

---

**Εννοιολογικό Μοντέλο**

Conceptual Model

# Εννοιολογικό Μοντέλο (1/2)

---

## Αποτελεί:

- βασικό εργαλείο για τη σχεδίαση
- εισαγωγή σε διάφορα μέσα τεκμηρίωσης (artifacts)
- απεικόνιση σημαντικών εννοιολογικών κλάσεων
- το σημαντικότερο εργαλείο τεκμηρίωσης κατά τη διάρκεια της ΑΣ ανάλυσης

**Το εννοιολογικό μοντέλο ή μοντέλο περιοχής προβλήματος (ΠΠ) είναι μια αναπαράσταση εννοιολογικών κλάσεων του πραγματικού κόσμου**

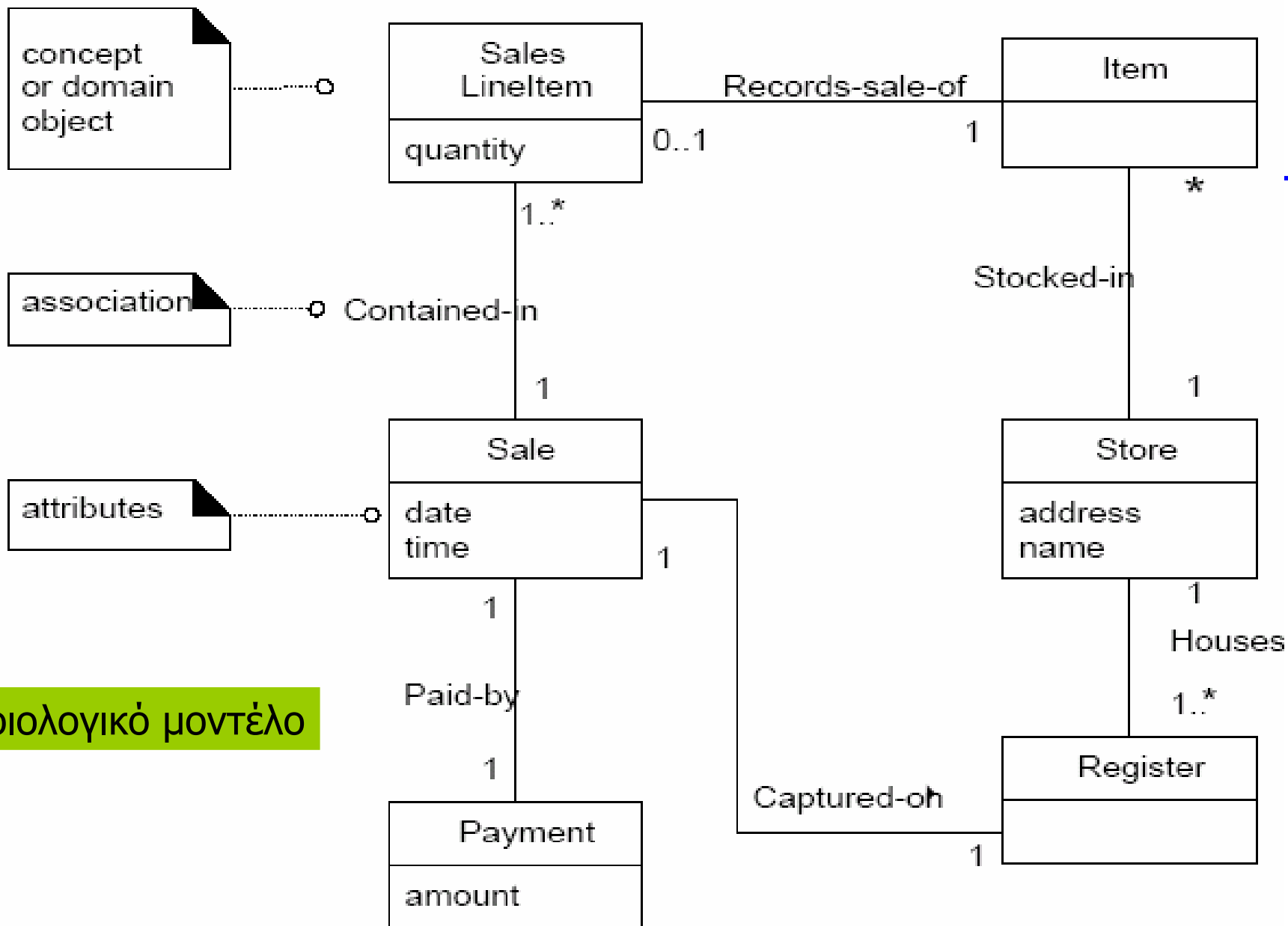
## Δεν είναι:

- τμήμα λογισμικού
- διάγραμμα κλάσεων λογισμικού

## Εννοιολογικό Μοντέλο (2/2)

---

- **Το σημαντικότερο βήμα στην αντικειμενοστραφή ανάλυση:**
  - Η διάσπαση της περιοχής προβλήματος σε ατομικές εννοιολογικές κλάσεις ή αντικείμενα
  - Τα πράγματα που αντιλαμβανόμαστε
- **Πως απεικονίζονται, στην UML;**
  - με διαγράμματα κλάσεων δίχως τις λειτουργίες (μέθοδοι).
  - Περιλαμβάνουν:
    - Αντικείμενα της περιοχής προβλήματος ή εννοιολογικές κλάσεις
    - Συσχετίσεις μεταξύ των εννοιολογικών κλάσεων
    - Χαρακτηριστικά των εννοιολογικών κλάσεων



Εννοιολογικό μοντέλο

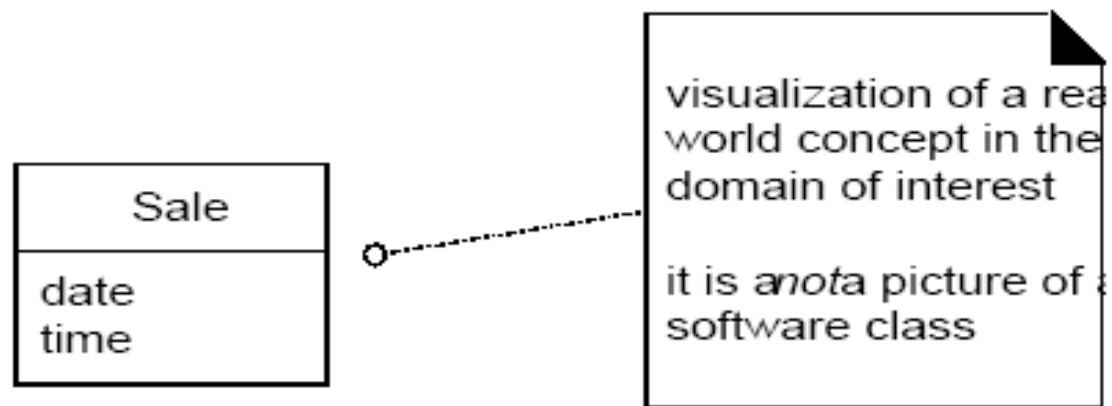


Figure 10.2 A domain model shows real-world conceptual classes, not software classes.

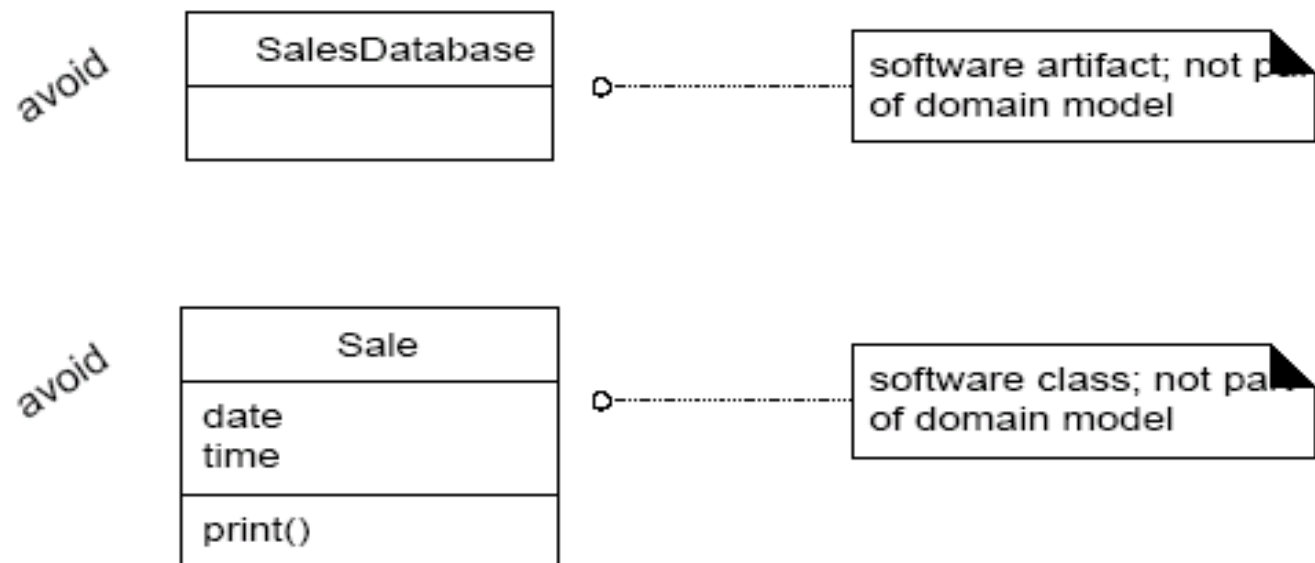


Figure 10.3 A domain model does not show software artifacts or classes.

# Εννοιολογικές κλάσεις

---

- **Τι αναπαριστούν;**
  - μια ιδέα, πράγμα, ή αντικείμενο.
  - θα πρέπει να εξετάζεται σύμφωνα με το σύμβολό της, το σκοπό της, και την έκτασή της
- **Σύμβολο – λέξεις ή εικόνες που εκφράζουν μια εννοιολογική κλάση**
- **Σκοπός – ο ορισμός μιας εννοιολογικής κλάσης**
- **Έκταση – το σύνολο των παραδειγμάτων στο οποίο εφαρμόζεται μια εννοιολογική κλάση**

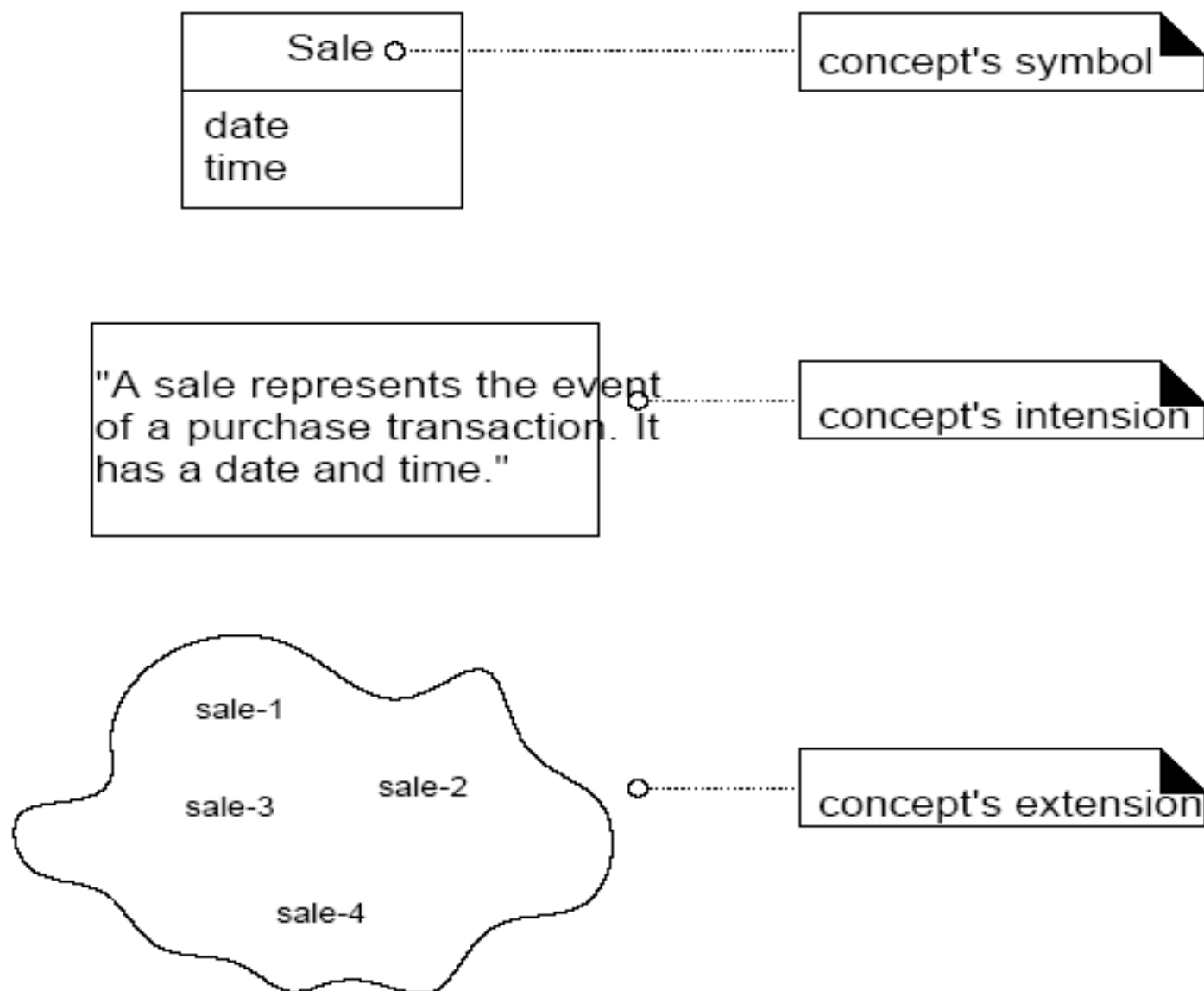


Figure 10.4 A conceptual class has a symbol, intension, and extension.

# Μοντέλα ΠΠ και Διάσπαση

---

- **Στρατηγική: «διαίρει και βασίλευε»**
  - χειρισμός της πολυπλοκότητας με την διάσπαση του προβλήματος σε κατανοητές ενότητες.
- **Η ουσιαστική διάκριση μεταξύ αντικειμενοστρεφούς σχεδίασης και δομημένης ανάλυσης είναι:**
  - διαίρεση με εννοιολογικές κλάσεις (αντικείμενα) *αντί* με λειτουργίες.



# Προσδιορισμός Εννοιολογικών κλάσεων

---

## **Βασικός Κανόνας**

*Είναι προτιμότερος ο υπερ-προσδιορισμός ενός μοντέλου ΠΠ με πολλές λεπτομερείς κλάσεις, από τον υπο-προσδιορισμό.*

# Τεχνικές εύρεσης (προσδιορισμού) εννοιολογικών κλάσεων

---

1. λίστα κατηγοριοποίησης εννοιολογικών κλάσεων
1. προσδιορισμός φράσεων με ουσιαστικά

## Λίστα κατηγοριοποίησης εννοιολογικών κλάσεων

---

Conceptual Class Category	Examples
physical or tangible objects	<i>Register</i> <i>Airplane</i>
specifications, designs, or descriptions of things	<i>ProductSpecification</i> <i>FlightDescription</i>
places	<i>Store</i> <i>Airport</i>
transactions	<i>Sale, Payment</i> <i>Reservation</i>
transaction line items	<i>SalesLineItem</i>
roles of people	<i>Cashier</i> <i>Pilot</i>
containers of other things	<i>Store, Bin</i> <i>Airplane</i>
things in a container	<i>Item</i> <i>Passenger</i>

organizations	<i>SalesDepartment</i> <i>ObjectAirline</i>
events	<i>Sale, Payment, Meeting</i> <i>Flight, Crash, Landing</i>
processes (often <i>not</i> represented as a concept, but may be)	<i>SellingAProduct</i> <i>BookingASeat</i>
rules and policies	<i>RefundPolicy</i> <i>CancellationPolicy</i>
records of finance, work, contracts, legal matters	<i>Receipt, Ledger, EmploymentContract</i> <i>MaintenanceLog</i>
financial instruments and services	<i>LineOfCredit</i> <i>Stock</i>
manuals, documents, reference papers, books	<i>DailyPriceChangeList</i> <i>RepairManual</i>

Table 10.1 Conceptual Class Category List.

# Εύρεση Εννοιολογικών κλάσεων από φράσεις με ουσιαστικά

---

- κατάλληλος χώρος εφαρμογής οι λεπτομερείς περιπτώσεις χρήσης (Προδιαγραφές περιπτώσεων χρήσης)
- **Μειονέκτημα:**
  - η έλλειψη ακρίβειας στη φυσική γλώσσα.
    - Διαφορετικές φράσεις ουσιαστικών μπορούν να παριστάνουν την ίδια εννοιολογική κλάση ή χαρακτηριστικό.
    - Συνιστάται ο συνδυασμός της με την τεχνική *Λίστας κατηγοριοποίησης Εννοιολογικών κλάσεων*.

### **Main Success Scenario (or Basic Flow):**

1. Customer arrives at POS checkout with goods and/or services to purchase.
2. Cashier starts a new sale.
3. Cashier enters item identifier.
4. System records sale line item and presents item description, price, and running total.  
Price calculated from a set of price rules.  
*Cashier repeats steps 3-4 until indicates done.*
5. System presents total with taxes calculated.
6. Cashier tells Customer the total, and asks for payment.
7. Customer pays and System handles payment.
8. System logs completed sale and sends sale and payment information to the external Accounting system (for accounting and commissions) and Inventory system (to update inventory).
9. System presents receipt.
10. Customer leaves with receipt and goods (if any).

### **Main Success Scenario (or Basic Flow):**

1. **Customer** arrives at a **POS checkout** with **goods** and/or **services** to purchase.
2. **Cashier** starts a new **sale**.
3. **Cashier** enters **item identifier**.
4. System records **sale line item** and presents **item description, price, and running total**. Price calculated from a set of price rules.  
Cashier repeats steps 2-3 until indicates done.
5. System presents total with **taxes** calculated.
6. Cashier tells Customer the total, and asks for **payment**.
7. Customer pays and System handles payment.
8. System logs the completed **sale** and sends sale and payment information to the external **Accounting** (for accounting and **commissions**) and **Inventory** systems (to update inventory).
9. System presents **receipt**.
10. Customer leaves with receipt and goods (if any).

### **Extensions (or Alternative Flows):**

...

#### **7a. Paying by cash:**

1. Cashier enters the cash **amount tendered**.

# Κανόνες Μοντελοποίησης ΠΠ

---

Συνιστώμενα βήματα:

1. Εφαρμογή της τεχνικής *Λίστα Κατηγ. Εννοιολ. Κλάσεων* στις υπό μελέτη απαιτήσεις
2. Σχεδιάσή τους σε ένα μοντέλο περιοχής προβλήματος
3. Προσθήκη των αναγκαίων επικοινωνιών, για την καταγραφή των συσχετίσεων, για τις οποίες είναι αναγκαίο να διατηρηθεί κάποια μνήμη.
4. Προσθήκη χαρακτηριστικών, αναγκαίων για την πλήρωση των πληροφοριακών απαιτήσεων



# Κοινό λάθος στον προσδιορισμό Εννοιολογικών κλάσεων

---

## Χαρακτηριστικό ή Έννοια;

- **Εμπειρικός κανόνας:**
  - *Εάν ΔΕΝ θεωρούμε μια εννοιολογική κλάση  $X$  ως αριθμό ή κείμενο στον πραγματικό κόσμο, τότε η  $X$  είναι πιθανότατα ενν. κλάση και όχι χαρακτηριστικό.*
  - **Παραδείγματα:**
    - Το *Store* είναι χαρακτηριστικό της *Sale* ή μια ξεχωριστή κλάση;
    - Η *Destination* είναι χαρακτηριστικό της *Flight* ή χαρακτηριστικό της *Airport*;
- Σε αμφιβολία δημιουργήσε μια ξεχωριστή κλάση. Τα χαρακτηριστικά σπανίζουν σε μια ΠΠ.

---

Flight
destination

or... ?

Flight

Airport
name

Sale
store

or... ?

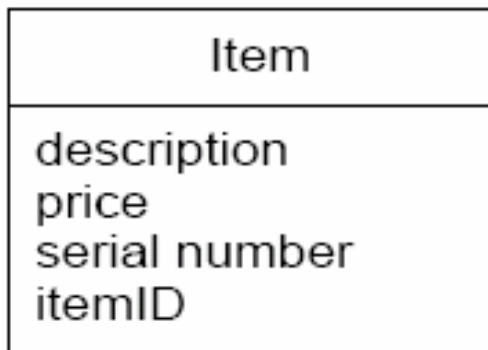
Sale

Store
phoneNumber

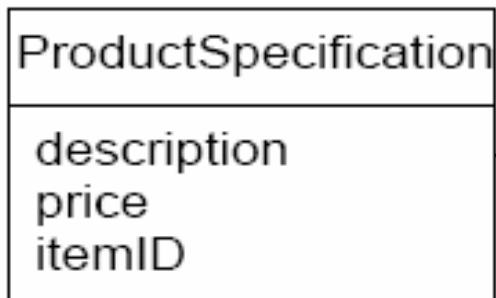
*Ένα μοντέλο ΠΠ δεν είναι απόλυτα σωστό ή λάθος, αλλά περισσότερο ή λιγότερο χρήσιμο. Είναι ένα εργαλείο επικοινωνίας.*

## Εννοιολ. κλάσεις Προσδιορισμού

---



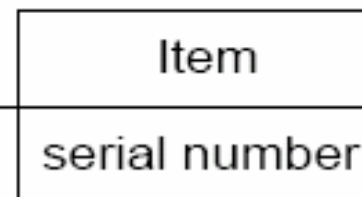
**Worse**



Describes

1

\*



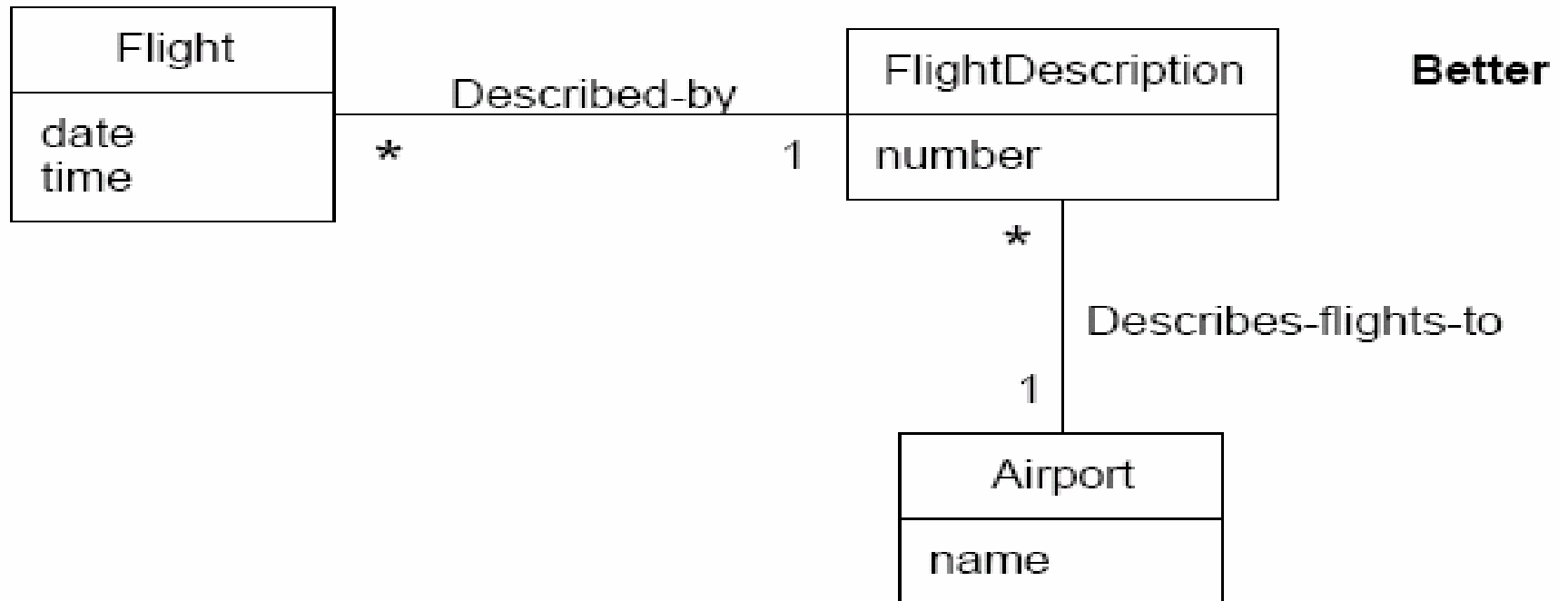
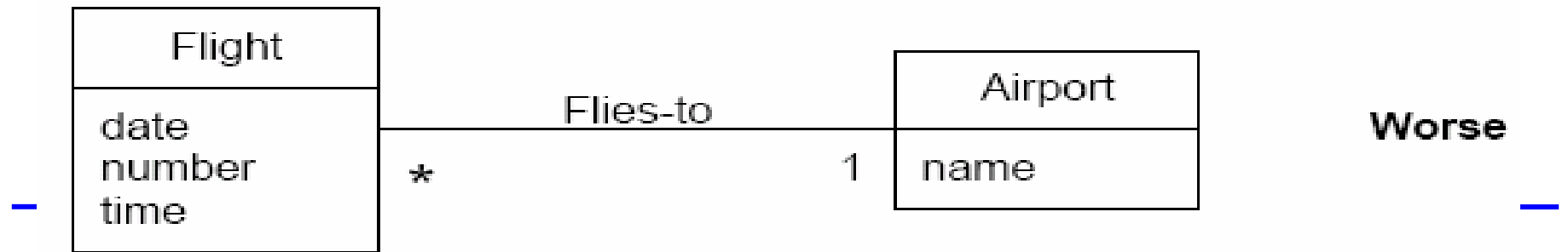
**Better**

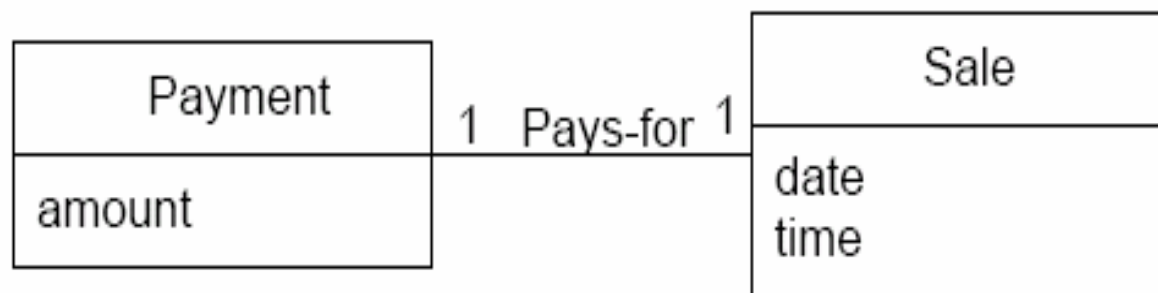
# Πότε απαιτούνται Ενν. Κλάσεις Προσδιορισμού;

---

*Κανόνας:*

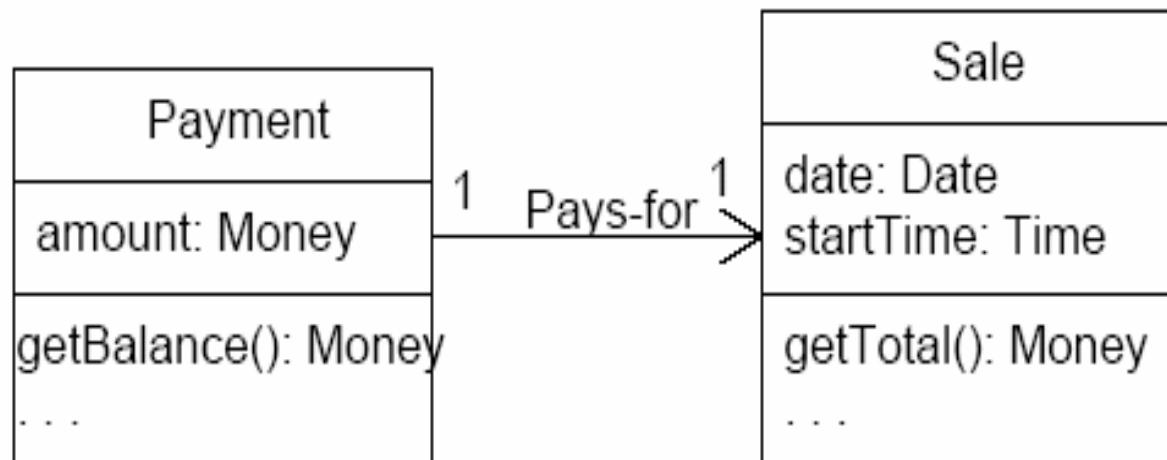
- Όταν υπάρχει ανάγκη προσδιορισμού ή περιγραφής μιας μονάδας ή υπηρεσίας, ανεξάρτητα από την ύπαρξη τέτοιων αντικειμένων στον πραγματικό κόσμο.
- Διαγράφοντας στιγμιότυπα από πράγματα που περιγράφουν (πχ. ένα *Τεμάχιο*), έχει ως αποτέλεσμα απώλεια αναγκαίας για τη διατήρηση πληροφορίας.
- Περιορίζει την επαναλαμβανόμενη πληροφορία





## UP Domain Model

Raw UML class diagram notation used in an essential model visualizing real-world concepts.



## UP Design Model

Raw UML class diagram notation used in a specification model visualizing software components.

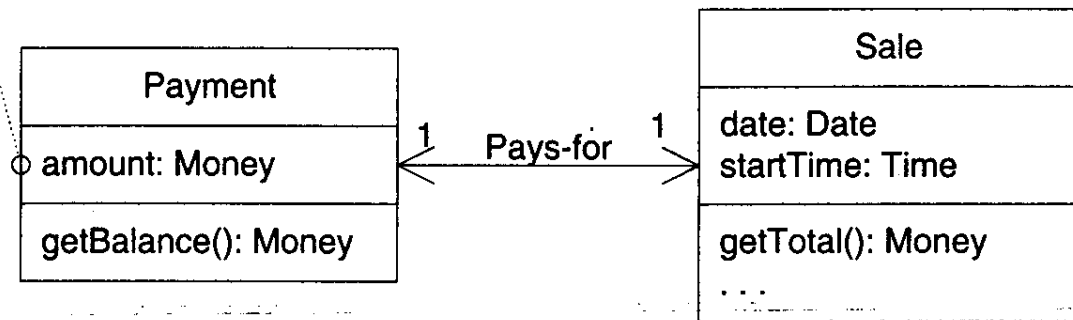
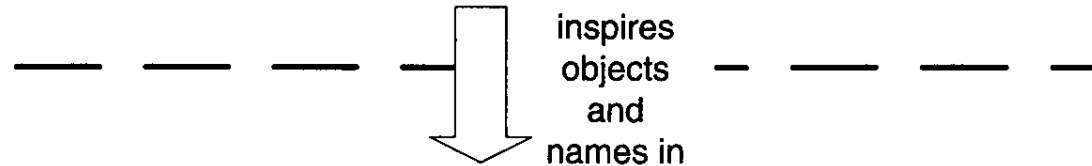
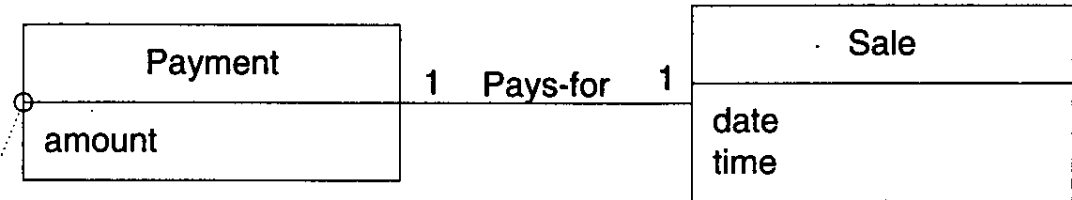
# Μειώνοντας το Χάσμα Αναπαράστασης

---

- Η επιλογή ονομάτων του λεξιλογίου της περιοχής προβλήματος (πχ. Sale) αυξάνει την κατανόηση και παρέχει αντιστοίχιση στην υλοποίηση κλάσης
- Το μοντέλο περιοχής προβλήματος, ως ένα οπτικό λεξιλόγιο εννοιών της περιοχής προβλήματος, μας εμπνέει για την ονομασία των πραγμάτων στη σχεδίαση λογισμικού.
- Αυτό σχετίζεται με το ζήτημα του χάσματος αναπαράστασης ή του χάσματος σημασιολογίας
  - το χάσμα μεταξύ του νοητικού μας μοντέλου και της αναπαράστασής του στο λογισμικό.

## UP Domain Model

Stakeholder's view of the noteworthy concepts in the domain.



## UP Design Model

The object-oriented developer has taken inspiration from the real world domain in creating software classes.

Therefore, the representational gap between how stakeholders conceive the domain, and its representation in software, has been lowered.

A Payment in the Domain Model is a concept, but a Payment in the Design Model is a software class. They are not the same thing, but the former *inspired* the naming and definition of the latter.

This reduces the representational gap.

This is one of the big ideas in object technology.

Figure 10.10 In object design and programming it is common to create software classes whose names and information is inspired from the real world domain.

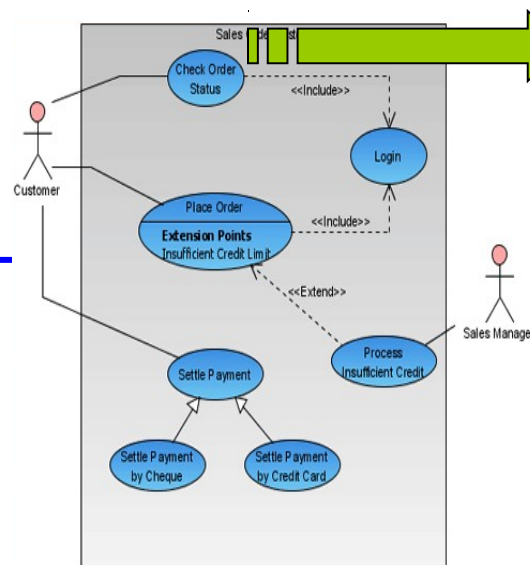
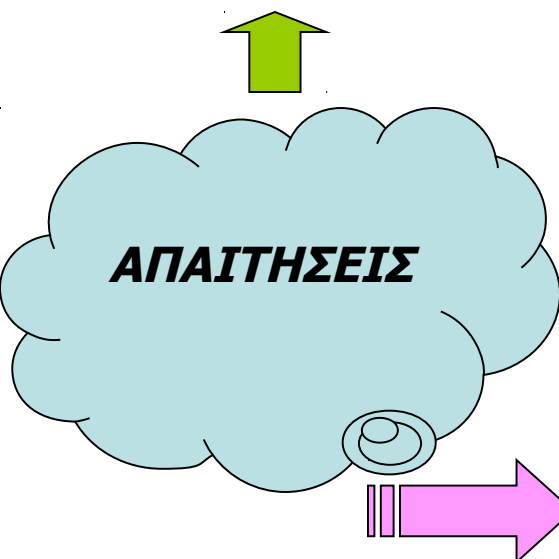


---

## Μείωση του Χάσματος αναπαράστασης

- Επιτυγχάνεται με την κατά το δυνατόν καλύτερη προσέγγιση μεταξύ του λεξιλογίου της περιοχής προβλήματος και του λεξιλογίου του λογισμικού
- Συμβάλει στην ταχύτερη κατανόηση του κώδικα και συνιστά «φυσικό» τρόπο επέκτασής του, με τρόπο που απλά αντιστοιχεί στην περιοχή προβλήματος.

Χρήστης	Στόχος
Ταμίας	διενέργεια πώλησης διενέργεια ενοικίασης
	εισαγωγή ταμείου εξαγωγή ταμείου



## Handle Returns

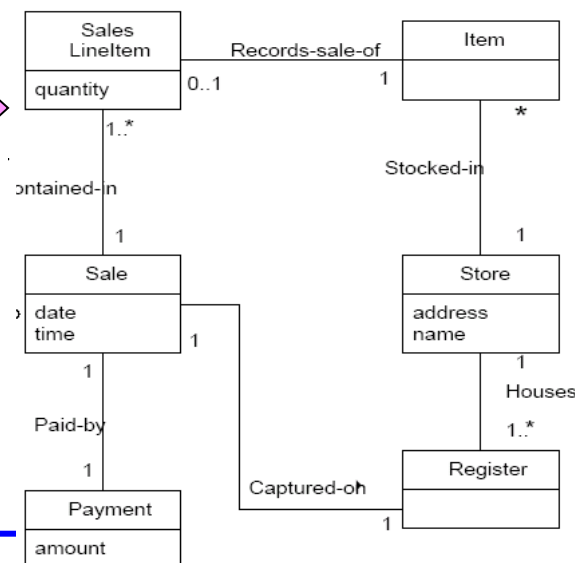
*Main Success Scenario:* A customer arrives at a checkout with items to return. The cashier uses the POS system to record each returned item ...

### Alternate Scenarios:

If the credit authorization is reject, inform the customer and ask for an alternate payment method.

If the item identifier is not found in the system, notify the Cashier and suggest manual entry of the identifier code (perhaps it is corrupted).

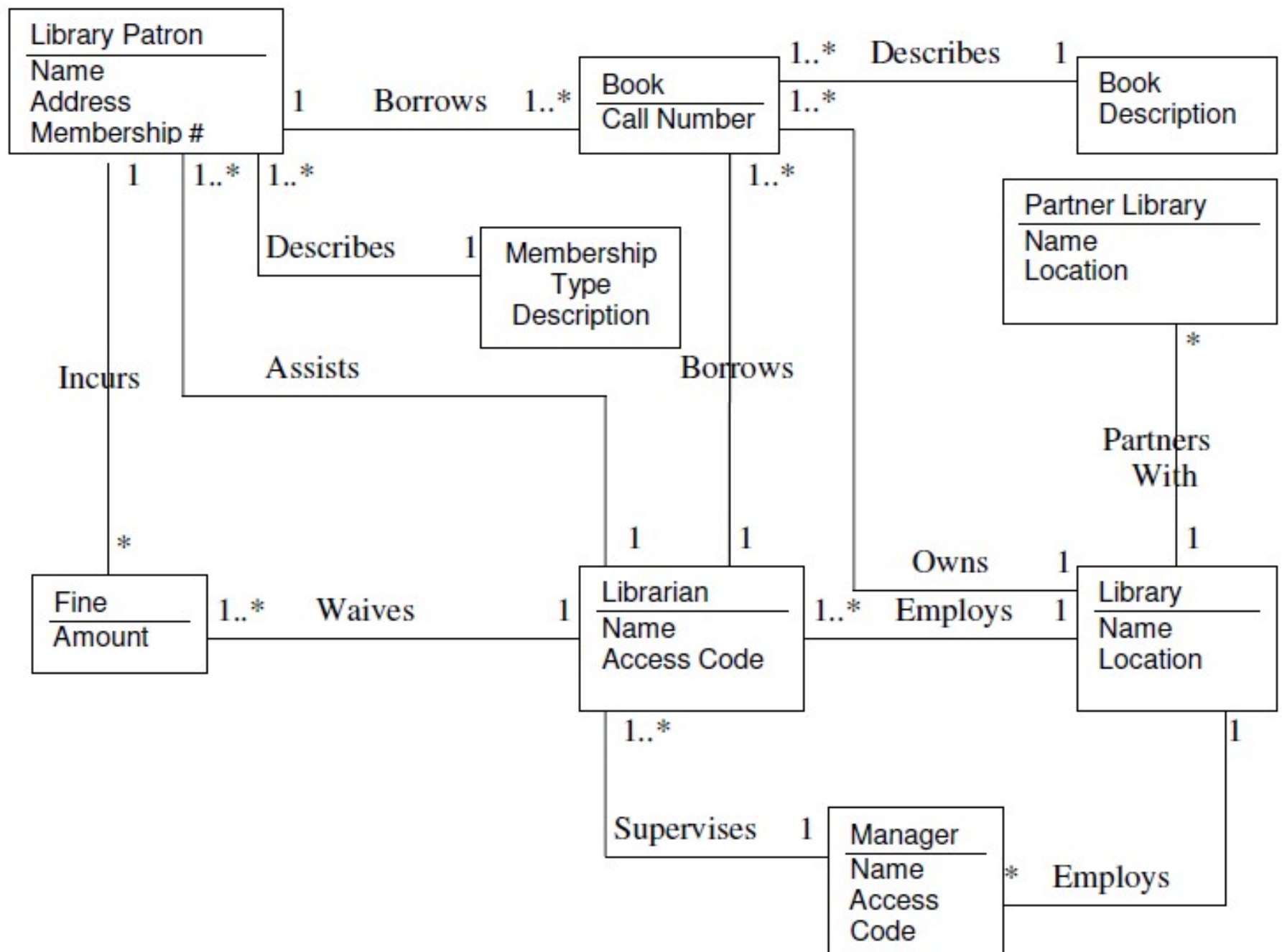
If the system detects failure to communicate with the external tax calculator system, ...



---

**Σε μια βιβλιοθήκη κάθε δανειζόμενος μπορεί να δανειστεί κάποια βιβλία. Η βιβλιοθήκη μπορεί να έχει κάποια παραρτήματα καθώς και ένα σύνολο υπαλλήλων και διευθυντών.**

**Τα μέλη της βιβλιοθήκης χρεώνονται για τις ημέρες που έχουν στη κατοχή τους ένα βιβλίο.**



---

ΔΙΑΓΡΑΜΜΑΤΑ ΔΡΑΣΤΗΡΙΟΤΗΤΩΝ

**Activity Diagrams**

# Δυναμικά Διαγράμματα

---

## Γιατί χρειάζονται;

- Για να δείξουν στοιχεία που δεν δείχνουν τα διαγράμματα κλάσεων
- Για την κατανόηση της συμπεριφοράς ενός αντικειμένου σε σχέση με το χρόνο
- Για την κατανόηση συμπεριφοράς που εξαρτάται από την κατάσταση του αντικειμένου
- Για την αποκάλυψη κρυμμένων ιδιοτήτων
- Για τον προσδιορισμό ιδιωτικών λειτουργιών ή λειτουργιών που λείπουν
- Για τον προσδιορισμό της αλληλουχίας των λειτουργιών και των μηνμάτων

## Είδη δυναμικών διαγραμμάτων

---

- *Διαγράμματα καταστάσεων (State Diagrams)*  
Περιγράφουν τις διάφορες καταστάσεις που μπορεί να κατέχει κάποιο αντικείμενο και τα γεγονότα που οδήγησαν στην αλλαγή
- *Διαγράμματα ακολουθίας (Sequence Diagrams)*  
Περιγράφουν πώς τα αντικείμενα αλληλεπιδρούν μεταξύ τους, εστιάζοντας περισσότερο σε θέματα χρόνου.
- *Διαγράμματα συνεργασίας (Collaboration Diagrams)* Περιγράφουν πως τα αντικείμενα επικοινωνούν μεταξύ τους, εστιάζοντας σε θέματα χώρου.
- *Διαγράμματα δραστηριοτήτων (Activity Diagrams)*  
Δίνουν έμφαση σε θέματα λειτουργιών. Τα αντικείμενα εκτελούν κάποιο έργο μέσω κάποιων δραστηριοτήτων

# Διάγραμμα Δραστηριοτήτων

---

- Ένας γράφος δραστηριοτήτων είναι μια ειδική μορφή μηχανής καταστάσεων για την μοντελοποίηση:
  - υπολογισμών και
  - ροής εργασιών
- Κάθε δραστηριότητα αναπαριστά την εκτέλεση μιας λειτουργίας (operation)
- Η ενεργοποίηση δραστηριοτήτων γίνεται από εσωτερικά γεγονότα ή με την ολοκλήρωση μιας δραστηριότητας



## Που χρησιμοποιούνται

---

- Η καταγραφή των *ενεργειών που εφαρμόζονται κατά την εκτέλεση μιας λειτουργίας* (στιγμιότυπο της υλοποίησης της λειτουργίας)
- Περιγραφή της *εσωτερικής λειτουργίας* ενός αντικειμένου
- Περιγραφή *συσχετιζόμενων ενεργειών* που εφαρμόζονται, και πως αυτές επηρεάζουν τα γύρω αντικείμενά τους.
- Υπόδειξη του τρόπου που ένα στιγμιότυπο κάποιου *use-case* μπορεί να εφαρμοστεί στο πλαίσιο των αλλαγών της κατάστασης του αντικειμένου.
- Αναφορά στο πώς μια *επιχείρηση λειτουργεί* (εργαζόμενοι, ροή εργασιών, οργάνωση και αντικείμενα)

# Τι δεν δείχνουν

---

- Τη συνεργασία αντικειμένων
- Την συμπεριφορά ενός αντικειμένου στη διάρκεια ζωής του

# Διαγράμματα Δραστηριότητας (Activity Diagrams)

---

- Ένα **διάγραμμα δραστηριότητας** μοντελοποιεί τη ροή της εργασίας, αναπαριστώντας τις διάφορες καταστάσεις εκτέλεσης ενός υπολογισμού (Bohm & Jacopini).
- Παρουσιάζεται η **ροή του ελέγχου** μεταξύ δραστηριοτήτων του ιδίου αντικειμένου ή πολλών αντικειμένων
- Μία **κατάσταση δραστηριότητας** συμβολίζεται ως ένα ορθογώνιο με καμπύλες γωνίες με περιγραφή της δραστηριότητας.
- Η **μετάβαση** κατά τη συμπλήρωση μιας δραστηριότητας συμβολίζεται ως ακμή.
- Οι διακλαδώσεις συμβολίζονται είτε με **συνθήκες φρουρούς** επί των μεταβάσεων είτε με **κόμβους απόφασης** (ρόμβους) με πολλαπλές εξερχόμενες ακμές.
- Μία **ένωση** (join) συμβολίζει συνένωση πολλών εισερχομένων μεταβάσεων σε μία εξερχόμενη ενώ μία **διχάλα** (fork) την ανάλυση μιας εισερχόμενης σε πολλές παράλληλες ταυτόχρονες.

## Συμβολισμοί (1/2)

---

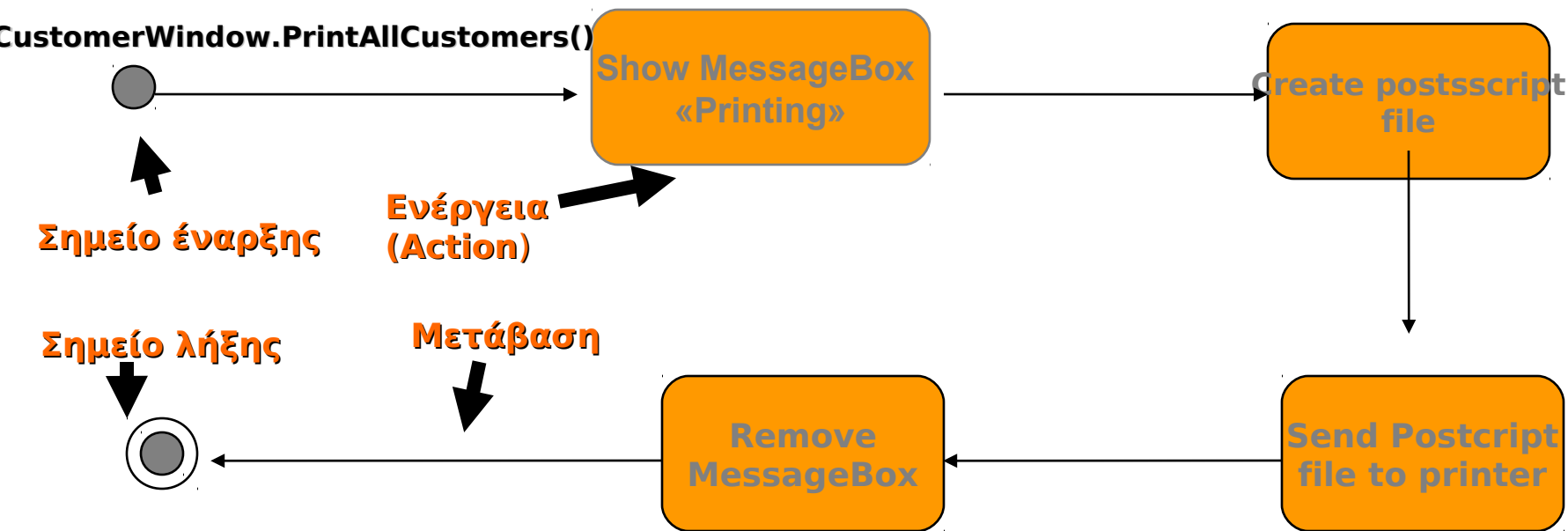
- **Αρχή (●) – Τέλος (◻)**
- **Δραστηριότητα (ορθογώνιο με καμπύλες γωνίες)**
- **Μετάβαση (ακμή)**
- **Διακλάδωση**
  - Με συνθήκες φρουρούς ([.....])
  - Με κόμβους απόφασης (ρόμβος) με πολλαπλές εξερχόμενες ακμές

## Συμβολισμοί (2/2)

---

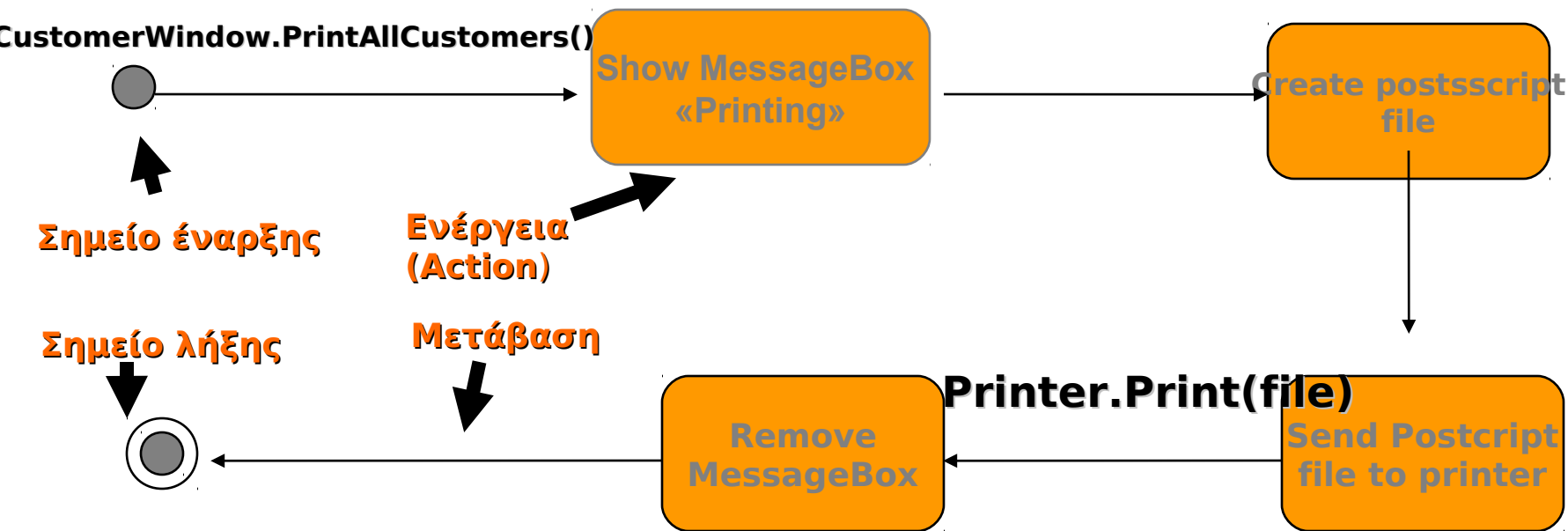
- **Ράβδοι συγχρονισμού (synchronization bar)**
  - Ένωση (join)
    - Συνένωση πολλών εισερχόμενων μεταβάσεων
  - Διχάλα (fork)
    - Ανάλυση μιας εισερχόμενης μετάβασης σε πολλές παράλληλες εξερχόμενες μεταβάσεις
- **Διάδρομος (swimlane)**
  - Οργάνωση δραστηριοτήτων με βάση τις αρμοδιότητες
  - Κατακόρυφη γραμμή
  - Αναπαριστά την ροή αντικειμένου

# Ενέργειες και μεταβάσεις (Actions and transitions)



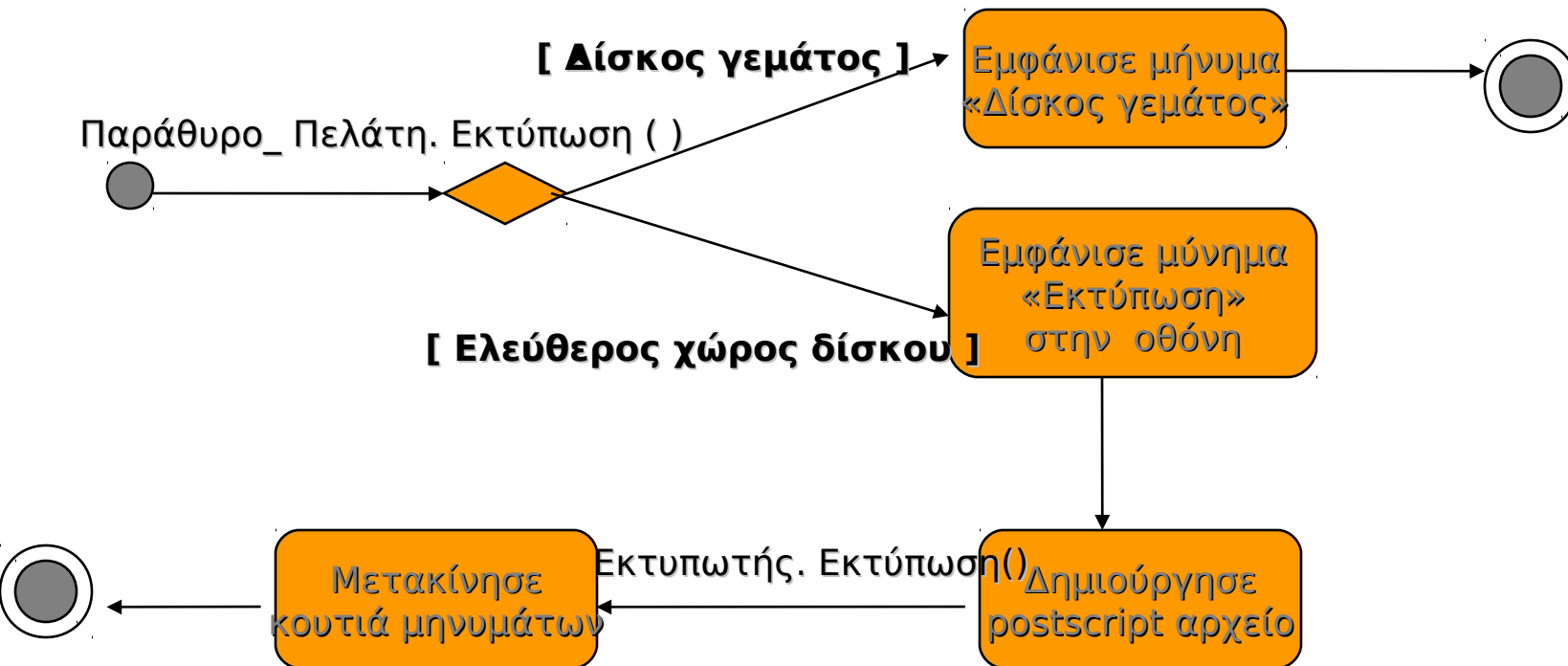
Όταν κάποιος καλεί την συνάρτηση *εκτύπωση()*, τότε αρχίζουν οι διάφορες ενέργειες. Η πρώτη ενέργεια είναι η εμφάνιση μηνύματος στην οθόνη, ενώ η δεύτερη η δημιουργία κάποιου αρχείου. Τρίτη ενέργεια η αποστολή αρχείου στον εκτυπωτή και τέταρτη η εξαφάνιση των κουτιών μηνυμάτων.

# Ενέργειες και μεταβάσεις (Actions and transitions)



Οι μεταβάσεις ανάμεσα στις καταστάσεις μπορούν να παρασταθούν με βέλη τα οποία πιθανώς να συνοδεύονται από κάποιες συνθήκες φύλαξης (guard-condition) , όροι αποστολής (send-clause) κτλ.

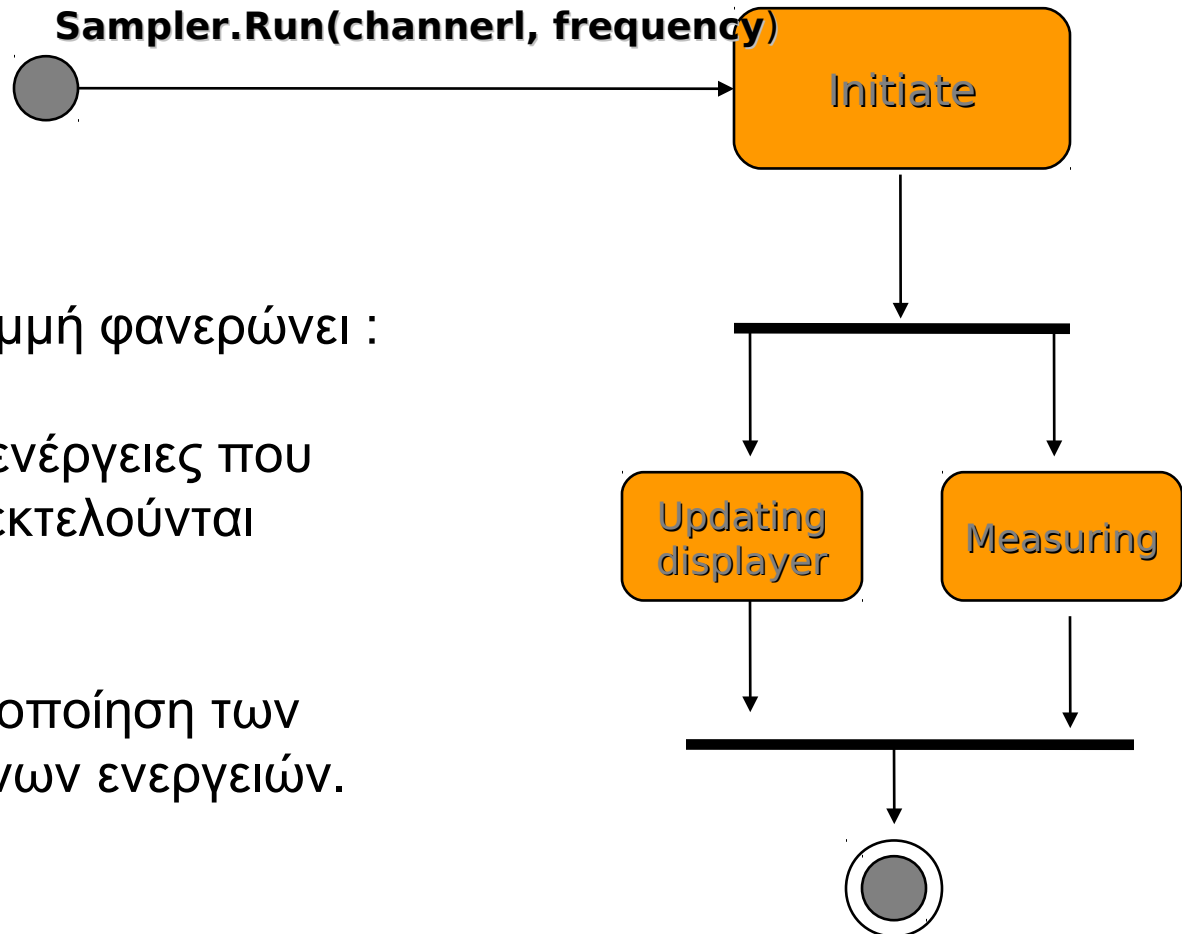
# Μετάβαση καταστάσεων μέσω συνθηκών



Ο ρόμβος φανερώνει ότι ακολουθεί κάποια διακλάδωση. Ανάλογα λοιπόν αν ισχύει ή όχι κάποια συνθήκη έπεται η αντίστοιχη ενέργεια



# Fork – Join



Η έντονη γραμμή φανερώνει :

α) είτε ότι οι ενέργειες που ακολουθούν εκτελούνται παράλληλα.

β) είτε την ενοποίηση των διακλαδιζόμενων ενεργειών.

# Swimlane

Το swimlane δείχνει που εφαρμόζονται οι διάφορες ενέργειες, δηλ. σε ποιο αντικείμενο ή σε ποιο κομμάτι της οργάνωσης

## Οθόνη

Updating  
displayer

## Δειγματολήπτης



Sampler.run(channel, freq.)

Initiate

Measuring

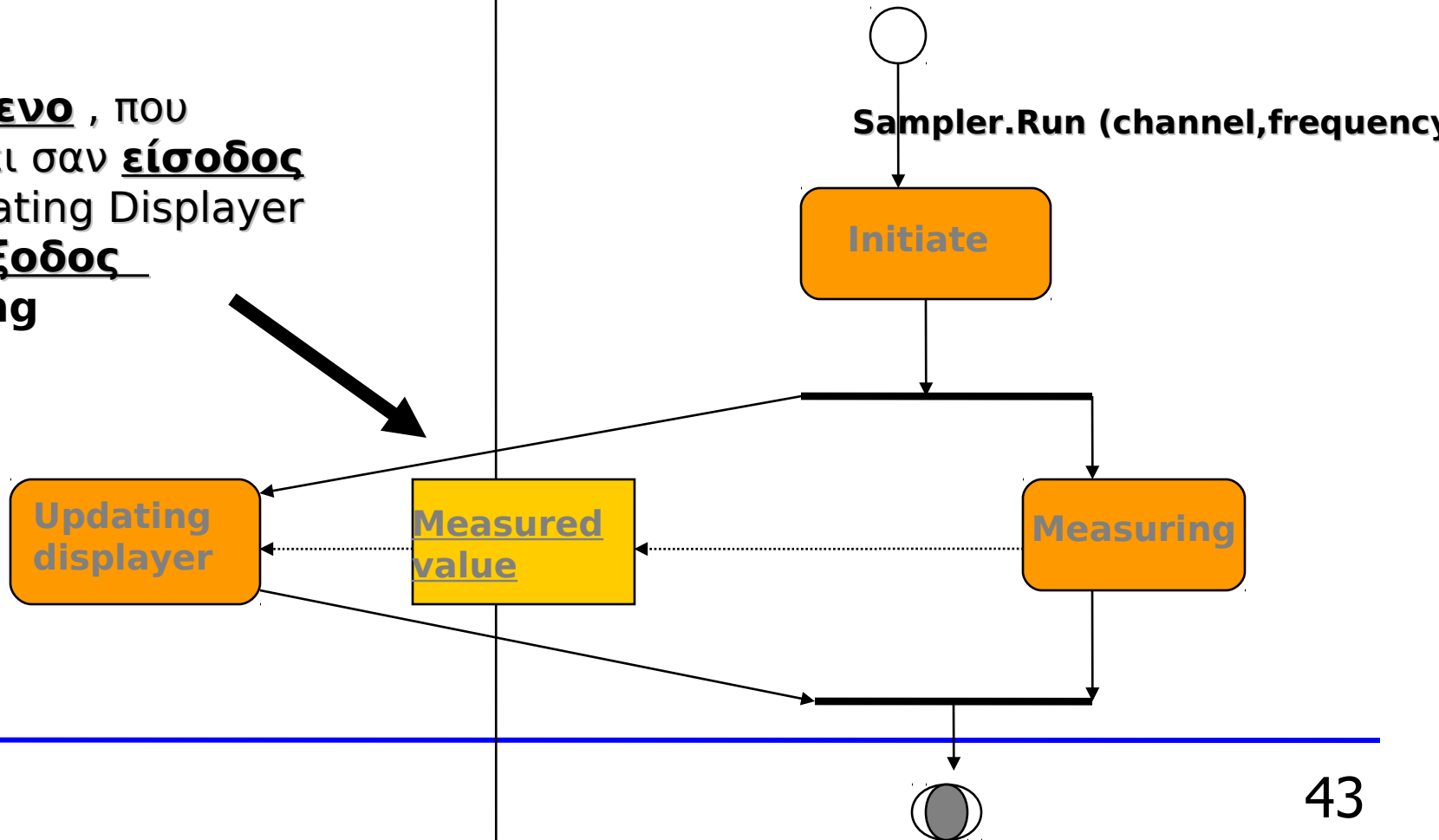


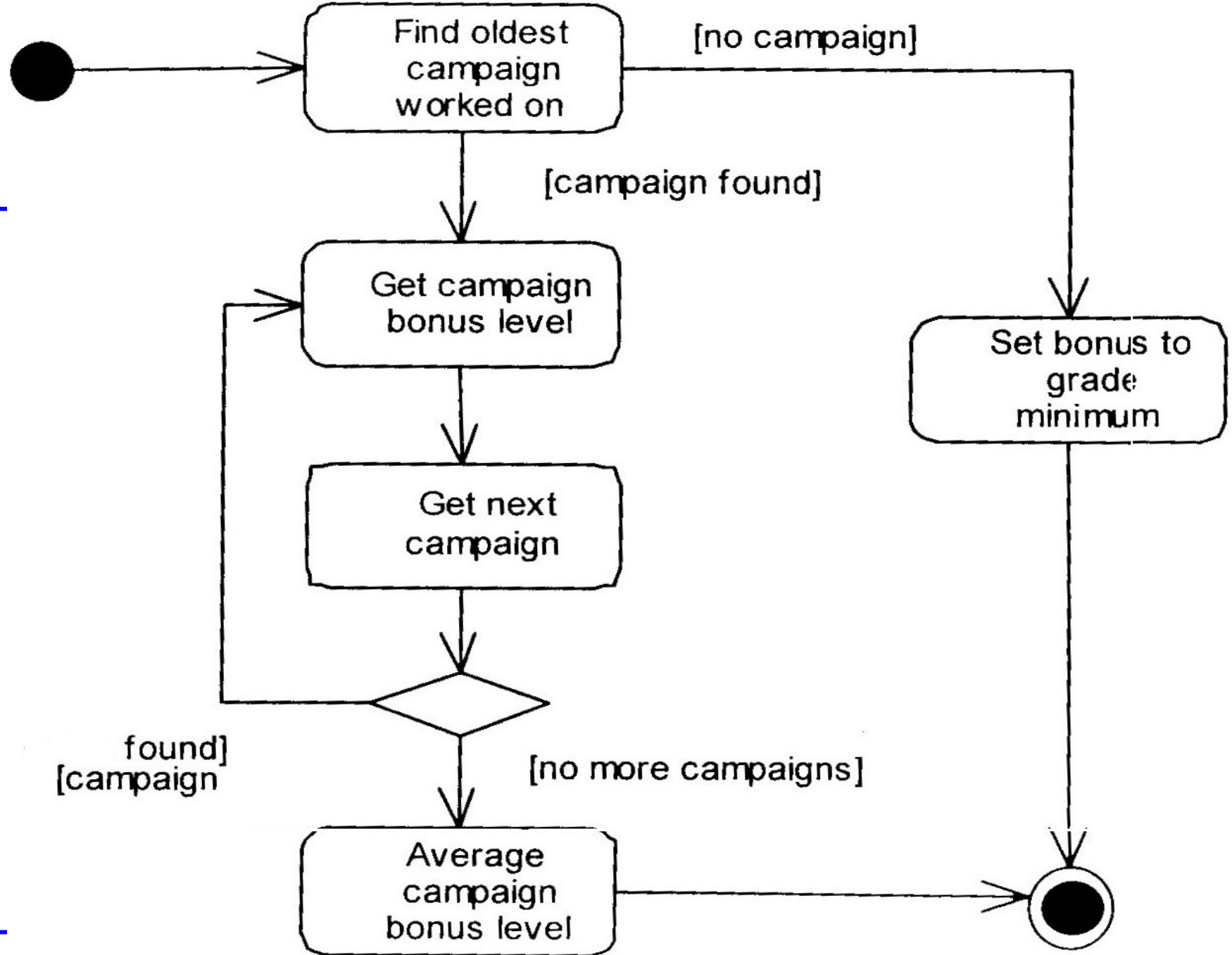
# Αντικείμενα στο διάγραμμα δραστηριοτήτων

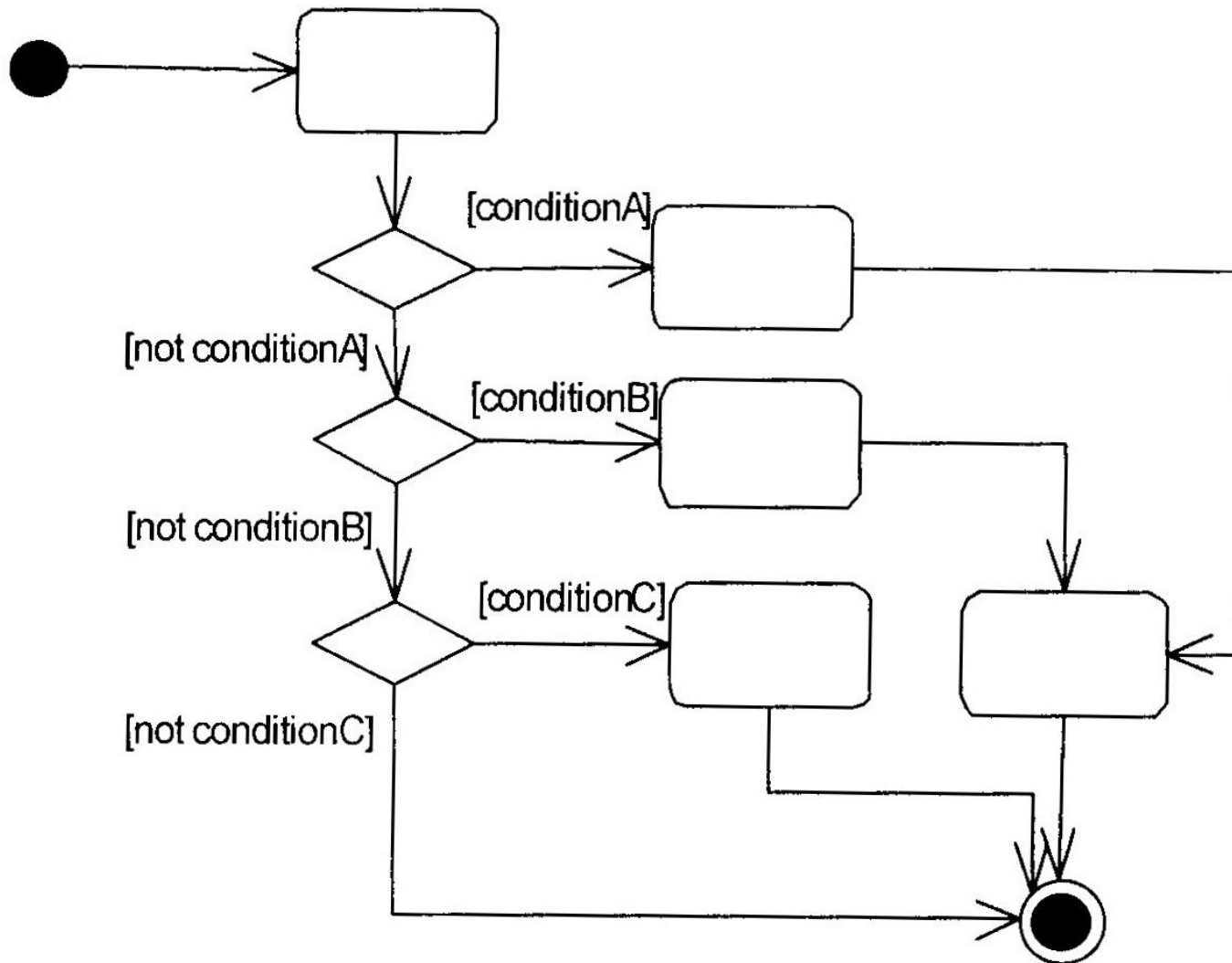
## Οθόνη

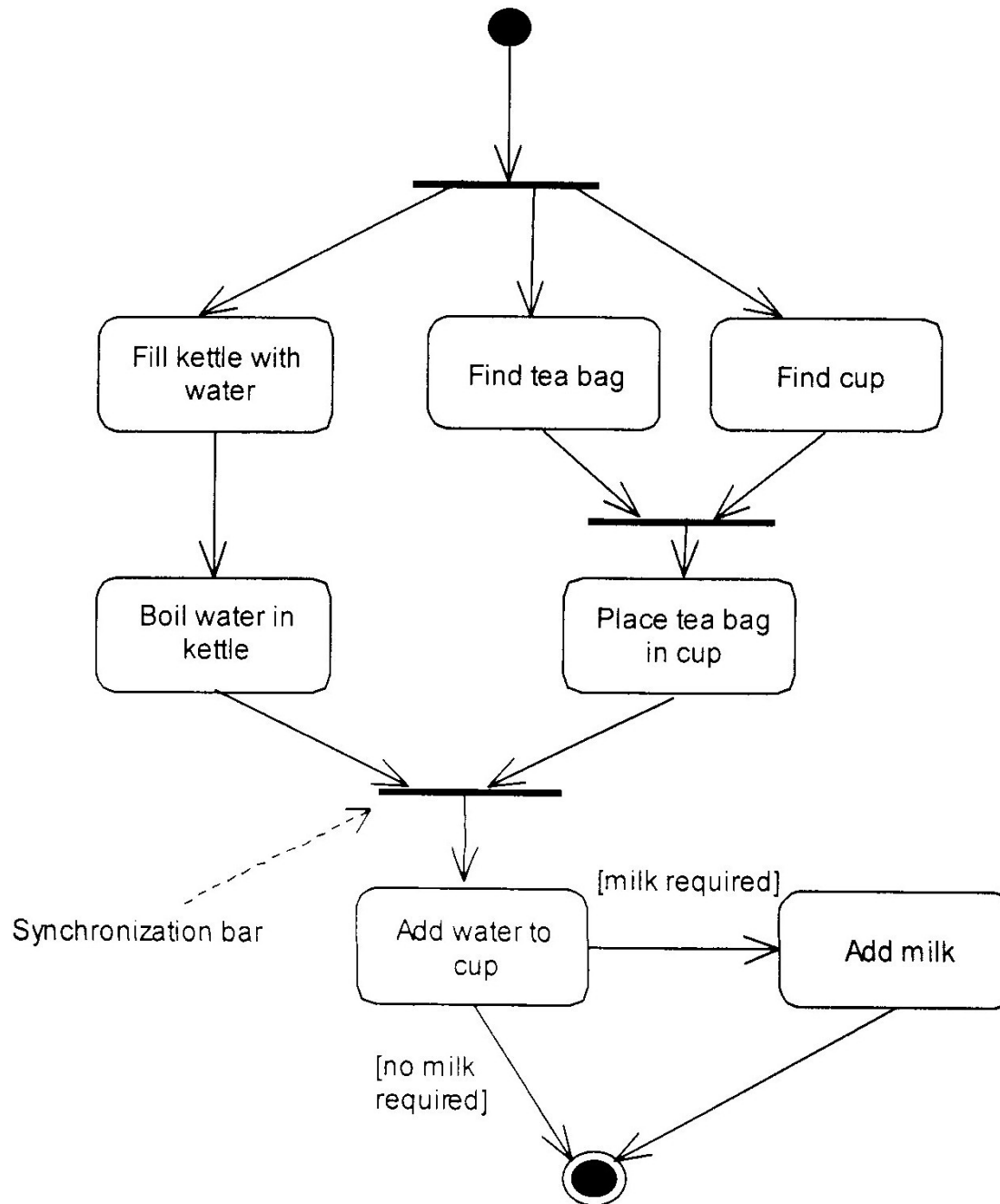
## Δειγματολήπτης

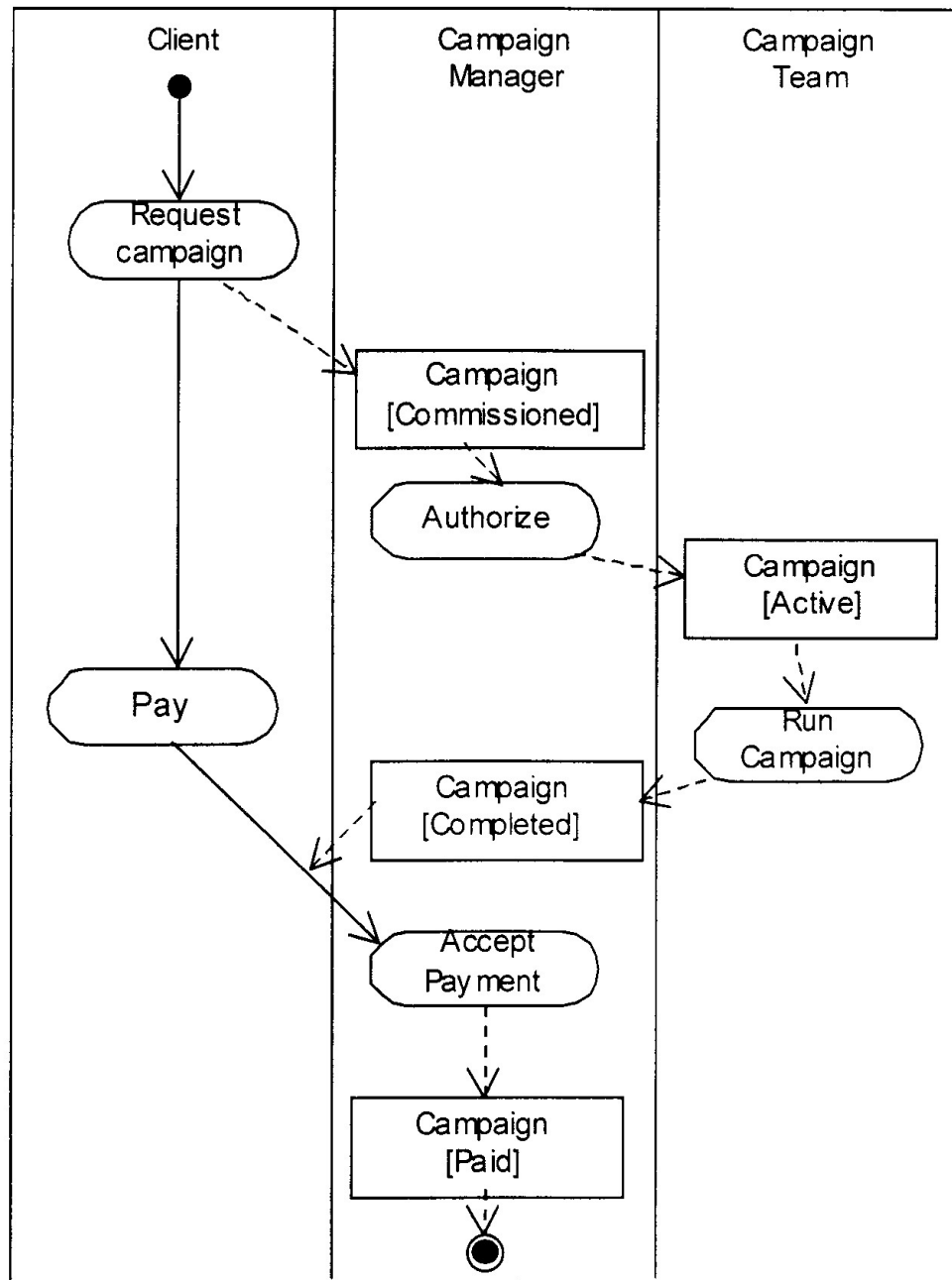
**Αντικείμενο** , που χρησιμεύει σαν είσοδος στην Updating Displayer και σαν έξοδος **measuring**



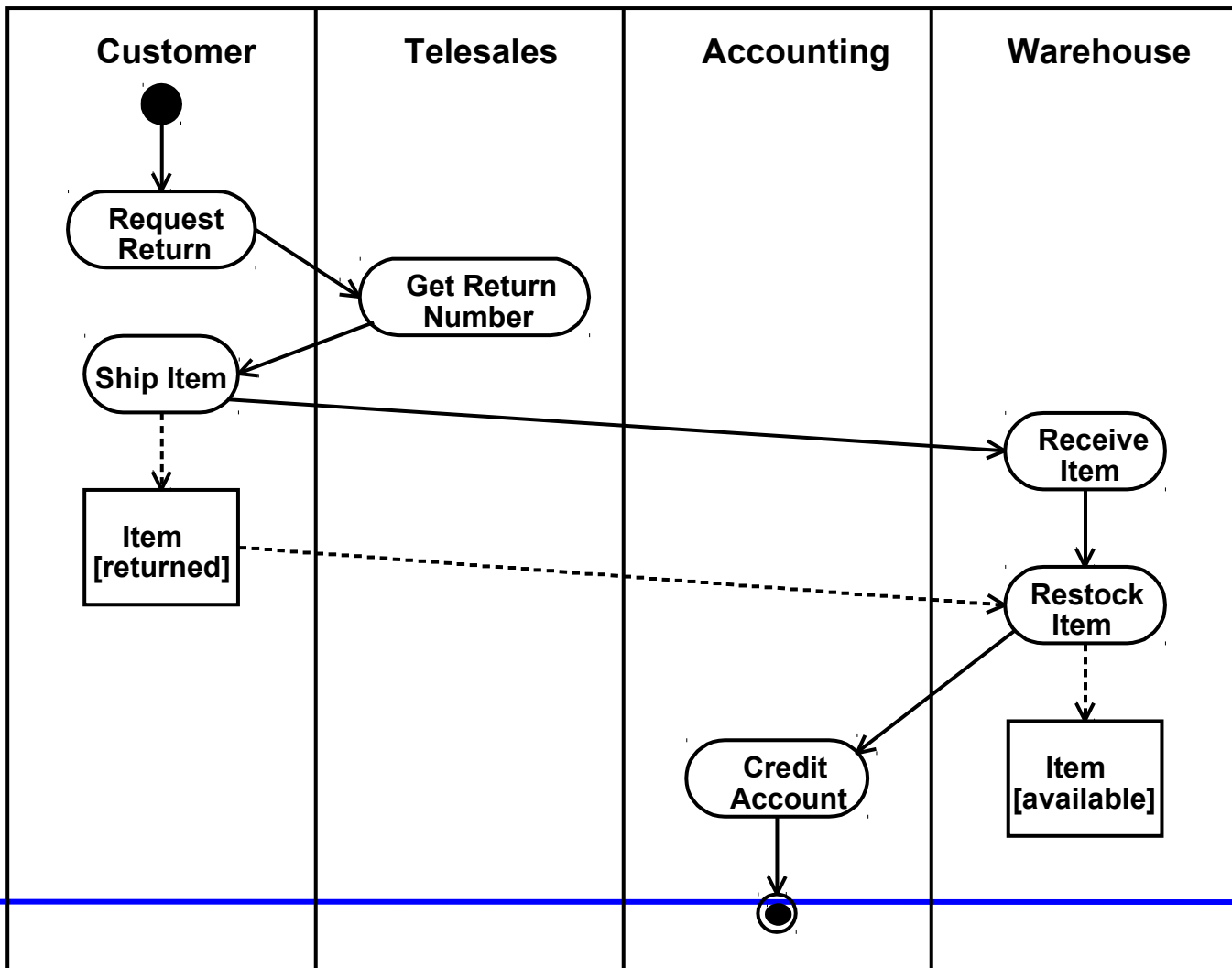




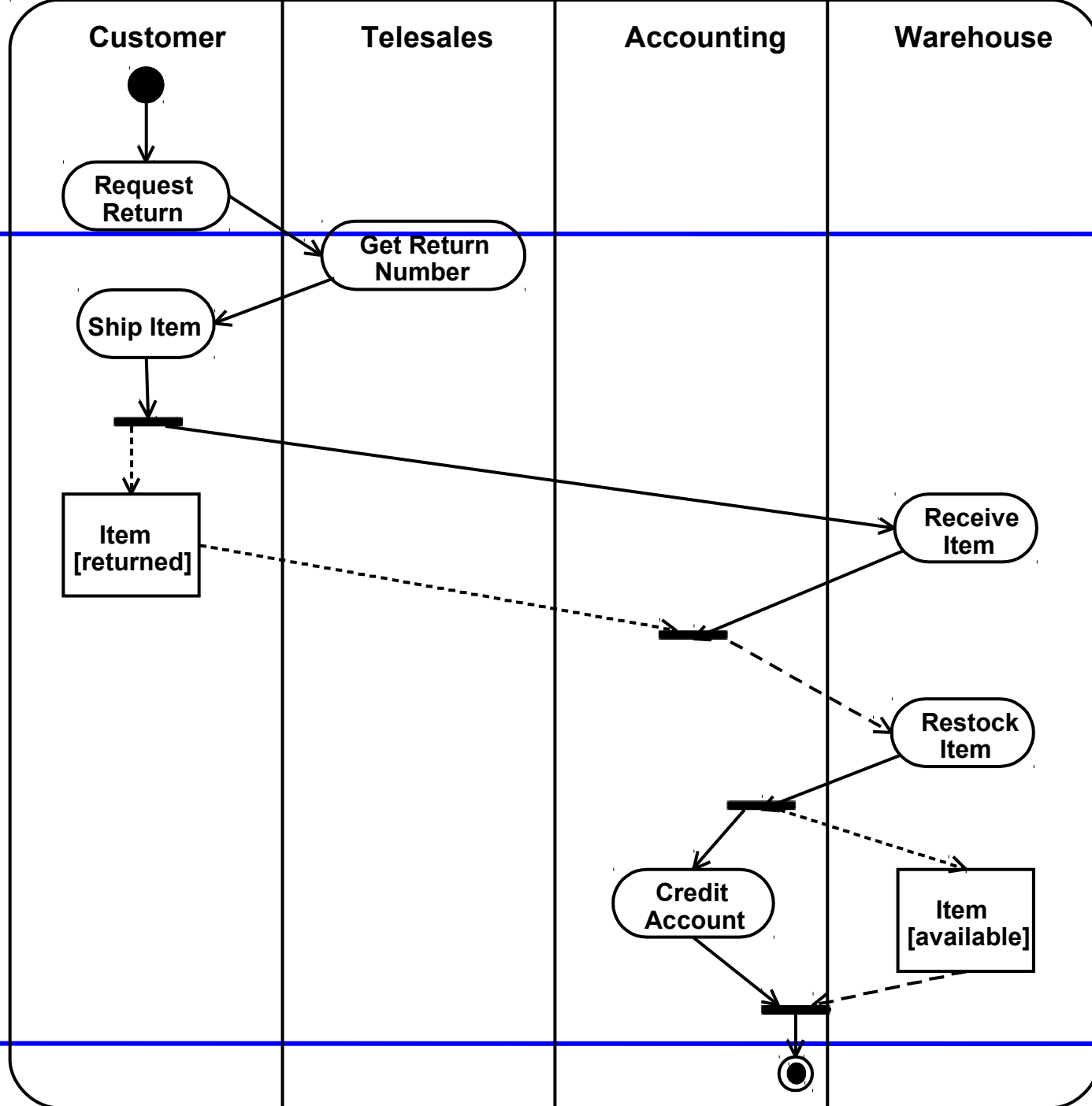




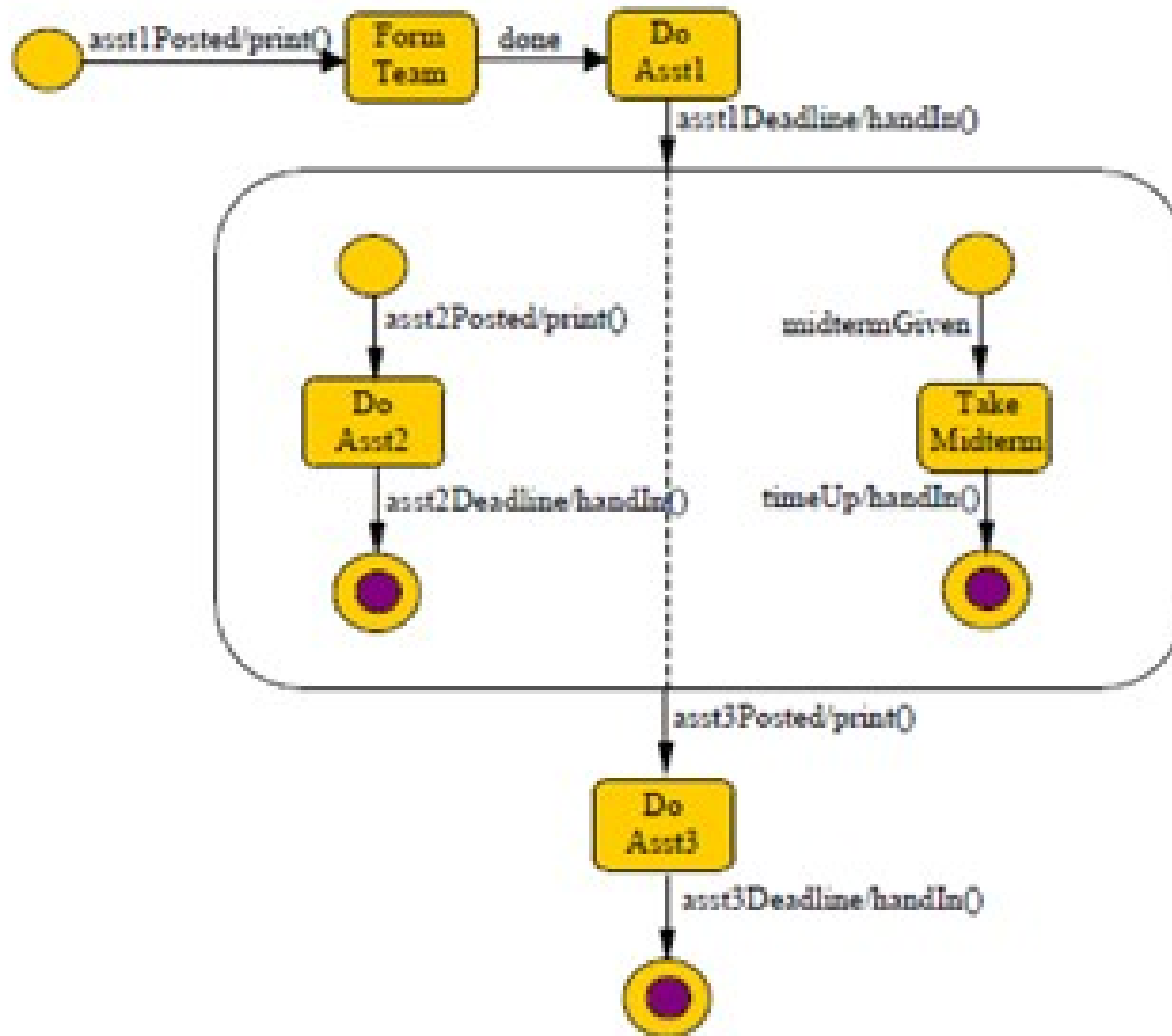
## Παράδειγμα Διαγράμματος Δραστηριότητας



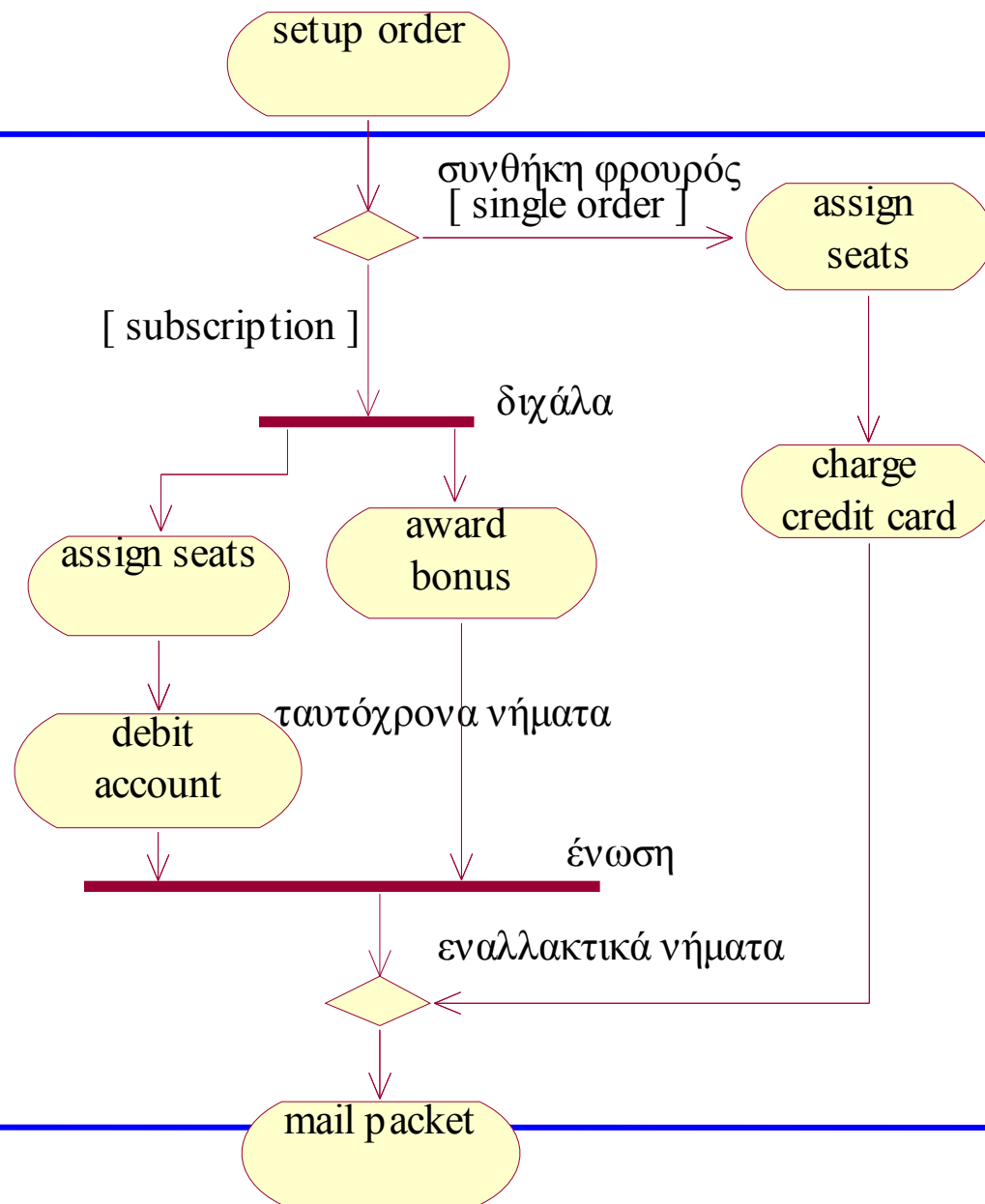




- 
- Στο μεταπτυχιακό μάθημα «Ειδικά Θέματα Πληροφορικής» το οποίο παρακολουθούν φοιτητές με πτυχίο πολυτεχνείου ή θετικών επιστημών ο καθηγητής ανακοινώνει τη 1η εργασία. Οι φοιτητές οργανώνονται σε ομάδες και παραδίδουν τη 1η εργασία. Στη συνέχεια ανακοινώνεται η 2η εργασία και παράλληλα οι φοιτητές εξετάζονται στη πρόοδο. Όταν ολοκληρωθεί η βαθμολόγηση της προόδου και η παράδοση της 2ης εργασίας τότε ανακοινώνεται η 3η εργασία την οποία θα πρέπει να παραδώσουν οι φοιτητές. Στο τέλος γίνεται έλεγχος αν οι φοιτητές έχουν κοπεί από απουσίες. Αν δεν έχουν κοπεί τότε ανακοινώνεται ο βαθμός τους αλλιώς ενημερώνονται τηλεφωνικά οι φοιτητές. Σχεδιάστε ένα διάγραμμα δραστηριοτήτων και ένα διάγραμμα κλάσεων το οποίο αντιπροσωπεύει τη παραπάνω διαδικασία.



# Διαγράμματα Δραστηριότητας (Activity Diagrams)

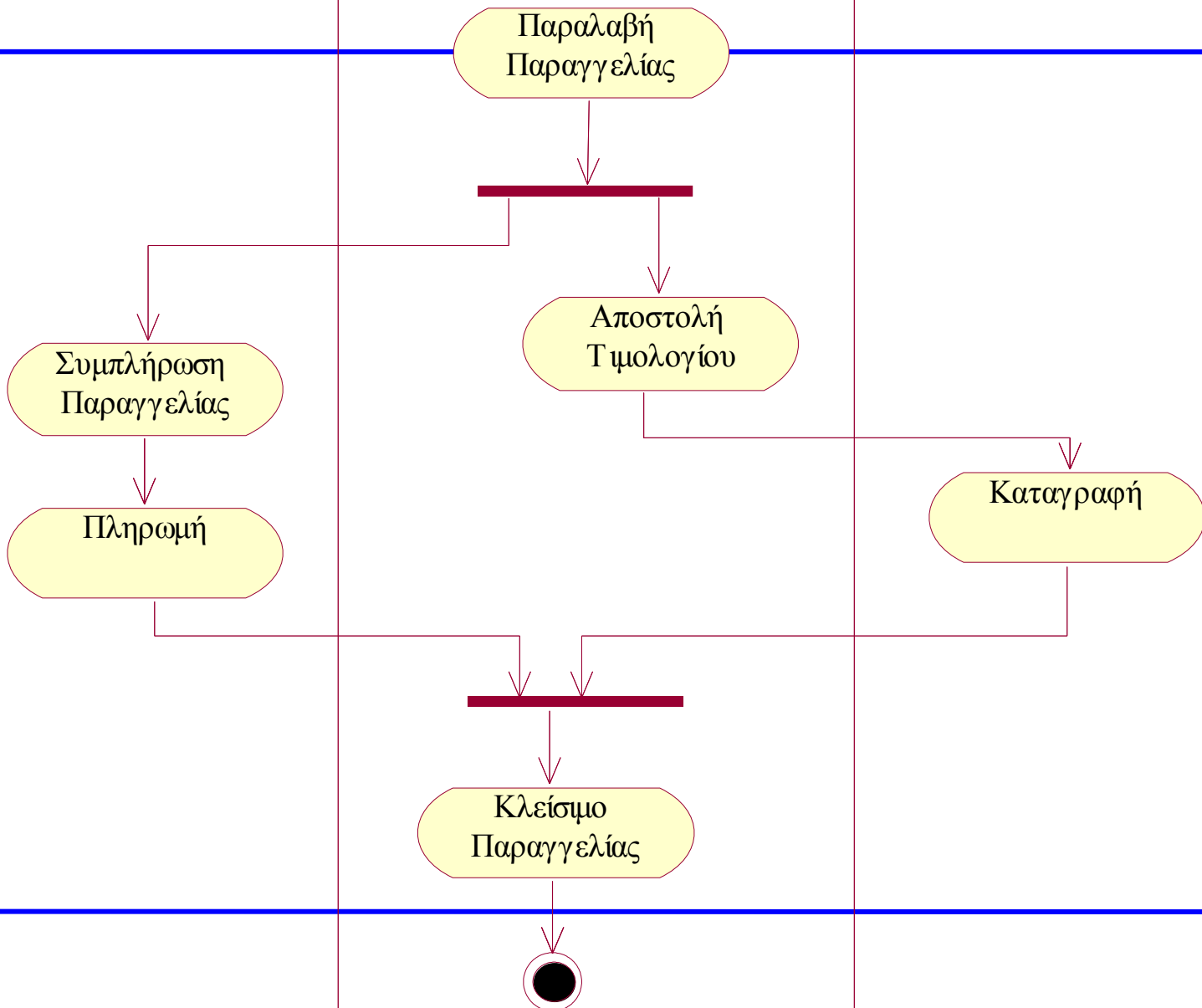


# Διάδρομοι (Swimlanes)

Τμήμα Διεκπεραίωσης

Εξυπηρέτηση Πελατών

Τμήμα Οικονομικού



# Άποψη Μηχανής Καταστάσεων

---

- Η άποψη μηχανής καταστάσεων (state machine view) ενός μοντέλου περιγράφει τη **δυναμική συμπεριφορά των αντικειμένων** στη διάρκεια του χρόνου, αναπαριστώντας τον κύκλο ζωής των αντικειμένων μιας κλάσης.
- Κάθε αντικείμενο αντιμετωπίζεται ως ξεχωριστή οντότητα που επικοινωνεί με το περιβάλλον ανιχνεύοντας γεγονότα και αντιδρώντας σε αυτά.
- **Γεγονός είναι οτιδήποτε μπορεί να γίνει αντιληπτό από ένα αντικείμενο**, όπως η λήψη σημάτων (κλήση μεθόδων) από άλλα αντικείμενα, αλλαγές σε συγκεκριμένες τιμές ή η πάροδος του χρόνου.
- **Μία κατάσταση είναι ένα σύνολο τιμών αντικειμένων** για μία δεδομένη κλάση που αντιδρούν ποιοτικά κατά τον ίδιο τρόπο σε ένα γεγονός. **Με άλλα λόγια, όλα τα αντικείμενα που βρίσκονται στην ίδια κατάσταση εκτελούν τις ίδιες ενέργειες με την ανίχνευση του ίδιου γεγονότος.**

## Γεγονότα (Events)

---

- Ένα γεγονός θα πρέπει να σημειωθεί ότι έχει χωρική και χρονική θέση στο σύστημα, αλλά **δεν έχει διάρκεια**.
- Η μοντελοποίηση κάποιου πράγματος ως γεγονός, συνεπάγεται ότι η εμφάνιση του θα έχει κάποιες συνέπειες. Ο όρος γεγονός αντιστοιχεί στην περιγραφή όλων των γεγονότων ιδίου τύπου που μπορεί να λάβουν χώρα, ακριβώς όπως μια κλάση αποτελεί την κατηγορία στην οποία ανήκουν αντικείμενα με κοινά χαρακτηριστικά. Ένα συγκεκριμένο γεγονός ονομάζεται και **στιγμιότυπο γεγονός** και μπορεί να έχει παραμέτρους που το χαρακτηρίζουν σε σχέση με άλλα στιγμιότυπα του ίδιου γεγονότος. Υπάρχουν διάφοροι τύποι γεγονότων:

## Γεγονότα (Events)

---

- **Γεγονός-σήμα:** Ένα σήμα είναι μία ονοματισμένη οντότητα που λειτουργεί ως μέσο επικοινωνίας μεταξύ δύο αντικειμένων. **Η λήψη ενός σήματος αποτελεί γεγονός για τον αποδέκτη.**
- Τα σήματα αποτελούν **ασύγχρονη** και **μονόδρομη** επικοινωνία: ο αποστολέας δεν περιμένει να επεξεργαστεί το σήμα ο αποδέκτης αλλά συνεχίζει ανεξάρτητα τη δική του εργασία. Για τη μοντελοποίηση αμφίδρομης επικοινωνίας, θα πρέπει να χρησιμοποιηθούν πολλαπλά σήματα, τουλάχιστον ένα σε κάθε κατεύθυνση.
- Τα σήματα είναι δυνατόν να δηλωθούν σε διαγράμματα κλάσεων ως οντότητες με τη λέξη <<signal>>. Οι παράμετροι του σήματος δηλώνονται ως ιδιότητες. Τα σήματα επίσης μπορούν να συμμετέχουν σε σχέσεις γενίκευσης και να κληρονομούν παραμέτρους από γονικά σήματα.



## Γεγονότα (Events)

---

- **Γεγονός-Αλλαγή:** Μία αλλαγή είναι η ικανοποίηση μιας λογικής συνθήκης που εξαρτάται από ορισμένες τιμές ιδιοτήτων. Με αυτό τον τρόπο επιτυγχάνεται αναμονή μέχρι την ικανοποίηση της συνθήκης αλλά θα πρέπει να χρησιμοποιείται με προσοχή καθώς συνεπάγεται έναν συνεχή και ενδεχομένως μή τοπικό υπολογισμό (επειδή οι τιμές που ελέγχονται μπορεί να είναι απομακρυσμένες).
- **Γεγονός-Χρόνος:** Ένα χρονικό γεγονός αναπαριστά την πάροδο του χρόνου. Ένα χρονικό γεγονός μπορεί να προσδιοριστεί είτε με **απόλυτο τρόπο** (ώρα, ημέρα) ή με **σχετικό τρόπο** (χρόνος που παρήλθε από ένα γεγονός).

# Μηχανή Καταστάσεων

---

- **Κατάσταση.** Μία κατάσταση περιγράφει μία χρονική περίοδο κατά τη διάρκεια ζωής ενός αντικειμένου. Μπορεί να χαρακτηριστεί με τρεις συμπληρωματικούς τρόπους:
  - ως ένα σύνολο τιμών αντικειμένων που είναι παρόμοιες από κάποια άποψη,
  - ως μία περίοδος κατά την οποία ένα αντικείμενο αναμένει την εμφάνιση ενός γεγονότος,
  - ως μία περίοδος κατά την οποία ένα αντικείμενο εκτελεί μία εργασία.
- Όταν ένα αντικείμενο βρίσκεται σε μία κατάσταση, αποκρίνεται σε γεγονότα που βρίσκονται στις μεταβάσεις που ξεκινούν από την κατάσταση αυτή. Μία κατάσταση συμβολίζεται ως ένα ορθογώνιο με καμπύλες γωνίες.

# Διάγραμμα Καταστάσεων

