

Ανάπτυξη & Σχεδίαση Λογισμικού (HY420)

Διάλεξη 9: Σχεδίαση Συστήματος



Χαρακτηριστικά Καλού Σχεδίου: Αναγνώριση & Χειρισμός Εξαιρέσεων



2

- Αμυντική σχεδίαση
- Τυπικές εξαιρέσεις:
 - Αδυναμία παροχής μιας υπηρεσίας.
 - Παροχή λανθασμένης υπηρεσίας.
 - Αλλοίωση δεδομένων.
- Τεχνικές σύλληψης εξαιρέσεων:
 - Αθροίσματα ελέγχου.
 - Πλεονασμός.
 - Timeouts.
- Χειρισμός εξαιρέσεων:
 - Οπισθοδρόμηση και επανάληψη (με διαφορετική στρατηγική).
 - Οπισθοδρόμηση και διόρθωση.
 - Οπισθοδρόμηση, αναφορά και άρνηση παροχής υπηρεσίας.

Χαρακτηριστικά Καλού Σχεδίου: Παρεμπόδιση & Ανοχή Σφαλμάτων



3

- **Παθητική** ανίχνευση σφαλμάτων:
 - Το σύστημα περιμένει έως ότου προκύψει δυσλειτουργία.
- **Ενεργητική** ανίχνευση σφαλμάτων:
 - Περιοδικός έλεγχος για σφάλματα ή συμπτώματα σφαλμάτων.
 - Π.χ. πολιτική αμοιβαίας υποψίας από τα συστατικά.
 - Πλεονασμός
 - Προγραμματισμός n-εκδόσεων (σχεδίαση & υλοποίηση από ανεξάρτητες ομάδες)
 - Όχι αξιόπιστο (κοινές παραστάσεις κατά την εκπαίδευση).
 - Κόστος
 - Ψηφοφορίες.
- **Διόρθωση** σφαλμάτων
 - Διόρθωση ζημιάς.
 - Αλλαγή προϊόντος ώστε να εξαλειφθεί το σφάλμα.
 - Άλλοι τρόποι χειρισμού: Restart (Windows ☺)
 - Αυτόνομα / Αυτοθεραπευόμενα συστήματα



Ανοχή στα Σφάλματα

- «Απομόνωση» της ζημιάς
- Προβλήματα:
 - Σε πολύπλοκα συστήματα δύσκολο να προληφθούν τα πιθανά σφάλματα.
 - Τα πολύπλοκα συστήματα όμως είναι αυτά που είναι πιθανό να έχουν σφάλματα.
 - Ο κώδικας που υλοποιεί την ανοχή μπορεί με τη σειρά του να έχει σφάλματα.



Δομημένη Σχεδίαση

- Yourdon – Constantine, 1979
- Σχεδιασμός Δομής Προγράμματος
- Κατευθυντήριες γραμμές
 - Όχι αλγόριθμος βήμα προς βήμα
- Σύστημα λογισμικού: έχει εισόδους τις οποίες μετασχηματίζει στις επιθυμητές εξόδους
 - Κύριο πρόβλημα: Υλοποίηση συνάρτησης μετασχηματισμού
- Απώτερος στόχος
 - Ιεραρχική δομή προγράμματος
 - Χαμηλή σύζευξη μεταξύ μονάδων
 - Υψηλή συνεκτικότητα μονάδων



Παραγοντοποίηση

- **Παραγοντοποίηση** (Factoring): Διαδικασία αποσύνθεσης μονάδας ώστε ο κύριος όγκος της εργασίας να πραγματοποιείται από τις υποκείμενες μονάδες
- Σύστημα **πλήρως παραγοντοποιημένο**:
 - Όλη η επεξεργασία πραγματοποιείται από ατομικές (αυτάρκειες) μονάδες του χαμηλοτέρου επιπέδου
 - Οι μη ατομικές μονάδες εκτελούν τον έλεγχο και συντονισμό
- Στόχος δομημένης σχεδίασης:
 - Δημιουργία πλήρως παραγοντοποιημένων συστημάτων

Στρατηγική Δομημένης Σχεδίασης



7

- Εντοπισμός ροών εισόδου, εξόδου και κεντρικών μετασχηματισμών
- Μονάδες υψηλού επιπέδου υλοποιούν αυτές τις δραστηριότητες
 - Στη συνέχεια **εκλεπτύνεται** η σχεδιάσή τους
- Τέσσερα βήματα:
 - Μοντελοποίηση του προβλήματος ως ΔΡΔ
 - Εντοπισμός εισόδων – εξόδων και αποσαφήνιση των ορίων
 - Παραγοντοποίηση πρώτου επιπέδου
 - Παραγοντοποίηση εισόδου, εξόδου, μετασχηματισμών



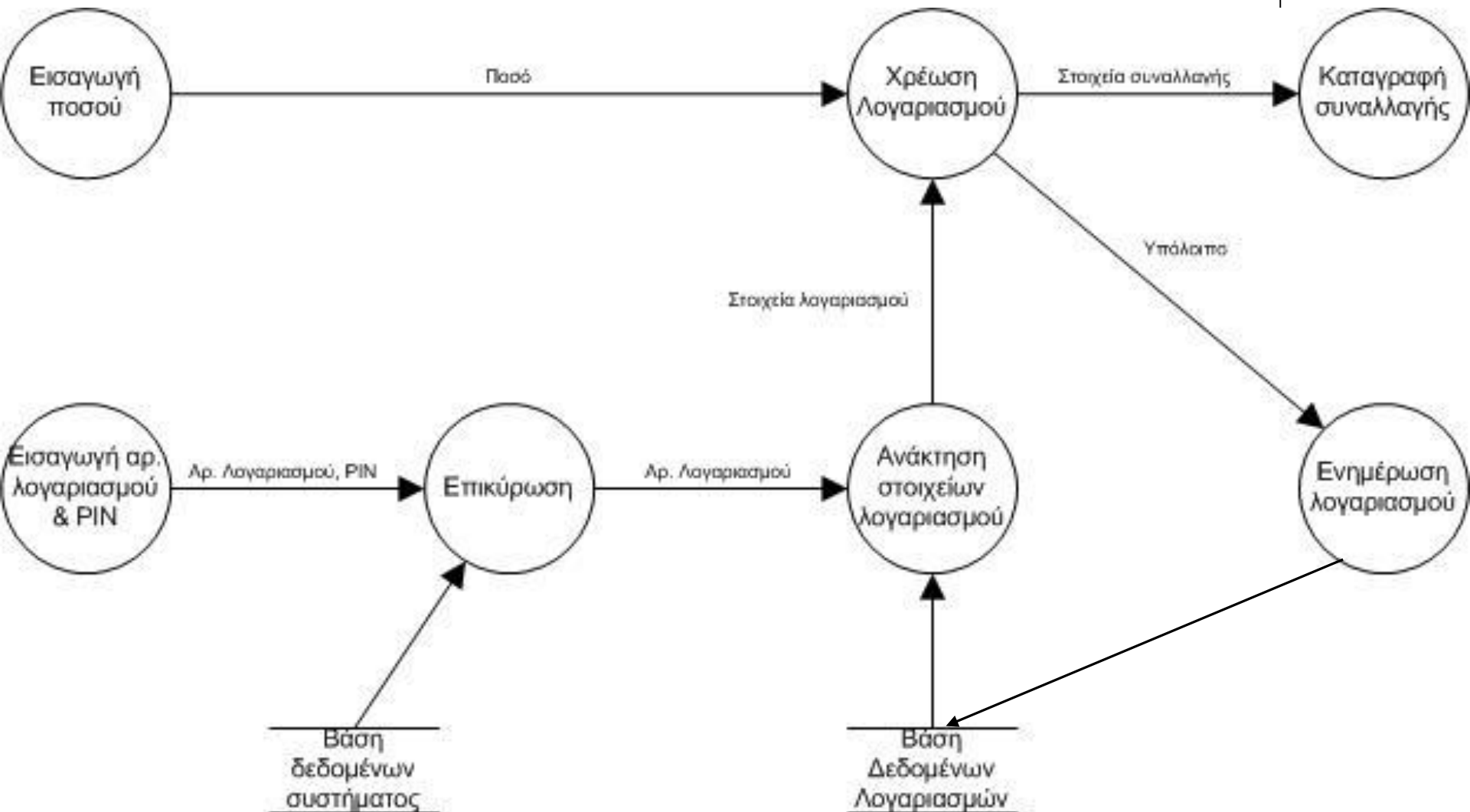
Μεθοδολογία (1)

- Συνοπτική περιγραφή συστήματος:
Μηχάνημα αυτόματων συναλλαγών (ATM)
που επιτρέπει ανάληψη από τους πελάτες
- Βήμα 1ο: Κατασκευή Διαγράμματος Ροής
Δεδομένων

Διάγραμμα Ροής Δεδομένων



9





Μεθοδολογία (2)

- Εντοπισμός των πιο **αφηρημένων** μορφών στοιχείων εισόδου και εξόδου
 - Οι μετασχηματισμοί του λογισμικού δεν επιδρούν απευθείας στην πρωτογενή μορφή της εισόδου
 - Οι είσοδοι μετατρέπονται σταδιακά σε μορφή επεξεργάσιμη από τους κεντρικούς μετασχηματισμούς
 - Αντίστοιχα και στην έξοδο
 - Στόχος: **διαχωρισμός** των μετασχηματισμών που “**προετοιμάζουν**” τα δεδομένα εισόδου και εξόδου από τους μετασχηματισμούς που **εκτελούν** τις κύριες διεργασίες



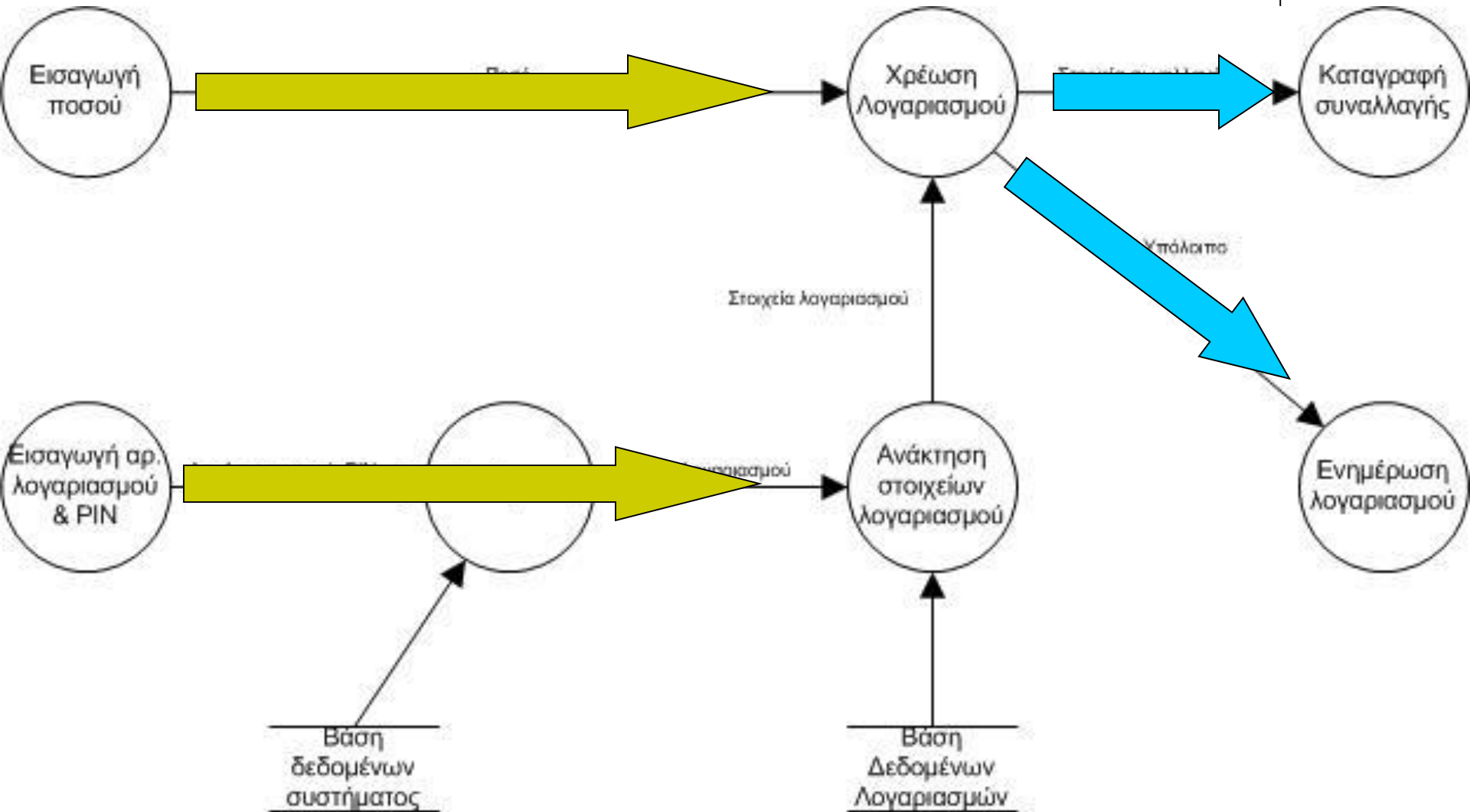
Μεθοδολογία (2)

- Εντοπισμός των πιο **αφηρημένων** μορφών στοιχείων εισόδου και εξόδου
 - Οι πιο αφηρημένες μορφές δεδομένων εισόδου προκύπτουν συνήθως μετά από λειτουργίες όπως έλεγχος σφαλμάτων, επικύρωση δεδομένων, μορφοποίηση και μετατροπή.
 - Π.χ. μία είσοδος (χρηματικό ποσό) η οποία αρχικά δίδεται σε \$ και στη συνέχεια μετατρέπεται σε €, παραμένει είσοδος του συστήματος
 - Οι πιο αφηρημένες μορφές εισόδου εντοπίζονται ξεκινώντας από τις φυσικές εισόδους και προχωρώντας προς τις εξόδους στο διαγράμματος ροής δεδομένων
 - Στόχος: Η μεγαλύτερη δυνατή απομάκρυνση από την φυσική είσοδο χωρίς να απολεσθεί η εισερχόμενη φύση των δεδομένων

Παράδειγμα



12





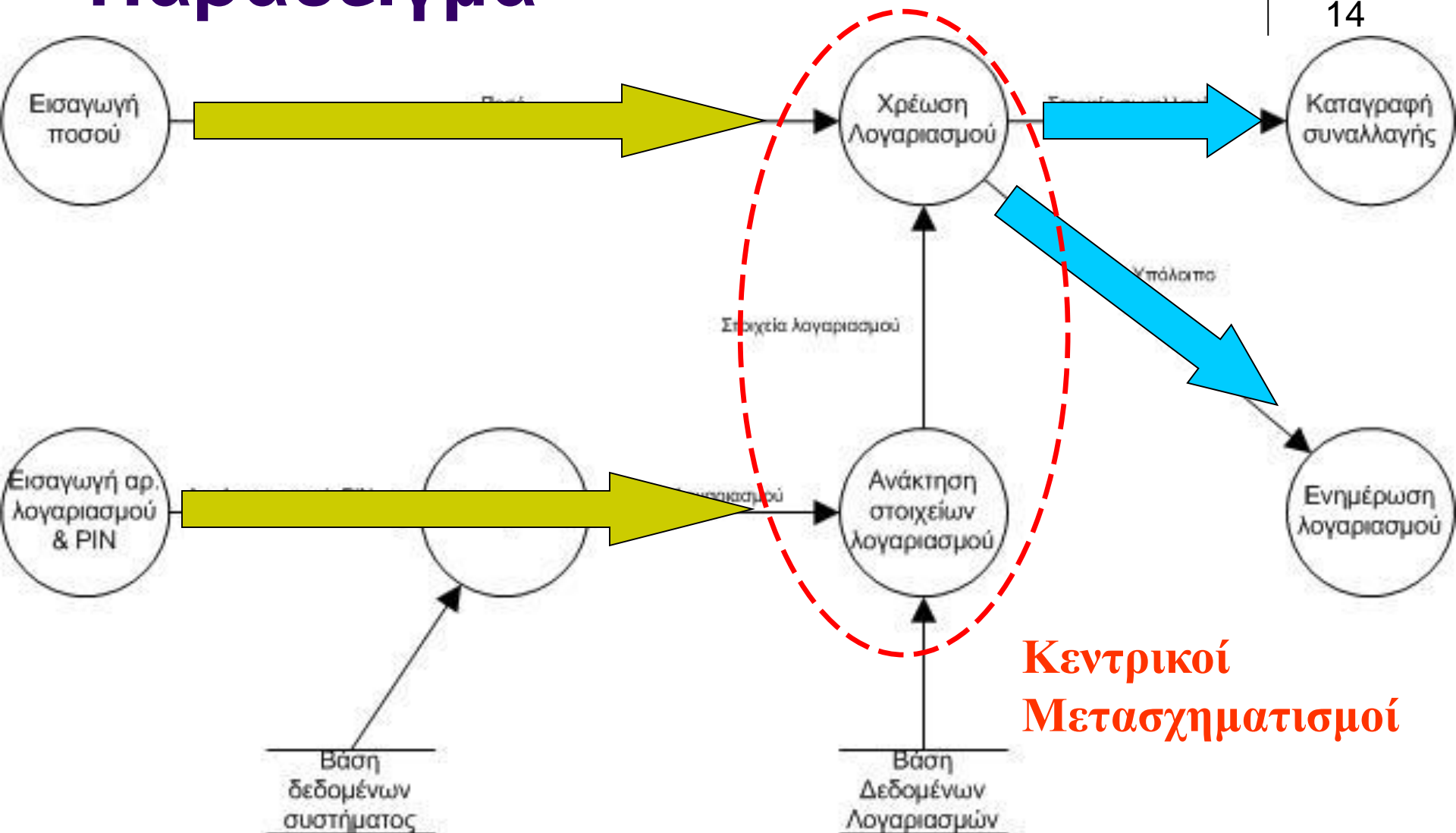
Μεθοδολογία (3)

- **Κεντρικοί μετασχηματισμοί**
 - Οι μετασχηματισμοί που απομένουν μεταξύ των πιο αφηρημένων μορφών εισόδου και εξόδου ονομάζονται κεντρικοί μετασχηματισμοί
 - Οι μονάδες λογισμικού που τους υλοποιούν **δεν** ασχολούνται με μετατροπή των εισόδων σε κατάλληλη μορφή, ελέγχου των δεδομένων κτλ

Παράδειγμα



14



**Κεντρικοί
Μετασχηματισμοί**

Παραγοντοποίηση 1^{ου} Επιπέδου



15

- Έχοντας εντοπίσει τις πιο αφηρημένες μορφές εισόδου και εξόδου καθώς και τους κεντρικούς μετασχηματισμούς, είναι δυνατόν να γίνει ο **εντοπισμός** των πρώτων **μονάδων λογισμικού**
 - Καθορίζεται **μονάδα (main) συντονισμού** που καλεί τις υποκείμενες μονάδες
 - Για κάθε μία από τις πιο αφηρημένες εισόδους καθορίζεται υποκείμενη μονάδα στη main (ή μία μονάδα εισόδου για όλες τις εισόδους)
 - Κάθε μονάδα εισόδου έχει ως σκοπό την παροχή στη μονάδα main των πιο **αφηρημένων δεδομένων** εισόδου για τα οποία δημιουργήθηκε
 - Αντιστοίχως για τις μονάδες εξόδου

Παραγοντοποίηση 1^{ου} Επιπέδου



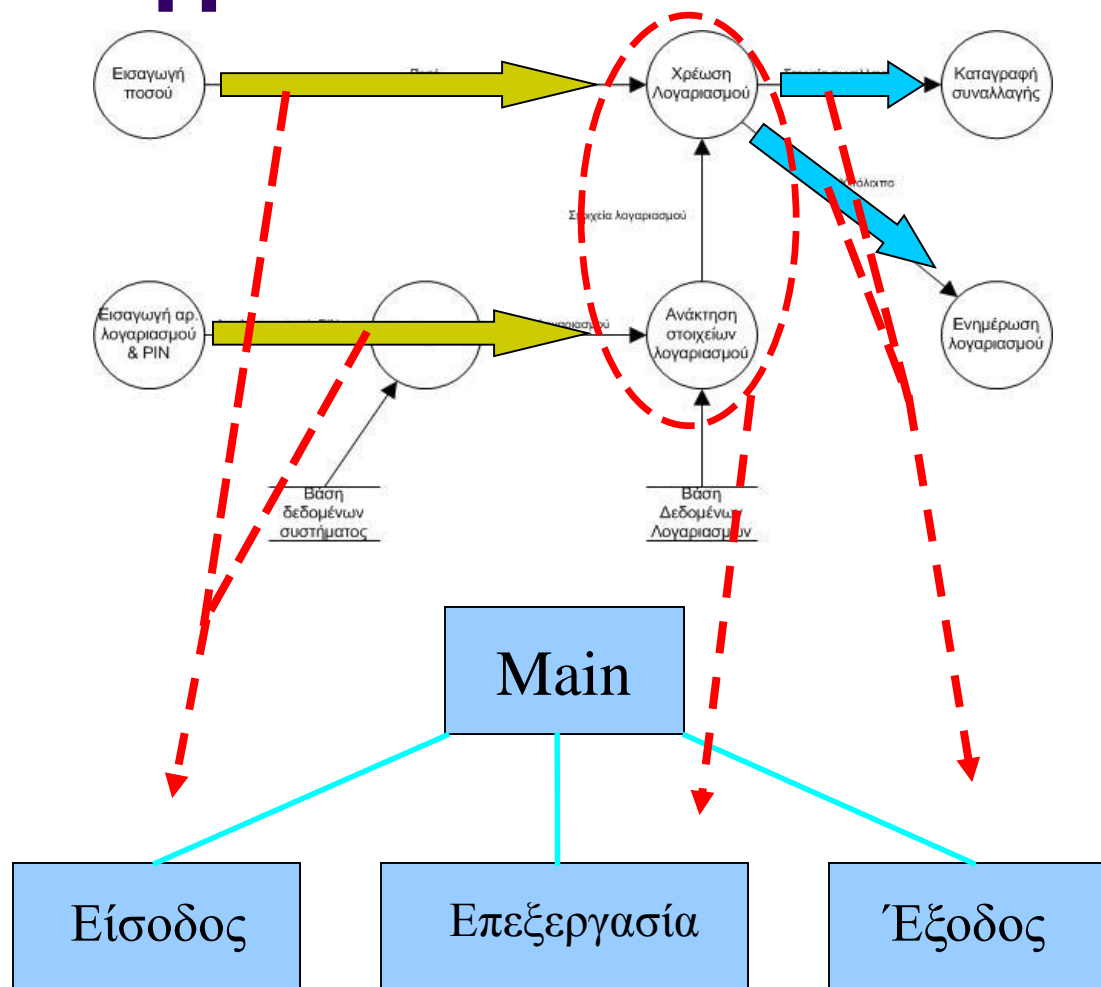
16

- Για κάθε **κεντρικό μετασχηματισμό** καθορίζεται μια υποκείμενη μονάδα στη main (ή μία μονάδα για όλους τους κεντρικούς μετασχηματισμούς)
 - Οι μονάδες αυτές, δεν λαμβάνουν μόνο δεδομένα από την main αλλά της παρέχουν και τα αποτελέσματα των μετασχηματισμών που υλοποιούν

Παράδειγμα

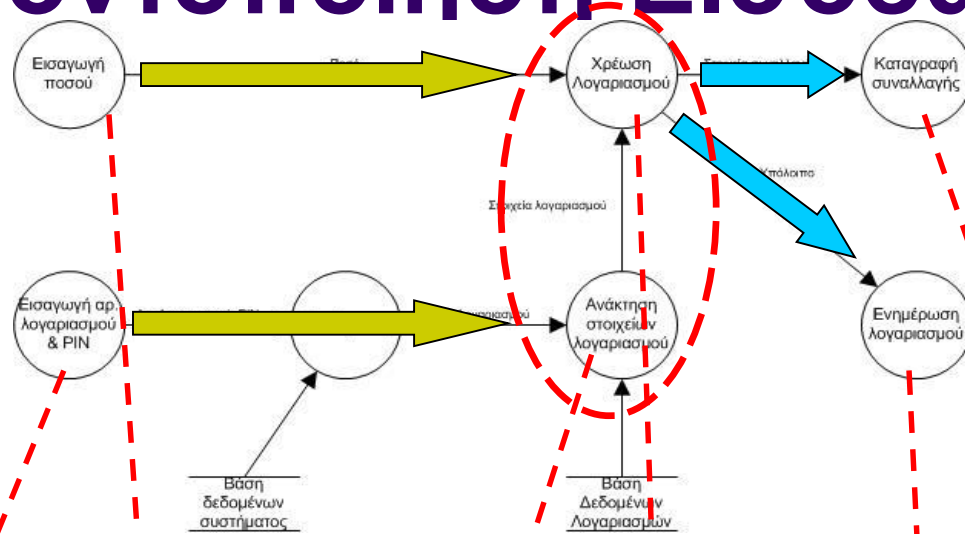


17





Παραγοντοποίηση Εισόδων



User:

User_ID : string

amount : int

verified: bool

Main

Είσοδος

Επεξεργασία

Έξοδος

account : int

amount : int

Χρήση Αρ. Λογαρ.

Λήψη Ποσού

Ανάκτ. στοιχ.

Χρέωση Λογαρ.

Ενημέρ. Λογαρ.

Καταγρ. Συναλ.

Χρήση Αρ. Λογαρ.

Λεπτομερής Σχεδίαση (Σχεδίαση Προγράμματος)



19

- Τελευταίο βήμα πριν την κωδικοποίηση
- Εκτενής περιγραφή των μονάδων
 - Όχι του συστήματος
- Έλεγχος πληρότητας προδιαγραφών σε επίπεδο μονάδος
- Οι προδιαγραφές προγράμματος μονάδας: Οδηγίες προς τον προγραμματιστή που περιγράφουν είσοδο, έξοδο και επεξεργασία για τη μονάδα
- Αναφορά σε συγκεκριμένους τύπους δεδομένων
- Περιγραφή των βημάτων του αλγορίθμου προς υλοποίηση



Διαγράμματα Ροής Ελέγχου

- Δεν αρκεί η περιγραφή της ροής των δεδομένων μέσω διαγραμμάτων ροής δεδομένων
 - Π.χ. Εφαρμογές των οποίων η λειτουργία εξαρτάται από γεγονότα (events) τα οποία παράγουν πληροφορία ελέγχου (control information).
- Μοντελοποίηση της **ροής του ελέγχου**
 - Παράλληλα με τη μοντελοποίηση της ροής των δεδομένων
- Σε διάγραμμα ροής δεδομένων παριστάνονται:
 - Στοιχεία ελέγχου (π.χ. γεγονότα) με **διακεκομμένα** βέλη
 - Διεργασίες ελέγχου με διακεκομμένους κύκλους



Διαγράμματα Καταστάσεων

- **Διαγράμματα καταστάσεων:** Για την περιγραφή μονάδων που υλοποιούν διεργασίες ελέγχου
 - Αναπαριστώνται οι δυνατές καταστάσεις μιας λειτουργίας ή ενός στοιχείου
 - Μοντελοποιεί τις μεταβάσεις μεταξύ των καταστάσεων που λαμβάνουν χώρα όταν συμβαίνουν κάποια γεγονότα
 - Κάθε διάγραμμα περιέχει μία αρχική κατάσταση και από αυτήν κάθε κατάσταση είναι προσβάσιμη
- Χρησιμοποιούνται:
 - Στη φάση της ανάλυσης
 - Στην αντικειμενοστραφή ανάλυση για την περιγραφή της συμπεριφοράς κλάσεων



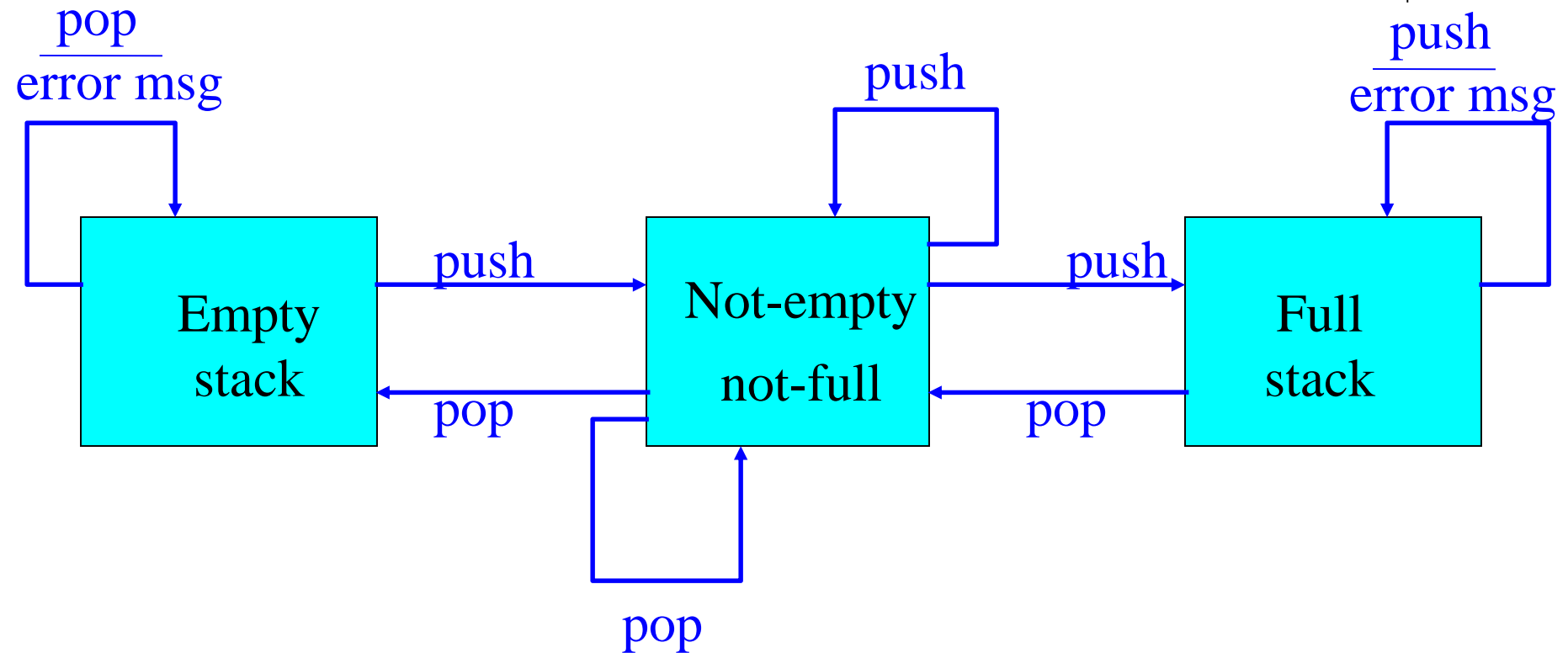
Διαγράμματα Καταστάσεων

- Σε ένα διάγραμμα καταστάσεων δεν αναπαριστώνται όλες οι πραγματικές καταστάσεις του συστήματος
 - Μόνο οι **λογικές καταστάσεις**
- **Λογική κατάσταση**: Σύνολο καταστάσεων για τις οποίες η συμπεριφορά (αντίδραση) σε όλα τα δυνατά γεγονότα είναι η ίδια
 - Δύο διαφορετικές λογικές καταστάσεις διαφέρουν στη συμπεριφορά τους για ένα τουλάχιστον γεγονός
- Παράδειγμα: Έστω ότι μοντελοποιούμε τη λειτουργία μιας στοίβας με χρήση διαγράμματος καταστάσεων
 - Η συμπεριφορά όλων των καταστάσεων για τις οποίες το μέγεθος της στοίβας δεν είναι μηδέν και είναι μικρότερο από το μέγιστο μέγεθος, είναι η ίδια
 - Άρα αντιστοιχούν στην ίδια λογική κατάσταση



23

Παράδειγμα: Stack





Ζητήματα Ταυτοχρονισμού

Υπόλοιπο λογαριασμού: 5000.00 €

Χρήστος (φίλος ;-) Ελένης

Ελένη

Τράπεζα

balance(5000) =
account balance

Ανάληψη 1000 €

balance(4000) =
balance-withdrawal

account balance(4000)
= balance

balance(5000) =
account balance

Ανάληψη 2000 €

balance(3000) =
balance-withdrawal

account balance(3000)
= balance

Υπόλοιπο Λογαριασμού : 4000 € ☺☺☺☺☺☺

☹ ~ @#\$%^ @!#

Αμοιβαίος Αποκλεισμός (locks)



25

Χρήστος

Lock (λογαριασμό καταθέσεων)

balance(5000) = account balance

Ανάληψη 1000 €

balance(4000) =

balance - withdrawal

Account balance(4000) =

balance

Unlock (λογαριασμό καταθέσεων)

Τελικό υπόλοιπο λογαριασμού: 2000.00 €

Ελένη

Lock (λογαριασμό καταθέσεων)

\$#@%%!#@\$^#\$@#\$^#\$!#\$^#\$

Αντεεεεεεεεε

^&@\$%*#(^*!#\$*^(\$#%*%#@!

Πλάκα κάνεις!!!!

#%@##%&^#%*(^(*@!%^*\$#&

Μπουνιά στο ATM

Εεεεε καιρός ήταν...

balance(4000) = account balance

Withdraw 2000 €

balance(2000) =

balance - withdrawal

Account balance(2000) =

balance

Unlock (λογαριασμό καταθέσεων)

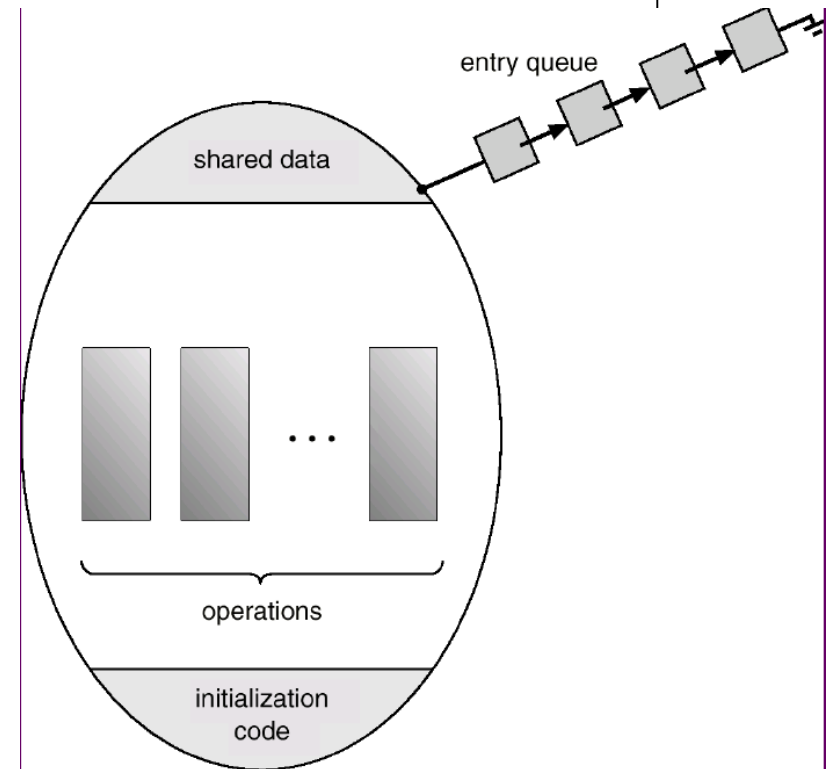
Τράπεζα: ☺☺☺☺☺☺☺☺☺



Ελεγκτές (Monitors)

- Ρίξε το μπαλάκι
 - Αυτή τη φορά στο μεταγλωττιστή
- Τι είναι;
 - **Κατασκευή** σε υψηλού επιπέδου γλώσσα
 - Κάτι σαν τις κλάσεις στις αντικειμενοστρεφείς γλώσσες
- Τι περιέχει;
 - **Δεδομένα, συναρτήσεις** που μπορούν να τα μεταβάλλουν
 - Όλα προστατευμένα από ένα (**υπονοούμενο**) **lock**.

```
monitor Queue {  
  int head, tail; // Διαμοιραζόμενα  
                  // δεδομένα  
  void enq(val) {} // Βάλε το val στην  
                  // ουρά  
  int deq() {} // Βγάλε κάτι από  
               // την ουρά  
}
```



Φρουροί (Guardians) / Φράγματα



27

- **Φρουρός** σε κάθε διαμοιραζόμενο πόρο.
 - Αίτηση για πρόσβαση στον πόρο στον φρουρό.
 - Πρόσβαση στον πόρο μόνο όταν ο φρουρός το επιτρέψει.
- **Φράγματα**:
 - Συγχρονισμός όλων των διεργασιών σε ένα σημείο.
 - Καμία διεργασία δεν προχωρά αν δε φτάσουν στο σημείο όλες οι υπόλοιπες.
 - Π.χ.: Να μαζευτούμε στο σπίτι του Νικόλα για να πάμε για μπάνιο...

ANSI/IEEE 1016



28

ΕΓΓΡΑΦΟ ΠΕΡΙΓΡΑΦΗΣ ΣΧΕΔΙΟΥ ΛΟΓΙΣΜΙΚΟΥ

1. Εισαγωγή

- 1.1 Σκοπός
- 1.2 Γενική άποψη
- 1.3 Ορισμοί, ακρωνύμια και συντομογραφίες
- 1.4 Αναφορές
- 1.5 Επισκόπηση

2. Περιγραφή Αποσύνθεσης

3. Σχέδιο Δεδομένων

- 3.1 Αρχεία και Βάσεις Δεδομένων
- 3.2 Δομές Δεδομένων

4. Σχέδιο Μονάδων

- 4.1 Μονάδα 1

...

5. Περιγραφή Διεπαφών

- 5.1 Διεπαφή 1

.....

6. Παραρτήματα



Παράδειγμα

ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ

1. Εισαγωγή

1.1 Σκοπός

Ο σκοπός αυτού του εγγράφου είναι ο καθορισμός του τρόπου σχεδίασης του προσομοιωτή ATSiM. Περιλαμβάνει όλες τις οντότητες του σχεδίου που σχετίζονται με το λογισμικό, τα αρχεία, τις δομές δεδομένων και τις διεπαφές του συστήματος.

1.2 Γενική Άποψη

Το έγγραφο περιλαμβάνει τον προσδιορισμό της σχεδίασης του ATSiM όπως αυτός περιγράφεται στο Έγγραφο Προδιαγραφής Απαιτήσεων.



Παράδειγμα

1.3 Ορισμοί, Ακρωνύμια και συντομογραφίες

1.3.1 Ορισμοί (αλφαβητικά)

- Εγγραφή πρόσβασης (Access record) – Πληροφορία για μία πρόσβαση ή απόπειρα πρόσβασης. Η πληροφορία περιλαμβάνει την ημερομηνία, ώρα, user ID και το αποτέλεσμα της πιστοποίησης
- Τύπος λογαριασμού (Account Type) – λογαριασμός όψεως ή ταμιευτηρίου
- Μήνυμα Έγκρισης (Approval message) – Το μήνυμα που εκδίδει η τράπεζα όταν η συναλλαγή εγκριθεί. Περιλαμβάνει πληροφορίες σχετικά με το χρήστη και τη συναλλαγή



Παράδειγμα

1.3 Ορισμοί, Ακρωνύμια και συντομογραφίες

1.3.2 Ακρωνύμια - Συντομογραφίες (αλφαβητικά)

- ABP - Authorized Bank Person
- ATM - Automatic Teller Machine
- ATsiM - ATM simulator
- IBM - International Business Machine
- MS-DOS - Microsoft Disk Operating System
- PIN - Personal Identification Number



Παράδειγμα

1.4 Αναφορές

- [1] Έγγραφο Περιγραφής Απαιτήσεων από το Λογισμικό του προσομοιωτή ATsiM
- [2] IEEE Standard 1016/1987, New York: The Institute of Electrical and Electronics Engineering.
- [3] Software Engineering I, Documentation Standards and Guidelines. Version 2.3, Department of Computer Science, Mississippi State University, January 1997.
- [4] Ian Sommerville, “Software Engineering”, 5th edn, Harlow, Addison-Wesley, 1995.



Παράδειγμα

1.5 Επισκόπηση

- Το έγγραφο αυτό βασίζεται στο πρότυπο IEEE 1016/1987 [2]. Το έγγραφο περιγράφει το σχέδιο λογισμικού του προσομοιωτή ATsiM
- Η ενότητα “Περιγραφή Αποσύνθεσης” περιγράφει τα συστατικά στοιχεία του σχεδίου συμπεριλαμβανομένων της δομής του συστήματος, των υποσυστημάτων, των βάσεων δεδομένων και της συσχέτισής τους με τις απαιτήσεις λογισμικού. Το διάγραμμα δομής του συστήματος παρουσιάζεται στο παράρτημα Δ, και το λεξικό δεδομένων στο παράρτημα Α.
- Η ενότητα “Σχέδιο Δεδομένων” παρέχει μία περιγραφή όλων των **εξωτερικών** δεδομένων που χρησιμοποιούνται από τις μονάδες λογισμικού.



Παράδειγμα

1.5 Επισκόπηση

- Η ενότητα “Σχέδιο Μονάδων” περιγράφει τις μονάδες λογισμικού που χρησιμοποιούνται στην ανάπτυξη του συστήματος. Συμπεριλαμβάνονται η ταυτότητα της κάθε μονάδας, η διεπαφή της και η εσωτερική της περιγραφή. Λεπτομερής περιγραφή των μονάδων παρέχεται στο Παράρτημα Β.
- Οι διεπαφές μεταξύ του ATSiM και εξωτερικών συστημάτων καθώς και η διεπαφή χρήστη περιγράφονται στην ενότητα “Περιγραφή Διεπαφών”. Στο παράρτημα Γ περιγράφεται λεπτομερώς η διεπαφή του ATSiM με το μηχανογραφικό σύστημα της τραπέζης και οι οθόνες χρήστη.



Παράδειγμα

2. Περιγραφή Αποσύνθεσης

- Στην ενότητα αυτή περιγράφεται η αποσύνθεση του ATsiM σε υποσυστήματα και μονάδες. Μία περιγραφή υψηλού επιπέδου παρουσιάζεται στο διάγραμμα του παραρτήματος Δ, ενώ αναλυτικότερη περιγραφή των μονάδων υπάρχει στην ενότητα 4. Ο συσχετισμός μεταξύ απαιτήσεων και μονάδων λογισμικού παρουσιάζεται στον Πίνακα Ιχνηλάτησης του Παραρτήματος Ε.
- Κατά τη σχεδίαση του ATsiM η μεγιστοποίηση της συνεκτικότητας και η ελαχιστοποίηση της σύζευξης αποτέλεσε τον κύριο στόχο. Η δυνατότητα επαναχρησιμοποίησης μονάδων του συστήματος εξετάστηκε επίσης στα πλαίσια της σχεδίασης.



Παράδειγμα

2. Περιγραφή Αποσύνθεσης (...συνέχεια)

Οι κύριες μονάδες του λογισμικού είναι:

- Βασική: Η μονάδα εκκίνησης (Παράγραφος 4.1 στη σελίδα)
- Αναγνώριση_Χρήστη: Αναγνωρίζει το χρήστη του ATsiM (Παράγραφος 4.2 στη σελίδα)
- Εκτέλεση_Λειτουργίας: Συντονίζει τις υπηρεσίες προς ένα χρήστη (Παράγραφος 4.3 στη σελίδα ...)
-
- Παρουσίαση_Αποτελεσμάτων: Παρουσιάζει τα αποτελέσματα της συναλλαγής στο χρήστη (Παράγραφος)



Παράδειγμα

3. Σχέδιο Δεδομένων

3.1 Αρχεία και Βάσεις Δεδομένων

- Τα δεδομένα που χρησιμοποιούνται από τον ATSiM αποθηκεύονται σε ένα αρχείο βάσης δεδομένων με το όνομα LOGFILE.TXT
- Υπάρχουν πέντε διαφορετικοί τύποι εγγραφών στη βάση δεδομένων:
- Εγγραφή Ανάληψης

Περιγραφή: Η εγγραφή περιλαμβάνει πληροφορίες για μία συναλλαγή ανάληψης δεδομένων. Κάθε συναλλαγή αποθηκεύεται σε ξεχωριστή εγγραφή

Δομή:

Όνομα Πεδίου	Τύπος	Μήκος	Περιγραφή
Έτος	Integer	4	Έτος συναλλαγής



3. Σχέδιο Δεδομένων

3.2 Δομές Δεδομένων

3.2.1 ATMState

Περιγραφή: Η δομή περιέχει πληροφορίες για την κατάσταση του προσομοιωτή (όρια αναλήψεων, καταθέσεων, μεταφορών καθώς και το διαθέσιμο ποσό του συστήματος)

Δομή:

Όνομα Πεδίου	Τύπος	Μήκος	Περιγραφή
DepLimit	Float		
WithLimit	Float		



4. Σχέδιο Μονάδων

4.1 Main

4.1.1 Ταυτότητα Μονάδας

Αριθμός Μονάδας: 1

Όνομα Μονάδας: Main

4.1.2 Διεπαφή

Είσοδοι: Καμία

Έξοδοι: Καμία (Θα μπορούσε να είναι π.χ. αναφορά)

4.1.3 Εξαρτήσεις

Κλήσεις: Identify_User, Do_Operation (Μονάδες)

Καλείται από: Τίποτα (Λειτουργικό Σύστημα)

4.1.4 Περιγραφή

Η μονάδα αυτή ξεκινά τον προσομοιωτή ATSiM. Η μονάδα πρώτα καλεί τη μονάδα Identify_User για την αναμονή ενός χρήστη και την αναγνώριση του.....



5. Περιγραφή Διεπαφών

Ξεχωριστή περιγραφή των μονάδων που υλοποιούν τις διεπαφές του συστήματος

5.1 Επαλήθευση Χρήστη (Verify_User)

5.1.1 Ταυτότητα Μονάδας

Αριθμός Μονάδας: B1

Όνομα Μονάδας: Identify_User

5.1.2 Διεπαφή

Είσοδοι: Κωδικός (από Χρήστη)

Έξοδοι: Λογική τιμή επαλήθευσης ή όχι

5.1.3 Εξαρτήσεις

Κλήσεις: Καμμία

Καλείται από: Identify_User (Αναγνώριση Χρήστη)

5.1.4 Περιγραφή

Η μονάδα αυτή συγκρίνει τις πληροφορίες που παρέχει ο χρήστης (ID, PAC κλπ) με τη βάση δεδομένων της τραπεζής. Αν υπάρχει το ίδιο ID



Παράρτημα Α – Λεξικό Δεδομένων

- Acc_Type: Integer – Το είδος του λογαριασμού. Μπορεί να είναι είτε λογαριασμός ταμιευτηρίου (1) είτε
- Amount: Float – Ποσό της συναλλαγής
- ATM_CashAvailable: NumberNotes – Τα μετρητά που είναι διαθέσιμα στο σύστημα. Υπάρχει άνω όριο του συνολικού ποσού καθώς και κατώτατο όριο διαθέσιμων μετρητών.
- ATMState: Structure {DepLimit + WithLimit + TranLimit + ATM_CashLimit + ATM_CashAvailable}. Η κατάσταση του προσομοιωτή ATSiM



Παράρτημα Β – Λεπτομερής Περιγραφή Μονάδων

xx. Do_Operation(IN User, OUT Status)

Όνομα Μονάδος: Do_Operation

Αριθμός Μονάδος: 1.2

Επισκόπηση Μονάδος: Η μονάδα προωθεί το επίπεδο του χρήστη στη μονάδα Get_Operation για τη λήψη του κωδικού Oper_Code και στη συνέχεια το είδος της συναλλαγής στη μονάδα Dispatch_Center για την εκτέλεση της συναλλαγής.

Υπερκείμενες Μονάδες: Main

Υποκείμενες Μονάδες: Get_Operation, Dispatch_Center

Τοπικές Μεταβλητές:

Όνομα	Τύπος	Περιγραφή
IOper_Code	Integer	Είδος συναλλαγής
....



Παράρτημα Β – Λεπτομερής Περιγραφή Μονάδων

Είσοδος: Χρήστης

Έξοδος: Status

Αλγόριθμος:

Function Do_Operation (IN IUser, OUT IStatus)

Start

 Initialize local variables

 loop while IStatus is different from 99 and 33

 Invoke Get_Operation(IUser.User_Rank, Oper_Code, IStatus);

 if IStatus is not 99 (quit) or 33 (exit) then

 Invoke Dispatch_Center(IUser, IOper_Code, IStatus);

 endif

 endloop

End



Παράρτημα Γ – Μορφή Αναφορών και Οθονών

Get_Operation

Κυρίως Μενού Κατόχου Κάρτας

F1 – Βοήθεια, Esc – Έξοδος, Return - Enter

Ανάληψη

Κατάθεση

Μεταφορά

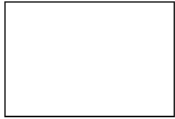
Ερώτηση υπολοίπου

Πληκτρολογήστε την επιλογή που επιθυμείτε

Παράρτημα Δ – Δομή Συστήματος



45



Μονάδα ATSiM



Εξωτερική Διεπαφή



Αναφορά



Είσοδος Χρήστη



Ροή Δεδομένων

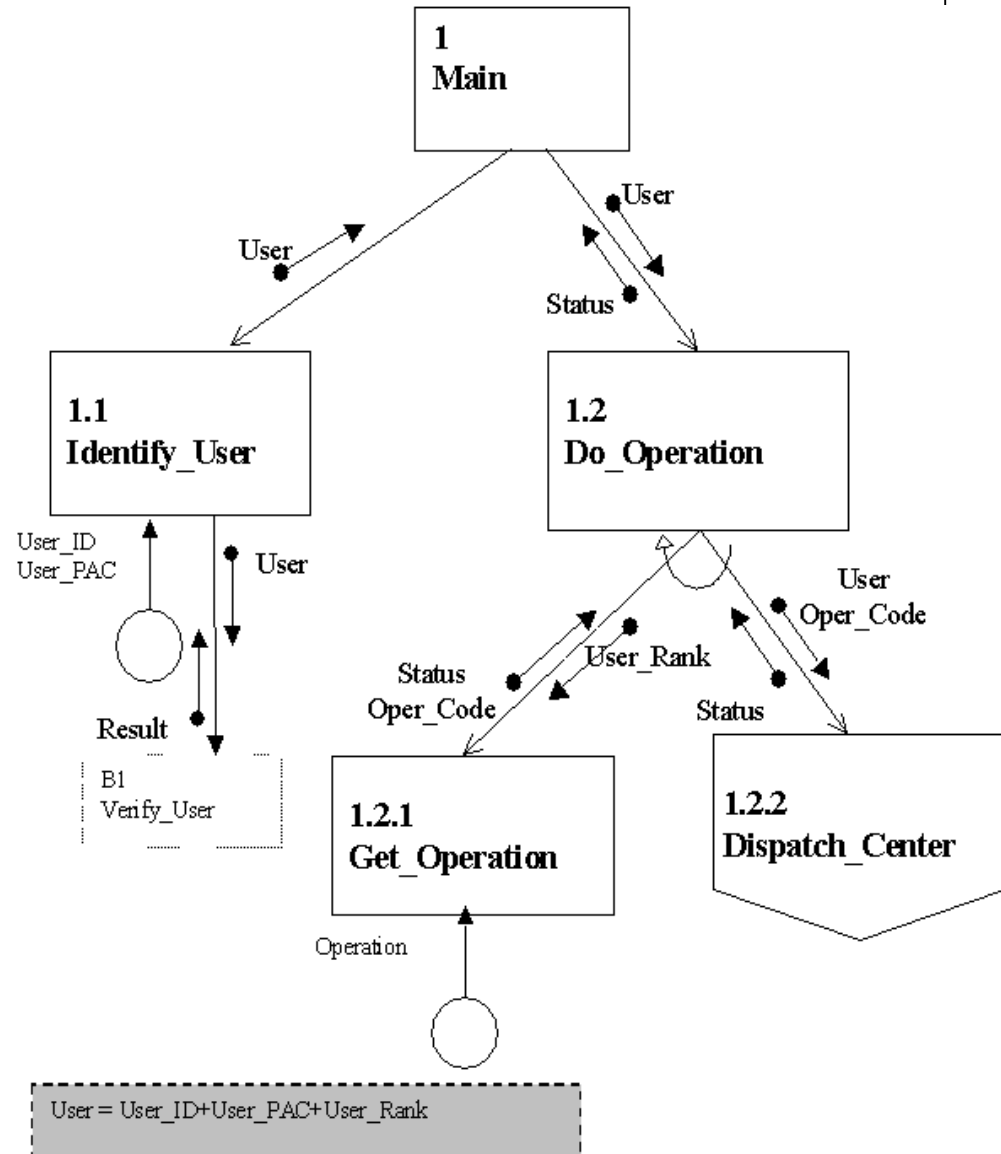


Υπολογισμός

Παράρτημα Δ – Δομή Συστήματος



46



Πίνακας Ιχνηλασιμότητας Απαιτήσεων



47

Λειτουργική Απαίτηση	1	1.1	1.2	1.2.1	1.2.2	1.2.2.1	1.2.2.2	1.2.2.3	1.2.2.4	1.2.2.1.1	1.2.2.1.2	1.2.2.1.3	1.2.2.1.4	1.2.2.4.1	1.2.2.4.2	1.2.2.4.3
3.1 Αναγνώριση Χρήστη	X	X														
3.2 Πιστοποίηση Κατόχου Κάρτας	X	X														
3.3 Πιστοποίηση Κατόχου Κάρτας	X	X														
3.4 Κατάθεση	X		X	X	X	X				X	X		X			
3.5 Ανάλυση	X		X	X	X	X				X	X	X	X			
3.6 Μεταφορά	X		X	X	X	X				X	X		X			
3.7 Ερώτηση υπολοίπου	X		X	X	X	X				X						