()
        PACK_ID_30 = CLng(lReader("NEW_PACK_ID"))
        PACK_DESC_30 = Now.Year & "_" & Format(Now.Month, "00") &
"_" & Format(Now.Day, "00") & "_ (30-89)"
        Sql = " INSERT INTO PACKS "
        Sql = Sql & " (PACK_ID, INST_ID, CUSTOMER_ID, PACK_DESC,
PACK_TYPE_ID, "
        Sql = Sql & " PRODUCT_TYPE_ID, ACTIVE_CASES, START_DATE)
"
        Sql = Sql & " VALUES (" & PACK_ID_30 & ", 1, 12, '" &
PACK_DESC_30 & "', 12, "
        Sql = Sql & " 1, " & Record_Count & ", SYSDATE )"
        Com = New OracleCommand(Sql.ToUpper, Con)
        Com.ExecuteNonQuery()

        Sql = " SELECT SEQ_PACK.NEXTVAL AS NEW_PACK_ID FROM DUAL "
        Show_Status("Creating New Pack for Loans... ")
        Com = New OracleCommand(Sql.ToUpper, Con)
        lReader = Com.ExecuteReader()
        lReader.Read()
        PACK_ID_90 = CLng(lReader("NEW_PACK_ID"))
        PACK_DESC_90 = Now.Year & "_" & Format(Now.Month, "00") &
"_" & Format(Now.Day, "00") & "_ (90-179)"
        Sql = " INSERT INTO PACKS "
        Sql = Sql & " (PACK_ID, INST_ID, CUSTOMER_ID, PACK_DESC,
PACK_TYPE_ID, "
        Sql = Sql & " PRODUCT_TYPE_ID, ACTIVE_CASES, START_DATE)
"
        Sql = Sql & " VALUES (" & PACK_ID_90 & ", 1, 12, '" &
PACK_DESC_90 & "', 12, "
        Sql = Sql & " 1, " & Record_Count & ", SYSDATE )"
        Com = New OracleCommand(Sql.ToUpper, Con)
        Com.ExecuteNonQuery()
    End Sub

    Private Sub Select_Cases_ExecuteCode(ByVal sender As
System.Object, ByVal e As System.EventArgs)
        Show_Status("Selecting Cases... ")
        Dim Sql As String = " SELECT * FROM
MCS_LOAD.Load_Demo_Bank_CASES "
        Dim Com As OracleCommand = New OracleCommand(Sql.ToUpper,
ConLoad)
        oReader = Com.ExecuteReader()
    End Sub

    Private Sub findCase_ExecuteCode(ByVal sender As
System.Object, ByVal e As System.EventArgs)
        I = I + 1
        v_flagItWasActive = True
        v_case_secID = CStr(oReader("ACCOUNTNO"))
        Show_Status("Processing Case " & I & "/" & Record_Count)
        Console.SetCursorPosition(1, 3)
        Console.Write("Case : " & v_case_secID)
    End Sub

```

```

        Dim Sql As String

        Sql = " SELECT MAX(CASE_ID) CASE_ID FROM CASES WHERE
CASE_SEC_ID =' " & v_case_secID & "' AND CUSTOMER_ID = 12 AND
IS_ACTIVE = 1"
        Dim Com As OracleCommand = New OracleCommand(Sql.ToUpper,
Con)
        CaseReader = Com.ExecuteReader()
        CaseReader.Read()

        Try
            CASE_ID = CLng(CaseReader("CASE_ID"))
        Catch ex1 As Exception
            Sql = " SELECT MAX(CASE_ID) CASE_ID FROM CASES WHERE
CASE_SEC_ID =' " & v_case_secID & "' AND CUSTOMER_ID = 12 "
            Com = New OracleCommand(Sql.ToUpper, Con)
            InactiveCaseReader = Com.ExecuteReader()
            InactiveCaseReader.Read()
            Try
                CASE_ID = CLng(InactiveCaseReader("CASE_ID"))
                v_flagItWasActive = False
            Catch ex2 As Exception
                CASE_ID = 0
            End Try
        End Try
    End Sub

    Private Sub Variables_ExecuteCode(ByVal sender As
System.Object, ByVal e As System.EventArgs)
        'TYPES
        v_LastOfeiliDays = CInt(oReader("LASTOFEILIDAYS"))
        If v_LastOfeiliDays <= 29 Then
            v_bucket = 1
            v_delequancy = 2
            PACK_ID = PACK_ID_30
            PACK_DESC = PACK_DESC_30
        ElseIf v_LastOfeiliDays >= 30 And v_LastOfeiliDays <= 59
Then
            v_bucket = 2
            v_delequancy = 2
            PACK_ID = PACK_ID_30
            PACK_DESC = PACK_DESC_30
        ElseIf v_LastOfeiliDays >= 60 And v_LastOfeiliDays <= 89
Then
            v_bucket = 3
            v_delequancy = 3
            PACK_ID = PACK_ID_90
            PACK_DESC = PACK_DESC_90
        ElseIf v_LastOfeiliDays >= 90 And v_LastOfeiliDays <= 119
Then
            v_bucket = 4
            v_delequancy = 3
            PACK_ID = PACK_ID_90
            PACK_DESC = PACK_DESC_90
        ElseIf v_LastOfeiliDays >= 120 And v_LastOfeiliDays <= 149
Then
            v_bucket = 5
            v_delequancy = 3
            PACK_ID = PACK_ID_90

```

```

        PACK_DESC = PACK_DESC_90
    Else
        v_bucket = 6
        v_delequancy = 3
        PACK_ID = PACK_ID_90
        PACK_DESC = PACK_DESC_90
    End If

    v_product_type = 1
    v_pack_type = 12

    If CInt(oReader("FLAGANATHESIS")) = 0 Then
        v_flagAnathesis = False
    Else
        v_flagAnathesis = True
    End If

    'BRANCH
    Try
        v_branch_id = CInt(oReader("BRANCH"))
    Catch ex As Exception
        v_branch_id = 0
    End Try

    Try
        v_poso_xorigisis = CDb1(oReader("POSODANEIOY"))
    Catch ex As Exception
        v_poso_xorigisis = 0
    End Try

    Try
        v_date_xorigisis =
        Mid(CStr(oReader("ACCOUNTSTARTDATE")), 7, 2) & "/" &
        Mid(CStr(oReader("ACCOUNTSTARTDATE")), 5, 2) & "/" &
        Mid(CStr(oReader("ACCOUNTSTARTDATE")), 1, 4)
    Catch ex As Exception
        v_date_xorigisis = ""
    End Try

    'CASES
    Try
        v_lentamount = CDb1(oReader("LENTAMOUNTDANEIOY"))
    Catch ex As Exception
        v_lentamount = 0
    End Try

    Try
        v_min_pay_amount = CDb1(oReader("SYNOLIKODELAYPOSO"))
    Catch ex As Exception
        v_min_pay_amount = 0
    End Try

    Try
        v_billing_date =
        CDate(Mid(CStr(oReader("LASTOFEILIDATE")), 7, 2) & "/" &
        Mid(CStr(oReader("LASTOFEILIDATE")), 5, 2) & "/" &
        Mid(CStr(oReader("LASTOFEILIDATE")), 1, 4))
    Catch ex As Exception
        v_billing_date = CDate("01/01/1899")
    End Try

```



```

End Try

Try
    v_last_pay_amount = CDb1(oReader("LASTPAYAMOUNT"))
Catch ex As Exception
    v_last_pay_amount = 0
End Try

Try
    v_last_pay_date =
CDate(Mid(CStr(oReader("LASTPAYMENTDATE")), 7, 2) & "/" &
Mid(CStr(oReader("LASTPAYMENTDATE")), 5, 2) & "/" &
Mid(CStr(oReader("LASTPAYMENTDATE")), 1, 4))
Catch ex As Exception
    v_last_pay_date = CDate("01/01/1899")
End Try

'CASE EXTRA DATA
Try
    v_EnhmeroLentamount =
CDbl(oReader("ENHMEROLENTAMOUNT"))
Catch ex As Exception
    v_EnhmeroLentamount = 0
End Try

Try
    v_TollAmount = CDb1(oReader("TOLLAMOUNT"))
Catch ex As Exception
    v_TollAmount = 0
End Try

Try
    v_NumberOfTolls_Left =
CInt(oReader("NUMBEROFTOLLS_LEFT"))
Catch ex As Exception
    v_NumberOfTolls_Left = 0
End Try

Try
    v_NumberOfDelay_Tolls =
CInt(oReader("NUMBEROFDELAY_TOLLS"))
Catch ex As Exception
    v_NumberOfDelay_Tolls = 0
End Try

Try
    v_LastOfeiliDays = CInt(oReader("LASTOFEILIDAYS"))
Catch ex As Exception
    v_LastOfeiliDays = 0
End Try

Try
    v_LoanDuration = CStr(oReader("LOANDURATION"))
Catch ex As Exception
    v_LoanDuration = ""
End Try

Try

```

```

        v_EndiksiPeriodouXaritos =
CStr(oReader("ENDIKSIPERIODOUXARITOS"))
    Catch ex As Exception
        v_EndiksiPeriodouXaritos = ""
    End Try

    Try
        v_LiksiPeriodouXaritos =
Mid(CStr(oReader("LIKSIPERIODOUXARITOS")), 7, 2) & "/" &
Mid(CStr(oReader("LIKSIPERIODOUXARITOS")), 5, 2) & "/" &
Mid(CStr(oReader("LIKSIPERIODOUXARITOS")), 1, 4)
    Catch ex As Exception
        v_LiksiPeriodouXaritos = ""
    End Try

End Sub

Private Sub findClient_ExecuteCode(ByVal sender As
System.Object, ByVal e As System.EventArgs)
    v_client_sec_id = CStr(oReader("CLIENT_CODE"))
    Dim Sql As String = " SELECT CLIENT_ID FROM CLIENT WHERE
CLIENT_SEC_ID='" & v_client_sec_id & "' AND CUSTOMER_ID = 12 AND
INST_ID = 1 "
    Dim Com As OracleCommand = New OracleCommand(Sql.ToUpper,
Con)
    ClientReader = Com.ExecuteReader()
    ClientReader.Read()
End Sub

Private Sub DeleteClientExData_ExecuteCode(ByVal sender As
System.Object, ByVal e As System.EventArgs)
    CLIENT_ID = CLng(ClientReader("CLIENT_ID"))
    Dim Sql As String = " DELETE FROM CLIENT_EX_DATA WHERE
CLIENT_ID=" & CLIENT_ID & " AND CUSTOMER_ID = 12 "
    Dim Com As OracleCommand = New OracleCommand(Sql.ToUpper,
Con)
    Com.ExecuteNonQuery()
End Sub

Private Sub UpdateClient_ExecuteCode(ByVal sender As
System.Object, ByVal e As System.EventArgs)
    Try
        v_client_name = " '" &
Trim(CStr(oReader("CLIENTNAME")))) & "' "
    Catch ex As Exception
        v_client_name = "NULL"
    End Try

    Try
        v_client_Afm = " '" & Trim(CStr(oReader("CLIENTAFM")))
& "' "
    Catch ex As Exception
        v_client_Afm = "NULL"
    End Try

    Try
        v_client_Adt = " '" & Trim(CStr(oReader("CLIENTADT")))
& "' "
    Catch ex As Exception

```



```

        v_client_Adt = "NULL"
    End Try

    Dim Sql As String = " UPDATE mcs.CLIENT SET "
    Sql = Sql & " SURNAME      = " & v_client_name & ", "
    Sql = Sql & " IDENTITYNO = " & v_client_Adt & ", "
    Sql = Sql & " TAX_NO     = " & v_client_Afm & ", "
    Sql = Sql & " CLIENT_EXTRA_DATA = 1 "
    Sql = Sql & " WHERE client_id = " & CLIENT_ID
    Sql = Sql & " AND CUSTOMER_ID = 12 "
    Sql = Sql & " AND INST_ID = 1 "

    Dim Com As OracleCommand = New OracleCommand(Sql.ToUpper,
Con)
    Com.ExecuteNonQuery()
    End Sub

    Private Sub InsertAddresses_ExecuteCode(ByVal sender As
System.Object, ByVal e As System.EventArgs)
        'ADDRESS
        Try
            v_client_Address = " '" &
Trim(CStr(oReader("CLIENTADDRESS")) & "' "
        Catch ex As Exception
            v_client_Address = ""
        End Try

        Try
            v_client_Town = " '" &
Trim(CStr(oReader("CLIENTTOWN"))) & "' "
        Catch ex As Exception
            v_client_Town = "NULL"
        End Try

        Try
            v_client_TK = " '" & Trim(CStr(oReader("CLIENTTK"))) &
"' "
        Catch ex As Exception
            v_client_TK = ""
        End Try

        Dim Sql As String = " INSERT INTO CLIENT_ADDRESSES "
        Sql = Sql & "(ADDRESS_ID ,CLIENT_ID, CUSTOMER_ID
,ADDRESS_TYPE_ID ,ADDRESS ,TOWN ,ZIP ,MALLING_ADDRESS ) "
        Sql = Sql & " VALUES (SEQ_CLIENT_ADDRESS.NEXTVAL, " &
CLIENT_ID & ", 12, 1, " & v_client_Address & ", " & v_client_Town
& ", " & v_client_TK & ", 1) "
        Dim Com As OracleCommand = New OracleCommand(Sql.ToUpper,
Con)
        Try
            Com.ExecuteNonQuery()
        Catch ex As Exception

        End Try

        'EXTRA ADDRESS
        Try
            v_client_ExAddress = " '" &
Trim(CStr(oReader("EXTRAITADDRESS"))) & "' "

```

```

Catch ex As Exception
    v_client_ExAddress = "NULL"
End Try

Try
    v_client_ExTown = " " &
Trim(CStr(oReader("EXTRAITTOWN"))) & " "
Catch ex As Exception
    v_client_ExTown = "NULL"
End Try

Try
    v_client_ExTK = " " &
Trim(CStr(oReader("EXTRAITTK"))) & " "
Catch ex As Exception
    v_client_ExTK = "NULL"
End Try

Sql = " INSERT INTO CLIENT_ADDRESSES "
Sql = Sql & "(ADDRESS_ID ,CLIENT_ID, CUSTOMER_ID
,ADDRESS_TYPE_ID ,ADDRESS ,TOWN ,ZIP ,MALLING_ADDRESS ) "
Sql = Sql & "      VALUES (SEQ_CLIENT_ADDRESS.NEXTVAL, " &
CLIENT_ID & ", 12, 3 ," & v_client_ExAddress & ", " &
v_client_ExTown & ", " & v_client_ExTK & ", 0) "
Com = New OracleCommand(Sql.ToUpper, Con)
Try
    Com.ExecuteNonQuery()
Catch ex As Exception
End Try

'phone
Try
    v_client_Phone = " " &
RTrim(CStr(oReader("CLIENTPHONE"))) & " "
Catch ex As Exception
    v_client_Phone = ""
End Try

If v_client_Phone <> "" Then 'insert phone
    Sql = " INSERT INTO CLIENT_PHONES "
    Sql = Sql & "(PHONE_ID, CUSTOMER_ID, PHONE_TYPE_ID,
CLIENT_ID, PHONE) "
    Sql = Sql & " VALUES (SEQ_CLIENT_PHONE.NEXTVAL, 12, 1,
" & CLIENT_ID & ", " & v_client_Phone & ") "
    Com = New OracleCommand(Sql.ToUpper, Con)
    Com.ExecuteNonQuery()
End If

End Sub

Private Sub InsertClientExtraData_ExecuteCode(ByVal sender As
System.Object, ByVal e As System.EventArgs)

Dim Sql As String
Dim Com As OracleCommand
Try
    v_client_Birthdate = CStr(oReader("BIRTHDATE"))
Catch ex As Exception

```

```

        v_client_Birthdate = ""
    End Try

    Try
        v_client_Profession = " ' " &
Replace(Trim(CStr(oReader("PROFFSSION"))), "''", "'") & "' "
    Catch ex As Exception
        v_client_Profession = ""
    End Try
    If v_client_Birthdate <> "" Then
        Sql = " INSERT INTO CLIENT_EX_DATA "
        Sql = Sql & " (EX_DATA_ID, CUSTOMER_ID, CLIENT_ID,
EX_DATA_DESC, EX_DATA) "
        Sql = Sql & " VALUES "
        Sql = Sql & " (SEQ_CLIENT_EX_DATA.NEXTVAL, 12, " &
CLIENT_ID & ", ' "
        Sql = Sql & "HMEP/NIA FENNHEHS', ' "
        Sql = Sql & v_client_Birthdate & "' ) "
        Com = New OracleCommand(Sql.ToUpper, Con)
        Com.ExecuteNonQuery()
    End If

    If Trim(v_client_Profession) <> "" And
Trim(v_client_Profession) <> "' ' " Then
        Sql = " INSERT INTO CLIENT_EX_DATA "
        Sql = Sql & " (EX_DATA_ID, CUSTOMER_ID, CLIENT_ID,
EX_DATA_DESC, EX_DATA) "
        Sql = Sql & " VALUES "
        Sql = Sql & " (SEQ_CLIENT_EX_DATA.NEXTVAL, 12, " &
CLIENT_ID & ", ' "
        Sql = Sql & "EHATTEAMA', "
        Sql = Sql & v_client_Profession & " ) "
    Try
        Com = New OracleCommand(Sql.ToUpper, Con)
        Com.ExecuteNonQuery()
    Catch ex As Exception
    End Try
    End If
End Sub

Private Sub InsertClient_ExecuteCode(ByVal sender As
System.Object, ByVal e As System.EventArgs)

    Dim Sql As String = " SELECT SEQ_CLIENT.NEXTVAL AS
NEW_CLIENT_ID FROM DUAL "
    Dim Com As OracleCommand = New OracleCommand(Sql.ToUpper,
Con)

    Dim lReader As OracleDataReader = Com.ExecuteReader()
    lReader.Read()
    CLIENT_ID = CLng(lReader("NEW_CLIENT_ID"))

    Try
        v_client_name = " ' " &
Trim(CStr(oReader("CLIENTNAME"))) & "' "
    Catch ex As Exception
        v_client_name = "NULL"
    End Try

    Try

```

## Msc Thesis NetCentric Systems

```
        v_client_Afm = " '" & Trim(CStr(oReader("CLIENTAFM")))
& "' "
    Catch ex As Exception
        v_client_Afm = "NULL"
    End Try

    Try
        v_client_Adt = " '" & Trim(CStr(oReader("CLIENTADT")))
& "' "
    Catch ex As Exception
        v_client_Adt = "NULL"
    End Try

    Sql = " INSERT INTO mcs.CLIENT( "
    Sql = Sql & "CLIENT_ID, CUSTOMER_ID, INST_ID,
CLIENT_SEC_ID, SURNAME, "
    Sql = Sql & "TAX_NO, IDENTITYNO, CLIENT_EXTRA_DATA) "
    Sql = Sql & "VALUES ( "
    Sql = Sql & CLIENT_ID & " , 12, 1, '" & v_client_sec_id &
"' , '" & v_client_name & ", "
    Sql = Sql & v_client_Afm & " , '" & v_client_Adt & " , 1) "
    Com = New OracleCommand(Sql.ToUpper, Con)
    Com.ExecuteNonQuery()
End Sub

Private Sub FindGuar_ExecuteCode(ByVal sender As
System.Object, ByVal e As System.EventArgs)
    Dim Sql As String = " SELECT SEC_PERSON_ID FROM SEC_PERSON
WHERE SURNAME =' " & Trim(CStr(oReader("GUARANTEENAME"))) & "' AND
CUSTOMER_ID = 12 AND SEC_PERSON_TYPE_ID = 1 AND CASE_ID = " &
CASE_ID
    Dim Com As OracleCommand = New OracleCommand(Sql.ToUpper,
Con)
    GuarReader = Com.ExecuteReader()
    GuarReader.Read()
End Sub

Private Sub DelGuarExtraData_ExecuteCode(ByVal sender As
System.Object, ByVal e As System.EventArgs)
    SEC_PERSON_ID = CLng(GuarReader("SEC_PERSON_ID"))
    Dim Sql As String = " DELETE FROM SEC_PERSON_EX_DATA WHERE
SEC_PERSON_ID=" & SEC_PERSON_ID & " AND CUSTOMER_ID = 12 "
    Dim Com As OracleCommand = New OracleCommand(Sql.ToUpper,
Con)
    Com.ExecuteNonQuery()
End Sub

Private Sub UpdateGuar_ExecuteCode(ByVal sender As
System.Object, ByVal e As System.EventArgs)
    GuarReader.Read()

    Try
        v_Guar_Adt = " '" &
Trim(CStr(oReader("GUARANTEEADT"))) & "' "
    Catch ex As Exception
        v_Guar_Adt = "NULL"
    End Try
```

```

Try
    v_Guar_Afm = Trim(CStr(oReader("GUARANTEEAFM")))
Catch ex As Exception
    v_Guar_Afm = "NULL"
End Try

Dim Sql As String = " UPDATE mcs.SEC_PERSON SET "
Sql = Sql & " IDENTITYNO = " & v_Guar_Adt & ", "
Sql = Sql & " TAX_NO = " & v_Guar_Afm & ", "
Sql = Sql & " SEC_PERSON_EXTRA_DATA = 1 "
Sql = Sql & " WHERE SEC_PERSON_ID = " & SEC_PERSON_ID
Sql = Sql & " AND CASE_ID = " & CASE_ID
Sql = Sql & " AND Customer_ID = 12 "
Sql = Sql & " AND INST_ID = 1 "

Dim Com As OracleCommand = New OracleCommand(Sql.ToUpper,
Con)
Com.ExecuteNonQuery()

Sql = "UPDATE CASES SET HAS_SEC_PERSON = 1 WHERE
CUSTOMER_ID = 12 AND CASE_ID = " & CASE_ID
Com = New OracleCommand(Sql.ToUpper, Con)
Com.ExecuteNonQuery()
End Sub

Private Sub InsertGuarExData_ExecuteCode(ByVal sender As
System.Object, ByVal e As System.EventArgs)
    Dim Sql As String
    Dim Com As OracleCommand
    Try
        v_Guar_Birthdate = " '" &
Trim(CStr(oReader("GUARANTEE BIRTHDATE")) & "' "
        Catch ex As Exception
            v_Guar_Birthdate = ""
        End Try

    Try
        v_Guar_Profession = " '" &
Trim(CStr(oReader("GUARANTEE PROFESSION")) & "' "
        Catch ex As Exception
            v_Guar_Profession = ""
        End Try

    If v_Guar_Birthdate <> "" Then
        Sql = " INSERT INTO SEC_PERSON_EX_DATA "
        Sql = Sql & " (EX_DATA_ID, CUSTOMER_ID, SEC_PERSON_ID,
EX_DATA_DESC, EX_DATA) "
        Sql = Sql & " VALUES "
        Sql = Sql & " (SEQ_SEC_PERSON_EX_DATA.NEXTVAL, 12, " &
SEC_PERSON_ID & ", '"
        Sql = Sql & "HMEP/NIA ΓΕΝΝΗΣΗΣ', "
        Sql = Sql & CStr(v_Guar_Birthdate) & " ) "
        Com = New OracleCommand(Sql.ToUpper, Con)
        Com.ExecuteNonQuery()
    End If

    If v_Guar_Profession <> "" Then
        Sql = " INSERT INTO SEC_PERSON_EX_DATA "

```

## Msc Thesis NetCentric Systems

```
        Sql = Sql & " (EX_DATA_ID, CUSTOMER_ID, SEC_PERSON_ID,
EX_DATA_DESC, EX_DATA) "
        Sql = Sql & " VALUES "
        Sql = Sql & " (SEQ_SEC_PERSON_EX_DATA.NEXTVAL, 12, " &
SEC_PERSON_ID & ", "
        Sql = Sql & "'EIIATTEAMA', "
        Sql = Sql & v_Guar_Profession.ToString & ") "
        Try
            Com = New OracleCommand(Sql.ToUpper, Con)
            Com.ExecuteNonQuery()
        Catch ex As Exception
        End Try

    End If
End Sub

Private Sub InsertGuar_ExecuteCode(ByVal sender As
System.Object, ByVal e As System.EventArgs)

    Try
        v_Guar_Adt = " '" &
Trim(CStr(oReader("GUARANTEEADT")) & "' "
        Catch ex2 As Exception
            v_Guar_Adt = "NULL"
        End Try

    Try
        v_Guar_Afm = Trim(CStr(oReader("GUARANTEEAFM")))
        Catch ex3 As Exception
            v_Guar_Afm = ""
        End Try
        If v_Guar_Name <> "" Then
            Dim Sql As String = " SELECT SEQ_SEC_PERSON.NEXTVAL AS
SEC_PERSON_ID FROM DUAL "
            Dim Com As OracleCommand = New
OracleCommand(Sql.ToUpper, Con)
            Dim lReader As OracleDataReader = Com.ExecuteReader()
            lReader.Read()
            SEC_PERSON_ID = CLng(lReader("SEC_PERSON_ID"))

            Sql = " INSERT INTO mcs.SEC_PERSON( "
            Sql = Sql & "SEC_PERSON_ID, SEC_PERSON_TYPE_ID,
CUSTOMER_ID, INST_ID, SURNAME, TAX_NO, "
            Sql = Sql & "IDENTITYNO, SEC_PERSON_EXTRA_DATA,
CASE_ID) "
            Sql = Sql & "VALUES ( "
            Sql = Sql & SEC_PERSON_ID & " ,1, 12, 1, '" &
v_Guar_Name & "', '"
            Sql = Sql & v_Guar_Afm & "', " & v_Guar_Adt & " , 1, "
& CASE_ID & ") "

            Com = New OracleCommand(Sql.ToUpper, Con)
            Com.ExecuteNonQuery()

            Sql = "UPDATE CASES SET HAS_SEC_PERSON = 1 WHERE
CUSTOMER_ID = 12 AND CASE_ID = " & CASE_ID
            Com = New OracleCommand(Sql.ToUpper, Con)
            Com.ExecuteNonQuery()
        End If
    End Try
End Sub
```

## Msc Thesis NetCentric Systems

```
        End If
    End Sub

    Private Sub Delete_Tolls_ExecuteCode(ByVal sender As
System.Object, ByVal e As System.EventArgs)
        Dim Sql As String = " DELETE FROM TOLLS WHERE CASE_ID=" &
CASE_ID & " AND CUSTOMER_ID = 12 "
        Dim Com As OracleCommand = New OracleCommand(Sql.ToUpper,
Con)
        Com.ExecuteNonQuery()
    End Sub

    Private Sub Insert_Tolls_ExecuteCode(ByVal sender As
System.Object, ByVal e As System.EventArgs)
        Try
            v_tollAmount1 = CDb1(oReader("POS01_30"))
        Catch ex As Exception
            v_tollAmount1 = 0
        End Try

        Try
            v_tollAmount31 = CDb1(oReader("POS031_90"))
        Catch ex As Exception
            v_tollAmount31 = 0
        End Try

        Try
            v_tollAmount91 = CDb1(oReader("POS091_180"))
        Catch ex As Exception
            v_tollAmount91 = 0
        End Try

        Try
            v_tollAmount181 = CDb1(oReader("POS0181_AFTER"))
        Catch ex As Exception
            v_tollAmount181 = 0
        End Try

        If v_tollAmount1 <> 0 Then
            Dim Sql As String = " INSERT INTO TOLLS "
            Sql = Sql & " (TOLLS_ID, CASE_ID, CUSTOMER_ID,
TOLLDATE, TOLLCAPTIAL, TOLLTOTAL, INST_ID, PACK_ID,
TOLL_STATUS_ID) "
            Sql = Sql & " VALUES "
            Sql = Sql & " (SEQ_TOLLS.NEXTVAL," & CASE_ID & ", 12,
SYSDATE, " & Replace(CStr(v_tollAmount1), ",", ".") & ", " &
Replace(CStr(v_tollAmount31), ",", ".") & ", 1," & PACK_ID & ",
100 ) "
            Dim Com As OracleCommand = New
OracleCommand(Sql.ToUpper, Con)
            Com.ExecuteNonQuery()
        End If

        If v_tollAmount31 <> 0 Then
            Dim Sql As String = " INSERT INTO TOLLS "
            Sql = Sql & " (TOLLS_ID, CASE_ID, CUSTOMER_ID,
TOLLDATE, TOLLCAPTIAL, TOLLTOTAL, INST_ID, PACK_ID,
TOLL_STATUS_ID) "
            Sql = Sql & " VALUES "
```

## Msc Thesis NetCentric Systems

```
        Sql = Sql & " (SEQ_TOLLS.NEXTVAL," & CASE_ID & ", 12,
SYSDATE, " & Replace(CStr(v_tollAmount1), ",", ".") & " , " &
Replace(CStr(v_tollAmount31), ",", ".") & " , 1," & PACK_ID & ",
101 ) "
        Dim Com As OracleCommand = New
OracleCommand(Sql.ToUpper, Con)
        Com.ExecuteNonQuery()
    End If

    If v_tollAmount91 <> 0 Then
        Dim Sql As String = " INSERT INTO TOLLS "
        Sql = Sql & " (TOLLS_ID, CASE_ID, CUSTOMER_ID,
TOLLDATE, TOLLCAPTIAL, TOLLTOTAL, INST_ID, PACK_ID,
TOLL_STATUS_ID) "
        Sql = Sql & " VALUES "
        Sql = Sql & " (SEQ_TOLLS.NEXTVAL," & CASE_ID & ", 12,
SYSDATE, " & Replace(CStr(v_tollAmount91), ",", ".") & " , " &
Replace(CStr(v_tollAmount91), ",", ".") & " , 1," & PACK_ID & ",
102 ) "
        Dim Com As OracleCommand = New
OracleCommand(Sql.ToUpper, Con)
        Com.ExecuteNonQuery()
    End If

    If v_tollAmount181 <> 0 Then
        Dim Sql As String = " INSERT INTO TOLLS "
        Sql = Sql & " (TOLLS_ID, CASE_ID, CUSTOMER_ID,
TOLLDATE, TOLLCAPTIAL, TOLLTOTAL, INST_ID, PACK_ID,
TOLL_STATUS_ID) "
        Sql = Sql & " VALUES "
        Sql = Sql & " (SEQ_TOLLS.NEXTVAL," & CASE_ID & ", 12,
SYSDATE, " & Replace(CStr(v_tollAmount181), ",", ".") & " , " &
Replace(CStr(v_tollAmount181), ",", ".") & " , 1," & PACK_ID & ",
103 ) "
        Dim Com As OracleCommand = New
OracleCommand(Sql.ToUpper, Con)
        Com.ExecuteNonQuery()
    End If
End Sub

Private Sub InsertCaseExtraData_ExecuteCode(ByVal sender As
System.Object, ByVal e As System.EventArgs)

    Dim Sql As String = " INSERT INTO CASE_EX_DATA "
    Sql = Sql & " (EX_DATA_ID, CUSTOMER_ID, CASE_ID,
EX_DATA_DESC, EX_DATA) "
    Sql = Sql & " VALUES "
    Sql = Sql & " (SEQ_CASE_EX_DATA.NEXTVAL, 12, " & CASE_ID &
", '"
    Sql = Sql & "'00 ΗΜΕΡΕΣ ΚΑΘΥΣΤΕΡΗΣΗ', '"
    Sql = Sql & v_LastOfeiliDays & "' ) "
    Dim Com As OracleCommand = New OracleCommand(Sql.ToUpper,
Con)
    Com.ExecuteNonQuery()

    Sql = " INSERT INTO CASE_EX_DATA "
    Sql = Sql & " (EX_DATA_ID, CUSTOMER_ID, CASE_ID,
EX_DATA_DESC, EX_DATA) "
    Sql = Sql & " VALUES "
```



## Msc Thesis NetCentric Systems

```
Sql = Sql & " (SEQ_CASE_EX_DATA.NEXTVAL, 12, " & CASE_ID &
", '"
Sql = Sql & "01 ΠΟΣΟ ΧΟΡΗΓΗΣΗΣ', '"
Sql = Sql & v_poso_xorigisis & "' ) "
Com = New OracleCommand(Sql.ToUpper, Con)
Com.ExecuteNonQuery()

Sql = " INSERT INTO CASE_EX_DATA "
Sql = Sql & " (EX_DATA_ID, CUSTOMER_ID, CASE_ID,
EX_DATA_DESC, EX_DATA) "
Sql = Sql & " VALUES "
Sql = Sql & " (SEQ_CASE_EX_DATA.NEXTVAL, 12, " & CASE_ID &
", '"
Sql = Sql & "02 ΗΜ/ΝΙΑ ΧΟΡΗΓΗΣΗΣ', '"
Sql = Sql & v_date_xorigisis & "' ) "
Com = New OracleCommand(Sql.ToUpper, Con)
Com.ExecuteNonQuery()

Sql = " INSERT INTO CASE_EX_DATA "
Sql = Sql & " (EX_DATA_ID, CUSTOMER_ID, CASE_ID,
EX_DATA_DESC, EX_DATA) "
Sql = Sql & " VALUES "
Sql = Sql & " (SEQ_CASE_EX_DATA.NEXTVAL, 12, " & CASE_ID &
", '"
Sql = Sql & "03 ΥΠΟΚΑΤΑΣΤΗΜΑ ΧΟΡΗΓΗΣΗΣ', '"
Sql = Sql & v_branch_id & " # " & Replace(v_branch_name,
"', "' ) & "' ) "
Com = New OracleCommand(Sql.ToUpper, Con)
Com.ExecuteNonQuery()

Sql = " INSERT INTO CASE_EX_DATA "
Sql = Sql & " (EX_DATA_ID, CUSTOMER_ID, CASE_ID,
EX_DATA_DESC, EX_DATA) "
Sql = Sql & " VALUES "
Sql = Sql & " (SEQ_CASE_EX_DATA.NEXTVAL, 12, " & CASE_ID &
", '"
Sql = Sql & "04 ΕΝΗΜΕΡΟ ΠΟΣΟ ΟΦΕΙΛΗΣ', '"
Sql = Sql & v_EnhmeroLentamount & "' ) "
Com = New OracleCommand(Sql.ToUpper, Con)

Sql = " INSERT INTO CASE_EX_DATA "
Sql = Sql & " (EX_DATA_ID, CUSTOMER_ID, CASE_ID,
EX_DATA_DESC, EX_DATA) "
Sql = Sql & " VALUES "
Sql = Sql & " (SEQ_CASE_EX_DATA.NEXTVAL, 12, " & CASE_ID &
", '"
Sql = Sql & "05 ΠΟΣΟ ΔΟΣΗΣ', '"
Sql = Sql & v_TollAmount & "' ) "
Com = New OracleCommand(Sql.ToUpper, Con)
Com.ExecuteNonQuery()

Sql = " INSERT INTO CASE_EX_DATA "
Sql = Sql & " (EX_DATA_ID, CUSTOMER_ID, CASE_ID,
EX_DATA_DESC, EX_DATA) "
Sql = Sql & " VALUES "
Sql = Sql & " (SEQ_CASE_EX_DATA.NEXTVAL, 12, " & CASE_ID &
", '"
Sql = Sql & "06 ΔΙΑΡΚΕΙΑ ΔΑΝΕΙΟΥ', '"
Sql = Sql & v_LoanDuration & "' ) "
```

## Msc Thesis NetCentric Systems

```
Com = New OracleCommand(Sql.ToUpper, Con)
Com.ExecuteNonQuery()

Sql = " INSERT INTO CASE_EX_DATA "
Sql = Sql & " (EX_DATA_ID, CUSTOMER_ID, CASE_ID,
EX_DATA_DESC, EX_DATA) "
Sql = Sql & " VALUES "
Sql = Sql & " (SEQ_CASE_EX_DATA.NEXTVAL, 12, " & CASE_ID &
", '"
Sql = Sql & "'07 AP.YΠΟΛΕΙΠΟΜΕΝΩΝ ΔΟΣΕΩΝ', '"
Sql = Sql & v_NumberOfTolls_Left & "' ) "
Com = New OracleCommand(Sql.ToUpper, Con)
Com.ExecuteNonQuery()

Sql = " INSERT INTO CASE_EX_DATA "
Sql = Sql & " (EX_DATA_ID, CUSTOMER_ID, CASE_ID,
EX_DATA_DESC, EX_DATA) "
Sql = Sql & " VALUES "
Sql = Sql & " (SEQ_CASE_EX_DATA.NEXTVAL, 12, " & CASE_ID &
", '"
Sql = Sql & "'08 AP.ΚΑΘΥΣΤΕΡΗΜΕΝΩΝ ΔΟΣΕΩΝ', '"
Sql = Sql & v_NumberOfDelay_Tolls & "' ) "
Com = New OracleCommand(Sql.ToUpper, Con)
Com.ExecuteNonQuery()

Sql = " INSERT INTO CASE_EX_DATA "
Sql = Sql & " (EX_DATA_ID, CUSTOMER_ID, CASE_ID,
EX_DATA_DESC, EX_DATA) "
Sql = Sql & " VALUES "
Sql = Sql & " (SEQ_CASE_EX_DATA.NEXTVAL, 12, " & CASE_ID &
", '"
Sql = Sql & "'09 ΕΝΔΕΙΞΗ ΠΕΡΙΟΔΟΥ ΧΑΡΙΤΟΣ # ΗΜ/ΝΙΑ ΛΗΞΗΣ
ΠΕΡΙΟΔΟΥ ΧΑΡΙΤΟΣ', '"
Sql = Sql & v_EndiksiPeriodouXaritos & " # " &
v_LiksiPeriodouXaritos & "' ) "
Com = New OracleCommand(Sql.ToUpper, Con)
Com.ExecuteNonQuery()
v_Guar_Name = Trim(CStr(oReader("GUARANTEENAME")))
End Sub

Private Sub DelCaseExtraData_ExecuteCode(ByVal sender As
System.Object, ByVal e As System.EventArgs)
Dim Sql As String = " DELETE FROM CASE_EX_DATA WHERE
CASE_ID=" & CASE_ID & " AND CUSTOMER_ID = 12 "
Dim Com As OracleCommand = New OracleCommand(Sql.ToUpper,
Con)
Com.ExecuteNonQuery()
End Sub

Private Sub Insert_10399_ExecuteCode(ByVal sender As
System.Object, ByVal e As System.EventArgs)
Dim Sql As String = " INSERT INTO CONTACT "
Sql = Sql & " (CONTACT_ID, ACTION_CODE, CASE_ID,
CUSTOMER_ID, INST_ID, USER_ID, COMMENTS, ACTION_TYPE_ID ) "
Sql = Sql & " VALUES (SEQ_CONTACT.NEXTVAL, 10399, " &
CASE_ID & ", 12, 1, 1, NULL, 3) "
Dim Com As OracleCommand = New OracleCommand(Sql.ToUpper,
Con)
Com.ExecuteNonQuery()
```

## Msc Thesis NetCentric Systems

End Sub

```
Private Sub _10399_ExecuteCode(ByVal sender As System.Object,
ByVal e As System.EventArgs)
    Dim Sql As String = " INSERT INTO CONTACT "
    Sql = Sql & "(CONTACT_ID, ACTION_CODE, CASE_ID,
CUSTOMER_ID, INST_ID, USER_ID, COMMENTS, ACTION_TYPE_ID ) "
    Sql = Sql & "          VALUES (SEQ_CONTACT.NEXTVAL, 10399, " &
CASE_ID & ", 12, 1, 1, NULL, 3) "
    Dim Com As OracleCommand = New OracleCommand(Sql.ToUpper,
Con)
    Com.ExecuteNonQuery()
End Sub
```

```
Private Sub Insert_Case_ExecuteCode(ByVal sender As
System.Object, ByVal e As System.EventArgs)
    Dim Sql As String = " SELECT SEQ_CASE.NEXTVAL AS
NEW_CASE_ID FROM DUAL "
    Dim Com As OracleCommand = New OracleCommand(Sql.ToUpper,
Con)

    Dim lReader As OracleDataReader = Com.ExecuteReader()
    lReader.Read()
    CASE_ID = CLng(lReader("NEW_CASE_ID"))
    Sql = ""
    Sql = " INSERT INTO MCS.CASES "
    Sql = Sql & "( "
    Sql = Sql & "APPLICATION_NO, "
    Sql = Sql & "CASE_ID, "
    Sql = Sql & "CLIENT_ID, "
    Sql = Sql & "CUSTOMER_ID, "
    Sql = Sql & "CUSTOMER_NAME, "
    Sql = Sql & "INST_ID, "
    Sql = Sql & "CASE_SEC_ID, "
    Sql = Sql & "PACK_ID, "
    Sql = Sql & "PACK_DESC, "
    Sql = Sql & "INIT_LENT_CAPITAL, "
    Sql = Sql & "INIT_LENT_INTEREST, "
    Sql = Sql & "INIT_LENT_EXPENSES, "
    Sql = Sql & "TOTAL_INIT_LENT_AMOUNT, "
    Sql = Sql & "INIT_MIN_PAY_AMOUNT, "
    Sql = Sql & "LENT_CAPITAL, "
    Sql = Sql & "LENT_INTEREST, "
    Sql = Sql & "LENT_EXPENSES, "
    Sql = Sql & "TOTAL_LENT_AMOUNT, "
    Sql = Sql & "MIN_PAY_AMOUNT, "
    Sql = Sql & "LAST_PAY_DATE, "
    Sql = Sql & "LAST_PAY_AMOUNT, "
    Sql = Sql & "PRODUCT_TYPE_ID, "
    Sql = Sql & "PACK_TYPE_ID, "
    Sql = Sql & "IS_LEGAL_CASE, "
    Sql = Sql & "DELEQUANCY_ID, "
    Sql = Sql & "INIT_BUCKET_ID, "
    Sql = Sql & "BUCKET_ID, "
    Sql = Sql & "CASE_EXT_DATA, "
    Sql = Sql & "BILLING_DATE, "
    Sql = Sql & "LAST_UPDATE_DATE, "
    Sql = Sql & "CREATE_DATE, "
    Sql = Sql & "LENT_DATE, "
    Sql = Sql & "IS_VISABLE, "
```

```

Sql = Sql & "IS_ACTIVE, "
Sql = Sql & "LOCATION_ID, "
Sql = Sql & "LOCATION_DESC, "
Sql = Sql & "WORKFLOW_NAME, "
Sql = Sql & "HAS_TOLLS, "
Sql = Sql & "USER_ID"
Sql = Sql & ") "
Sql = Sql & "VALUES "
Sql = Sql & "( "
Sql = Sql & "" & v_LastOfeiliDays & ", "
Sql = Sql & CASE_ID & ", "
Sql = Sql & CLIENT_ID & ", "
Sql = Sql & "12, "
Sql = Sql & "'ALPHA BANK - REAL ESTATE', "
Sql = Sql & "1, "
Sql = Sql & "" & v_case_secID & ", "
Sql = Sql & PACK_ID & ", "
Sql = Sql & "" & PACK_DESC & ", "
Sql = Sql & Replace(CStr(v_lentamount), ",", ".") & ", "
Sql = Sql & " 0 , "
Sql = Sql & " 0 , "
Sql = Sql & Replace(CStr(v_lentamount), ",", ".") & ", "
Sql = Sql & Replace(CStr(v_min_pay_amount), ",", ".") & ",
"

Sql = Sql & Replace(CStr(v_lentamount), ",", ".") & ", "
Sql = Sql & " 0 , "
Sql = Sql & " 0 , "
Sql = Sql & Replace(CStr(v_lentamount), ",", ".") & ", "
Sql = Sql & Replace(CStr(v_min_pay_amount), ",", ".") & ",
"

Sql = Sql & "" & v_last_pay_date & "" , "
Sql = Sql & Replace(CStr(v_last_pay_amount), ",", ".") &
", "

Sql = Sql & v_product_type & " , "
Sql = Sql & v_pack_type & " , "
Sql = Sql & "0, "
Sql = Sql & v_delequancy & ", "
Sql = Sql & v_bucket & ", "
Sql = Sql & v_bucket & ", "
Sql = Sql & "1, "
Sql = Sql & "" & v_billing_date & "" , "
Sql = Sql & "sysdate, "
Sql = Sql & "sysdate, "
Sql = Sql & "sysdate, "
Sql = Sql & "1, "
If v_flagAnathesis = True Then
    Sql = Sql & "1, "
    Sql = Sql & LOCATION_ID & ", "
    Sql = Sql & "" & LOCATION_DESC & ", "
Else
    Sql = Sql & "0, "
    Sql = Sql & " 188, "
    Sql = Sql & "'INACTIVE QUEUE', "
End If
Sql = Sql & "'Alpha_Home_v1', "
Sql = Sql & "1, "
Sql = Sql & "1"

```

## Msc Thesis NetCentric Systems

```
Sql = Sql & ") "  
  
Com = New OracleCommand(Sql.ToUpper, Con)  
Com.ExecuteNonQuery()  
End Sub  
  
Private Sub Insert_Case_History_ExecuteCode(ByVal sender As  
System.Object, ByVal e As System.EventArgs)  
Dim Sql As String = " INSERT INTO CASE_HISTORY "  
Sql = Sql & "( "  
Sql = Sql & "HISTORY_ID, "  
Sql = Sql & "CASE_ID, "  
Sql = Sql & "CLIENT_ID, "  
Sql = Sql & "CUSTOMER_ID, "  
Sql = Sql & "CUSTOMER_NAME, "  
Sql = Sql & "INST_ID, "  
Sql = Sql & "CASE_SEC_ID, "  
Sql = Sql & "PACK_ID, "  
Sql = Sql & "PACK_DESC, "  
Sql = Sql & "INIT_LENT_CAPITAL, "  
Sql = Sql & "INIT_LENT_INTEREST, "  
Sql = Sql & "INIT_LENT_EXPENSES, "  
Sql = Sql & "TOTAL_INIT_LENT_AMOUNT, "  
Sql = Sql & "INIT_MIN_PAY_AMOUNT, "  
Sql = Sql & "LENT_CAPITAL, "  
Sql = Sql & "LENT_INTEREST, "  
Sql = Sql & "LENT_EXPENSES, "  
Sql = Sql & "TOTAL_LENT_AMOUNT, "  
Sql = Sql & "MIN_PAY_AMOUNT, "  
Sql = Sql & "LAST_PAY_DATE, "  
Sql = Sql & "LAST_PAY_AMOUNT, "  
Sql = Sql & "PRODUCT_TYPE_ID, "  
Sql = Sql & "PACK_TYPE_ID, "  
Sql = Sql & "IS_LEGAL_CASE, "  
Sql = Sql & "DELEQUANCY_ID, "  
Sql = Sql & "INIT_BUCKET_ID, "  
Sql = Sql & "BUCKET_ID, "  
Sql = Sql & "BILLING_DATE, "  
Sql = Sql & "LAST_UPDATE_DATE, "  
Sql = Sql & "CREATE_DATE, "  
Sql = Sql & "IS_ACTIVE "  
Sql = Sql & ") "  
Sql = Sql & "VALUES "  
Sql = Sql & "( "  
Sql = Sql & "SEQ_CASE_HISTORY.nextval, "  
Sql = Sql & CASE_ID & ", "  
Sql = Sql & CLIENT_ID & ", "  
Sql = Sql & "12, "  
Sql = Sql & "'ALPHA BANK', "  
Sql = Sql & "1, "  
Sql = Sql & "" & v_case_secID & "", "  
Sql = Sql & PACK_ID & ", "  
Sql = Sql & "" & PACK_DESC & "", "  
Sql = Sql & Replace(CStr(v_lentamount), ",", ".") & ", "  
Sql = Sql & " 0 , "  
Sql = Sql & " 0 , "  
Sql = Sql & Replace(CStr(v_lentamount), ",", ".") & ", "  
Sql = Sql & Replace(CStr(v_min_pay_amount), ",", ".") & ", "  
"
```

```

        Sql = Sql & Replace(CStr(v_lentamount), ",", ".") & ", "
        Sql = Sql & " 0 , "
        Sql = Sql & " 0 , "
        Sql = Sql & Replace(CStr(v_lentamount), ",", ".") & ", "
        Sql = Sql & Replace(CStr(v_min_pay_amount), ",", ".") & ",
    "

        Sql = Sql & "" & v_last_pay_date & " , "
        Sql = Sql & Replace(CStr(v_last_pay_amount), ",", ".") &
    ", "

        Sql = Sql & v_product_type & " , "
        Sql = Sql & v_pack_type & " , "
        Sql = Sql & "0, "
        Sql = Sql & v_delequancy & ", "
        Sql = Sql & v_bucket & ", "
        Sql = Sql & v_bucket & ", "
        Sql = Sql & "" & v_billing_date & " , "
        Sql = Sql & "sysdate, "
        Sql = Sql & "sysdate, "
        If v_flagAnathesis = True Then
            Sql = Sql & "1 "
        Else
            Sql = Sql & "0 "
        End If
        Sql = Sql & ") "

        Dim Com As OracleCommand = New OracleCommand(Sql.ToUpper,
Con)
        Com.ExecuteNonQuery()
    End Sub

    Private Sub OLD_package_ExecuteCode(ByVal sender As
System.Object, ByVal e As System.EventArgs)
        Dim Sql As String = " SELECT TRUNC(MAX(START_DATE)) AS
START_DATE "
        Sql = Sql & " FROM MCS.PACKS "
        Sql = Sql & " WHERE CUSTOMER_ID = 12 "
        Sql = Sql & " AND PACK_TYPE_ID = 12 "
        Dim Com As OracleCommand = New OracleCommand(Sql.ToUpper,
Con)

        Dim lReader As OracleDataReader = Com.ExecuteReader()
        lReader.Read()
        START_DATE = CDate(lReader("START_DATE"))
        Sql = ""
        Sql = " SELECT PACK_ID, PACK_DESC "
        Sql = Sql & " FROM MCS.PACKS "
        Sql = Sql & " WHERE CUSTOMER_ID = 12 "
        Sql = Sql & " AND PACK_TYPE_ID = 12"
        Sql = Sql & " AND PACK_DESC LIKE '%30%'"
        Sql = Sql & " AND TRUNC(START_DATE) = '" & START_DATE &
    ""

        Com = New OracleCommand(Sql.ToUpper, Con)
        lReader = Com.ExecuteReader()
        lReader.Read()
        PACK_ID_30 = CLng(lReader("PACK_ID"))
        PACK_DESC_30 = CStr(lReader("PACK_DESC"))

        Sql = " SELECT TRUNC(MAX(START_DATE)) AS START_DATE "
    
```

## Msc Thesis NetCentric Systems

```
Sql = Sql & " FROM MCS.PACKS "  
Sql = Sql & " WHERE CUSTOMER_ID = 12 "  
Sql = Sql & " AND PACK_TYPE_ID = 12 "  
Com = New OracleCommand(Sql.ToUpper, Con)  
lReader = Com.ExecuteReader()  
lReader.Read()  
START_DATE = CDate(lReader("START_DATE"))  
  
Sql = ""  
Sql = " SELECT PACK_ID, PACK_DESC "  
Sql = Sql & " FROM MCS.PACKS "  
Sql = Sql & " WHERE CUSTOMER_ID = 12 "  
Sql = Sql & " AND PACK_TYPE_ID = 12 "  
Sql = Sql & " AND PACK_DESC LIKE '%90%'"  
Sql = Sql & " AND TRUNC(START_DATE) = '" & START_DATE &  
""  
  
Com = New OracleCommand(Sql.ToUpper, Con)  
lReader = Com.ExecuteReader()  
lReader.Read()  
PACK_ID_90 = CLng(lReader("PACK_ID"))  
PACK_DESC_90 = CStr(lReader("PACK_DESC"))  
End Sub  
  
Private Sub Update_Case_ExecuteCode(ByVal sender As  
System.Object, ByVal e As System.EventArgs)  
Dim Sql As String = " UPDATE CASES "  
Sql = Sql & "SET "  
Sql = Sql & "INIT_LENT_CAPITAL = " &  
Replace(CStr(v_lentamount), ",", ".") & ", "  
Sql = Sql & "INIT_LENT_INTEREST = 0, "  
Sql = Sql & "INIT_LENT_EXPENSES = 0, "  
Sql = Sql & "TOTAL_INIT_LENT_AMOUNT = " &  
Replace(CStr(v_lentamount), ",", ".") & ", "  
Sql = Sql & "INIT_MIN_PAY_AMOUNT = " &  
Replace(CStr(v_min_pay_amount), ",", ".") & ", "  
Sql = Sql & "INIT_BUCKET_ID = " & v_bucket & ", "  
Sql = Sql & "LENT_CAPITAL = " &  
Replace(CStr(v_lentamount), ",", ".") & ", "  
Sql = Sql & "LENT_INTEREST = 0, "  
Sql = Sql & "LENT_EXPENSES = 0, "  
Sql = Sql & "TOTAL_LENT_AMOUNT = " &  
Replace(CStr(v_lentamount), ",", ".") & ", "  
Sql = Sql & "MIN_PAY_AMOUNT = " &  
Replace(CStr(v_min_pay_amount), ",", ".") & ", "  
Sql = Sql & "PACK_ID = " & PACK_ID & ", "  
Sql = Sql & "PACK_DESC = '" & PACK_DESC & "', "  
If v_flagAnathesis = True Then  
Sql = Sql & "IS_ACTIVE = 1, "  
If v_flagItWasActive = False Then  
Sql = Sql & "LOCATION_ID = " & LOCATION_ID & ", "  
Sql = Sql & "LOCATION_DESC = '" & LOCATION_DESC &  
"', "  
End If  
Else  
Sql = Sql & "IS_ACTIVE = 0, "  
Sql = Sql & "LOCATION_ID = 188, "  
Sql = Sql & "LOCATION_DESC = 'INACTIVE QUEUE', "  
End If
```

## Msc Thesis NetCentric Systems

```
        Sql = Sql & "CREATE_DATE = sysdate, "
        Sql = Sql & "CASE_EXT_DATA = 1, "
        Sql = Sql & "LAST_PAY_DATE = '" & CStr(v_last_pay_date) &
"', "
        Sql = Sql & "LAST_PAY_AMOUNT = " &
Replace(CStr(v_last_pay_amount), ",", ".") & ", "
        Sql = Sql & "APPLICATION_NO = '" & CStr(v_LastOfeilliDays)
& "', "
        Sql = Sql & "BILLING_DATE = '" & v_billing_date & "', "
        Sql = Sql & "BUCKET_ID = " & v_bucket & ", "
        Sql = Sql & "DELEQUANCY_ID = " & v_delequancy & ", "
        Sql = Sql & "USER_ID = 1, "
        Sql = Sql & "LAST_UPDATE_DATE = sysdate, "

        Sql = Sql & "IS_VISABLE = 1 "
        Sql = Sql & "WHERE CASE_ID = " & CASE_ID & " "
        Sql = Sql & "AND CUSTOMER_ID = 12"
        Dim Com As OracleCommand = New OracleCommand(Sql.ToUpper,
Con)
        Com.ExecuteNonQuery()
    End Sub

    Private Sub InsertGuarAddr_ExecuteCode(ByVal sender As
System.Object, ByVal e As System.EventArgs)
        'ADDRESS
        Try
            v_Guar_Address = " '" &
Trim(CStr(oReader("GUARANTEEADDRESS"))) & "' "
        Catch ex As Exception
            v_Guar_Address = ""
        End Try

        Try
            v_Guar_Town = " '" &
Trim(CStr(oReader("GUARANTEEETOWN"))) & "' "
        Catch ex As Exception
            v_Guar_Town = "NULL"
        End Try

        Try
            v_Guar_TK = " '" & Trim(CStr(oReader("GUARANTEEETK")))
& "' "
        Catch ex As Exception
            v_Guar_TK = ""
        End Try

        Dim Sql As String = " INSERT INTO SEC_PERSON_ADDRESSES "
        Sql = Sql & "(ADDRESS_ID ,SEC_PERSON_ID, CUSTOMER_ID
,ADDRESS_TYPE_ID ,ADDRESS ,TOWN ,ZIP ,MALLING_ADDRESS ) "
        Sql = Sql & " VALUES (SEQ_CLIENT_ADDRESS.NEXTVAL, " &
SEC_PERSON_ID & ", 12, 1, " & v_Guar_Address & ", " & v_Guar_Town
& ", " & v_Guar_TK & ", 1) "
        Dim Com As OracleCommand = New OracleCommand(Sql.ToUpper,
Con)
        Try
            Com.ExecuteNonQuery()
        Catch ex As Exception
        End Try
    End Sub
```



```

        'phone
    Try
        v_Guar_Phone = " '" &
RTrim(CStr(oReader("GUARANTEEPHONE")) & "' "
    Catch ex As Exception
        v_Guar_Phone = ""
    End Try

    If v_Guar_Phone <> "" Then 'insert phone
        Sql = " INSERT INTO SEC_PERSON_PHONES "
        Sql = Sql & "(PHONE_ID, CUSTOMER_ID, PHONE_TYPE_ID,
SEC_PERSON_ID, PHONE) "
        Sql = Sql & " VALUES (SEQ_CLIENT_PHONE.NEXTVAL, 12, 1,
" & SEC_PERSON_ID & ", " & v_Guar_Phone & ") "
        Com = New OracleCommand(Sql.ToUpper, Con)
        Com.ExecuteNonQuery()
    End If
End Sub

Private Sub InsertCaseHistory_ExecuteCode(ByVal sender As
System.Object, ByVal e As System.EventArgs)
    Dim Sql As String = " INSERT INTO CASE_HISTORY "
    Sql = Sql & "( "
    Sql = Sql & "HISTORY_ID, "
    Sql = Sql & "CASE_ID, "
    Sql = Sql & "CLIENT_ID, "
    Sql = Sql & "CUSTOMER_ID, "
    Sql = Sql & "CUSTOMER_NAME, "
    Sql = Sql & "INST_ID, "
    Sql = Sql & "CASE_SEC_ID, "
    Sql = Sql & "PACK_ID, "
    Sql = Sql & "PACK_DESC, "
    Sql = Sql & "INIT_LENT_CAPITAL, "
    Sql = Sql & "INIT_LENT_INTEREST, "
    Sql = Sql & "INIT_LENT_EXPENSES, "
    Sql = Sql & "TOTAL_INIT_LENT_AMOUNT, "
    Sql = Sql & "INIT_MIN_PAY_AMOUNT, "
    Sql = Sql & "LENT_CAPITAL, "
    Sql = Sql & "LENT_INTEREST, "
    Sql = Sql & "LENT_EXPENSES, "
    Sql = Sql & "TOTAL_LENT_AMOUNT, "
    Sql = Sql & "MIN_PAY_AMOUNT, "
    Sql = Sql & "LAST_PAY_DATE, "
    Sql = Sql & "LAST_PAY_AMOUNT, "
    Sql = Sql & "PRODUCT_TYPE_ID, "
    Sql = Sql & "PACK_TYPE_ID, "
    Sql = Sql & "IS_LEGAL_CASE, "
    Sql = Sql & "DELEQUANCY_ID, "
    Sql = Sql & "INIT_BUCKET_ID, "
    Sql = Sql & "BUCKET_ID, "
    Sql = Sql & "BILLING_DATE, "
    Sql = Sql & "LAST_UPDATE_DATE, "
    Sql = Sql & "CREATE_DATE, "
    Sql = Sql & "IS_ACTIVE "
    Sql = Sql & ") "
    Sql = Sql & "VALUES "
    Sql = Sql & "( "
    Sql = Sql & "SEQ_CASE_HISTORY.nextval, "
    Sql = Sql & CASE_ID & ", "

```

```

Sql = Sql & CLIENT_ID & ", "
Sql = Sql & "12, "
Sql = Sql & "'ALPHA BANK', "
Sql = Sql & "1, "
Sql = Sql & "'" & v_case_secID & "'", "
Sql = Sql & PACK_ID & ", "
Sql = Sql & "'" & PACK_DESC & "'", "
Sql = Sql & Replace(CStr(v_lentamount), ",", ".") & ", "
Sql = Sql & " 0 , "
Sql = Sql & " 0 , "
Sql = Sql & Replace(CStr(v_lentamount), ",", ".") & ", "
Sql = Sql & Replace(CStr(v_min_pay_amount), ",", ".") & ",
"

Sql = Sql & Replace(CStr(v_lentamount), ",", ".") & ", "
Sql = Sql & " 0 , "
Sql = Sql & " 0 , "
Sql = Sql & Replace(CStr(v_lentamount), ",", ".") & ", "
Sql = Sql & Replace(CStr(v_min_pay_amount), ",", ".") & ",
"

Sql = Sql & "'" & v_last_pay_date & "' , "
Sql = Sql & Replace(CStr(v_last_pay_amount), ",", ".") &
", "

Sql = Sql & v_product_type & " , "
Sql = Sql & v_pack_type & " , "
Sql = Sql & "0, "
Sql = Sql & v_delequancy & " , "
Sql = Sql & v_bucket & " , "
Sql = Sql & v_bucket & " , "
Sql = Sql & "'" & v_billing_date & "' , "
Sql = Sql & "sysdate, "
Sql = Sql & "sysdate, "
If v_flagAnathesis = True Then
    Sql = Sql & "1 "
Else
    Sql = Sql & "0 "
End If
Sql = Sql & ") "

Dim Com As OracleCommand = New OracleCommand(Sql.ToUpper,
Con)
    Com.ExecuteNonQuery()
End Sub
End Class

```

## 2. Κώδικας Production Workflow

```

Imports System.Data.OracleClient
Imports System.IO
Imports System.Xml
Imports System.Xml.XPath

Public Class Workflow1

```

## Msc Thesis NetCentric Systems

```
Inherits SequentialWorkflowActivity
Private LOCATION_DESC As String
Private LOCATION_ID As Long
Private CASE_BEHAVIOR_SCORE As Integer
Private LAST_CONTACT_ID As Long
Private LAST_STRONG_CONTACT_ID As Long
Private LAST_ACTION_CODE As Long
Private LAST_STRONG_ACTION_CODE As Long
Private LAST_CONTACT_DATE As Date
Private _Mode As String
Private _Customer_ID As Long
Private _Case_ID As Long
Private _Client_ID As Long
Private _User_ID As Long
Private _Inst_ID As Long
Private _Contact_ID As Long
Private _Action_Code As Long
Private _Action_Type_id As Long
Private _Pay_Date As Date
Private _Pay_Amount As Double
Private _Next_Contact_Date As Date
Private _Comments As String
Private _Is_Reminder As Long
Private _Next_Contact_Time As String
Private _Data1 As String
Private _Data2 As String
Private _Data3 As String
Private _Data4 As String

Private _LEGAL_REFERS_TO As Long
Private _LEGAL_LAND_REGISTRY As String
Private _LEGAL_ORDER_DATE As Date
Private _LEGAL_PREFECT As String
Private _LEGAL_DATE_COMPLETED As Date
Private _LEGAL_PROPERTY_TYPE As Long
Private _LEGAL_PAY_AMOUNT As Double
Private _LEGAL_CITY_COURT As String
Private _LEGAL_RESULTS As Long
Private _LEGAL_TEXT1 As String
Private _LEGAL_TEXT2 As String
Private _LEGAL_TEXT3 As String
Private _LEGAL_DATE_SENT As Date
Private _LEGAL_DATE_RETURNED As Date
Private _LEGAL_LAWER As String
Private _LEGAL_ACTION_DATE As Date
Private _LEGAL_COURT As String

'Return variables
Private _Status As String
Private _Description As String

Private Con As OracleConnection
Private conString As String
Private OLD_LOCATION_ID As Long

Public Sub New()
```

## Msc Thesis NetCentric Systems

```
        MyBase.New()
        InitializeComponent()
    End Sub

    Public Property STATUS() As String
        Get
            STATUS = _Status
        End Get
        Set(ByVal value As String)
            _Status = value
        End Set
    End Property

    Public Property STAUSDESCRIPTION() As String
        Get
            STAUSDESCRIPTION = _Description
        End Get
        Set(ByVal value As String)
            _Description = value
        End Set
    End Property

    Public WriteOnly Property Mode() As String
        Set(ByVal value As String)
            _Mode = value
        End Set
    End Property

    Public WriteOnly Property Customer_ID() As Long
        Set(ByVal value As Long)
            _Customer_ID = value
        End Set
    End Property
    Public WriteOnly Property Case_ID() As Long
        Set(ByVal value As Long)
            _Case_ID = value
        End Set
    End Property
    Public WriteOnly Property Client_ID() As Long
        Set(ByVal value As Long)
            _Client_ID = value
        End Set
    End Property
    Public WriteOnly Property User_ID() As Long
        Set(ByVal value As Long)
            _User_ID = value
        End Set
    End Property
    Public WriteOnly Property Inst_ID() As Long
        Set(ByVal value As Long)
            _Inst_ID = value
        End Set
    End Property
    Public Property Contact_ID() As Long
        Get
            Contact_ID = _Contact_ID
        End Get
        Set(ByVal value As Long)
            _Contact_ID = value
        End Set
    End Property
```

```
        End Set
    End Property
    Public WriteOnly Property Action_Code() As Long
        Set(ByVal value As Long)
            _Action_Code = value
        End Set
    End Property
    Public WriteOnly Property Action_Type() As Long
        Set(ByVal value As Long)
            _Action_Type_id = value
        End Set
    End Property
    Public WriteOnly Property Pay_Date() As Date
        Set(ByVal value As Date)
            _Pay_Date = value
        End Set
    End Property
    Public WriteOnly Property Pay_Amount() As Double
        Set(ByVal value As Double)
            _Pay_Amount = value
        End Set
    End Property
    Public WriteOnly Property Next_Contact_Date() As Date
        Set(ByVal value As Date)
            _Next_Contact_Date = value
        End Set
    End Property
    Public WriteOnly Property Comments() As String
        Set(ByVal value As String)
            _Comments = value
            _Comments = Replace(_Comments, "'", " ")
        End Set
    End Property
    Public WriteOnly Property Is_Reminder() As Long
        Set(ByVal value As Long)
            _Is_Reminder = value
        End Set
    End Property
    Public WriteOnly Property Next_Contact_Time() As String
        Set(ByVal value As String)
            _Next_Contact_Time = value
        End Set
    End Property
    Public WriteOnly Property Data1() As String
        Set(ByVal value As String)
            _Data1 = value
        End Set
    End Property
    Public WriteOnly Property Data2() As String
        Set(ByVal value As String)
            _Data2 = value
        End Set
    End Property
    Public WriteOnly Property Data3() As String
        Set(ByVal value As String)
            _Data3 = value
        End Set
    End Property
    Public WriteOnly Property Data4() As String
```

```
        Set(ByVal value As String)
            _Data4 = value
        End Set
    End Property

    Public WriteOnly Property LEGAL_REFERS_TO() As Long
        Set(ByVal value As Long)
            _LEGAL_REFERS_TO = value
        End Set
    End Property
    Public WriteOnly Property LEGAL_LAND_REGISTRY() As String
        Set(ByVal value As String)
            _LEGAL_LAND_REGISTRY = value
        End Set
    End Property
    Public WriteOnly Property LEGAL_ORDER_DATE() As Date
        Set(ByVal value As Date)
            _LEGAL_ORDER_DATE = value
        End Set
    End Property
    Public WriteOnly Property LEGAL_PREFECT() As String
        Set(ByVal value As String)
            _LEGAL_PREFECT = value
        End Set
    End Property
    Public WriteOnly Property LEGAL_DATE_COMPLETED() As Date
        Set(ByVal value As Date)
            _LEGAL_DATE_COMPLETED = value
        End Set
    End Property
    Public WriteOnly Property LEGAL_PROPERTY_TYPE() As Long
        Set(ByVal value As Long)
            _LEGAL_PROPERTY_TYPE = value
        End Set
    End Property
    Public WriteOnly Property LEGAL_PAY_AMOUNT() As Double
        Set(ByVal value As Double)
            _LEGAL_PAY_AMOUNT = value
        End Set
    End Property
    Public WriteOnly Property LEGAL_CITY_COURT() As String
        Set(ByVal value As String)
            _LEGAL_CITY_COURT = value
        End Set
    End Property
    Public WriteOnly Property LEGAL_RESULTS() As Long
        Set(ByVal value As Long)
            _LEGAL_RESULTS = value
        End Set
    End Property
    Public WriteOnly Property LEGAL_TEXT1() As String
        Set(ByVal value As String)
            _LEGAL_TEXT1 = value
        End Set
    End Property
    Public WriteOnly Property LEGAL_TEXT2() As String
        Set(ByVal value As String)
            _LEGAL_TEXT2 = value
        End Set
    End Set
```

```

End Property

Public WriteOnly Property LEGAL_TEXT3() As String
    Set(ByVal value As String)
        _LEGAL_TEXT3 = value
    End Set
End Property
Public WriteOnly Property LEGAL_DATE_SENT() As Date
    Set(ByVal value As Date)
        _LEGAL_DATE_SENT = value
    End Set
End Property
Public WriteOnly Property LEGAL_DATE_RETURNED() As Date
    Set(ByVal value As Date)
        _LEGAL_DATE_RETURNED = value
    End Set
End Property
Public WriteOnly Property LEGAL_LAWER() As String
    Set(ByVal value As String)
        _LEGAL_LAWER = value
    End Set
End Property
Public WriteOnly Property LEGAL_ACTION_DATE() As Date
    Set(ByVal value As Date)
        _LEGAL_ACTION_DATE = value
    End Set
End Property
Public WriteOnly Property LEGAL_COURT() As String
    Set(ByVal value As String)
        _LEGAL_COURT = value
    End Set
End Property

Private Sub ReadConfig_ExecuteCode(ByVal sender As
System.Object, ByVal e As System.EventArgs)
    Dim AppPath As String =
System.IO.Path.GetDirectoryName(System.Reflection.Assembly.GetExec
utingAssembly().Location())
    Dim document As XPathDocument = New XPathDocument(AppPath
& "\wf_config.xml")
    Dim navigator As XPathNavigator =
document.CreateNavigator()
    Dim iterator As XPathNodeIterator =
navigator.Select("//Connection[@name='MainDB']")

    iterator.MoveNext()
    conString =
iterator.Current.GetAttribute("connectionString", "")
End Sub

Private Sub OpenDB_ExecuteCode(ByVal sender As System.Object,
ByVal e As System.EventArgs)
    Con = New OracleConnection(conString) 'connect to
database
    Con.Open()
End Sub

```

## Msc Thesis NetCentric Systems

```
Private Sub DeleteContact_ExecuteCode(ByVal sender As
System.Object, ByVal e As System.EventArgs)
    Dim Sql As String
    Dim Com As OracleCommand

    Sql = " delete contact where contact.contact_id=" &
Me._Contact_ID & " and contact.customer_id=" & Me._Customer_ID
    Com = New OracleCommand(Sql.ToUpper, Con)
    Com.ExecuteNonQuery()
End Sub

Private Sub SetOKDeleteStatus_ExecuteCode(ByVal sender As
System.Object, ByVal e As System.EventArgs)
    Me._Status = "OK"
End Sub

Private Sub GetNewContactID_ExecuteCode(ByVal sender As
System.Object, ByVal e As System.EventArgs)
    Dim Sql As String = " SELECT SEQ_CONTACT.NEXTVAL AS
NEW_CONTACT_ID FROM DUAL "
    Dim Com As OracleCommand = New OracleCommand(Sql.ToUpper,
Con)

    Dim oReader As OracleDataReader = Com.ExecuteReader()

    oReader.Read()
    Me._Contact_ID = CLng(oReader("NEW_CONTACT_ID"))
    oReader.Close()
End Sub

Private Sub InsetContact_ExecuteCode(ByVal sender As
System.Object, ByVal e As System.EventArgs)
    Dim Sql As String
    Dim Com As OracleCommand

    Sql = " INSERT INTO CONTACT "
    Sql = Sql & "          (CONTACT_ID, ACTION_CODE,
CASE_ID, CUSTOMER_ID, USER_ID, ACTION_TYPE_ID, "
    Sql = Sql & "          PAY_DATE, PAY_AMOUNT,
NEXT_CONTACT_DATE, NEXT_CONTACT_TIME ,IS_REMINDER,"
    Sql = Sql & "          COMMENTS, DATA1, DATA2, DATA3,
DATA4, "
    Sql = Sql & " LEGAL_REFERS_TO,"
    Sql = Sql & " LEGAL_LAND_REGISTRY,"
    Sql = Sql & " LEGAL_ORDER_DATE ,"
    Sql = Sql & " LEGAL_PREFECT,"
    Sql = Sql & " LEGAL_DATE_COMPLETED,"
    Sql = Sql & " LEGAL_PROPERTY_TYPE,"
    Sql = Sql & " LEGAL_PAY_AMOUNT,"
    Sql = Sql & " LEGAL_CITY_COURT,"
    Sql = Sql & " LEGAL_RESULTS,"
    Sql = Sql & " LEGAL_TEXT1,"
    Sql = Sql & " LEGAL_TEXT2,"
    Sql = Sql & " LEGAL_TEXT3,"
    Sql = Sql & " LEGAL_DATE_SENT,"
    Sql = Sql & " LEGAL_DATE_RETURNED,"
    Sql = Sql & " LEGAL_LAWER,"
    Sql = Sql & " LEGAL_ACTION_DATE,"
    Sql = Sql & " LEGAL_COURT"
    Sql = Sql & " ) "
```



```

        Sql = Sql & "        VALUES (" & _Contact_ID & "," &
        _Action_Code & "," & _Case_ID & "," & _Customer_ID & "," &
        _User_ID & "," & _Action_Type_id & "," "

        If _Pay_Date.Year = 1980 Then
            Sql = Sql & "NULL,"
        Else
            Sql = Sql & " to_date('" & _Pay_Date &
            "','dd/mm/yyyy'),' "
        End If

        If _Pay_Amount = 0 Then
            Sql = Sql & "NULL,"
        Else
            Sql = Sql & _Pay_Amount * 100 & "/100,"
        End If

        If Not CBool(_Is_Reminder) Then
            Sql = Sql & "NULL,"
            Sql = Sql & "NULL,"
            Sql = Sql & "0,"
        Else
            Sql = Sql & " to_date('" & _Next_Contact_Date & " " &
            _Next_Contact_Time & "','dd/mm/yyyy HH24:MI'),' "
            Sql = Sql & "'" & _Next_Contact_Time & "','1,"
        End If

        If _Comments = "" Then
            Sql = Sql & "NULL,"
        Else
            Sql = Sql & "'" & _Comments & "',' "
        End If

        If _Data1 = "" Then
            Sql = Sql & "NULL,"
        Else
            Sql = Sql & "'" & _Data1 & "',' "
        End If

        If _Data2 = "" Then
            Sql = Sql & "NULL,"
        Else
            Sql = Sql & "'" & _Data2 & "',' "
        End If

        If _Data3 = "" Then
            Sql = Sql & "NULL,"
        Else
            Sql = Sql & "'" & _Data3 & "',' "
        End If

        If _Data4 = "" Then
            Sql = Sql & "NULL,"
        Else
            Sql = Sql & "'" & _Data4 & "',' "
        End If
    
```

## Msc Thesis NetCentric Systems

```
If _LEGAL_REFERS_TO = 0 Then
    Sql = Sql & "NULL,"
Else
    Sql = Sql & _LEGAL_REFERS_TO & ", "
End If

If _LEGAL_LAND_REGISTRY = "" Then
    Sql = Sql & "NULL,"
Else
    Sql = Sql & "'" & _LEGAL_LAND_REGISTRY & "', "
End If

If _LEGAL_ORDER_DATE.Year = 1980 Then
    Sql = Sql & "NULL,"
Else
    Sql = Sql & " to_date('" & _LEGAL_ORDER_DATE &
"', 'dd/mm/yyyy'), "
End If

If _LEGAL_PREFECT = "" Then
    Sql = Sql & "NULL,"
Else
    Sql = Sql & "'" & _LEGAL_PREFECT & "', "
End If

If _LEGAL_DATE_COMPLETED.Year = 1980 Then
    Sql = Sql & "NULL,"
Else
    Sql = Sql & " to_date('" & _LEGAL_DATE_COMPLETED &
"', 'dd/mm/yyyy'), "
End If

If _LEGAL_PROPERTY_TYPE = 0 Then
    Sql = Sql & "NULL,"
Else
    Sql = Sql & _LEGAL_PROPERTY_TYPE & ", "
End If

If _LEGAL_PAY_AMOUNT = 0 Then
    Sql = Sql & "NULL,"
Else
    Sql = Sql & _LEGAL_PAY_AMOUNT * 100 & "/100,"
End If

If _LEGAL_CITY_COURT = "" Then
    Sql = Sql & "NULL,"
Else
    Sql = Sql & "'" & _LEGAL_CITY_COURT & "', "
End If

If _LEGAL_RESULTS = 0 Then
    Sql = Sql & "NULL,"
Else
    Sql = Sql & _LEGAL_RESULTS & ", "
End If

If _LEGAL_TEXT1 = "" Then
    Sql = Sql & "NULL,"
Else
```

## Msc Thesis NetCentric Systems

```
        Sql = Sql & "'" & _LEGAL_TEXT1 & "',"
    End If

    If _LEGAL_TEXT2 = "" Then
        Sql = Sql & "NULL,"
    Else
        Sql = Sql & "'" & _LEGAL_TEXT2 & "',"
    End If

    If _LEGAL_TEXT3 = "" Then
        Sql = Sql & "NULL,"
    Else
        Sql = Sql & "'" & _LEGAL_TEXT3 & "',"
    End If

    If _LEGAL_DATE_SENT.Year = 1980 Then
        Sql = Sql & "NULL,"
    Else
        Sql = Sql & " to_date('" & _LEGAL_DATE_SENT &
        "','dd/mm/yyyy'),',"
    End If

    If _LEGAL_DATE_RETURNED.Year = 1980 Then
        Sql = Sql & "NULL,"
    Else
        Sql = Sql & " to_date('" & _LEGAL_DATE_RETURNED &
        "','dd/mm/yyyy'),',"
    End If

    If _LEGAL_LAWER = "" Then
        Sql = Sql & "NULL,"
    Else
        Sql = Sql & "'" & _LEGAL_LAWER & "',"
    End If

    If _LEGAL_ACTION_DATE.Year = 1980 Then
        Sql = Sql & "NULL,"
    Else
        Sql = Sql & " to_date('" & _LEGAL_ACTION_DATE &
        "','dd/mm/yyyy'),',"
    End If

    If _LEGAL_COURT = "" Then
        Sql = Sql & "NULL)"
    Else
        Sql = Sql & "'" & _LEGAL_COURT & "')"
    End If

    Com = New OracleCommand(Sql.ToUpper, Con)
    Com.ExecuteNonQuery()
End Sub

Private Sub UpdateContact_ExecuteCode(ByVal sender As
System.Object, ByVal e As System.EventArgs)
    Dim Com As OracleCommand
    Dim Sql As String = " UPDATE CONTACT SET ACTION_CODE=" &
_Action_Code & ", "
```

## Msc Thesis NetCentric Systems

```
If _Pay_Date.Year = 1980 Then
    Sql = Sql & "PAY_DATE=NULL,"
Else
    Sql = Sql & "PAY_DATE= to_date('" & _Pay_Date &
"', 'dd/mm/yyyy'),"
End If

If _Pay_Amount = 0 Then
    Sql = Sql & "PAY_AMOUNT=NULL,"
Else
    Sql = Sql & " PAY_AMOUNT=" & _Pay_Amount * 100 &
"/100,"
End If

If Not CBool(_Is_Reminder) Then
    Sql = Sql & "NEXT_CONTACT_DATE=NULL,"
    Sql = Sql & "NEXT_CONTACT_TIME=NULL,"
    Sql = Sql & "IS_REMINDER=0,"
Else
    Sql = Sql & "NEXT_CONTACT_DATE= to_date('" &
_Next_Contact_Date & " " & _Next_Contact_Time & "', 'dd/mm/yyyy
HH24:MI'),"
    Sql = Sql & "NEXT_CONTACT_TIME='" & _Next_Contact_Time
& "', IS_REMINDER=1,"
End If

If _Comments = "" Then
    Sql = Sql & "COMMENTS=NULL,"
Else
    Sql = Sql & "COMMENTS='" & _Comments & "',"
End If

If _Data1 = "" Then
    Sql = Sql & "DATA1=NULL,"
Else
    Sql = Sql & "DATA1='" & _Data1 & "',"
End If

If _Data2 = "" Then
    Sql = Sql & "DATA2=NULL,"
Else
    Sql = Sql & "DATA2='" & _Data2 & "',"
End If

If _Data3 = "" Then
    Sql = Sql & "DATA3=NULL,"
Else
    Sql = Sql & "DATA3='" & _Data3 & "',"
End If

If _Data4 = "" Then
    Sql = Sql & "DATA4=NULL,"
Else
    Sql = Sql & "DATA4='" & _Data4 & "',"
End If

If _LEGAL_REFERS_TO = 0 Then
    Sql = Sql & "LEGAL_REFERS_TO=NULL,"
```

```

Else
    Sql = Sql & "LEGAL_REFERS_TO=" & _LEGAL_REFERS_TO &
", "
End If

If _LEGAL_LAND_REGISTRY = "" Then
    Sql = Sql & "LEGAL_LAND_REGISTRY=NULL,"
Else
    Sql = Sql & "LEGAL_LAND_REGISTRY='" &
_LEGAL_LAND_REGISTRY & "',"
End If

If _LEGAL_ORDER_DATE.Year = 1980 Then
    Sql = Sql & "LEGAL_ORDER_DATE=NULL,"
Else
    Sql = Sql & "LEGAL_ORDER_DATE= to_date('" &
_LEGAL_ORDER_DATE & "', 'dd/mm/yyyy'),"
End If

If _LEGAL_PREFECT = "" Then
    Sql = Sql & "LEGAL_PREFECT=NULL,"
Else
    Sql = Sql & "LEGAL_PREFECT='" & _LEGAL_PREFECT & "',"
End If

If _LEGAL_DATE_COMPLETED.Year = 1980 Then
    Sql = Sql & "LEGAL_DATE_COMPLETED=NULL,"
Else
    Sql = Sql & "LEGAL_DATE_COMPLETED= to_date('" &
_LEGAL_DATE_COMPLETED & "', 'dd/mm/yyyy'),"
End If

If _LEGAL_PROPERTY_TYPE = 0 Then
    Sql = Sql & "LEGAL_PROPERTY_TYPE=NULL,"
Else
    Sql = Sql & "LEGAL_PROPERTY_TYPE=" &
_LEGAL_PROPERTY_TYPE & ","
End If

If _LEGAL_PAY_AMOUNT = 0 Then
    Sql = Sql & "LEGAL_PAY_AMOUNT=NULL,"
Else
    Sql = Sql & "LEGAL_PAY_AMOUNT=" & _LEGAL_PAY_AMOUNT *
100 & "/100,"
End If

If _LEGAL_CITY_COURT = "" Then
    Sql = Sql & "LEGAL_CITY_COURT=NULL,"
Else
    Sql = Sql & "LEGAL_CITY_COURT='" & _LEGAL_CITY_COURT &
"', "
End If

If _LEGAL_RESULTS = 0 Then
    Sql = Sql & "LEGAL_RESULTS=NULL,"
Else
    Sql = Sql & "LEGAL_RESULTS=" & _LEGAL_RESULTS & ","
End If

```

## Msc Thesis NetCentric Systems

```
If _LEGAL_TEXT1 = "" Then
    Sql = Sql & "LEGAL_TEXT1=NULL,"
Else
    Sql = Sql & "LEGAL_TEXT1='" & _LEGAL_TEXT1 & "',"
End If

If _LEGAL_TEXT2 = "" Then
    Sql = Sql & "LEGAL_TEXT2=NULL,"
Else
    Sql = Sql & "LEGAL_TEXT2='" & _LEGAL_TEXT2 & "',"
End If

If _LEGAL_TEXT3 = "" Then
    Sql = Sql & "LEGAL_TEXT3=NULL,"
Else
    Sql = Sql & "LEGAL_TEXT3='" & _LEGAL_TEXT3 & "',"
End If

If _LEGAL_DATE_SENT.Year = 1980 Then
    Sql = Sql & "LEGAL_DATE_SENT=NULL,"
Else
    Sql = Sql & "LEGAL_DATE_SENT= to_date('" &
    _LEGAL_DATE_SENT & "', 'dd/mm/yyyy'),"
End If

If _LEGAL_DATE_RETURNED.Year = 1980 Then
    Sql = Sql & "LEGAL_DATE_RETURNED=NULL,"
Else
    Sql = Sql & "LEGAL_DATE_RETURNED= to_date('" &
    _LEGAL_DATE_RETURNED & "', 'dd/mm/yyyy'),"
End If

If _LEGAL_LAWER = "" Then
    Sql = Sql & "LEGAL_LAWER=NULL,"
Else
    Sql = Sql & "LEGAL_LAWER='" & _LEGAL_LAWER & "',"
End If

If _LEGAL_ACTION_DATE.Year = 1980 Then
    Sql = Sql & "LEGAL_ACTION_DATE=NULL,"
Else
    Sql = Sql & "LEGAL_ACTION_DATE= to_date('" &
    _LEGAL_ACTION_DATE & "', 'dd/mm/yyyy'),"
End If

If _LEGAL_COURT = "" Then
    Sql = Sql & "LEGAL_COURT=NULL "
Else
    Sql = Sql & "LEGAL_COURT='" & _LEGAL_COURT & "' "
End If

Sql = Sql & " WHERE Contact_ID=" & _Contact_ID & " AND
CUSTOMER_ID=" & _Customer_ID

Com = New OracleCommand(Sql.ToUpper, Con)
Com.ExecuteNonQuery()

End Sub
```

## Msc Thesis NetCentric Systems

```
Private Sub Update_Last_Contact_ExecuteCode(ByVal sender As
System.Object, ByVal e As System.EventArgs)
    Dim Prev_contact_id As Long
    Dim Prev_Action_Code As Long

    Dim Sql As String = " SELECT CONTACT.CONTACT_ID,
CONTACT.ACTION_CODE, CONTACT.CONTACT_DATE "
    Sql = Sql & " FROM CONTACT "
    Sql = Sql & " WHERE ((CONTACT.CASE_ID = " & _Case_ID &
") AND (CONTACT.CUSTOMER_ID = " & _Customer_ID & ")) "
    Sql = Sql & "ORDER BY CONTACT.CONTACT_DATE ASC "

    Dim Com As OracleCommand = New OracleCommand(Sql.ToUpper,
Con)
    Dim oReader As OracleDataReader = Com.ExecuteReader()

    'go through updating the previoud contacts
    If oReader.HasRows Then 'found previous last contact
        Do While oReader.Read()
            Last_Contact_ID = CLng(oReader("CONTACT_ID"))
            If Prev_contact_id = Nothing Then
                Sql = " UPDATE CONTACT SET
CONTACT.LAST_CONTACT=0 "
                Sql = Sql & " WHERE CONTACT_ID=" &
CLng(oReader("CONTACT_ID")) & " AND CUSTOMER_ID=" & _Customer_ID
                Com = New OracleCommand(Sql.ToUpper, Con)
                Com.ExecuteNonQuery()
            Else 'update prev contact id
                Sql = " UPDATE CONTACT SET
CONTACT.LAST_CONTACT=0, PREVIOUS_CONTACT_ID= " & Prev_contact_id &
", PREVIOUS_CONTACT_CODE= " & Prev_Action_Code
                Sql = Sql & " WHERE CONTACT_ID=" &
CLng(oReader("CONTACT_ID")) & " AND CUSTOMER_ID=" & _Customer_ID
                Com = New OracleCommand(Sql.ToUpper, Con)
                Com.ExecuteNonQuery()
            End If
            Prev_contact_id = CLng(oReader("CONTACT_ID"))
            Prev_Action_Code = CLng(oReader("ACTION_CODE"))
            LAST_CONTACT_DATE = CDate(oReader("CONTACT_DATE"))
            LAST_ACTION_CODE = CLng(oReader("ACTION_CODE"))
        Loop
        Sql = " UPDATE CONTACT SET CONTACT.LAST_CONTACT=1 "
        Sql = Sql & " WHERE CONTACT_ID=" & last_contact_id & "
AND CUSTOMER_ID=" & _Customer_ID
        'update last contact=1
        Com = New OracleCommand(Sql.ToUpper, Con)
        Com.ExecuteNonQuery()

    End If

End Sub
```

## Msc Thesis NetCentric Systems

```
Private Sub Update_Last_Strong_ExecuteCode(ByVal sender As
System.Object, ByVal e As System.EventArgs)
    Dim Prev_contact_id As Long
    Dim Prev_Action_Code As Long
    Dim Prev_Hier As Long

    Dim Sql As String = " SELECT CONTACT.CONTACT_ID,
CONTACT.ACTION_CODE, action_code.HIERARCHY "
    Sql = Sql & " FROM contact, action_code "
    Sql = Sql & " WHERE ( (contact.action_code =
action_code.action_code) and (CONTACT.CASE_ID = " & _Case_ID & ")
"
    Sql = Sql & " AND (CONTACT.CUSTOMER_ID = " &
_Customer_ID & ") "
    Sql = Sql & " AND (action_code.lang_id = 'EN')) "
    Sql = Sql & " ORDER BY CONTACT.CONTACT_DATE ASC "

    Dim Com As OracleCommand = New OracleCommand(Sql.ToUpper,
Con)
    Dim oReader As OracleDataReader = Com.ExecuteReader()

    'go through updating the previoud contacts
    If oReader.HasRows Then 'found previous last contact
        Do While oReader.Read()
            If Prev_contact_id = Nothing Then 'just update
last strong contact
                Sql = " UPDATE CONTACT SET
CONTACT.LAST_STRONG_CONTACT=0 "
                Sql = Sql & " WHERE CONTACT_ID=" &
CLng(oReader("CONTACT_ID")) & " AND CUSTOMER_ID=" & _Customer_ID
                Com = New OracleCommand(Sql.ToUpper, Con)
                Com.ExecuteNonQuery()
                Prev_Hier = CLng(oReader("HIERARCHY")) 'set
initial strong contact
                Last_Strong_Contact_ID =
CLng(oReader("CONTACT_ID"))
                Prev_Action_Code =
CLng(oReader("ACTION_CODE"))
                Prev_contact_id = CLng(oReader("CONTACT_ID"))
                LAST_STRONG_ACTION_CODE = Prev_Action_Code
            Else
                Sql = " UPDATE CONTACT SET
CONTACT.LAST_STRONG_CONTACT=0, PREVIOUS_STRONG_CONTACT_ID= " &
Last_Strong_Contact_ID & ", PREVIOUS_STRONG_CONTACT_CODE= " &
Prev_Action_Code
                Sql = Sql & " WHERE CONTACT_ID=" &
CLng(oReader("CONTACT_ID")) & " AND CUSTOMER_ID=" & _Customer_ID
                Com = New OracleCommand(Sql.ToUpper, Con)
                Com.ExecuteNonQuery()
                If CLng(oReader("HIERARCHY")) <= Prev_Hier
Then
                    Prev_Hier = CLng(oReader("HIERARCHY"))
                'set initial strong contact
                Last_Strong_Contact_ID =
CLng(oReader("CONTACT_ID"))
                Prev_Action_Code =
CLng(oReader("ACTION_CODE"))
                LAST_STRONG_ACTION_CODE = Prev_Action_Code
            End If
        End While
    End If
End Sub
```



## Msc Thesis NetCentric Systems

```
                End If
            End If
        Loop

        'update the last strong contact
        Sql = " UPDATE CONTACT SET
CONTACT.LAST_STRONG_CONTACT=1 "
        Sql = Sql & " WHERE CONTACT_ID=" &
Last_Strong_Contact_ID & " AND CUSTOMER_ID=" & _Customer_ID
        Com = New OracleCommand(Sql.ToUpper, Con)
        Com.ExecuteNonQuery()

    End If
End Sub

Private Sub UpdateCase_ExecuteCode(ByVal sender As
System.Object, ByVal e As System.EventArgs)

    Dim Sql As String = " UPDATE CASES "
    Sql = Sql & " SET LAST_CONTACT_ID = " & LAST_CONTACT_ID
& ", "
    Sql = Sql & " LAST_STRONG_CONTACT_ID = " &
LAST_STRONG_CONTACT_ID & ", "
    Sql = Sql & " LAST_ACTION_CODE = " &
LAST_ACTION_CODE & ", "
    Sql = Sql & " LAST_STRONG_ACTION_CODE = " &
LAST_STRONG_ACTION_CODE & ", "
    Sql = Sql & " CASE_BEHAVIOR_SCORE = " &
CASE_BEHAVIOR_SCORE & ", "
    Sql = Sql & " LOCATION_DESC = '" & LOCATION_DESC &
"', "
    Sql = Sql & " LOCATION_ID = " & LOCATION_ID & ", "
    Sql = Sql & " LAST_CONTACT_DATE = SYSDATE "
    Sql = Sql & " WHERE CASE_ID = " & _Case_ID & " AND
CUSTOMER_ID =" & _Customer_ID

    Dim Com As OracleCommand = New OracleCommand(Sql.ToUpper,
Con)
    Com.ExecuteNonQuery()
End Sub

Private Sub CloseDB_ExecuteCode(ByVal sender As System.Object,
ByVal e As System.EventArgs)
    Con.Close()
End Sub

Private Sub CheckSkip_ExecuteCode(ByVal sender As
System.Object, ByVal e As System.EventArgs)
    Dim InString As String = "10104"
    If LOCATION_ID = 0 Then
        If InStr(InString, LAST_ACTION_CODE.ToString) > 0 Then
            LOCATION_ID = 171
            LOCATION_DESC = "SKIP QUEUE"
        End If
    End If
End Sub
```

## Msc Thesis NetCentric Systems

```
Private Sub CheckWorking_ExecuteCode(ByVal sender As System.Object, ByVal e As System.EventArgs)
    If LOCATION_ID = 0 Then
        LOCATION_ID = 170
        LOCATION_DESC = "WORKING QUEUE"
    End If
End Sub

Private Sub Init_location_ExecuteCode(ByVal sender As System.Object, ByVal e As System.EventArgs)
    LOCATION_ID = 0
End Sub

Private Sub _10102_ExecuteCode(ByVal sender As System.Object, ByVal e As System.EventArgs)
    Dim InString As String = "10102"
    If LOCATION_ID = 0 Then
        If InStr(InString, LAST_ACTION_CODE.ToString) > 0 Then
            LOCATION_ID = 168
            LOCATION_DESC = "PAID OFF PRE ASSIGNMENT QUEUE"
        End If
    End If
End Sub

Private Sub _60700_ExecuteCode(ByVal sender As System.Object, ByVal e As System.EventArgs)
    Dim InString As String = "60700"
    If LOCATION_ID = 0 Then
        If InStr(InString, Me.LAST_ACTION_CODE.ToString) > 0
Then
            LOCATION_ID = 181
            LOCATION_DESC = "COLLECTION TEMPORARILY PAUSED
QUEUE"
            CASE_BEHAVIOR_SCORE = 1
        End If
    End If
End Sub

Private Sub _10800_ExecuteCode(ByVal sender As System.Object, ByVal e As System.EventArgs)
    Dim InString As String = "10800"
    If LOCATION_ID = 0 Then
        If InStr(InString, LAST_ACTION_CODE.ToString) > 0 Then
            LOCATION_ID = 167
            LOCATION_DESC = "LETTER SHOP QUEUE"
        End If
    End If
End Sub

Private Sub _11104_ExecuteCode(ByVal sender As System.Object, ByVal e As System.EventArgs)
    Dim InString As String = "11104"
    If LOCATION_ID = 0 Then
        If InStr(InString, LAST_ACTION_CODE.ToString) > 0 Then
            LOCATION_ID = 173
            LOCATION_DESC = "SEND BANK QUEUE"
        End If
    End If
End Sub
```

```

    Private Sub _12109_ExecuteCode(ByVal sender As System.Object,
ByVal e As System.EventArgs)
        Dim InString As String = "10600"
        If LOCATION_ID = 0 Then
            If InStr(InString, LAST_ACTION_CODE.ToString) > 0 Then
                If OLD_LOCATION_ID <> 169 And OLD_LOCATION_ID <>
189 Then
                    CASE_BEHAVIOR_SCORE = 4
                    LOCATION_ID = 172
                    LOCATION_DESC = "NON WORKING QUEUE"
                End If
            End If
        End If
    End Sub

    Private Sub _NextContact_ExecuteCode(ByVal sender As
System.Object, ByVal e As System.EventArgs)
        Dim InString As String =
"10300,10301,10302,10303,10304,60000,60200,60300,60302,12104,12106
"
        If LOCATION_ID = 0 Then
            If InStr(InString, LAST_ACTION_CODE.ToString) > 0 Then
                LOCATION_ID = 174
                LOCATION_DESC = "REST QUEUE"
            End If
        End If
    End Sub

    Private Sub _NoContact_ExecuteCode(ByVal sender As
System.Object, ByVal e As System.EventArgs)
        Dim InString As String = "10500,10501"
        If LOCATION_ID = 0 Then
            If InStr(InString, LAST_ACTION_CODE.ToString) > 0 Then
                LOCATION_ID = 174
                LOCATION_DESC = "REST QUEUE"
            End If
        End If
    End Sub

    Private Sub _Promise_ExecuteCode(ByVal sender As
System.Object, ByVal e As System.EventArgs)
        Dim InString As String = "10105"
        If LOCATION_ID = 0 Then
            If InStr(InString, LAST_ACTION_CODE.ToString) > 0 Then
                If CASE_BEHAVIOR_SCORE <> 3 Then
                    LOCATION_ID = 176
                    LOCATION_DESC = "PROMISE QUEUE"
                Else
                    CASE_BEHAVIOR_SCORE = 0
                    LOCATION_ID = 176
                    LOCATION_DESC = "PROMISE QUEUE"
                End If
            End If
        End If
    End Sub

    Private Sub _Settlement_ExecuteCode(ByVal sender As
System.Object, ByVal e As System.EventArgs)

```

## Msc Thesis NetCentric Systems

```
    Dim InString As String = "80105,80151"
    If LOCATION_ID = 0 Then
        If InStr(InString, LAST_ACTION_CODE.ToString) > 0 Then
            LOCATION_ID = 177
            LOCATION_DESC = "SETTLEMENT QUEUE"
            CASE_BEHAVIOR_SCORE = 2
        End If
    End If
End Sub

Private Sub _NonWorking2_ExecuteCode(ByVal sender As
System.Object, ByVal e As System.EventArgs)
    Dim InString As String = "10401,77777,60400"
    If LOCATION_ID = 0 Then
        If InStr(InString, LAST_ACTION_CODE.ToString) > 0 Then
            LOCATION_ID = 180
            LOCATION_DESC = "NON WORKING 2 QUEUE"
        End If
    End If
End Sub

Private Sub _10101_ExecuteCode(ByVal sender As System.Object,
ByVal e As System.EventArgs)
    Dim InString As String = "10101"
    If LOCATION_ID = 0 Then
        If InStr(InString, LAST_ACTION_CODE.ToString) > 0 Then
            If CASE_BEHAVIOR_SCORE <> 3 Then
                LOCATION_ID = 186
                LOCATION_DESC = "PAY OFF QUEUE"
            Else
                CASE_BEHAVIOR_SCORE = 0
                LOCATION_ID = 186
                LOCATION_DESC = "PAY OFF QUEUE"
            End If
        End If
    End If
End Sub

Private Sub Check60300_ExecuteCode(ByVal sender As
System.Object, ByVal e As System.EventArgs)
    Dim InString As String = "60300"
    If LOCATION_ID = 0 Then
        If InStr(InString, LAST_ACTION_CODE.ToString) > 0 Then
            LOCATION_ID = 182
            LOCATION_DESC = "AFTERNOON QUEUE"
        End If
    End If
End Sub

Private Sub _20200_ExecuteCode(ByVal sender As System.Object,
ByVal e As System.EventArgs)
    Dim InString As String = "20200,20300"
    If LOCATION_ID = 0 Then
        If InStr(InString, LAST_ACTION_CODE.ToString) > 0 Then
            LOCATION_ID = 176
            LOCATION_DESC = "PROMISE QUEUE"
        End If
    End If
End Sub
```

```
Private Sub Find_CBS_ExecuteCode(ByVal sender As
System.Object, ByVal e As System.EventArgs)
    Dim Sql As String = " SELECT PACK_ID, CASE_BEHAVIOR_SCORE,
INIT_BUCKET_ID, PACK_TYPE_ID, LOCATION_ID "

    Sql = Sql & "FROM CASES WHERE CUSTOMER_ID = 12"

    Sql = Sql & "AND CASE_ID = " & _Case_ID

    Dim Com As OracleCommand = New OracleCommand(Sql.ToUpper,
Con)

    Dim lReader As OracleDataReader = Com.ExecuteReader()

    lReader.Read()

    'PACK_ID = CLng(lReader("PACK_ID"))

    CASE_BEHAVIOR_SCORE = CInt(lReader("CASE_BEHAVIOR_SCORE"))

    'PACK_TYPE_ID = CLng(lReader("PACK_TYPE_ID"))

    'INIT_BUCKET_ID = CLng(lReader("INIT_BUCKET_ID"))

    OLD_LOCATION_ID = CLng(lReader("LOCATION_ID"))

    lReader.Close()

End Sub
End Class
```

### 3. *Κώδικας StateMachine Workflow*

```
Imports System.Data.OracleClient
Imports System.IO
Imports System.Xml
Imports System.Xml.XPath
Imports System.Net.Mail

Public Class Workflow1
    Inherits SequentialWorkflowActivity
    Private NEXT_CONTACT_DATE As Date
    Private CONTACT_ID As Long
    Private LOCATION_DESC As String
    Private LOCATION_ID As Long
    Private LAST_CONTACT_ID As Long
    Private LAST_STRONG_CONTACT_ID As Long
    Private LAST_ACTION_CODE As Long
    Private LAST_STRONG_ACTION_CODE As Long
    Private LAST_CONTACT_DATE As Date
    Private AppPath As String
    Private TO_LAWYER As Long
    Private OLD_LOCATION_ID As Long
    Private Customer_ID As String
    Private Con As OracleConnection
```

## Msc Thesis NetCentric Systems

```
Private conString As String
Private CONTACT_TIME As String
Private oReader As OracleDataReader
Private PAY_DAY As String
Private PAY_AMOUNT As Double
Private MIN_PAY_AMOUNT As Double
Private PROMISE_AMOUNT As Double
Private PAY_DATE As Date
Private _Action_Type_id As Long
Private _Case_ID As Long
Private _Case_count As Long
Private _Pay_Date As Date
Private IS_VISABLE As Integer = 1
Private Assign_TO As Long = 0
Private CASE_BEHAVIOR_SCORE As Integer

Private Sub Show_Status(ByVal Msg As String)
    Console.SetCursorPosition(0, 24)
    Console.ForegroundColor = ConsoleColor.Red
    Console.Write("
")
    Console.SetCursorPosition(0, 24)
    Console.Write(Msg)
    Console.ForegroundColor = ConsoleColor.White
End Sub
Private Sub Initialise_ExecuteCode(ByVal sender As
System.Object, ByVal e As System.EventArgs)
    Console.Clear()
    Console.SetCursorPosition(0, 0)
    Console.ForegroundColor = ConsoleColor.Red
    Console.Write("                Alpha Bank
Stegastika State Machine      ")

    Customer_ID = "12"
    LOCATION_ID = 0
    _Case_count = 0
End Sub

Private Sub ReadConfig_ExecuteCode(ByVal sender As
System.Object, ByVal e As System.EventArgs)
    Dim AppPath As String =
System.IO.Path.GetDirectoryName(System.Reflection.Assembly.GetExec
utingAssembly().Location())
    Dim document As XPathDocument = New XPathDocument(AppPath
& "\wf_config.xml")
    Dim navigator As XPathNavigator =
document.CreateNavigator()
    Dim iterator As XPathNodeIterator =
navigator.Select("//Connection[@name='MainDB']")

    iterator.MoveNext()
    conString =
iterator.Current.GetAttribute("connectionString", "")
End Sub

Private Sub OpenDB_ExecuteCode(ByVal sender As System.Object,
ByVal e As System.EventArgs)
    Con = New OracleConnection(conString) 'connect to
database
```

## Msc Thesis NetCentric Systems

```
Con.Open()
End Sub

Private Sub SelectCases_ExecuteCode(ByVal sender As
System.Object, ByVal e As System.EventArgs)
    Show_Status("Selecting Cases. Plaes Wait....")
    Dim Sql As String = " SELECT cases.case_id,
MIN_PAY_AMOUNT, LENT_CAPITAL, PACK_TYPE_ID, LOCATION_ID,
LAST_UPDATE_DATE, BUCKET_ID, TOTAL_PAY_AMOUNT "
    Sql = Sql & " FROM cases "
    Sql = Sql & " WHERE ((cases.customer_id = " & Customer_ID
& ") and is_active = 1 ) "
    Dim Com As OracleCommand = New OracleCommand(Sql.ToUpper,
Con)
    oReader = Com.ExecuteReader()
End Sub

Private Sub ReadCaseID_ExecuteCode(ByVal sender As
System.Object, ByVal e As System.EventArgs)
    Me._Case_ID = CLng(oReader("CASE_ID"))
    _Case_count = _Case_count + 1
    Show_Status("Processing Case : " & _Case_count.ToString)
    OLD_LOCATION_ID = CLng(oReader("LOCATION_ID"))
    MIN_PAY_AMOUNT = CLng(oReader("MIN_PAY_AMOUNT"))
    LOCATION_ID = 0
    IS_VISABLE = 1
    CASE_BEHAVIOR_SCORE = 0
    Me.Assign_TO = 0
End Sub

Private Sub ReadLastContact_ExecuteCode(ByVal sender As
System.Object, ByVal e As System.EventArgs)
    Dim Sql As String = " SELECT CONTACT.CONTACT_ID,
CONTACT.ACTION_CODE, CONTACT.LAST_CONTACT,ACTION_TYPE_ID,
CONTACT.CUSTOMER_ID, "
    Sql = Sql & " CONTACT.CASE_ID,
CONTACT.LAST_STRONG_CONTACT, PREVIOUS_CONTACT_CODE,
trunc(CONTACT.NEXT_CONTACT_DATE) NEXT_CONTACT_DATE ,
trunc(CONTACT.PAY_DATE) PAY_DATE, CONTACT.PAY_AMOUNT PAY_AMOUNT"
    Sql = Sql & " FROM CONTACT "
    Sql = Sql & " WHERE ((CONTACT.CASE_ID = " & _Case_ID & ")
"
    Sql = Sql & " AND (CONTACT.CUSTOMER_ID = " & Customer_ID
& ") "
    Sql = Sql & " AND (CONTACT.LAST_CONTACT = 1) "
    Sql = Sql & " ) "
    Dim Com As OracleCommand = New OracleCommand(Sql.ToUpper,
Con)
    Dim lReader As OracleDataReader = Com.ExecuteReader()
    If lReader.HasRows Then
        lReader.Read()
        LAST_CONTACT_ID = CLng(lReader("CONTACT_ID"))
        LAST_ACTION_CODE = CLng(lReader("ACTION_CODE"))
        Me._Action_Type_id = CLng(lReader("ACTION_TYPE_ID"))
        If lReader("NEXT_CONTACT_DATE") IsNot DBNull.Value
Then
            NEXT_CONTACT_DATE =
CDate(lReader("NEXT_CONTACT_DATE"))

```

```

        End If
        If lReader("PAY_DATE") IsNot DBNull.Value Then
            PAY_DATE = CDate(lReader("PAY_DATE"))
        End If
        If lReader("PAY_AMOUNT") IsNot DBNull.Value Then
            PAY_AMOUNT = CDb1(lReader("PAY_AMOUNT"))
        End If
    Else
        LAST_ACTION_CODE = 7777777
        Me._Action_Type_id = -1
    End If
End Sub

Private Sub ReadLastStrong_ExecuteCode(ByVal sender As
System.Object, ByVal e As System.EventArgs)
    On Error Resume Next
    Dim Sql As String = " SELECT CONTACT.ACTION_CODE,
CONTACT.LAST_CONTACT, CONTACT.CUSTOMER_ID, "
    Sql = Sql & " CONTACT.CASE_ID, "
CONTACT.LAST_STRONG_CONTACT, PAY_DATE "
    Sql = Sql & " FROM CONTACT "
    Sql = Sql & " WHERE ((CONTACT.CASE_ID = " & _Case_ID & "
"
& ") "
    Sql = Sql & " AND (CONTACT.CUSTOMER_ID = " & Customer_ID
& ") "
    Sql = Sql & " AND (CONTACT.LAST_STRONG_CONTACT = 1) "
    Sql = Sql & " ) "

    Dim Com As OracleCommand = New OracleCommand(Sql.ToUpper,
Con)

    Dim lReader As OracleDataReader = Com.ExecuteReader()
    If lReader.HasRows Then
        lReader.Read()
        LAST_STRONG_ACTION_CODE = CLng(lReader("ACTION_CODE"))
        _Pay_Date = CDate(lReader("PAY_DATE"))
    Else
        LAST_STRONG_ACTION_CODE = 7777777
    End If
End Sub

Private Sub CheckSkip_ExecuteCode(ByVal sender As
System.Object, ByVal e As System.EventArgs)
    Dim InString As String = "10104"
    If LOCATION_ID = 0 Then
        If InStr(InString, LAST_ACTION_CODE.ToString) > 0 Then
            LOCATION_ID = 171
            LOCATION_DESC = "SKIP QUEUE"
        End If
    End If
End Sub

Private Sub CheckSendBank_ExecuteCode(ByVal sender As
System.Object, ByVal e As System.EventArgs)
    Dim InString As String = "11104"
    If LOCATION_ID = 0 Then
        If InStr(InString, LAST_ACTION_CODE.ToString) > 0 Then
            LOCATION_ID = 173
            LOCATION_DESC = "SEND BANK QUEUE"
        End If
    End If
End Sub

```



## Msc Thesis NetCentric Systems

```
End If
End Sub

Private Sub CheckNonWorking_ExecuteCode(ByVal sender As
System.Object, ByVal e As System.EventArgs)
    Dim InString As String = "10600"
    If LOCATION_ID = 0 Then
        If InStr(InString, LAST_ACTION_CODE.ToString) > 0 Then
            LOCATION_ID = 172
            LOCATION_DESC = "NON WORKING QUEUE"
        End If
    End If
End Sub

Private Sub CheckNextContact_ExecuteCode(ByVal sender As
System.Object, ByVal e As System.EventArgs)
    Dim InString As String =
"10300,10301,10302,10303,10304,60000,60200,60302,12104,12106"
    If LOCATION_ID = 0 Then
        If InStr(InString, LAST_ACTION_CODE.ToString) > 0 Then
            If NEXT_CONTACT_DATE <= Now.Date Then
                LOCATION_ID = 175
                LOCATION_DESC = "NEXT CONTACT QUEUE"
            Else
                LOCATION_ID = 174
                LOCATION_DESC = "REST QUEUE"
            End If
        End If
    End If
End Sub

Private Sub CheckNoContact_ExecuteCode(ByVal sender As
System.Object, ByVal e As System.EventArgs)
    Dim InString As String = "10500,10501"
    If LOCATION_ID = 0 Then
        If InStr(InString, LAST_ACTION_CODE.ToString) > 0 Then
            If NEXT_CONTACT_DATE <= Now.Date Then
                LOCATION_ID = 178
                LOCATION_DESC = "NO CONTACT QUEUE"
            Else
                LOCATION_ID = 174
                LOCATION_DESC = "REST QUEUE"
            End If
        End If
    End If
End Sub

Private Sub CheckPromise_ExecuteCode(ByVal sender As
System.Object, ByVal e As System.EventArgs)
    Dim InString As String = "10105"
    If LOCATION_ID = 0 Then
        If InStr(InString, LAST_ACTION_CODE.ToString) > 0 Then
            If Now.Date >= PAY_DATE.AddDays(+1) Then
                LOCATION_ID = 179
                LOCATION_DESC = "BROKEN PROMISE"
                CASE_BEHAVIOR_SCORE = 3
            Else
                LOCATION_ID = 176
                LOCATION_DESC = "PROMISE QUEUE"
            End If
        End If
    End If
End Sub
```

## Msc Thesis NetCentric Systems

```
        End If
    End If
End If
End Sub

Private Sub CheckSettlement_ExecuteCode(ByVal sender As
System.Object, ByVal e As System.EventArgs)
    Dim InString As String = "80105,80151"
    If LOCATION_ID = 0 Then
        If InStr(InString, LAST_ACTION_CODE.ToString) > 0 Then
            LOCATION_ID = 177
            LOCATION_DESC = "SETTLEMENT QUEUE"
        End If
    End If
End Sub

Private Sub CheckPayment_ExecuteCode(ByVal sender As
System.Object, ByVal e As System.EventArgs)
    Dim InString As String = "10107"
    If LOCATION_ID = 0 Then
        If InStr(InString, LAST_ACTION_CODE.ToString) > 0 Then

            If CLng(oReader("BUCKET_ID")) <= 3 Then '30-89
                If CLng(oReader("TOTAL_PAY_AMOUNT")) <= (80 *
MIN_PAY_AMOUNT) / 100 Then
                    LOCATION_ID = 183
                    LOCATION_DESC = "PAYMENT QUEUE"
                ElseIf MIN_PAY_AMOUNT -
CLng(oReader("TOTAL_PAY_AMOUNT")) <= 50 Then
                    LOCATION_ID = 186
                    LOCATION_DESC = "PAY OFF QUEUE"
                Else
                    LOCATION_ID = 184
                    LOCATION_DESC = "PAYMENT 1 QUEUE"
                End If
            Else '90-180
                If CLng(oReader("TOTAL_PAY_AMOUNT")) <= (50 *
MIN_PAY_AMOUNT) / 100 Then
                    LOCATION_ID = 183
                    LOCATION_DESC = "PAYMENT QUEUE"
                ElseIf MIN_PAY_AMOUNT -
CLng(oReader("TOTAL_PAY_AMOUNT")) <= 50 Then
                    LOCATION_ID = 186
                    LOCATION_DESC = "PAY OFF QUEUE"
                Else
                    LOCATION_ID = 184
                    LOCATION_DESC = "PAYMENT 1 QUEUE"
                End If
            End If
        End If
    End If
End Sub

Private Sub CheckNonWorking2_ExecuteCode(ByVal sender As
System.Object, ByVal e As System.EventArgs)
    Dim InString As String = "10401,60400,77777"
    If LOCATION_ID = 0 Then
        If InStr(InString, LAST_ACTION_CODE.ToString) > 0 Then
            LOCATION_ID = 180
        End If
    End If
End Sub
```

## Msc Thesis NetCentric Systems

```
        LOCATION_DESC = "NON WORKING 2 QUEUE"
    End If
End If
End Sub

Private Sub CheckWorking_ExecuteCode(ByVal sender As
System.Object, ByVal e As System.EventArgs)
    If LOCATION_ID = 0 Then
        LOCATION_ID = 170
        LOCATION_DESC = "WORKING QUEUE"
    End If
End Sub

Private Sub CheckLetterShop_ExecuteCode(ByVal sender As
System.Object, ByVal e As System.EventArgs)
    Dim InString As String = "10800"
    If LOCATION_ID = 0 Then
        If InStr(InString, LAST_ACTION_CODE.ToString) > 0 Then
            LOCATION_ID = 167
            LOCATION_DESC = "LETTER SHOP QUEUE"
        End If
    End If
End Sub

Private Sub CheckPaidOffAssign_ExecuteCode(ByVal sender As
System.Object, ByVal e As System.EventArgs)
    Dim InString As String = "10102"
    If LOCATION_ID = 0 Then
        If InStr(InString, LAST_ACTION_CODE.ToString) > 0 Then
            LOCATION_ID = 168
            LOCATION_DESC = "PAID OFF PRE ASSIGNMENT QUEUE"
        End If
    End If
End Sub

Private Sub UpdateCase_ExecuteCode(ByVal sender As
System.Object, ByVal e As System.EventArgs)
    Dim Sql As String = " UPDATE CASES "
    Sql = Sql & " SET LAST_CONTACT_ID = " & LAST_CONTACT_ID &
", "
    ' Sql = Sql & " LAST_STRONG_CONTACT_ID = " &
LAST_STRONG_CONTACT_ID & ", "
    Sql = Sql & " LAST_ACTION_CODE = " & LAST_ACTION_CODE &
", "
    'Sql = Sql & " LAST_STRONG_ACTION_CODE = " &
LAST_STRONG_ACTION_CODE & ", "
    If CASE_BEHAVIOR_SCORE = 3 Then
        Sql = Sql & " CASE_BEHAVIOR_SCORE = 3, "
    End If
    Sql = Sql & " LOCATION_DESC = '" & LOCATION_DESC & "', "
    Sql = Sql & " LOCATION_ID = " & LOCATION_ID & " "
    Sql = Sql & " WHERE CASE_ID = " & _Case_ID & " AND
CUSTOMER_ID = " & Customer_ID & " "

    Dim Com As OracleCommand = New OracleCommand(Sql.ToUpper,
Con)
    Com.ExecuteNonQuery()
End Sub
```

## Msc Thesis NetCentric Systems

```
Private Sub UpdateInactive_ExecuteCode(ByVal sender As
System.Object, ByVal e As System.EventArgs)
    Show_Status("
")
    Show_Status("Updating Inactive Cases ")
    Dim Sql As String = " UPDATE CASES SET
LOCATION_DESC='INACTIVE QUEUE', LOCATION_ID=188, USER_ID=1 WHERE
IS_ACTIVE = 0 and customer_id = " & Customer_ID & " "
    Dim Com As OracleCommand = New OracleCommand(Sql.ToUpper,
Con)
    Com.ExecuteNonQuery()

    Sql = " UPDATE MCS.CASES "
    Sql = Sql & "SET "
    Sql = Sql & "USER_ID = 1, "
    Sql = Sql & "LOCATION_ID = 188, "
    Sql = Sql & "LOCATION_DESC = 'INACTIVE QUEUE' "
    Sql = Sql & "WHERE CASE_ID IN "
    Sql = Sql & "( "
    Sql = Sql & "select case_id "
    Sql = Sql & "from mcs.case_ex_data "
    Sql = Sql & "where ex_data_desc = 'INACTIVE CASES' "
    Sql = Sql & "AND EX_DATA = 0 "
    Sql = Sql & "AND CUSTOMER_ID = " & Customer_ID & ") "
    Sql = Sql & "AND CUSTOMER_ID = " & Customer_ID & " "
    Com = New OracleCommand(Sql.ToUpper, Con)
    Com.ExecuteNonQuery()
End Sub

Private Sub Send_Mail_ExecuteCode(ByVal sender As
System.Object, ByVal e As System.EventArgs)
    Dim MsgString As String = "Locations completed
successfully."
    Dim Subject As String = "Alpha Bank Home - Locations Info"
    Dim Recivers As String = "a.pateli@mellon.com.gr"
    Dim mailSender As String = "conceptus_desc@mellon.com.gr"
    Try
        Dim message As New MailMessage(mailSender, Recivers,
Subject, MsgString)
        message.CC.Add("conceptus_desc@mellon.com.gr")
        message.CC.Add("mellonc_desc@mellon.com.gr")
        Dim emailClient As New SmtplibClient("192.168.0.4")
        emailClient.Send(message)
    Catch ex As Exception
        Console.WriteLine("Error in Sending Report. " &
ex.Message)
        Console.ReadLine()
    End Try
End Sub

Private Sub CloseConnections_ExecuteCode(ByVal sender As
System.Object, ByVal e As System.EventArgs)
    Con.Close()
    Show_Status("
")
    Show_Status("Done Processing ")
End Sub
```

## Msc Thesis NetCentric Systems

```
Private Sub Check60300_ExecuteCode(ByVal sender As
System.Object, ByVal e As System.EventArgs)
    Dim InString As String = "60300"
    If LOCATION_ID = 0 Then
        If InStr(InString, LAST_ACTION_CODE.ToString) > 0 Then
            LOCATION_ID = 182
            LOCATION_DESC = "AFTERNOON QUEUE"
        End If
    End If
End Sub
```

```
Private Sub Special_ExecuteCode(ByVal sender As System.Object,
ByVal e As System.EventArgs)
    Dim Sql As String
    Dim Com As OracleCommand
    '= " UPDATE CASES SET "
    'Sql = Sql & "LOCATION_ID = 180, "
    'Sql = Sql & "LOCATION_DESC = 'NON WORKING 2 QUEUE' "
    'Sql = Sql & "WHERE CASES.LAST_ACTION_CODE IN
(10401,10600,77777) "
    'Sql = Sql & "AND CUSTOMER_ID = 12 "
    'Sql = Sql & "AND IS_ACTIVE= 1 "
    'Dim Com As OracleCommand = New OracleCommand(Sql.ToUpper,
Con)
    'Com.ExecuteNonQuery()
```

```
Sql = " UPDATE CASES "
Sql = Sql & "SET "
Sql = Sql & "LOCATION_ID = 169, "
Sql = Sql & "LOCATION_DESC = 'SPECIAL CASES QUEUE' "
Sql = Sql & "WHERE CLIENT_ID IN "
Sql = Sql & "(select DISTINCT(client_ID) FROM CASES "
Sql = Sql & "WHERE CLIENT_ID IN "
Sql = Sql & "( SELECT CLIENT_ID FROM CASES HAVING
SUM(TOTAL_LENT_AMOUNT ) >= 200000 "
Sql = Sql & "AND CUSTOMER_ID = 12 AND IS_ACTIVE = 1 GROUP
BY CLIENT_ID,CUSTOMER_ID,IS_ACTIVE)) "
Sql = Sql & "AND IS_ACTIVE = 1 "
Sql = Sql & "AND CASES.LAST_ACTION_CODE <> 10104 "
Com = New OracleCommand(Sql.ToUpper, Con)
Com.ExecuteNonQuery()
```

```
Sql = " UPDATE CASES "
Sql = Sql & "SET "
Sql = Sql & "LOCATION_ID = 169, "
Sql = Sql & "LOCATION_DESC = 'SPECIAL CASES QUEUE' "
Sql = Sql & "WHERE CLIENT_ID IN "
Sql = Sql & "(select DISTINCT(client_ID) FROM CASES "
Sql = Sql & "WHERE CLIENT_ID IN "
Sql = Sql & "( SELECT CLIENT_ID FROM CASES HAVING
SUM(TOTAL_LENT_AMOUNT ) >= 200000 "
Sql = Sql & "AND CUSTOMER_ID = 12 AND IS_ACTIVE = 1 GROUP
BY CLIENT_ID,CUSTOMER_ID,IS_ACTIVE)) "
Sql = Sql & "AND LAST_ACTION_CODE IN
(10101,10600,10401,77777) "
Sql = Sql & "AND IS_ACTIVE = 1 "
Sql = Sql & "AND CASES.LAST_ACTION_CODE <> 10104 "
Com = New OracleCommand(Sql.ToUpper, Con)
```

```
Com.ExecuteNonQuery()

Sql = " UPDATE CASES  "
Sql = Sql & "SET "
Sql = Sql & "LOCATION_ID = 189,"
Sql = Sql & "LOCATION_DESC = 'SPECIAL NON WORKING QUEUE'"
Sql = Sql & "WHERE CLIENT_ID IN "
Sql = Sql & "(select DISTINCT(client_ID) FROM CASES "
Sql = Sql & "WHERE CLIENT_ID IN "
Sql = Sql & "( SELECT CLIENT_ID FROM CASES HAVING
SUM(TOTAL_LENT_AMOUNT      ) >= 200000 "
Sql = Sql & "AND CUSTOMER_ID = 12 AND IS_ACTIVE = 1 GROUP
BY CLIENT_ID,CUSTOMER_ID,IS_ACTIVE)) "
Sql = Sql & "AND IS_ACTIVE = 1 "
Sql = Sql & "AND CASES.LAST_ACTION_CODE IN
(10101,10600,10401,77777) "
Com = New OracleCommand(Sql.ToUpper, Con)
Com.ExecuteNonQuery()

Sql = " UPDATE CASES  "
Sql = Sql & "SET "
Sql = Sql & "LOCATION_ID = 189,"
Sql = Sql & "LOCATION_DESC = 'SPECIAL NON WORKING QUEUE'"
Sql = Sql & "WHERE CLIENT_ID IN "
Sql = Sql & "(select DISTINCT(client_ID) FROM CASES "
Sql = Sql & "WHERE CLIENT_ID IN "
Sql = Sql & "( SELECT CLIENT_ID FROM CASES HAVING
SUM(TOTAL_LENT_AMOUNT      ) >= 200000 "
Sql = Sql & "AND CUSTOMER_ID = 12 AND IS_ACTIVE = 1 GROUP
BY CLIENT_ID,CUSTOMER_ID,IS_ACTIVE)) "
Sql = Sql & "AND LAST_ACTION_CODE IN
(10101,10600,10401,77777) "
Sql = Sql & "AND IS_ACTIVE = 1 "
Sql = Sql & "AND CASES.LAST_ACTION_CODE IN
(10101,10600,10401,77777) "
Com = New OracleCommand(Sql.ToUpper, Con)
Com.ExecuteNonQuery()

Sql = " UPDATE CASES  "
Sql = Sql & "SET "
Sql = Sql & "LOCATION_ID = 177,"
Sql = Sql & "LOCATION_DESC = 'SETTLEMENT QUEUE' "
Sql = Sql & "WHERE CUSTOMER_ID = 12 "
Sql = Sql & "AND CASE_BEHAVIOR_SCORE = 2 "
Sql = Sql & "AND IS_ACTIVE = 1 "
Com = New OracleCommand(Sql.ToUpper, Con)
Com.ExecuteNonQuery()

Sql = " UPDATE CASES  "
Sql = Sql & "SET "
Sql = Sql & "LOCATION_ID = 179,"
Sql = Sql & "LOCATION_DESC = 'BROKEN PROMISE QUEUE' "
Sql = Sql & "WHERE CUSTOMER_ID = 12 "
Sql = Sql & "AND CASE_BEHAVIOR_SCORE = 3 "
Sql = Sql & "AND IS_ACTIVE = 1 "
Com = New OracleCommand(Sql.ToUpper, Con)
Com.ExecuteNonQuery()
```

```

        Sql = " UPDATE CASES SET "
        Sql = Sql & "LOCATION_ID = 168, "
        Sql = Sql & "LOCATION_DESC = 'PAID OFF PRE ASSIGNMENT
QUEUE' "
        Sql = Sql & "WHERE CUSTOMER_ID = 12 "
        Sql = Sql & "AND INIT_MIN_PAY_AMOUNT <= 50 "
        Sql = Sql & "AND IS_ACTIVE = 1 "
        Sql = Sql & "AND LAST_ACTION_CODE = 10399 "
        Com = New OracleCommand(Sql.ToUpper, Con)
        Com.ExecuteNonQuery()

        Sql = " UPDATE CASES "
        Sql = Sql & "SET "
        Sql = Sql & "LOCATION_ID = 172,"
        Sql = Sql & "LOCATION_DESC = 'NON WORKING QUEUE' "
        Sql = Sql & "WHERE CUSTOMER_ID = 12 "
        Sql = Sql & "AND CASE_BEHAVIOR_SCORE = 4 "
        Sql = Sql & "AND IS_ACTIVE = 1 "
        Com = New OracleCommand(Sql.ToUpper, Con)
        Com.ExecuteNonQuery()
    End Sub

    Private Sub _10101_ExecuteCode(ByVal sender As System.Object,
ByVal e As System.EventArgs)
        Dim InString As String = "10101"
        If LOCATION_ID = 0 Then
            If InStr(InString, LAST_ACTION_CODE.ToString) > 0 Then
                LOCATION_ID = 186
                LOCATION_DESC = "PAY OFF QUEUE"
            End If
        End If
    End Sub

    Private Sub ReturnDates_ExecuteCode(ByVal sender As
System.Object, ByVal e As System.EventArgs)
        Dim Sql As String = " UPDATE CASES "
        Sql = Sql & "SET USER_ID = OLD_USER_ID "
        If Now.DayOfWeek = DayOfWeek.Monday Then
            Sql = Sql & "WHERE TRUNC(RETURN_DATE) = TRUNC(SYSDATE-
3) "
        Else
            Sql = Sql & "WHERE TRUNC(RETURN_DATE) = TRUNC(SYSDATE-
1) "
        End If
        Sql = Sql & "AND CUSTOMER_ID = " & Customer_ID
        Dim Com As OracleCommand = New OracleCommand(Sql.ToUpper,
Con)
        Com.ExecuteNonQuery()
    End Sub
End Class

```

#### 4. Κώδικας Workflow Αναφορών

```

Imports System.Data.OracleClient
Imports System.IO
Imports System.Xml
Imports System.Xml.XPath

```

## Msc Thesis NetCentric Systems

```
Imports System.DBNull
Imports System.Text
Imports Microsoft.VisualBasic
Imports System.Net.Mail
Public Class Workflow1
    Inherits SequentialWorkflowActivity
    Private Con As OracleConnection
    Private ConHistory As OracleConnection
    Private ConLoad As OracleConnection
    Private conString As String
    Private conStringHistory As String
    Private conStringLoad As String
    Private AppPath As String
    Private oReader As OracleDataReader
    Private lReader As OracleDataReader
    Private START_DATE As String
    Private END_DATE As String
    Private Contacts_Count As Long
    Private customer_id As Long = 12
    Private fileWriter As StreamWriter
    Private SUM_PAY_AMOUNT As String
    Private ACTION_CODE As String
    Private OUTCOME As String
    Private COUNT_CONTACTS As String

    Private Sub Initialise_ExecuteCode(ByVal sender As
System.Object, ByVal e As System.EventArgs)
        Console.Clear()
        Console.SetCursorPosition(0, 0)
        Console.ForegroundColor = ConsoleColor.Red
        Console.Write("
Actions File
")
        Console.ForegroundColor = ConsoleColor.White
        Console.SetCursorPosition(1, 3)
        Console.Write("Enter Start Date (FORMAT: DD/MM/YYYY): ")
        START_DATE = Console.ReadLine()
        Console.Write("Enter End Date (FORMAT: DD/MM/YYYY): ")
        END_DATE = Console.ReadLine()
    End Sub

    Private Sub Show_Status(ByVal Msg As String)
        Console.SetCursorPosition(0, 24)
        Console.ForegroundColor = ConsoleColor.Red
        Console.Write("
")
        Console.SetCursorPosition(0, 24)
        Console.WriteLine(Msg)
        Console.ForegroundColor = ConsoleColor.White
    End Sub

    Private Sub ReadConfig_ExecuteCode(ByVal sender As
System.Object, ByVal e As System.EventArgs)
        AppPath =
System.IO.Path.GetDirectoryName(System.Reflection.Assembly.GetExec
utingAssembly().Location())
        Dim document As XPathDocument = New XPathDocument(AppPath
& "\wf_config.xml")
        Dim navigator As XPathNavigator =
document.CreateNavigator()
```



## Msc Thesis NetCentric Systems

```
        Dim iterator As XPathNodeIterator =
navigator.Select("//Connection[@name='MainDB']")

        iterator.MoveNext()
        conString =
iterator.Current.GetAttribute("connectionString", "")

        iterator =
navigator.Select("//Connection[@name='HistoryDB']")
        iterator.MoveNext()
        conStringHistory =
iterator.Current.GetAttribute("connectionString", "")
        iterator =
navigator.Select("//Connection[@name='LoadDB']")
        iterator.MoveNext()
        conStringLoad =
iterator.Current.GetAttribute("connectionString", "")
    End Sub

    Private Sub OpenDB_ExecuteCode(ByVal sender As System.Object,
ByVal e As System.EventArgs)
        Con = New OracleConnection(conString) 'connect to
database
        Con.Open()
        ConHistory = New OracleConnection(conStringHistory)
'connect to database
        ConHistory.Open()
        ConLoad = New OracleConnection(conStringLoad) 'connect to
database
        ConLoad.Open()

    End Sub

    Private Sub DeleteOldFiles_ExecuteCode(ByVal sender As
System.Object, ByVal e As System.EventArgs)
        If My.Computer.FileSystem.FileExists(AppPath &
"\ALPHA_HOME_90_180_" & Mid(END_DATE, 7, 4) & "_" & Mid(END_DATE,
4, 2) & "_" & Mid(END_DATE, 1, 2) & ".txt") Then
            My.Computer.FileSystem.DeleteFile(AppPath &
"\ALPHA_HOME_90_180_" & Mid(END_DATE, 7, 4) & "_" & Mid(END_DATE,
4, 2) & "_" & Mid(END_DATE, 1, 2) & ".txt")
        End If
        If My.Computer.FileSystem.FileExists(AppPath &
"\ALPHA_HOME_30_90_" & Mid(END_DATE, 7, 4) & "_" & Mid(END_DATE,
4, 2) & "_" & Mid(END_DATE, 1, 2) & ".txt") Then
            My.Computer.FileSystem.DeleteFile(AppPath &
"\ALPHA_HOME_30_90_" & Mid(END_DATE, 7, 4) & "_" & Mid(END_DATE,
4, 2) & "_" & Mid(END_DATE, 1, 2) & ".txt")
        End If
    End Sub

    Private Sub ContactCount_ExecuteCode(ByVal sender As
System.Object, ByVal e As System.EventArgs)
        Show_Status("Selecting From Contact Count... ")
        Dim Sql As String = " SELECT COUNT(*) CONTACT_COUNT "
        Sql = Sql & "FROM CONTACT, ACTION_CODE,CASES,PACKS "
        Sql = Sql & "WHERE
ACTION_CODE.ACTION_CODE=CONTACT.ACTION_CODE "
```

```

        Sql = Sql & "AND CASES.CASE_ID=CONTACT.CASE_ID "
        Sql = Sql & "AND PACKS.PACK_ID=CASES.PACK_ID "
        Sql = Sql & "AND PACKS.CUSTOMER_ID = 12 "
        Sql = Sql & "AND PACKS.PACK_TYPE_ID = 12 "
        Sql = Sql & "AND CASES.IS_ACTIVE = 1 "
        Sql = Sql & "AND TRUNC (CONTACT.CONTACT_DATE) >= '" &
START_DATE & "'"
        Sql = Sql & "AND TRUNC (CONTACT.CONTACT_DATE) <= '" &
END_DATE & "'"
        Sql = Sql & "GROUP BY ACTION_CODE.DESCRPTION,
CONTACT.ACTION_CODE "

        Dim Com As OracleCommand = New OracleCommand(Sql.ToUpper,
Con)

        Dim lReader As OracleDataReader = Com.ExecuteReader()
        lReader.Read()
        Contacts_Count = CLng(lReader("CONTACT_COUNT"))
        Console.SetCursorPosition(1, 6)
        Console.Write("Contact Count = " & Contacts_Count)
    End Sub

    Private Sub CreateEmptyFile_ExecuteCode(ByVal sender As
System.Object, ByVal e As System.EventArgs)
        fileWriter =
My.Computer.FileSystem.OpenTextFileWriter(AppPath &
"\ALPHA_NULL_HOME_" & Mid(END_DATE, 7, 4) & "_" & Mid(END_DATE, 4,
2) & "_" & Mid(END_DATE, 1, 2) & ".txt", True,
Encoding.GetEncoding(1253))
        fileWriter.AutoFlush = True
        fileWriter.Flush()
        fileWriter.Close()
    End Sub

    Private Sub CloseConnections_ExecuteCode(ByVal sender As
System.Object, ByVal e As System.EventArgs)
        Con.Close()
        ConHistory.Close()
        ConLoad.Close()
        Show_Status("
")
        Show_Status("Done Processing ")
    End Sub

    Private Sub SelectActionData_ExecuteCode(ByVal sender As
System.Object, ByVal e As System.EventArgs)
        Dim Sql As String = " SELECT CONTACT.ACTION_CODE
ACTION_CODE,ACTION_CODE.DESCRPTION OUTCOME,
COUNT(CONTACT.CONTACT_ID) COUNT_CONTACTS,SUM(CONTACT.PAY_AMOUNT)
SUM_PAY_AMOUNT "
        Sql = Sql & "FROM CONTACT, ACTION_CODE,CASES,PACKS "
        Sql = Sql & "WHERE
ACTION_CODE.ACTION_CODE=CONTACT.ACTION_CODE "
        Sql = Sql & "AND CASES.CASE_ID=CONTACT.CASE_ID "
        Sql = Sql & "AND PACKS.PACK_ID=CASES.PACK_ID "
        Sql = Sql & "AND PACKS.CUSTOMER_ID = 12 "
        Sql = Sql & "AND PACKS.PACK_TYPE_ID = 12 "
        Sql = Sql & "AND ACTION_CODE.LANG_ID = 'GR' "
        Sql = Sql & "AND CASES.IS_ACTIVE = 1 "

```

## Msc Thesis NetCentric Systems

```
        Sql = Sql & "AND TO_NUMBER(CASES.APPLICATION_NO) >= '91'
"
        Sql = Sql & "AND TRUNC(CONTACT.CONTACT_DATE) >= '" &
START_DATE & "' "
        Sql = Sql & "AND TRUNC(CONTACT.CONTACT_DATE) <= '" &
END_DATE & "' "
        Sql = Sql & "GROUP BY ACTION_CODE.DESCRPTION,
CONTACT.ACTION_CODE "

        If My.Computer.FileSystem.FileExists(AppPath &
"\ALPHA_HOME_90_180_" & Mid(END_DATE, 7, 4) & "_" & Mid(END_DATE,
4, 2) & "_" & Mid(END_DATE, 1, 2) & ".csv") Then
            My.Computer.FileSystem.DeleteFile(AppPath &
"\ALPHA_HOME_90_180_" & Mid(END_DATE, 7, 4) & "_" & Mid(END_DATE,
4, 2) & "_" & Mid(END_DATE, 1, 2) & ".csv")
        End If

        Dim Com As OracleCommand = New OracleCommand(Sql.ToUpper,
Con)
        Dim Report1Reader As OracleDataReader =
Com.ExecuteReader()

        AppPath =
System.IO.Path.GetDirectoryName(System.Reflection.Assembly.GetExec
utingAssembly().Location())
        Dim fileWriter As StreamWriter =
My.Computer.FileSystem.OpenTextFileWriter(AppPath &
"\ALPHA_HOME_90_180_" & Now.Year & "_" & Format(Now.Month, "00") &
 "_" & Format(Now.Day, "00") & ".csv", True,
Encoding.GetEncoding(1253))
        fileWriter.AutoFlush = True
        fileWriter.WriteLine("ΚΩΔΙΚΟΣ;ΠΕΡΙΓΡΑΦΗ;ΑΡΙΘΜΟΣ
ΕΠΑΦΩΝ;ΠΟΣΟ ΠΛΗΡΩΜΗΣ - ΥΠΟΣΧΕΣΗΣ")
        fileWriter.Flush()
        While Report1Reader.Read() = True
            fileWriter.AutoFlush = True
            Try
                SUM_PAY_AMOUNT =
CStr(Report1Reader("SUM_PAY_AMOUNT"))
            Catch ex As Exception
                SUM_PAY_AMOUNT = ""
            End Try

            fileWriter.WriteLine(CStr(Report1Reader("ACTION_CODE")) & ";" &
CStr(Report1Reader("OUTCOME")) & ";" &
CStr(Report1Reader("COUNT_CONTACTS")) & ";" & SUM_PAY_AMOUNT)
            fileWriter.Flush()
        End While
        fileWriter.Close()
    End Sub

    Private Sub Select30_90_Data_ExecuteCode(ByVal sender As
System.Object, ByVal e As System.EventArgs)
        Dim Sql As String = " SELECT CONTACT.ACTION_CODE
ACTION_CODE, ACTION_CODE.DESCRPTION OUTCOME,
COUNT(CONTACT.CONTACT_ID) COUNT_CONTACTS, SUM(CONTACT.PAY_AMOUNT)
SUM_PAY_AMOUNT "
        Sql = Sql & "FROM CONTACT, ACTION_CODE, CASES, PACKS "
```

```

        Sql = Sql & "WHERE
ACTION_CODE.ACTION_CODE=CONTACT.ACTION_CODE "
        Sql = Sql & "AND CASES.CASE_ID=CONTACT.CASE_ID "
        Sql = Sql & "AND PACKS.PACK_ID=CASES.PACK_ID "
        Sql = Sql & "AND PACKS.CUSTOMER_ID = 12 "
        Sql = Sql & "AND PACKS.PACK_TYPE_ID = 12 "
        Sql = Sql & "AND ACTION_CODE.LANG_ID = 'GR' "
        Sql = Sql & "AND CASES.IS_ACTIVE = 1 "
        Sql = Sql & "AND TO_NUMBER(CASES.APPLICATION_NO) < '91' "
        Sql = Sql & "AND TRUNC(CONTACT.CONTACT_DATE) >= '" &
START_DATE & "' "
        Sql = Sql & "AND TRUNC(CONTACT.CONTACT_DATE) <= '" &
END_DATE & "' "
        Sql = Sql & "GROUP BY ACTION_CODE.DESCRPTION,
CONTACT.ACTION_CODE "

        If My.Computer.FileSystem.FileExists(AppPath &
"\ALPHA_HOME_30_90_" & Mid(END_DATE, 7, 4) & "_" & Mid(END_DATE,
4, 2) & "_" & Mid(END_DATE, 1, 2) & ".csv") Then
            My.Computer.FileSystem.DeleteFile(AppPath &
"\ALPHA_HOME_30_90_" & Mid(END_DATE, 7, 4) & "_" & Mid(END_DATE,
4, 2) & "_" & Mid(END_DATE, 1, 2) & ".csv")
        End If

        Dim Com As OracleCommand = New OracleCommand(Sql.ToUpper,
Con)
        Dim Report2Reader As OracleDataReader =
Com.ExecuteReader()

        AppPath =
System.IO.Path.GetDirectoryName(System.Reflection.Assembly.GetExec
utingAssembly().Location())
        Dim fileWriter As StreamWriter =
My.Computer.FileSystem.OpenTextFileWriter(AppPath &
"\ALPHA_HOME_30_90_" & Now.Year & "_" & Format(Now.Month, "00") &
 "_" & Format(Now.Day, "00") & ".csv", True,
Encoding.GetEncoding(1253))
        fileWriter.AutoFlush = True
        fileWriter.WriteLine("ΚΩΔΙΚΟΣ;ΠΕΡΙΓΡΑΦΗ;ΑΡΙΘΜΟΣ
ΕΠΑΦΩΝ;ΠΟΣΟ ΠΛΗΡΩΜΗΣ - ΥΠΟΣΧΕΣΗΣ")
        fileWriter.Flush()
        While Report2Reader.Read() = True
            fileWriter.AutoFlush = True
            Try
                SUM_PAY_AMOUNT =
CStr(Report2Reader("SUM_PAY_AMOUNT"))
            Catch ex As Exception
                SUM_PAY_AMOUNT = ""
            End Try

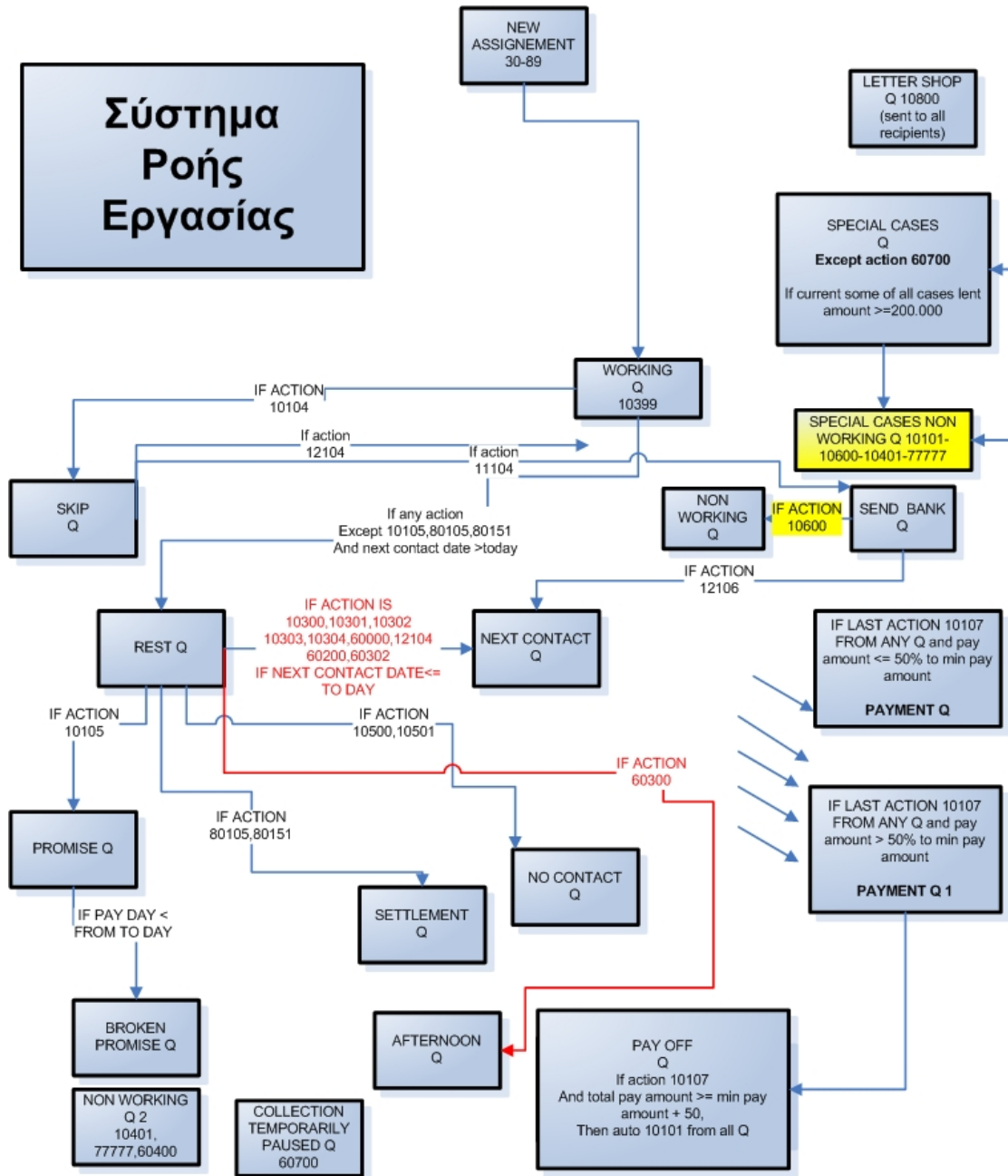
            fileWriter.WriteLine(CStr(Report2Reader("ACTION_CODE")) & ";" &
CStr(Report2Reader("OUTCOME")) & ";" &
CStr(Report2Reader("COUNT_CONTACTS")) & ";" & SUM_PAY_AMOUNT)
            fileWriter.Flush()
        End While
        fileWriter.Close()
    End Sub

```

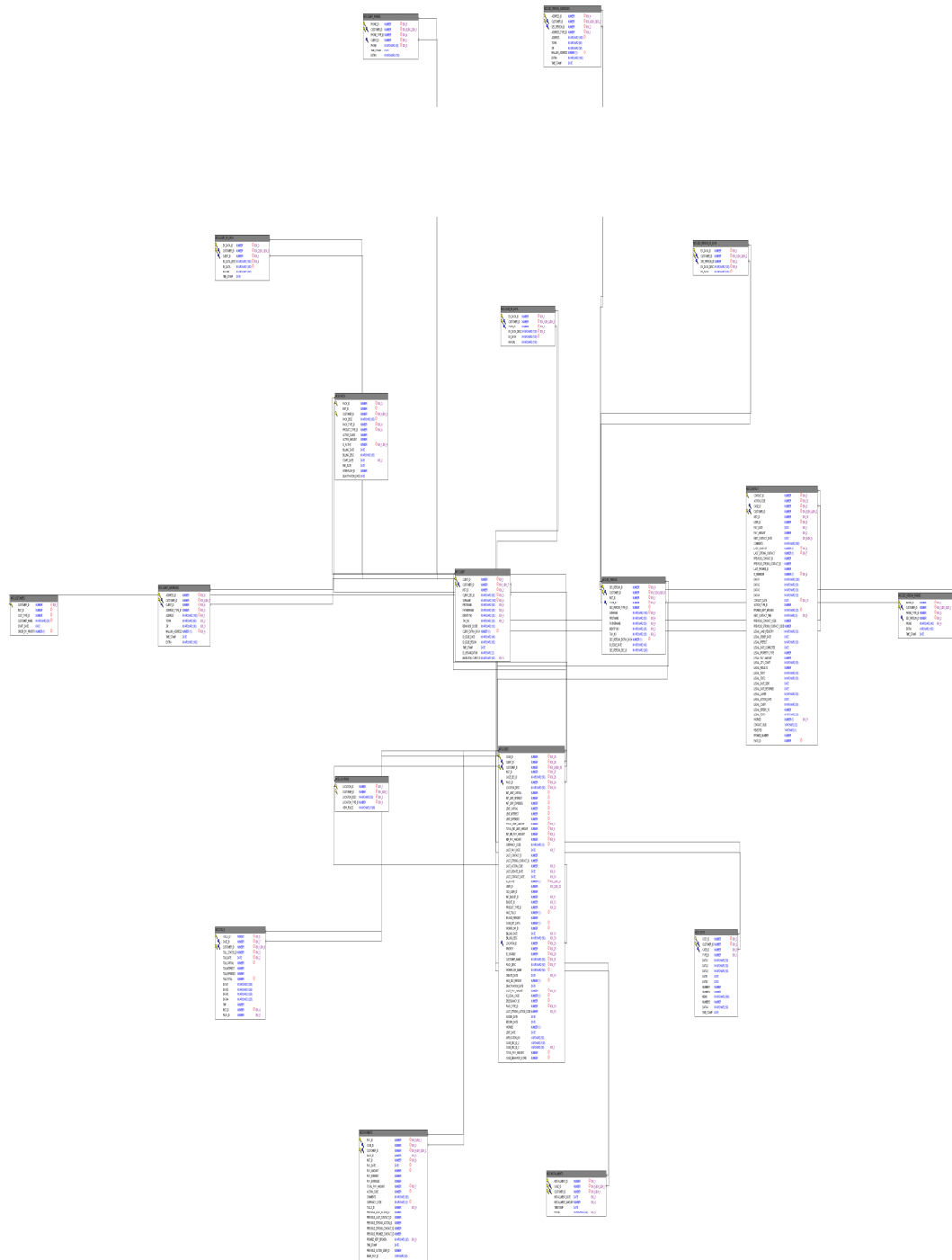
## Msc Thesis NetCentric Systems

```
Private Sub SendMail_ExecuteCode(ByVal sender As
System.Object, ByVal e As System.EventArgs)
    Dim MsgString As String = "Daily Action Alpha Real
Estate."
    Dim Subject As String = "Alpha Bank Home - Daily Actions"
    Dim Recivers As String = "a.pateli@mellon.com.gr"
    Dim mailSender As String = "mellonc_desc@mellon.com.gr"
    Dim mailAttachment1 As System.Net.Mail.Attachment = New
Attachment(AppPath & "\ALPHA_HOME_30_90_" & Now.Year & "_" &
Format(Now.Month, "00") & "_" & Format(Now.Day, "00") & ".csv",
"")
    Dim mailAttachment2 As System.Net.Mail.Attachment = New
Attachment(AppPath & "\ALPHA_HOME_90_180_" & Now.Year & "_" &
Format(Now.Month, "00") & "_" & Format(Now.Day, "00") & ".csv",
"")
    Try
        Dim message As New MailMessage(mailSender, Recivers,
Subject, MsgString)
        message.CC.Add("conceptus_desc@mellon.com.gr")
        message.CC.Add("mellonc_desc@mellon.com.gr")
        message.Attachments.Add(mailAttachment1)
        message.Attachments.Add(mailAttachment2)
        Dim emailClient As New SmtpClient("192.168.0.4")
        emailClient.Send(message)
    Catch ex As Exception
        Console.WriteLine("Error in Sending Report. " &
ex.Message)
        Console.ReadLine()
    End Try
End Sub
End Class
```

## 5. Σκαρίφιμα Διαχείρισης Συνεργασίας



## 6. Σχισιακό Σχήμα Βάσης



## Βιβλιογραφία

- [1] Workflow Management Coalition. (1999). *Terminology and Glossary*. Issue 3.0, No. WFMC-TC-1011.
- [2] E. Bertino, E. Ferrari, V. Atluri. (1999). "Specification and Enforcement of Authorization Constraints in Workflow Management Systems." In: *ACM Transactions on Information and Systems Security*, 2(1). pp. 65-104.  
<http://www.acm.org/pubs/articles/journals/tissec/1999-2-1/p65-bertino/p65-bertino.pdf>
- [3] V. Atluri, W-K. Huang. (1996). "An Authorization Model for Workflows,"  
In: *Proceedings of the Fifth European Symposium on Research in Computer Security*. Rome, Italy. pp. 44-64.  
<http://cimic.rutgers.edu/atluri/esorics96.ps>
- [4] Workflow Management Coalition, Author: David Hollingsworth. (1995). *The Workflow Reference Model*. Issue 1.1, No. TC00-1003.
- [5] V. Atluri and W-K. Huang. (1996). "Modeling and Analysis of Workflows using Petri Nets." *Technical report*. CMIC, Rutgers University.
- [6] M.S. Olivier, R.P. van de Riet and E. Gudes. (1998). "Specifying Application-level Security in Workflow Systems." In: *R. Wagner), Database and Expert System Applications, IEE Computer Society*.
- [7] R. Allen. (2001). "Workflow: An Introduction." In: *The Workflow Handbook 2001*. Published in association with the Workflow Management Coalition. <http://www.wfmc.org/information/info.htm>
- [8] The Workflow Management Coalition. (2001). *Introduction to the Workflow Management Coalition*.  
<http://www.wfmc.org/about.htm>
- [9] Workflow Management Coalition. (1998). *Workflow Security Considerations*.  
White Paper Issue 1.0, No. WFMC-TC-1019.  
<http://www.aiim.org/wfmc/standards/docs/secure1.pdf>
- [10] D. Parker. (1995). "A New Framework for Information Security to Avoid Information Anarchy." In: *Proceedings of IFIP TC 11 Eleventh International Conference on Information Security*. Cape Town. pp. 155-164.
- [11] "Information Processing Systems - Open System Interconnection -



- Basic Reference Model - Part 2: Security Architecture.” (1989).
- [12] B. Schneier. (2000). *Secrets and Lies. Digital Security in a Networked World*. Wiley.
- [13] C.P. Pfleeger. (1997). *Security in Computing*. (International Edition).  
Prentice Hall International.
- [14] B. Davis. (1998). “In Certificates We Trust.” In: *Information Week Southern Africa*. May 1998. pp. 54-56.
- [15] S. Osborne, R. Sandhu and Q. Munawer. (2000) “Configuring Rolebased Access Control to Enforce Mandatory and Discretionary Access Control Policies.” In: *ACM Transactions on Information and Systems Security*, Vol. 3, No. 2.
- [16] F. Leyman and D. Roller. (2000). *Production Workflow: Concepts and Techniques*. Prentice-Hall.
- [17] R.A. Botha, J.H.P. Eloff. (2001). “Designing Role Hierarchies for Access Control in Workflow Systems.”  
In: *Proceedings of COMPSAC 2001*. Chicago, USA.  
<http://www.petech.ac.za/csresearch/publications/pub44.html>
- [18] D.G. Cholewka, R.A. Botha, J.H.P. Eloff. (August 2000). “A Context-sensitive Access Control Model and Prototype Implementation.”  
In: *Information Security for Global Information Infrastructures: IFIP TC 11 Sixteenth Annual Working Conference on Information Security*. Beijing, China. pp. 341-350.
- [19] B. Kuechler and B. Burg. (1998). “Using Intentional Information to Coordinate Inter-operating Workflows.” In: *OOPSLA '98 Business Object Workshop 4*.  
<http://jeffsutherland.org/oopsla98/kuechler.html>
- [20] F. Rabbitti, E. Bertino, W. Kim and D. Woelk. (1991). “A Model of Authorization for Next-Generation Database Systems.” In: *ACM Transactions on Database Systems*. pp. 88-131.
- [21] E. Gudes, M.S. Olivier and R.P. van de Riet. (1999). “Modelling, Specifying and Implementing Workflow Security in Cyberspace.” In: *Journal of Computer Security*. 7, 4, pp. 287-315.
- [22] R. Kuhn, D. Ferriolo and J. Cugini. (December 1995). “Role-

- based  
Access Control (RBAC): Features and Motivations.” In:  
*Proceedings  
of 11<sup>th</sup> Annual Computer Security Application Conference*. New  
Orleans,  
LA, pp. 241-248.
- [23] R. Sandhu. (1997). “Rationale for the RBAC96 Family of Access  
Control Models.” In: *Proceedings of the First ACM Workshop on  
Role-based Access Control*.
- [24] R. Sandhu, E. Coyne, H. Feinstein, C. Youman. (1996). “Role-  
based  
Access Control Models.” In: *IEEE Computer* 29(2). pp. 38-47.  
<http://dlib.computer.org/co/books/co1996/pdf/r2038.pdf>
- [25] J.D. Moffett. (October 1998). “Control Principles and Role  
Hierarchies.”  
In: *3<sup>rd</sup> ACM Workshop on Role-based Access Control*. George  
Mason University, Fairfax, VA.
- [26] W-K. Huang and V. Atluri. (1999). “SecureFlow: A Secure  
Webenabled  
Workflow Management System.” In: *Proceedings of the  
Fourth ACM Workshop on Role-based Access Control*. Fairfax, VA  
USA, pp. 83-94.  
[http://www.acm.org/pubs/citations/proceedings/commsec  
/319171/p83-huang](http://www.acm.org/pubs/citations/proceedings/commsec/319171/p83-huang)
- [26] Lars M. Kristensen, Soren Christensen, Kurt Jensen, “The  
practitioner’s guide to coloured Petri nets”, CPN  
Group, Department of Computer Science, University of  
Aarhus, Denmark; 1998 Springer-Verlag
- [27] W.M.P. van der Aalst, “The Application of Petri Nets to  
Workflow Management”, Department of Mathematics  
and Computing Science, Eindhoven University of  
Technology
- [28] Dongsheng Liu, Jianmin Wang, Stephen C.F. Chan,  
Jiaguang Sun, Li Zhang, “Modeling workflow processes  
with colored Petri nets”, Department of Computer  
Science and Technology, Tsinghua University,  
Department of Computing, The Hong Kong Polytechnic  
University, 2002 – Computers In Industry 49 – Elsevier
- [29]

Lisa Wells, "Performance Analysis using Coloured Petri Nets", PhD Dissertation, 2002, Department of Computer Science, University of Aarhus, Denmark

[30]

Mariska Netjes, Wil M.P. van der Aalst, Hajo A. Reijers, "Analysis of resource-constrained processes with Colored Petri Nets", Eindhoven University of Technology

[31]

Christian W. Guenther, Wil M.P. van der Aalst, "Modeling The Case Handling Principles with Colored Petri Nets", Eindhoven University of Technology

[32]

Irene Vanderfeesten, Wil van der Aalst, Hajo A. Reijers, "Modelling a product based workflow system in CPN tools", Eindhoven University of Technology

[33]

M. Pesic, W.M.P. van der Aalst, "Modeling Work Distribution Mechanisms Using Colored Petri Nets", Department of Technology Management, Eindhoven University of Technology

[34]

CPN Tools Homepage,  
<http://wiki.daimi.au.dk/cpntools/cpntools.wiki>

[35]

CPN Tools Help Files (version 2.0.0)

[36]

WikiPedia – The Free Online Encyclopedia,  
<http://www.wikipedia.org/>

