Μεθοδολογία Προγραμματισμού

UML Διαγράμματα

Νικόλαος Πεταλίδης

Τμήμα Μηχανικών Πληροφορικής και Επικοινωνιών
Διεθνές Πανεπιστήμιο της Ελλάδος

Εισαγωγή
Εαρινό Εξάμηνο

Τα υπόλοιπα διαγράμματα UML

- Η UML εκτός από τα διαγράμματα κλάσεων έχει και μια σωρεία άλλων διαγραμμάτων που χρησιμοποιούνται σε διαφορετικές συνθήκες
- Δομικά διαγράμματα
- Διαγράμματα συμπεριφοράς
- Διαγράμματα αλληλεπίδρασης

Δομικά (Structural) Διαγράμματα

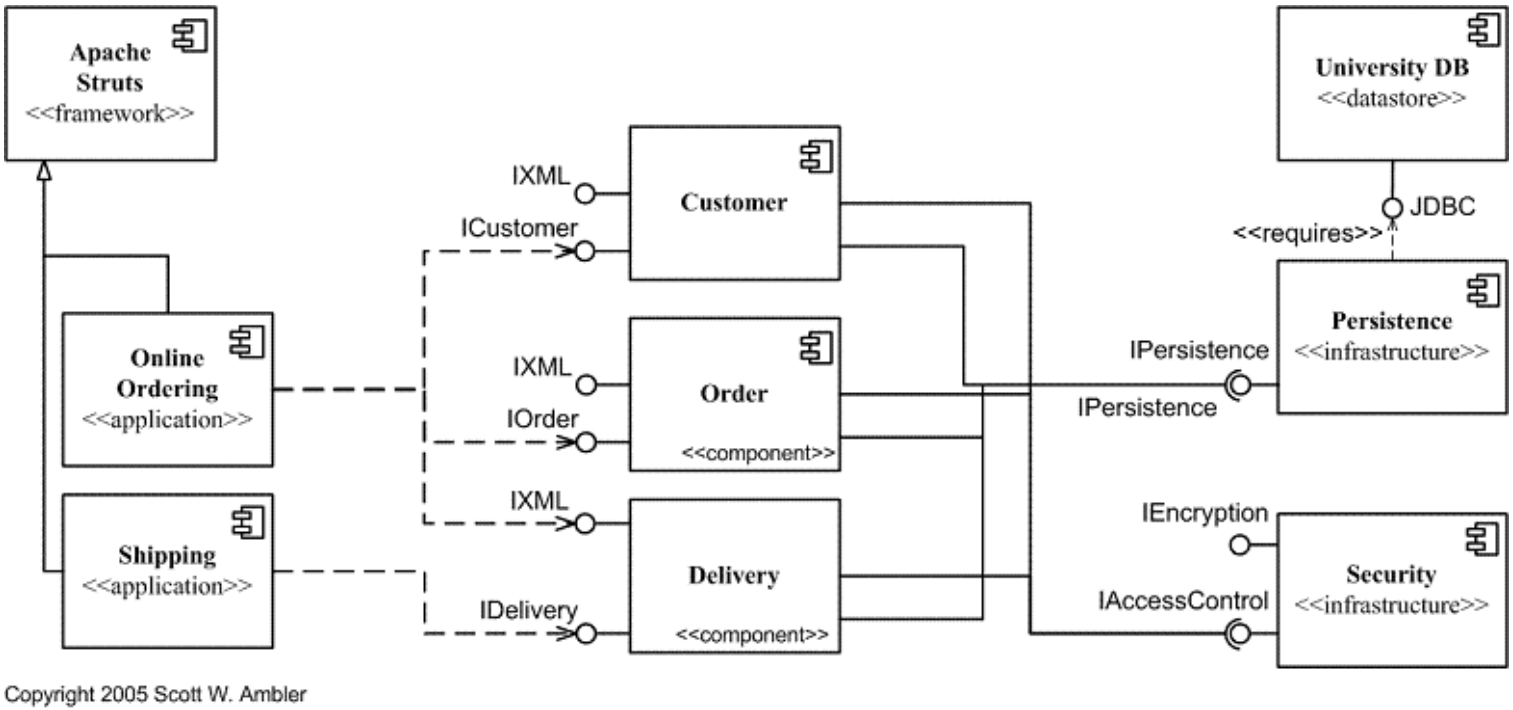
Δίνουν έμφαση στη δομή του συστήματος ανεξάρτητα από το πως μεταβάλλεται στο χρόνο

- Κλάσεων
- Συστατικών
- Διάταξης
- Αντικειμένων
- Πακέτων
- Σύνθετων δομών

Διαγράμματα συστατικών

- Δείχνουν τις εξαρτήσεις μεταξύ των συστατικών του λογισμικού
- Δείχνουν τόσο τις κλάσεις αλλά όσο και τα αρχεία από τα οποιά αποτελείται το λογισμικό (scripts, binary files κτλ)

Παράδειγμα διαγράμματος συστατικών



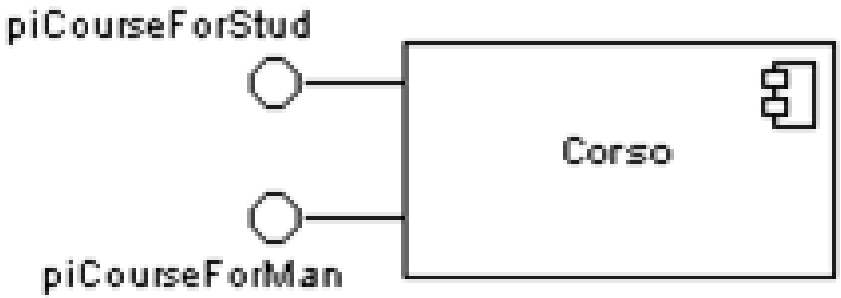
Συστατικά

- Ένα συστατικό είναι μια *αυτόνομη* μονάδα
- Ένα συστατικό έχει

Interfaces	Δηλαδή ένα σύνολο από λειτουργίες και υποχρεώσεις
Εξαρτήσεις	Σχέσεις με άλλα συστατικά
Ports	Σημεία διεπαφής με άλλα συστατικά και το περιβάλλον
Connectors	Συνδέουν το εξωτερικό ενός συστατικού με το εσωτερικό του

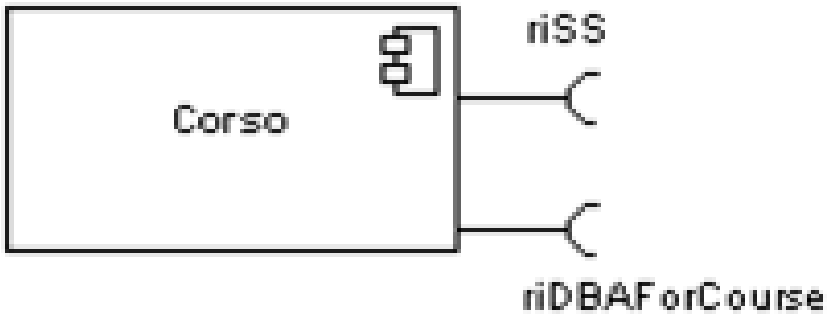
Παρεχόμενες διεπαφές

- Ένα συστατικό μπορεί να δηλώσει ότι παρέχει μια διεπαφή



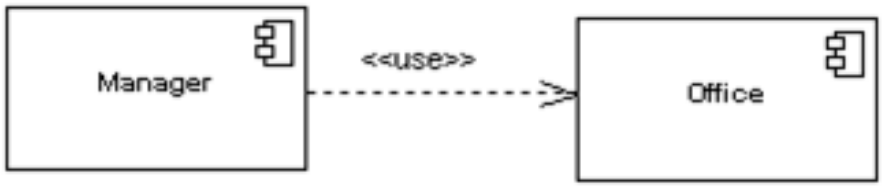
Απαιτούμενες διεπαφές

- Ένα συστατικό μπορεί να δηλώσει ότι απαιτεί μια διεπαφή



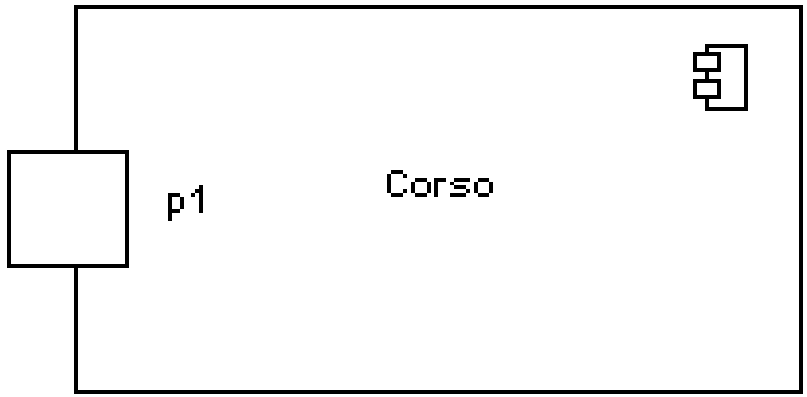
Εξαρτήσεις

- Μια εξάρτηση δηλώνει ότι ένα συστατικό απαιτεί ένα άλλο προκειμένου να λειτουργήσει



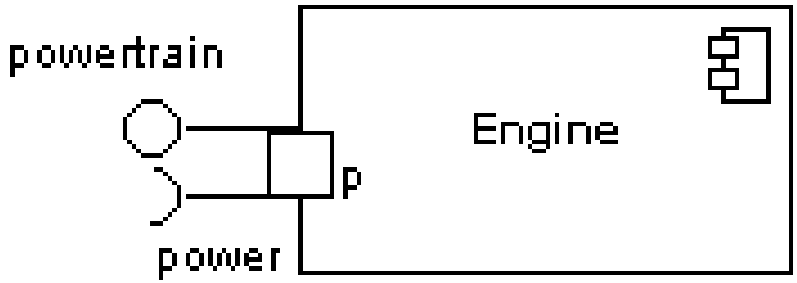
Ports

- Μια πόρτα δηλώνει ένα ξεχωριστό σημείο επαφής μεταξύ ενός συστατικού και του περιβάλλοντός του ή μεταξύ ενός συστατικού και των εσωτερικών τμημάτων του



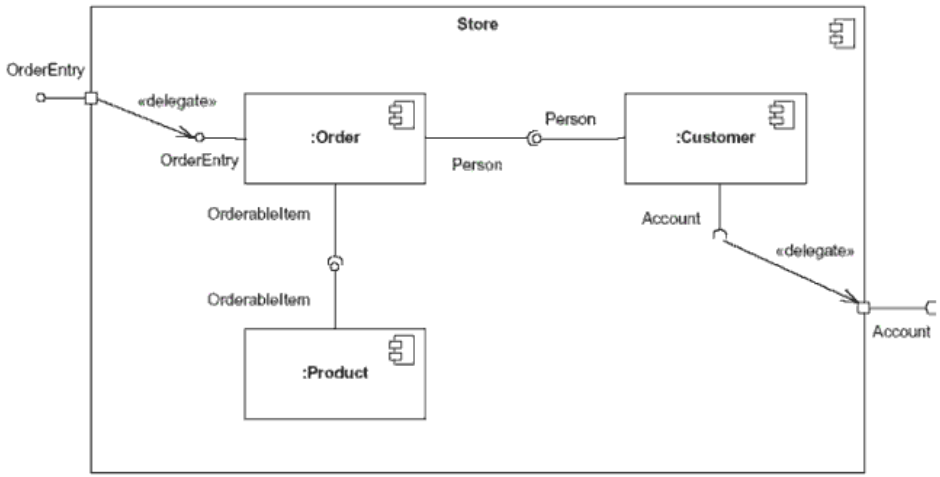
Ports

- Μια πόρτα μπορεί να είναι μόνο για εισαγωγή ή εξαγωγή δεδομένων ή και για αμφίδρομη επικοινωνία. Μπορείτε να θεωρήσετε τις πόρτες ως ένα σύνολο από interfaces με συγκεκριμένο όνομα



Connectors

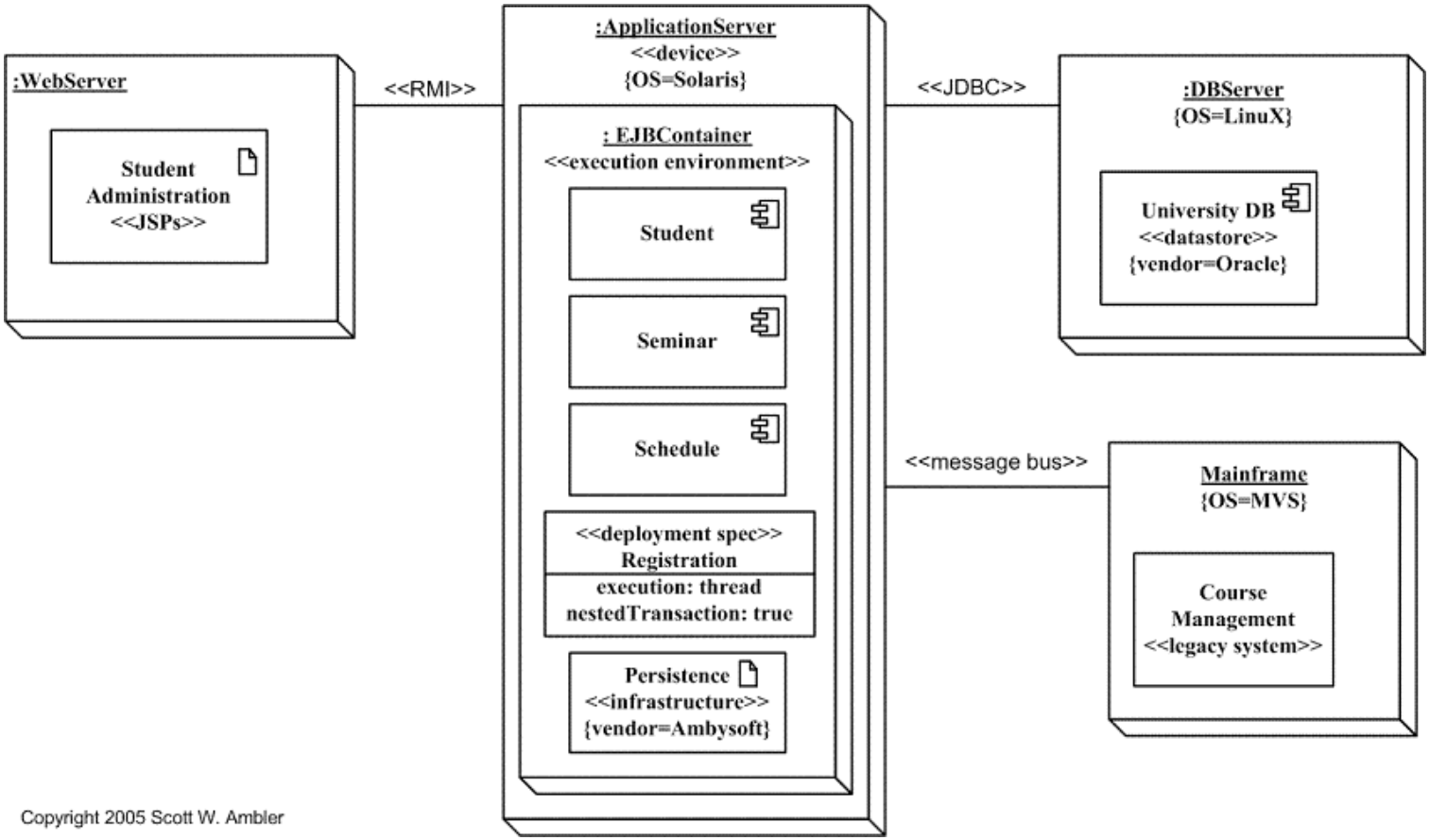
- Πολλές φορές είναι χρήσιμο να δείξουμε την εσωτερική δομή ενός συστατικού



Διαγράμματα διάταξης

- Παρουσιάζουν μια στατική δομή της διάταξης του συστήματος
- Δείχνουν την παραμετροποίηση του hardware καθώς και ποιά κομμάτια λογισμικού εκτελούνται σε ποιό hardware

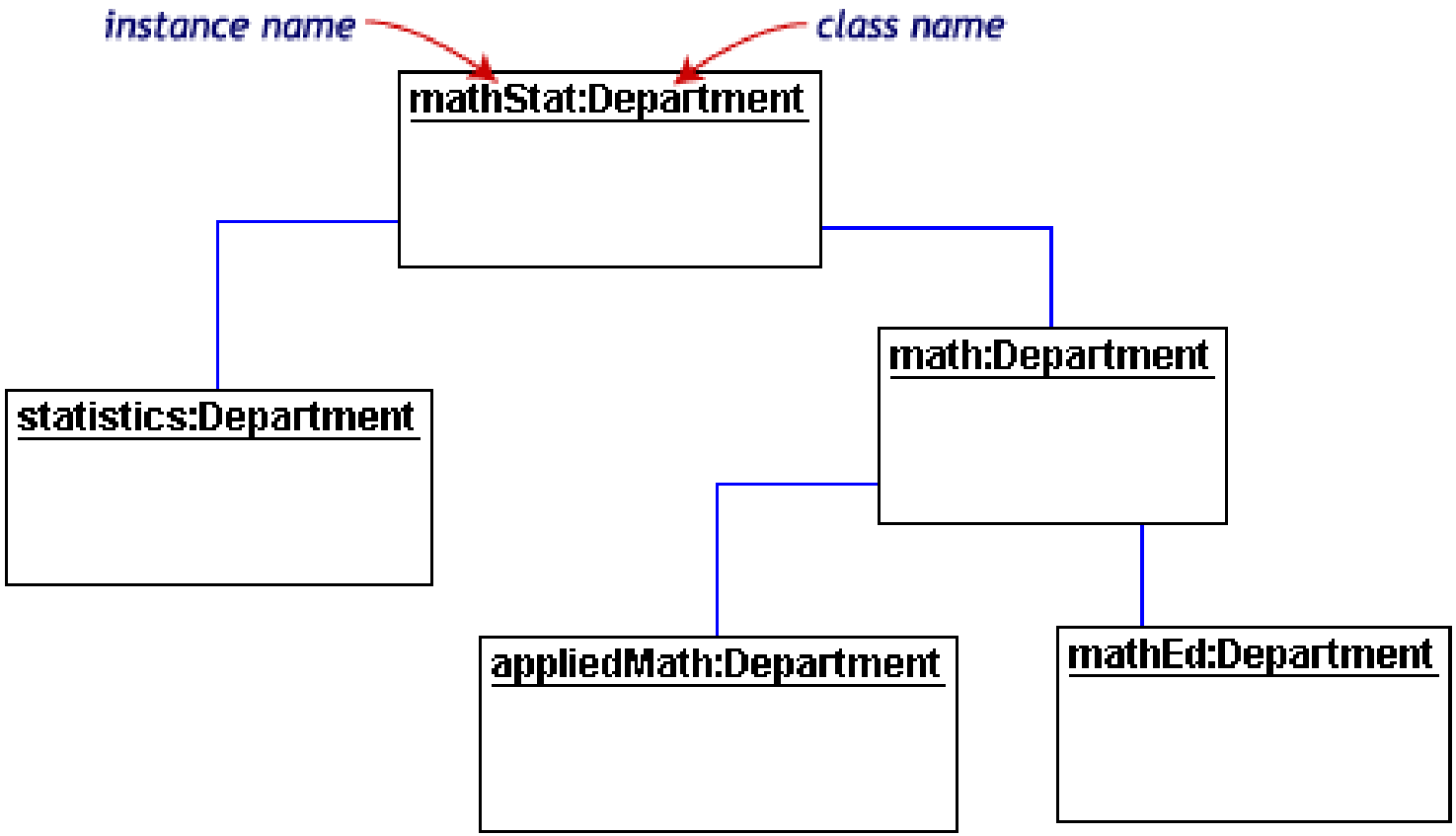
Παράδειγμα διαγράμματος διάταξης



Διαγράμματα αντικειμένων

- Παρόμοια με τα διαγράμματα κλάσεων μόνο που δείχνουν αντικείμενα και όχι κλάσεις
- Δείχνουν πώς συγκεκριμένα αντικείμενα σχετίζονται με άλλα σε συγκεκριμένα σενάρια

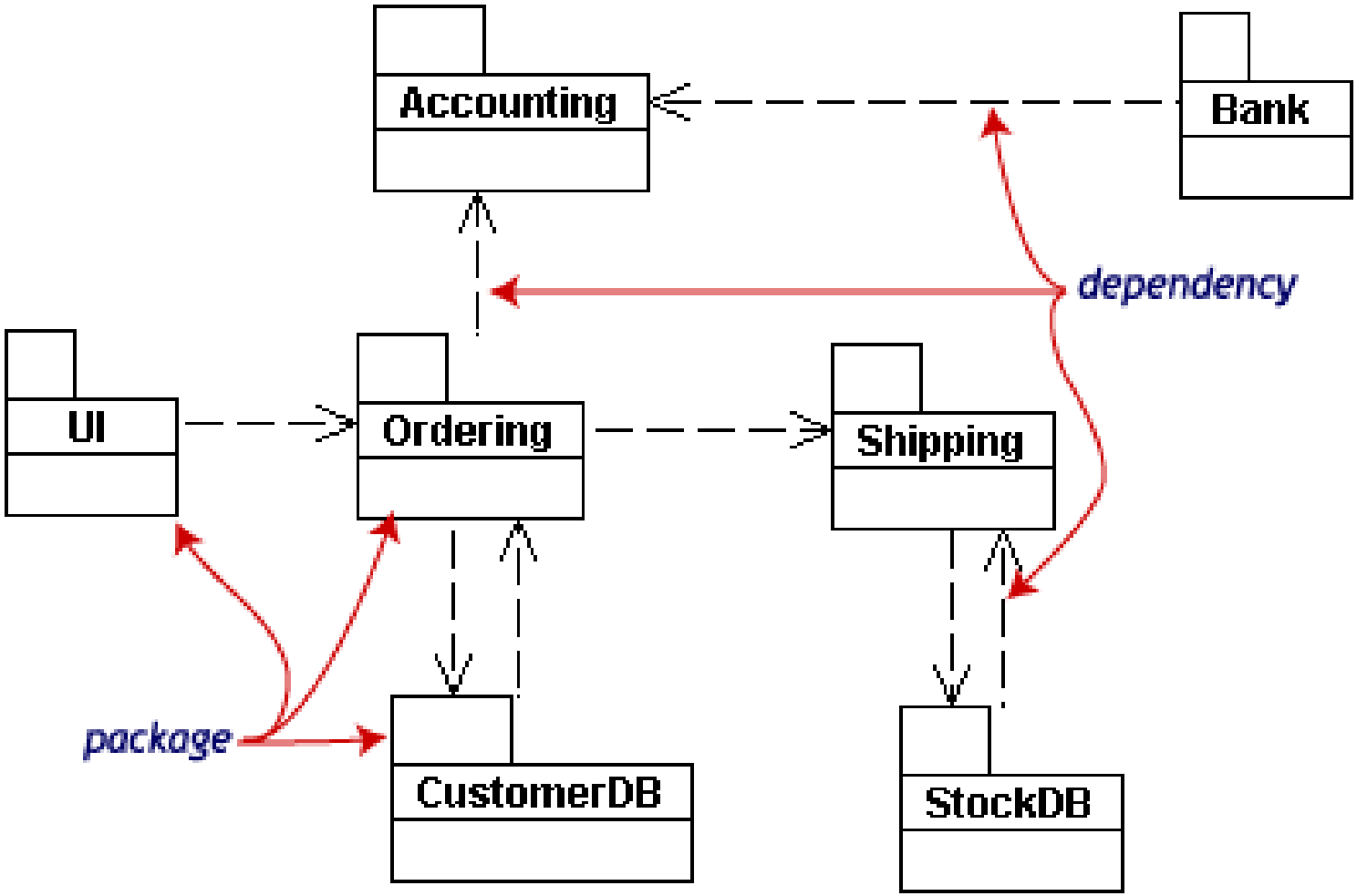
Παράδειγμα διαγράμματος αντικειμένων



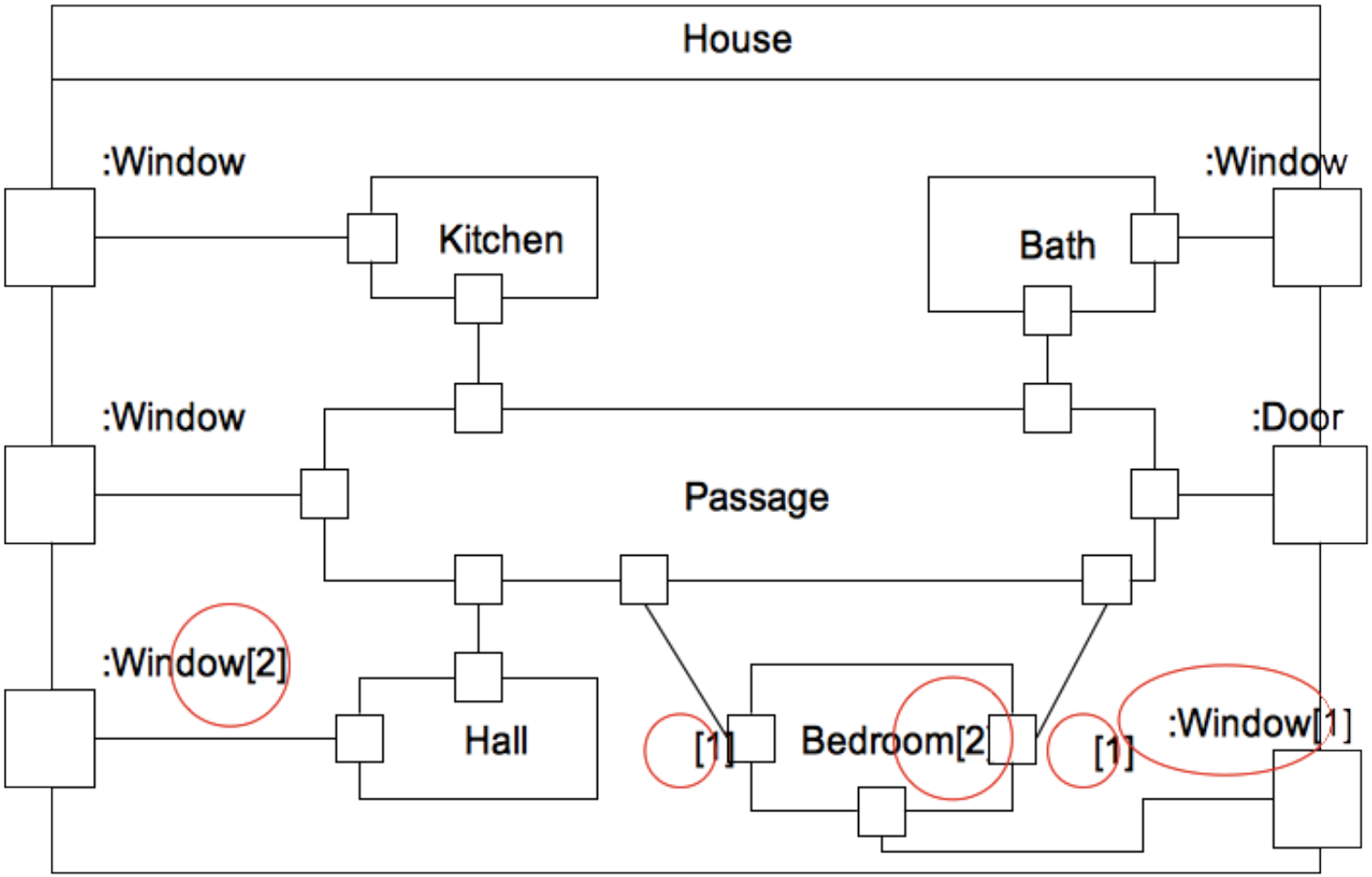
Διαγράμματα πακέτων

- Παρόμοια με τα διαγράμματα κλάσεων
- Ομαδοποιούν κλάσεις σε πακέτα και αναπαριστούν το σύστημα με λιγότερη λεπτομέρεια

Παράδειγμα διαγράμματος πακέτων



Παράδειγμα διαγράμματος σύνθετης δομής



Διαγράμματα συμπεριφοράς (behavioural)

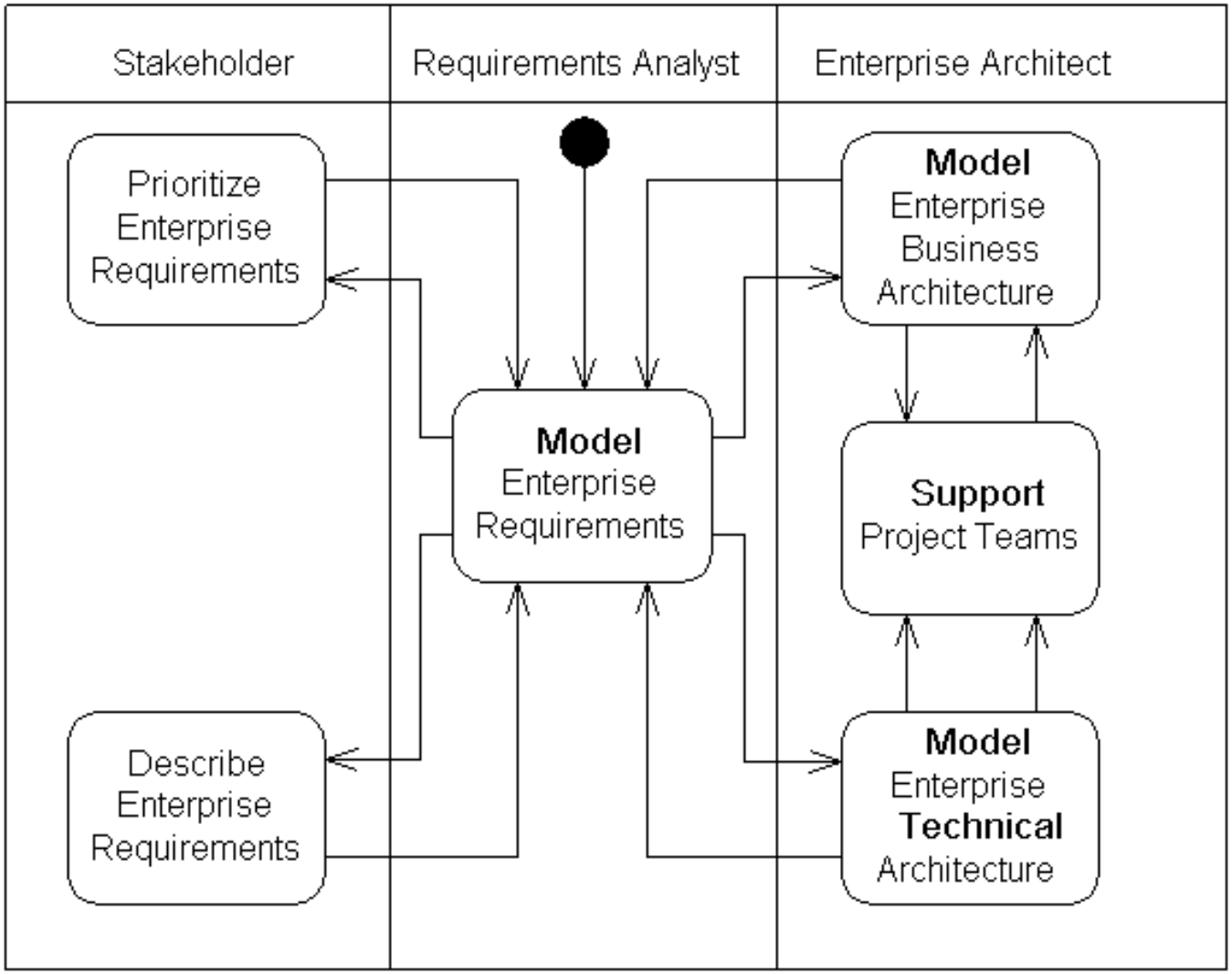
Δείχνουν τη συμπεριφορά μιας συγκεκριμένης διαδικασίας

- Δραστηριότητας
- Χάρτες καταστάσεων
- Περιπτώσεων χρήσης
- Αλληλεπίδρασης

Διαγράμματα δραστηριότητας

- Είναι το ισοδύναμο των παραδοσιακών διαγραμμάτων ροής
- Δείχνουν τη ροή του ελέγχου μέσα στο σύστημα

Παράδειγμα διαγράμματος δραστηριότητας



Χάρτες καταστάσεων

- Η UML μας δίνει σαν εργαλείο τους χάρτες καταστάσεων για να δείξουμε πως μπορεί να αλλάξει η κατάσταση ενός αντικειμένου
- Δείχνουν τις καταστάσεις από τις οποίες περνά ένα αντικείμενο

Παράδειγμα: Οι καταστάσεις μιας παραγγελίας

Διαγράμματα κατάστασης

- Σε ένα διάγραμμα κατάστασης συμβαίνουν κάποιες ενέργειες οι οποίες αλλάζουν την κατάσταση του συστήματος και τις δραστηριότητες που μπορούν να εκτελεστούν
- Συνήθως η τριπλέτα Συμβάν[Συνθήκη]/Ενέργεια σημαίνει ότι αν συμβεί το "Συμβάν" και η "Συνθήκη" είναι αληθής τότε εκτελείται η "Ενέργεια" και ολοκληρώνεται η μετάβαση από μια κατάσταση σε μια άλλη.
- Ενόσω το σύστημα είναι σε μια κατάσταση μπορεί να ολοκληρώσει διάφορες δραστηριότητες

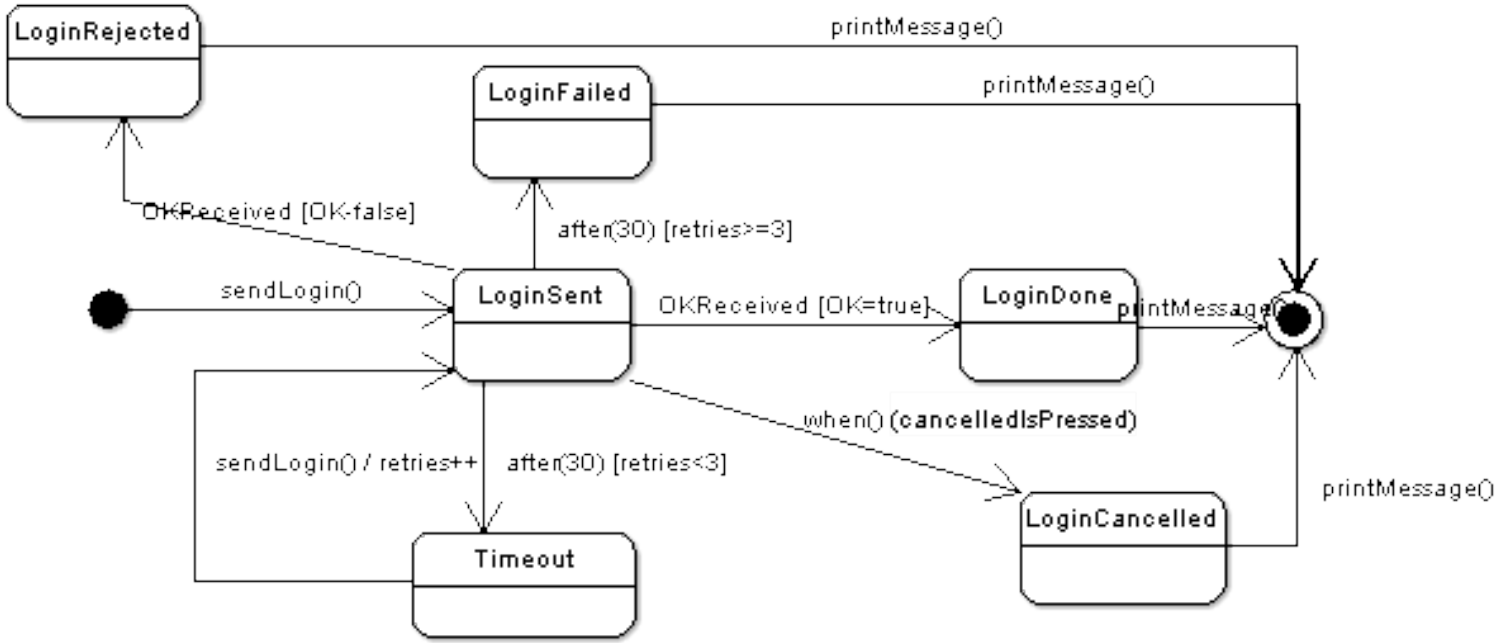
Παράδειγμα

- Έστω ότι έχουμε μία κλάση που αφορά ένα πρωτόκολλο εισόδου (Login Protocol)
- Σε αυτό το πρωτόκολλο αφού πάρουμε εντολή στέλνουμε ένα αίτημα Login περιμένουμε μια απάντηση OK. (LoginSent)
- Αν η απάντηση είναι θετική θεωρούμε ότι η είσοδος είναι πετυχημένη (LoginDone)
- Αν η απάντηση είναι αρνητική θεωρούμε ότι η είσοδος είναι αποτυχημένη (LoginRejected)

Παράδειγμα (συνέχεια)

- Αν αργήσουμε να πάρουμε απάντηση για πάνω από 30sec τότε έχουμε ένα Timeout και ξαναπροσπαθούμε για το πολύ 3 φορές.
- Την 3η φορά θεωρούμε ότι αποτύχαμε στην είσοδο (LoginFailed)
- Επίσης ακυρώνουμε την ενέργεια οποτεδήποτε η συνθήκη cancelIsPressed γίνει αληθής οπότε και ακυρώνουμε την είσοδο (LoginCancelled)

Παράδειγμα: LoginProtocol



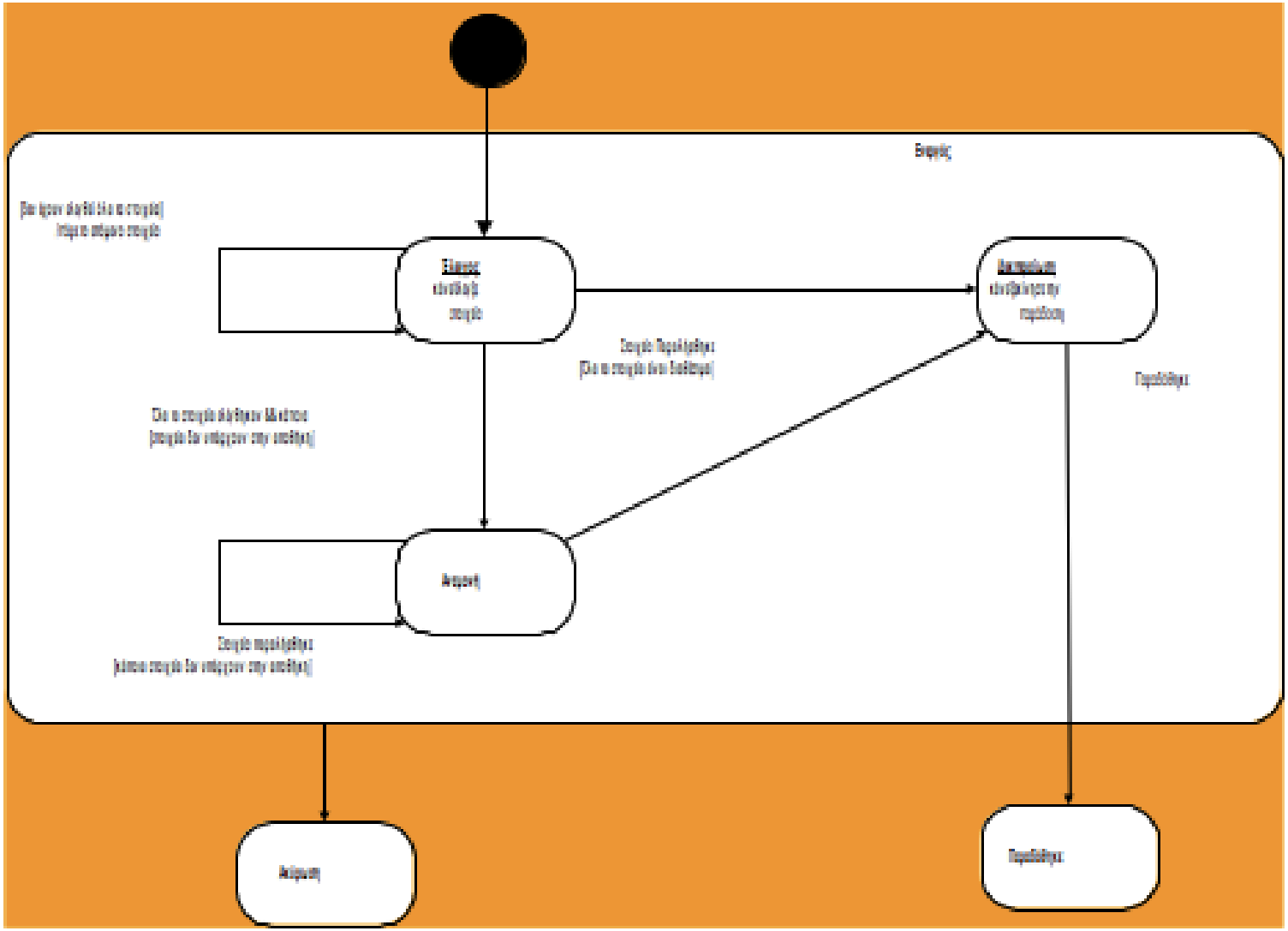
Είδη συμβάντων

- CallEvent: λήψη ενός μηνύματος, π.χ. sendLogin
- SignalEvent: λήψη ενός σήματος, π.χ. OKReceived
- ChangeEvent: συνθήκη γίνεται αληθής, π.χ. when(cancelledPressed)
- TimeEvent: σχετικό ή απόλυτο σημείο στο χρόνο, π.χ. after(30 sec)

Υπερκαταστάσεις

- Όταν από πολλές καταστάσεις υπάρχει η ίδια μετάβαση στην ίδια κατάσταση τότε μπορούμε να χρησιμοποιήσουμε την έννοια της υπερκατάστασης

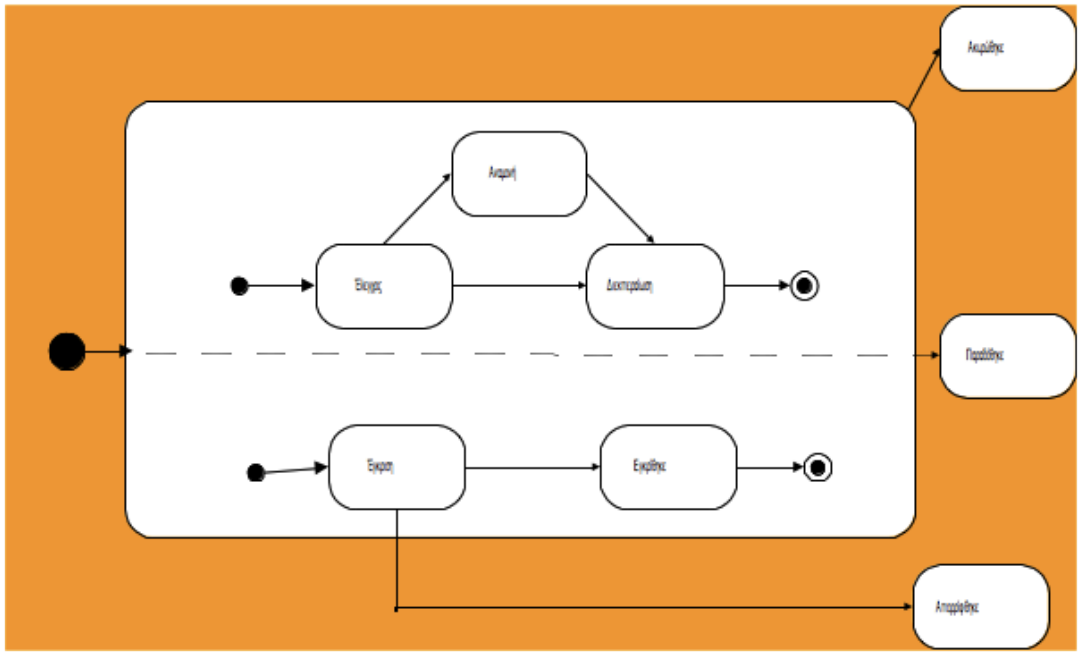
Παράδειγμα υπερκατάστασης



Ταυτοχρονισμός

- Υπάρχουν επίσης πολλά προβλήματα τα οποία εμπεριέχουν την έννοια της παραλληλίας
- Για παράδειγμα όσο γίνεται ο έλεγχος για μια παραγγελία στην αποθήκη, ταυτόχρονα γίνεται έλεγχος για την πληρωμή της παραγγελίας

Παράδειγμα ταυτοχρονισμού



Διαγράμματα Περιπτώσεων χρήσης

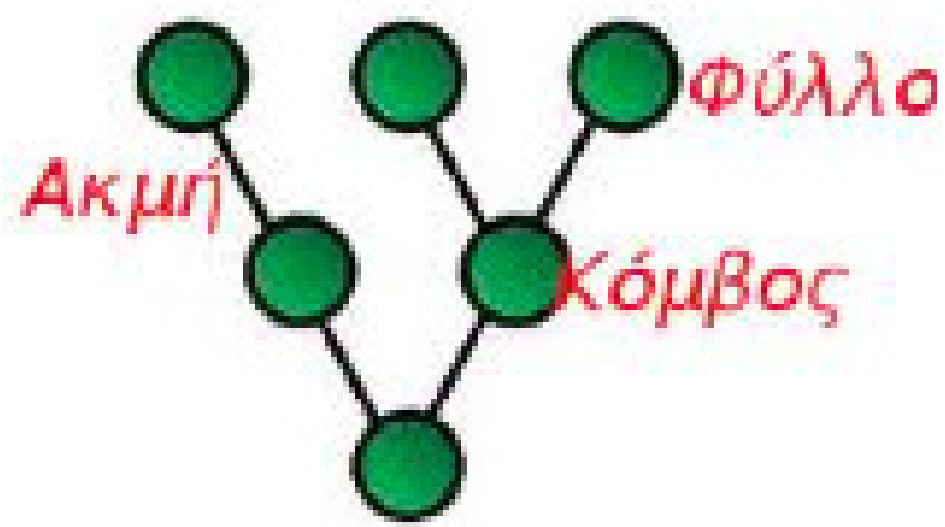
- Χρησιμοποιούνται για την αναπαράσταση της λειτουργίας ενός συστήματος, υποσυστήματος ή κλάσης, όπως αυτή γίνεται αντιληπτή από τον εξωτερικό χρήστη (χαρακτήρας).
- Χωρίζουν τη λειτουργία του συστήματος σε συναλλαγές που έχουν νόημα για τους χαρακτήρες (actors). Κάθε χωριστή και ολοκληρωμένη λειτουργία είναι μια περίπτωση χρήσης.
- Το σύνολο των περιπτώσεων χρήσης συνιστούν τη συμπεριφορά του συστήματος.

Διαγράμματα Περιπτώσεων χρήσης

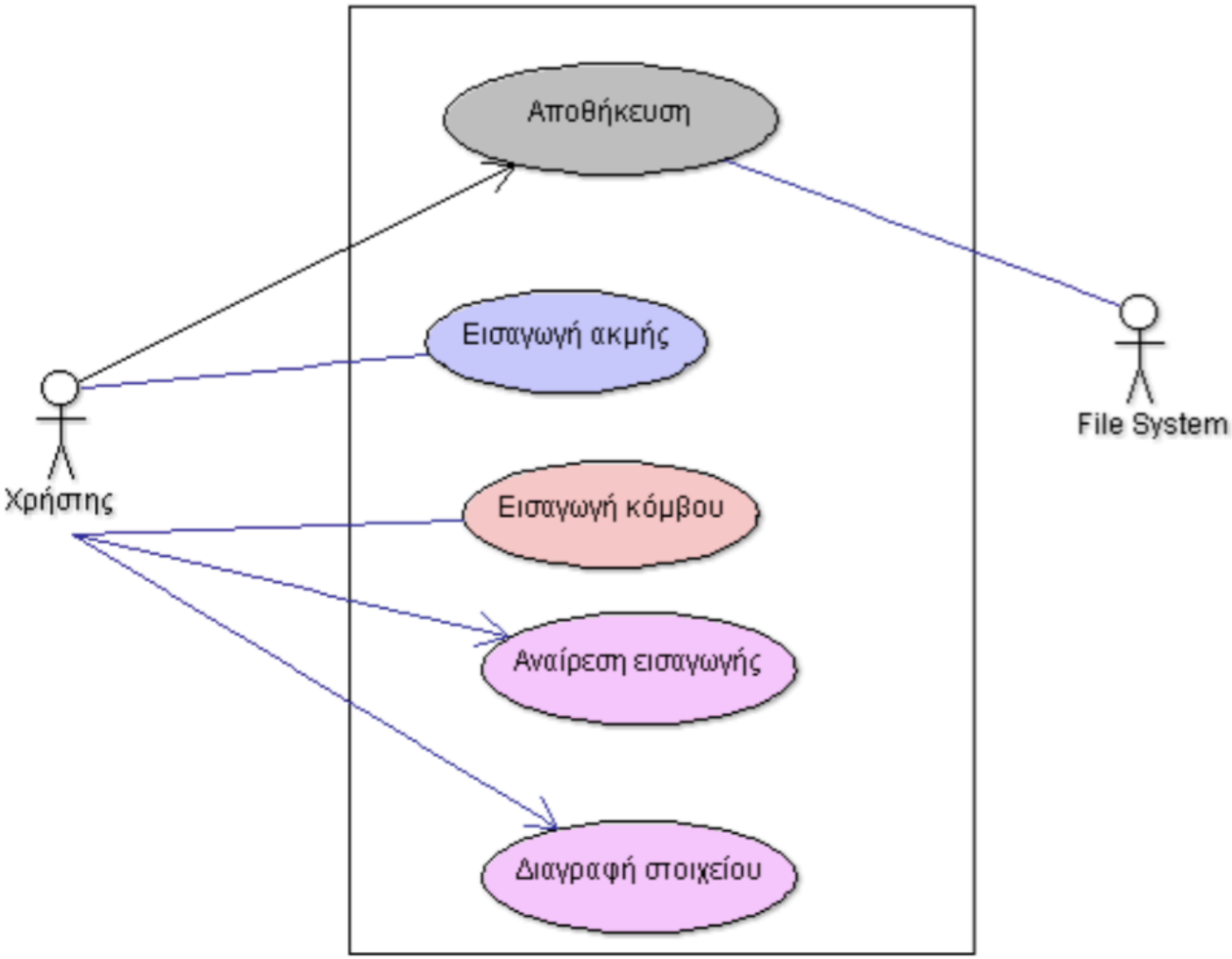
- Σε κάθε διάγραμμα περίπτωσης χρήσης απεικονίζεται ένας χρήστης του συστήματος ως ένα ανθρωπάκι
- Η περίπτωση χρήσης απεικονίζεται ως μία έλλειψη.
- Ο χρήστης «ξεκινά» μία περίπτωση χρήσης αναμένοντας την εκτέλεση κάποια λειτουργίας. Η συσχέτιση μεταξύ χρήστη και περίπτωσης χρήσης απεικονίζεται με μία γραμμή μεταξύ τους ενώ η φορά της ενεργοποίησης με τη χρήση κατευθυνόμενης γραμμής.

Παράδειγμα

- Έστω ότι πρέπει να φτιάξετε ένα πρόγραμμα δημιουργίας γράφων. Ένα τέτοιο πρόγραμμα θα σας επιτρέπει να «ζωγραφίζετε» στην οθόνη γράφους σαν αυτούς πιο κάτω



Μοντέλο παραδείγματος



Κανονική και εναλλακτικές ροές

- Μια περίπτωση χρήσης έχει μια κανονική και πολλές εναλλακτικές συμπεριφορές
- Σε ένα μοντέλο περιπτώσεων χρήσης θεωρείται ότι η εκτέλεση κάθε περίπτωσης χρήσης είναι ανεξάρτητη από τις υπόλοιπες (ορθογωνικότητα)
- Στην πράξη βέβαια, κατά την υλοποίηση του συστήματος μπορεί να υπάρχουν εξαρτήσεις μεταξύ τους.

Περιγραφή μιας περίπτωσης χρήσης

Σύντομη Περιγραφή Περιγραφή σε 1 ή 2 σειρές της συμπεριφοράς που εκτελείται και των χρηστών της συγκεκριμένης περίπτωσης

Προαπαιτούμενα Τι πρέπει να ισχύει ώστε να είναι δυνατή η έναρξη της περίπτωσης χρήσης

Κύρια ροή Περιγραφή σε μορφή κειμένου των γεγονότων με τη σειρά που θα συμβούν

Εναλλακτικές ροές Περιγραφή εξαιρέσεων ή λανθασμένων καταστάσεων

Αποτελέσματα Συνθήκες που θα ισχύουν μετά την ομαλή εκτέλεση της εν λόγω περίπτωσης χρήσης.

Παράδειγμα περιγραφής

Σύντομη Περιγραφή Η περίπτωση χρήσης επιτρέπει σε έναν Χρήστη να εισάγει ένα νέο κόμβο στο γράφο

Προαπαιτούμενα Ο χρήστης έχει τρέξει την περίπτωση χρήσης «Δημιουργία νέου γράφου» (δε φαίνεται στην προηγούμενη διαφάνεια)

Κύρια ροή

1. Η περίπτωση χρήσης ξεκινά όταν ο Χρήστης αποφασίσει να προσθέσει ένα κόμβο.
2. Το σύστημα δίνει τη δυνατότητα στον Χρήστη να τοποθετήσει ένα μικρό κύκλο σε οποιοδήποτε σημείο του καμβά του προγράμματος
3. Ο Χρήστης τοποθετεί τον κόμβο και το σύστημα τον ρωτάει να εισάγει το facebookId του χρήστη στον οποίο αντιστοιχεί ο κόμβος

Παράδειγμα περιγραφής

Εναλλακτική ροή γεγονότων Ο χρήστης προσπαθεί να προσθέσει έναν κόμβο αλλά το σύστημα δεν του επιτρέπει λόγω περιορισμών μνήμης. Πρέπει να εμφανίζεται ανάλογο μήνυμα

Αποτέλεσμα Αν η περίπτωση χρήσης είναι επιτυχής, ο γράφος περιέχει έναν περισσότερο κόμβο

Συνεργασία μεταξύ περιπτώσεων χρήσης

Συσχέτιση Το μονοπάτι επικοινωνίας μεταξύ ενός χαρακτήρα και μιας περίπτωσης χρήσης στην οποία συμμετέχει

Επέκταση (extends) Η προσθήκη λειτουργικότητας σε μία βασική περίπτωση χρήσης (η οποία δεν γνωρίζει για αυτή)

Γενίκευση Μία συσχέτιση μεταξύ μιας γενικής περίπτωσης χρήσης και μίας ειδικότερης που κληρονομεί στοιχεία συμπεριφοράς και προσθέτει νέα χαρακτηριστικά.

Περιεκτικότητα (includes) Η προσθήκη λειτουργικότητας σε μία βασική περίπτωση χρήσης (η οποία περιγράφει σαφώς την περίπτωση που εμπεριέχεται)

Πότε να χρησιμοποιείτε τις σχέσεις

Περιεκτικότητα Όταν επαναλαμβάνετε τα ίδια σε δύο ή περισσότερες περιπτώσεις χρήσης για να αποφύγετε την επανάληψη

Επέκταση (extends) Όταν περιγράφετε μια παραλλαγή μιας κανονικής συμπεριφοράς και θέλετε να ελέγξετε ακριβώς τα σημεία στα οποία αυτή διαφέρει από την κανονική (σημεία επέκτασης)

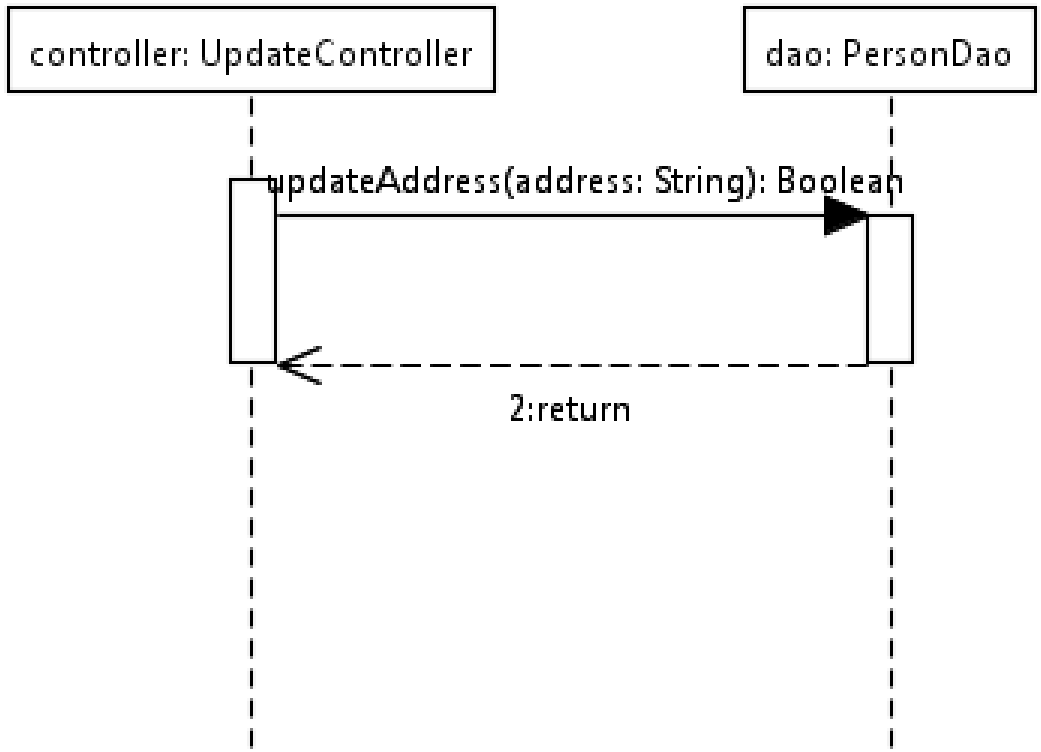
Γενίκευση Όταν περιγράφετε μια παραλλαγή μιας κανονικής συμπεριφοράς και θέλετε να το κάνετε με κομψό τρόπο

Διαγράμματα αλληλεπίδρασης (interaction)

Δείχνουν πώς αλληλεπιδρούν μεταξύ τους τα αντικείμενα

- Ακολουθίας
- Επικοινωνίας
- Χρονισμού (δεν θα αναλυθούν)
- Σύνοψης (δεν θα αναλυθούν)

Παράδειγμα διαγράμματος ακολουθίας



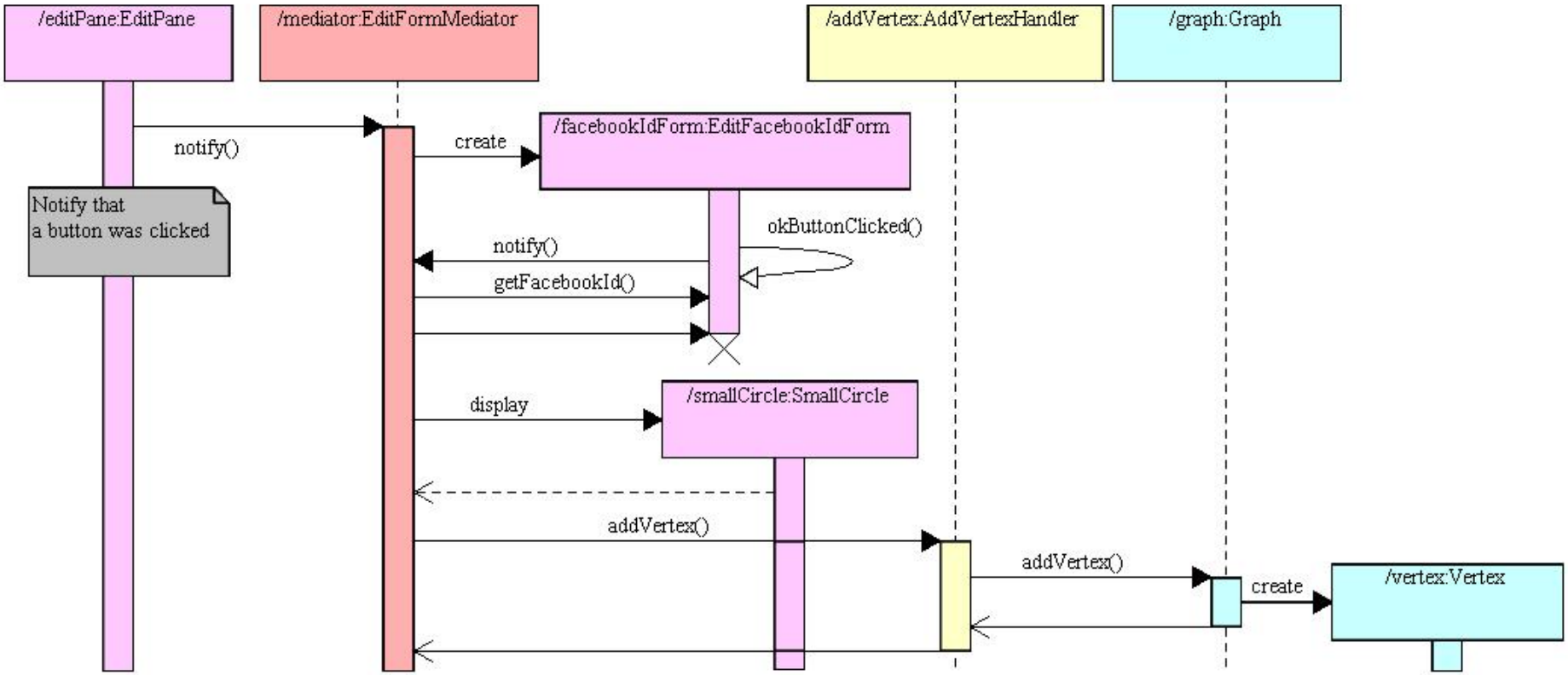
Επεξήγηση

- Τα παραλληλόγραμμα στην κορυφή δηλώνουν τους συμμετέχοντες (συνήθως αντικείμενα)
- Η γραμμή κάτω από κάθε παραλληλόγραμμο είναι γνωστή ως γραμμή ζωής (lifeline)
- Ένα παραλληλόγραμμο ενεργοποίησης δηλώνει απλώς ότι τη συγκεκριμένη στιγμή το αντικείμενο εκτελεί κάποια ενέργεια
- Ο κατακόρυφος άξονας είναι ο άξονας του χρόνου.
- Ο χρόνος είναι σημαντικός μόνο ως προς την αλληλουχία των γεγονότων εκτός και αν αναφέρεται κάτι διαφορετικό
- Οι οριζόντιες γραμμές αναπαριστούν μηνύματα

Ονομασία αντικειμένων

- Η μέθοδος ονομασίας των αντικειμένων που εμφανίζεται στα διαγράμματα ακολουθεί τη μορφή: `Όνομα αντικειμένου: όνομα κλάσης`
- Πολλές φορές μπορεί να αναγράψουμε μόνο το ένα από τα δύο, δηλ. `Όνομα αντικειμένου:` ή `:Όνομα κλάσης` (είτε `Όνομα κλάσης`)

2ο Παράδειγμα διαγράμματος ακολουθίας



Επεξήγηση

1. Το αντικείμενο `mediator` λαμβάνει ένα μήνυμα μέσω της `notify()` (Εναλλακτικά: το αντικείμενο `editPane` καλεί τη συνάρτηση `notify()` του `mediator`)
2. Το αντικείμενο `mediator` δημιουργεί ένα νέο αντικείμενο `facebookIdForm` της κλάσης `EditFacebookIdForm`
3. Το αντικείμενο `facebookIdForm` στέλνει ένα μήνυμα στον εαυτό του `okButtonClicked`
4. ...

Είδη μηνυμάτων

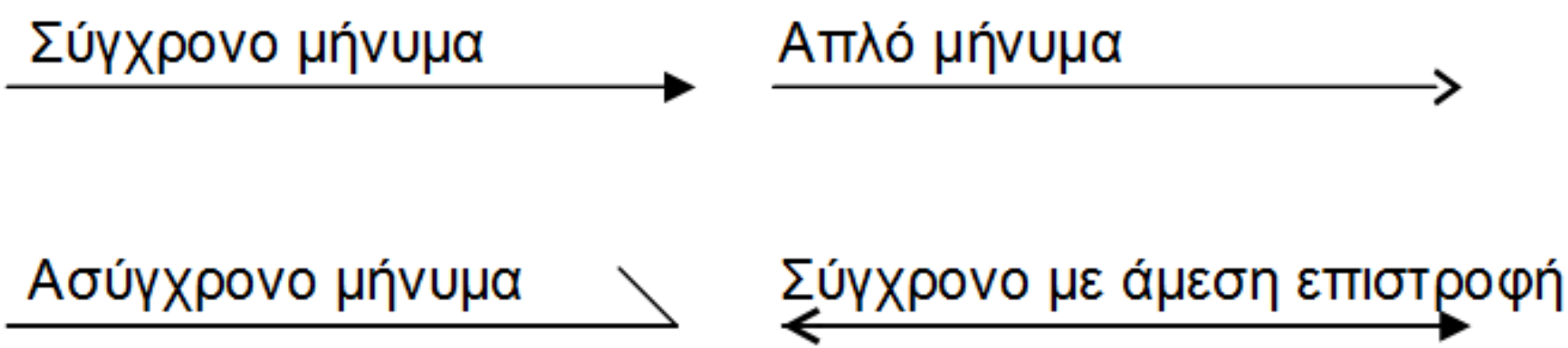
- Ένα μήνυμα στην πιο απλή του μορφή μπορεί να εννοηθεί ως η κλήση μιας συνάρτησης. Υπάρχουν όμως πολλά είδη μηνυμάτων που μπορούν να ανταλλαχθούν

Απλά Περιγράφουν πως ο έλεγχος περνάει από το ένα αντικείμενο στο άλλο, χωρίς λεπτομέρειες για την επικοινωνία

Σύγχρονα Υλοποιούνται σαν μια κλήση συνάρτησης.

Ασύγχρονα Ο αποστολέας συνεχίζει να εκτελεί τις υπόλοιπες λειτουργίες του αφού στείλει το μήνυμα , χωρίς να περιμένει για την διαχείρισή του.

Αναπαράσταση μηνυμάτων



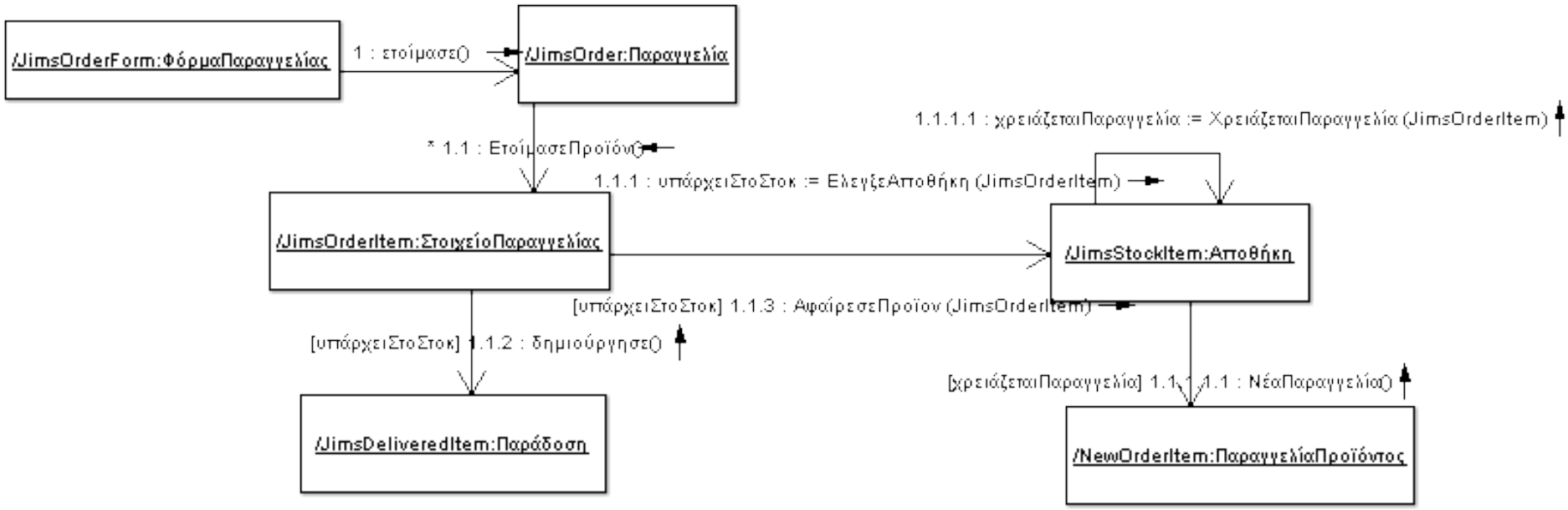
Εναλλακτικές ροές

- Στα διαγράμματα ακολουθίας προσπαθήστε να δείξετε μία συγκεκριμένη περίπτωση φορά
- Αν υπάρχουν πολλές εναλλακτικές ροές κάντε και πολλά διαγράμματα

Διαγράμματα επικοινωνίας

- Σε ένα διάγραμμα επικοινωνίας τα αντικείμενα απεικονίζονται ως εικονίδια
- Όπως και σε ένα διάγραμμα ακολουθίας τα βέλη απεικονίζουν αποστολές μηνυμάτων στο πλαίσιο της αναπαράστασης μιας συγκεκριμένης λειτουργίας

Παράδειγμα διαγράμματος επικοινωνίας



Επεξήγηση

- Το αντικείμενο `JimsOrderForm` στέλνει ένα μήνυμα `ετοίμασε()` στο αντικείμενο `JimsOrder`
- Για κάθε στοιχείο παραγγελίας, το αντικείμενο `JimsOrder` στέλνει ένα μήνυμα `ετοίμασεΠροϊον()` στο αντικείμενο `JimsOrderItem`
- Το αντικείμενο `JimsOrderItem` στέλνει ένα μήνυμα `έλεγξεΑποθήκη()` στο αντικείμενο `JimsStockITem` και αποθηκεύει το αποτέλεσμα στη μεταβλητή `ΥπάρχειΣτοΣτοκ`
- Το αντικείμενο `JimsStockITem` στέλνει ένα μήνυμα στον εαυτό του `χρειάζεταιΠαραγγελία()` και αποθηκεύει το αποτέλεσμα στη μεταβλητή `χρειάζεταιΠαραγγελία`
- Αν η μεταβλητή `χρειάζεταιΠαραγγελία` είναι αληθής τότε το αντικείμενο `AthensStock` στέλνει ένα μήνυμα νέο στο αντικείμενο `NewOrderItem` ...

Παρατηρήσεις

- Στα διαγράμματα επικοινωνίας η αρίθμηση των μηνυμάτων δείχνει την ακολουθία τους
- Αυτό μπορεί να κάνει πιο δύσκολη την παρακολούθηση της ακολουθίας σε σχέση με την παράθεση των μηνυμάτων από πάνω προς τα κάτω
- Όμως μπορείτε να δείξετε πιο εύκολα πως συνδέονται τα αντικείμενα μεταξύ τους