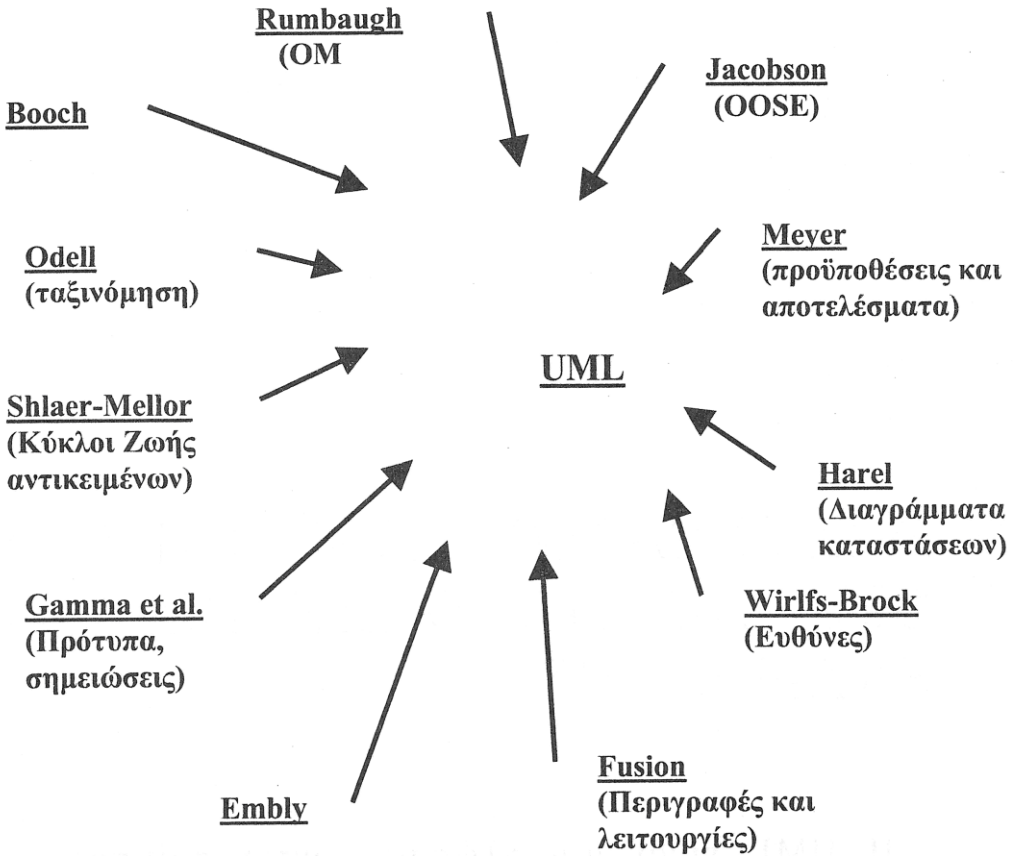
ΤΕΧΝΟΛΟΓΙΑ ΛΟΓΙΣΜΙΚΟΥ

Μαρία Βίρβου
Καθηγήτρια
mvirvou@unipi.gr
http://www.unipi.gr/faculty/mvirvou/

Η ΓΛΩΣΣΑ ΜΟΝΤΕΛΟΠΟΙΗΣΗΣ UML ΚΑΙ ΜΙΑ ΔΙΑΔΙΚΑΣΙΑ ΕΦΑΡΜΟΓΗΣ

- Η UML είναι μια γλώσσα μοντελοποίησης για ανάλυση και σχεδιασμό η οποία προέκυψε από επιρροές της μεθοδολογίας Booch, της OMT (Object Modeling Technique) και της OOSE (Object Oriented Software Engineering).
- Η UML παρέχει το **συμβολισμό** για ανάλυση και σχεδιασμό. Μερικά στοιχεία του συμβολισμού (π.χ. Τάξεις, συσχετισμοί, αθροίσματα, κληρονομικότητα) εισάγονται στην ανάλυση. Άλλα στοιχεία του συμβολισμού (π.χ. ιδιότητες) εισάγονται στο σχεδιασμό.
- Η UML είναι γλώσσα συμβολισμού. Δεν είναι ολόκληρη μεθοδολογία. Η γλώσσα UML αναπτύχθηκε από τους κύριους συντελεστές των μεθοδολογιών που προαναφέρθηκαν δηλ. από τον G. Booch, τον J. Rumbaugh και τον I. Jacobson. Όμως η γλώσσα UML περιέχει στοιχεία και από άλλες μεθοδολογίες όπως φαίνεται στο Σχήμα 3.1.

Η ΓΛΩΣΣΑ ΜΟΝΤΕΛΟΠΟΙΗΣΗΣ UML ΚΑΙ ΜΙΑ ΔΙΑΔΙΚΑΣΙΑ ΕΦΑΡΜΟΓΗΣ



Σχήμα 3.1 Στοιχεία που ενσωματώνει η UML

Σύντομη ιστορία της UML

- Η UML εκδόθηκε από μια σύμπραξη εταιριών της οποίας ηγήθηκε εταιρεία **Rational Software Corporation**.
- Η UML επιβεβαιώθηκε ως "industry standard" από την OMG (Object Management Group) το Νοέμβριο 1997.
- Η πρώτη δοκιμαστική έκδοση έγινε τον Οκτώβριο του 1995. Τον Ιούλιο και τον Οκτώβριο 1996 έγιναν οι δύο επόμενες εκδόσεις οι οποίες ενσωμάτωναν τα σχόλια του Jacobson και του κοινού.
- Τέλος το Σεπτέμβρη 1997 έγινε η έκδοση 1.1.

Τι ορίζει η UML

- Οι Booch, Rumbaugh και Jacobson (οι «τρεις» amigos, όπως αποκαλούνται στην κοινότητα της Τεχνολογίας Λογισμικού) λένε ρητώς ότι: Η UML είναι **γλώσσα** μοντελοποίησης και όχι μεθοδολογία.

- Η μεθοδολογία αποτελείται από μια **γλώσσα μοντελοποίησης** και μια **διαδικασία**. Η UML ορίζει μόνο τη γλώσσα μοντελοποίησης και όχι τη διαδικασία. Η γλώσσα μοντελοποίησης βοηθάει στην **περιγραφή του σχεδιασμού**. Η διαδικασία ορίζει τον **τρόπο δημιουργίας του σχεδιασμού.**

- Πολλοί λένε ότι η ενοποιημένη γλώσσα μοντελοποίησης (UML) σήμανε το τέλος στους **πολέμους μοντελοποίησης** αλλά σήμανε και την αρχή στους **πολέμους για τη διαδικασία**. Οι ίδιοι οι συγγραφείς της UML εκφράζουν την πεποίθηση ότι δεν υπάρχει μία και μοναδική διαδικασία. Πάντως οι συγγραφείς της UML έχουν ορίσει μια διαδικασία η οποία είναι ανεξάρτητη από τη UML. Η διαδικασία αυτή λέγεται Rational Unified Process και αναλύεται παρακάτω στο κεφάλαιο αυτό. Όμως μια **γλώσσα μοντελοποίησης** δεν είναι μεθοδολογία γιατί δεν ορίζει από μόνη της τη διαδικασία.

Τι ορίζει η UML

- Η UML ορίζει δύο εργαλεία:
  - Ένα συμβολισμό.
  - Ένα μεταμοντέλο.
- Ο συμβολισμός ορίζει τα διαγράμματα της UML. Το μεταμοντέλο είναι ο ορισμός του συμβολισμού (Liberty J., 1998). Π.χ. μπορούμε να χρησιμοποιήσουμε το συμβολισμό της UML για να ορίσουμε τον ίδιο συμβολισμό.

Κριτική της UML

- Οπωσδήποτε η UML θα αλλάξει λίγο καθώς θα χρησιμοποιείται σαν ένα standard στη βιομηχανία αλλά οι αλλαγές δεν αναμένεται να είναι πολλές. Οι περισσότεροι αναλυτές αντικειμένων πιστεύουν ότι μπορούν άνετα να χρησιμοποιήσουν την UML ως γλώσσα μοντελοποίησης.

- Καθώς γράφονται βιβλία για την UML και παράγονται εργαλεία που την υποστηρίζουν θα γίνεται πιο κατανοητή και εύχρηστη. Ήδη η διεθνής βιβλιογραφία έχει εμπλουτιστεί πολύ. Όμως υπάρχουν μερικές απόψεις του μεταμοντέλου που δεν έχουν ξεκαθαρίσει ακόμα. Οι εμπειρογνώμονες διαφωνούν για μερικές από τις λεπτές διαφορές.

Η διαδικασία Rational Unified Process

- Η διαδικασία Rational Unified Process είναι η διαδικασία που προτείνουν ο Booch, Rumbaugh και Jacobson για την ανάπτυξη λογισμικού.
- Ο κύκλος ζωής λογισμικού προτείνεται να είναι επαναληπτικός. Η ανάπτυξη δηλαδή να προχωρεί σε μια σειρά επαναλήψεων μέχρι να εξελιχθεί το τελικό προϊόν.
- Η διαδικασία Rational Unified Process αποτελείται από ένα σύνολο οδηγιών σχετικά με τις τεχνικές και οργανωτικές απόψεις της ανάπτυξης λογισμικού. Η διαδικασία αυτή αφορά κυρίως στην Ανάλυση Απαιτήσεων και στο Σχεδιασμό.
- Ο κύκλος ζωής λογισμικού όπως προτείνεται από τη Rational Unified Process φαίνεται στο Σχήμα 3.2.
- Η διαδικασία Rational Unified Process είναι δομημένη σε δύο διαστάσεις:

1) **Χρόνο - Χωρισμός του κύκλου ζωής σε φάσεις και επαναλήψεις.**
2) **Τμήματα διαδικασίας - Καλά ορισμένες εργασίες.**

Η διαδικασία Rational Unified Process

- Η δόμηση ενός έργου σε σχέση με το **χρόνο** ακολουθεί τις εξής φάσεις που έχουν σχέση με το χρόνο:

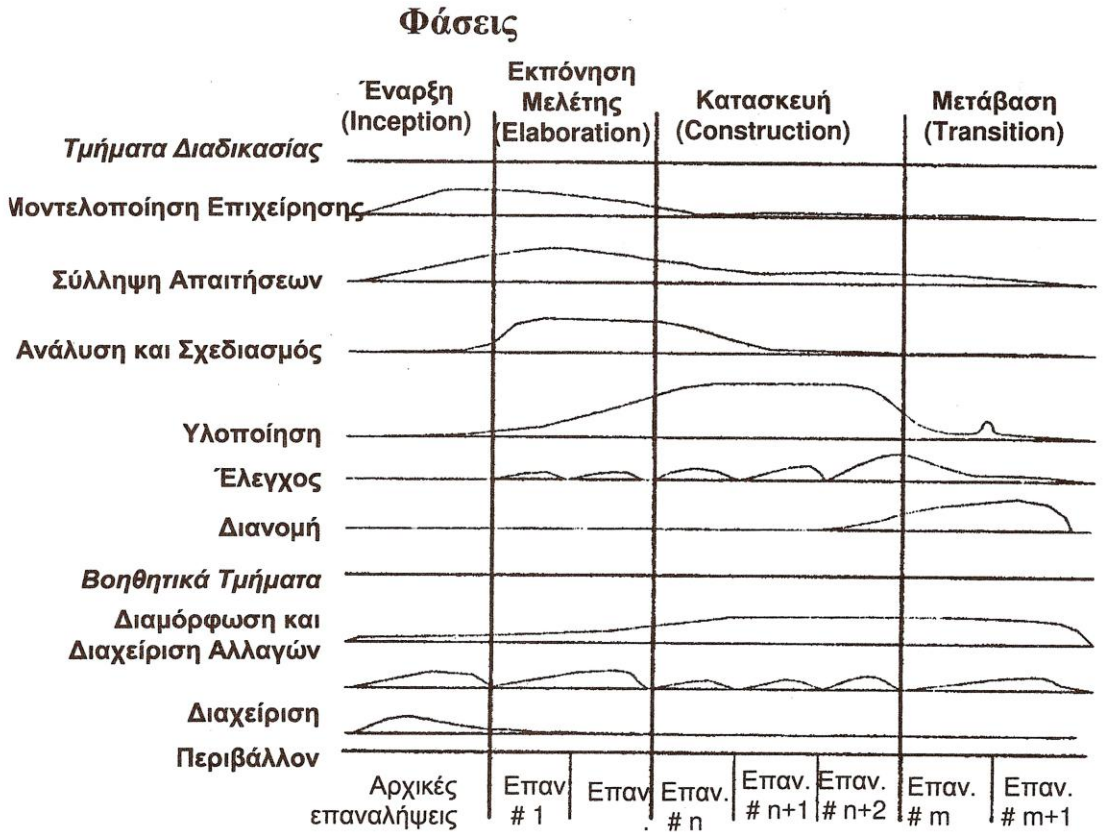
1) **Έναρξη (Inception):**
   Καθορίζει την προοπτική του έργου.
2) **Εκπόνηση μελέτης (Elaboration):**
   Σχεδιασμός των απαιτούμενων δραστηριοτήτων και πόρων. Καθορισμός των χαρακτηριστικών και σχεδιασμός της αρχιτεκτονικής.
3) **Κατασκευή (Construction):**
   Ανάπτυξη του προϊόντος σε μια σειρά βηματικών επαναλήψεων.
4) **Μετάβαση (Transition):**
   Προμήθευση του προϊόντος στην κοινότητα χρηστών (παραγωγή, διανομή, εκπαίδευση).

Η διαδικασία Rational Unified Process

- Η δόμηση έργου σύμφωνα με τη **διάσταση** των τμημάτων διαδικασίας περιλαμβάνει τις ακόλουθες δραστηριότητες:

1) **Σύλληψη απαιτήσεων (Requirements capture):**
   Μια αφήγηση του τι πρέπει να κάνει το σύστημα.
2) **Ανάλυση και σχεδιασμός (Analysis and design):**
   Μια περιγραφή του πως θα υλοποιηθεί το σύστημα.
3) **Υλοποίηση (Implementation):**
   Η παραγωγή του κώδικα.
4) **Έλεγχος (Test):**
   Η επαλήθευση του συστήματος.

Η διαδικασία Rational Unified Process



Σχήμα 3.2 Κύκλος Ζωής Ανάπτυξης Λογισμικού

Τάξεις και αντικείμενα (Παράδειγμα)

- Τα χαρακτηριστικά για κάθε ράτσα αποθηκεύονται στο ίδιο επίπεδο με τη ράτσα αλλά γενικές πληροφορίες για σκύλους υπάρχουν σε υψηλότερο επίπεδο απ' όπου **«κληρονομούνται»** από κάθε ράτσα.

Ερωτήσεις όπως:

1. **Μπορεί ένα μαλτεζάκι να αναπνέει;** ή
2. **Είναι η Λάσι τετράποδο;**

μπορούν να απαντηθούν ανατρέχοντας στις ιδιότητες του ζώου ή του σκύλου που βρίσκονται πιο ψηλά στην ιεραρχία "είναι".

Ο συμβολισμός UML

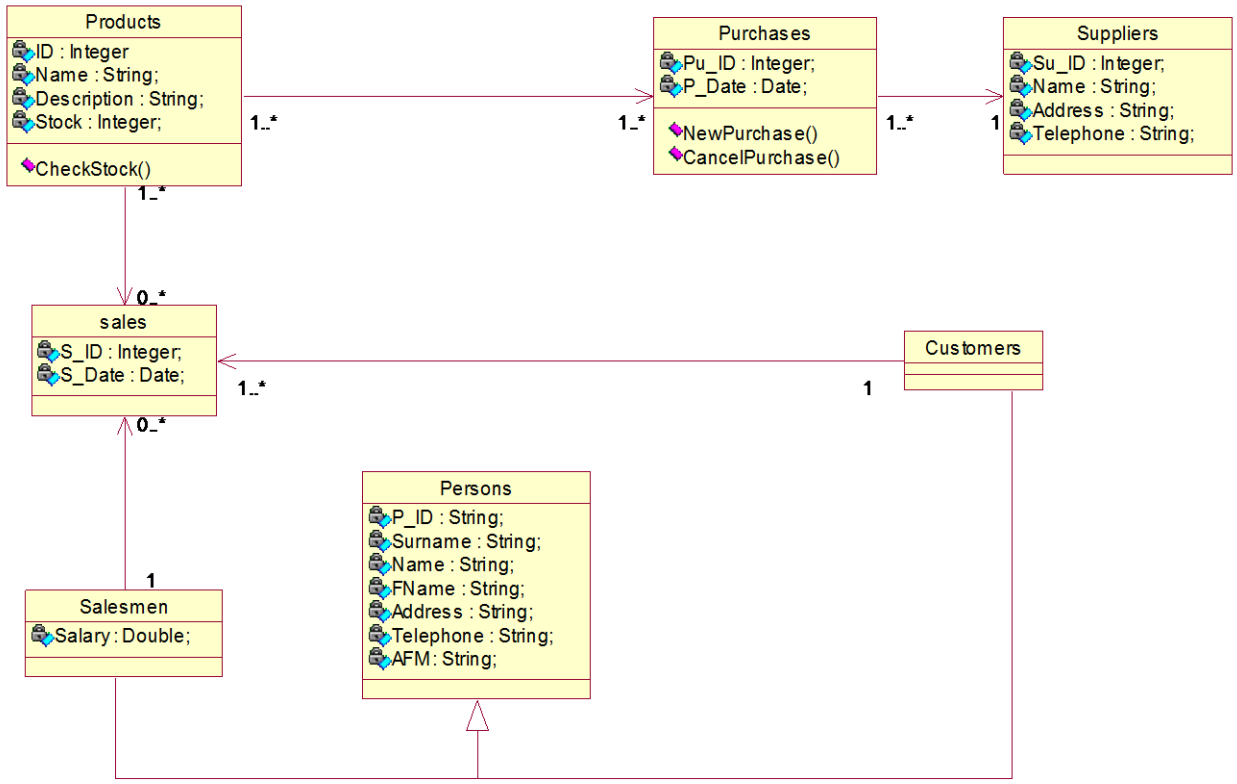
- Η UML ορίζει 9 είδη διαγραμμάτων για να αναπαραστήσει τις διαφορετικές απόψεις μοντελοποίησης:
  1) **Διαγράμματα τάξεων (Class Diagrams):**
     Αναπαριστούν τη στατική δομή όσον αφορά στις τάξεις και τις σχέσεις τους.
  2) **Διαγράμματα αντικειμένων (Object Diagrams):**
     Αναπαριστούν αντικείμενα και τις σχέσεις τους και αντιστοιχούν σε απλοποιημένα διαγράμματα συνεργασίας που δεν αναπαριστούν μετάδοση μηνυμάτων.
  3) **Διαγράμματα συνεργασίας (Collaboration Diagrams):**
     Η αναπαράσταση των αντικειμένων, συνδέσεων και αλληλεπιδράσεων.
  4) **Διαγράμματα σειράς (Sequence diagrams):**
     Χρονική αναπαράσταση των αντικειμένων και των αλληλεπιδράσεών τους.

- Τα **διαγράμματα συνεργασίας** και **διαγράμματα σειράς** μπορούν να ομαδοποιηθούν κάτω από τον τίτλο **διαγράμματα αλληλεπίδρασης**.

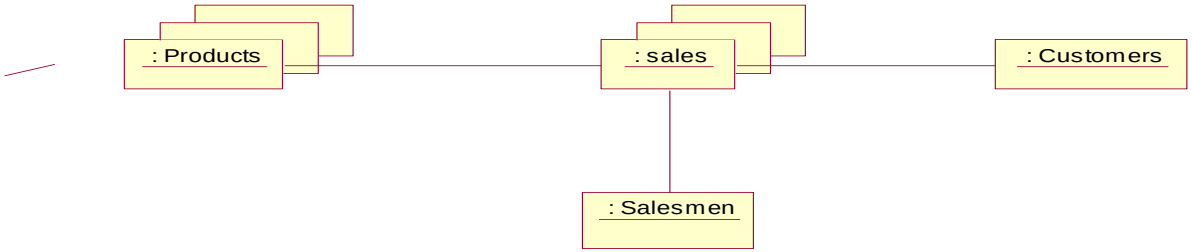
Ο συμβολισμός UML

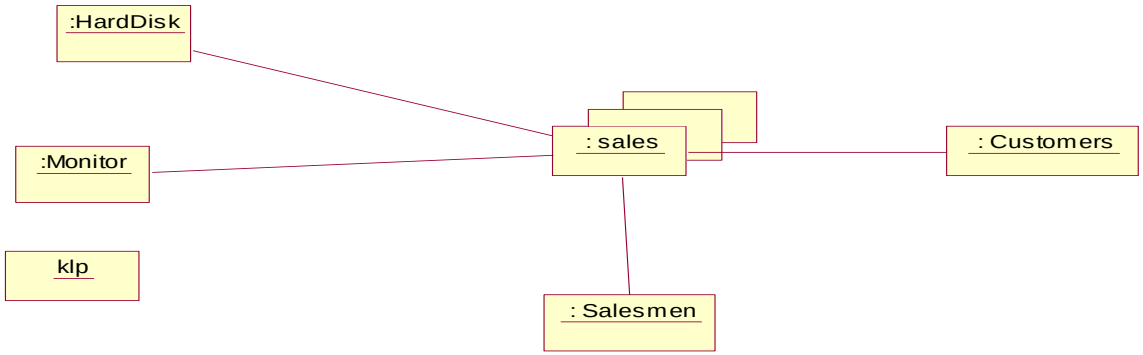
5) **Διαγράμματα καταστάσεων (Statechart diagrams):**
   Αναπαριστούν τη συμπεριφορά της τάξης όσον αφορά στις καταστάσεις της.
6) **Διαγράμματα δραστηριοτήτων (Activity diagrams):**
   Αναπαριστούν τη συμπεριφορά μιας λειτουργίας ως σύνολο ενεργειών.
7) **Διαγράμματα εξαρτημάτων (Component diagrams):**
   Αναπαριστούν τα φυσικά εξαρτήματα μιας εφαρμογής.
8) **Διαγράμματα διανομής (Deployment diagrams):**
   Αναπαριστούν τη διανομή των εξαρτημάτων σε συγκεκριμένα τεμάχια του hardware (υλικού).
9) **Διαγράμματα περιπτώσεων χρήσης (Use case diagrams):**
   Αναπαριστούν τις λειτουργίες ενός συστήματος από την οπτική γωνία του χρήστη.

Διαγράμματα τάξεων (Class Diagrams)

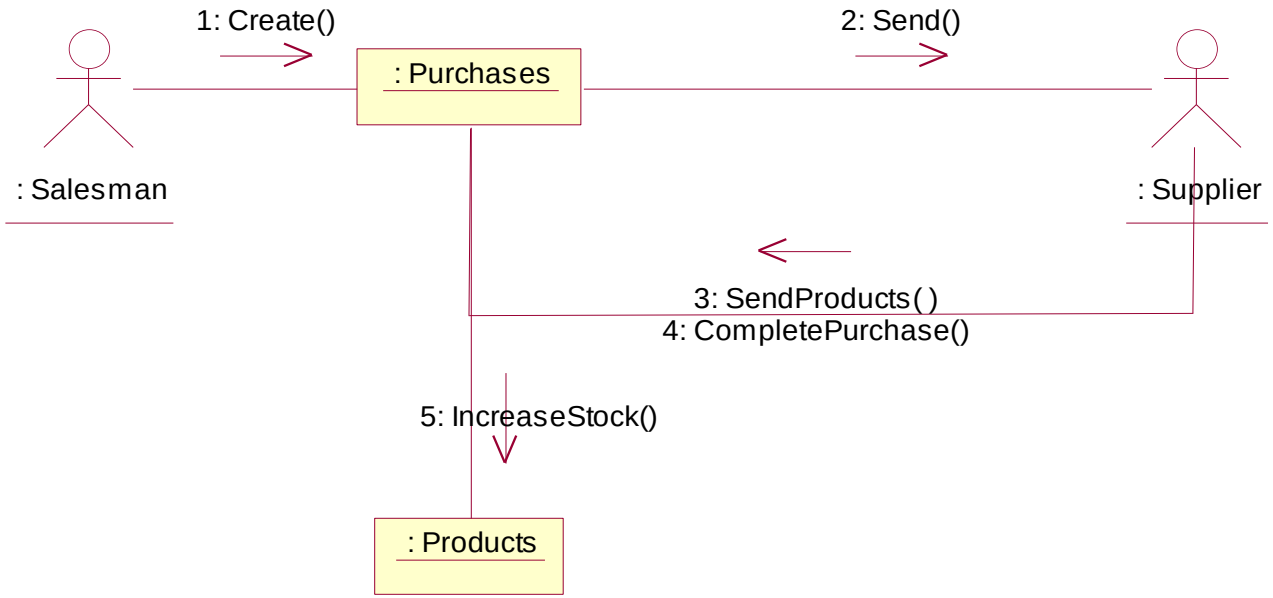


Διαγράμματα αντικειμένων (Object Diagrams)

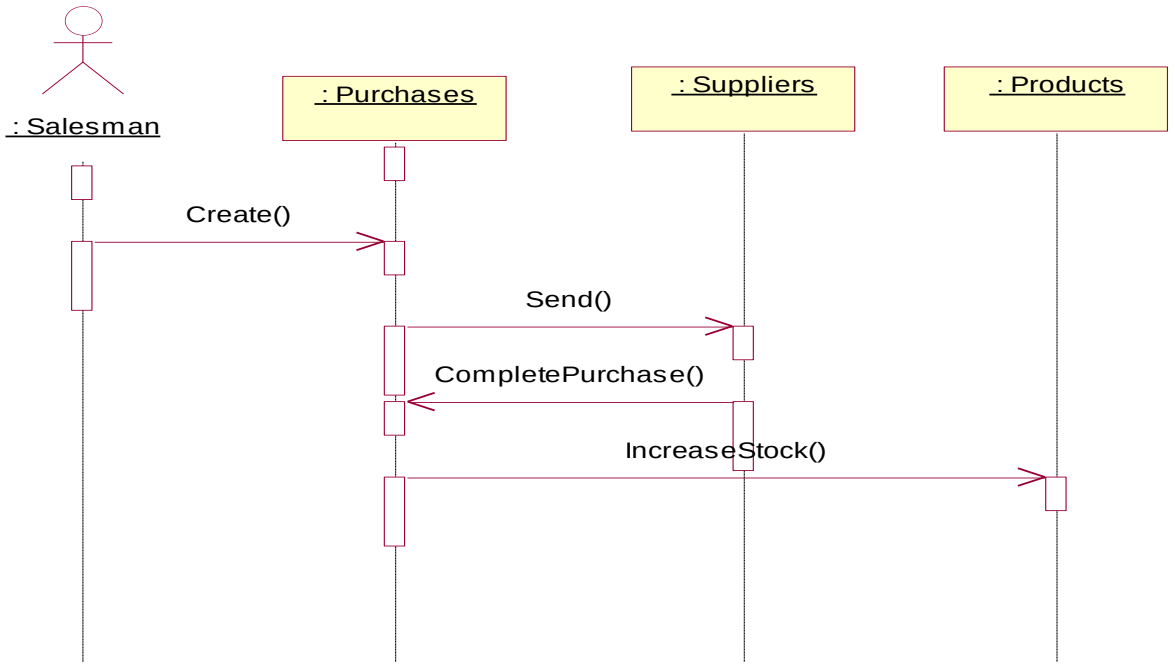




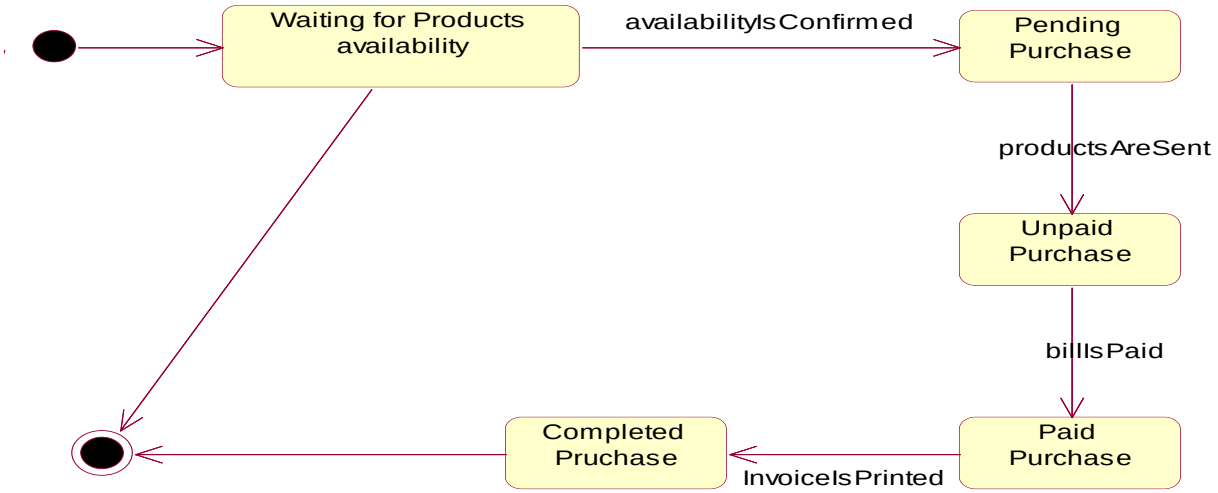
Διαγράμματα συνεργασίας (Collaboration Diagrams)



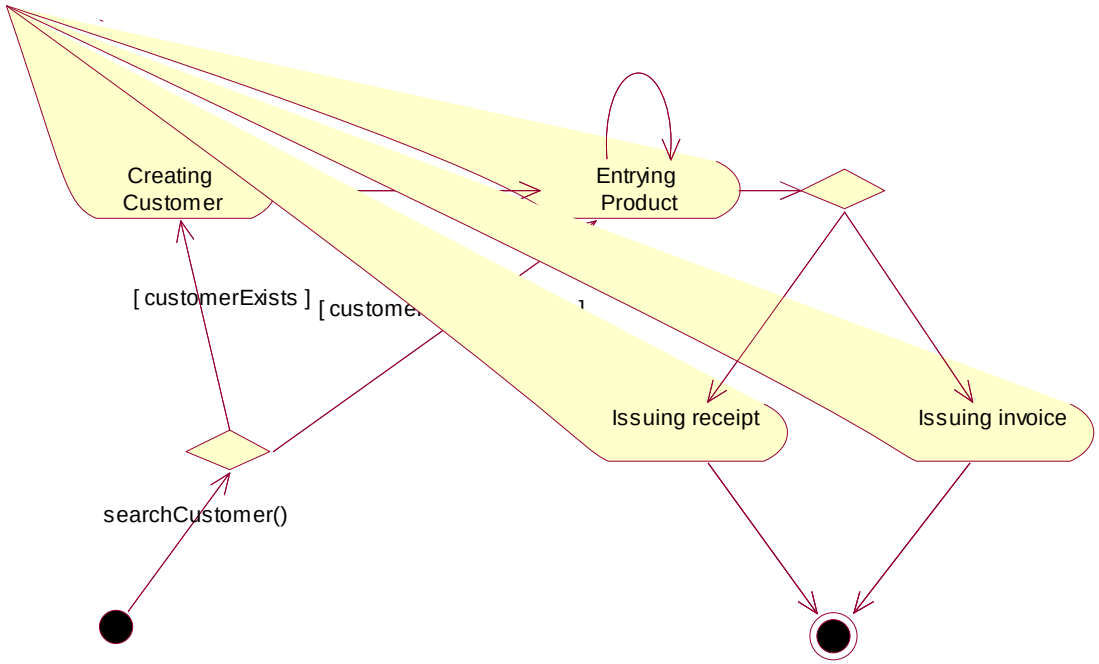
Διαγράμματα σειράς (Sequence diagrams)



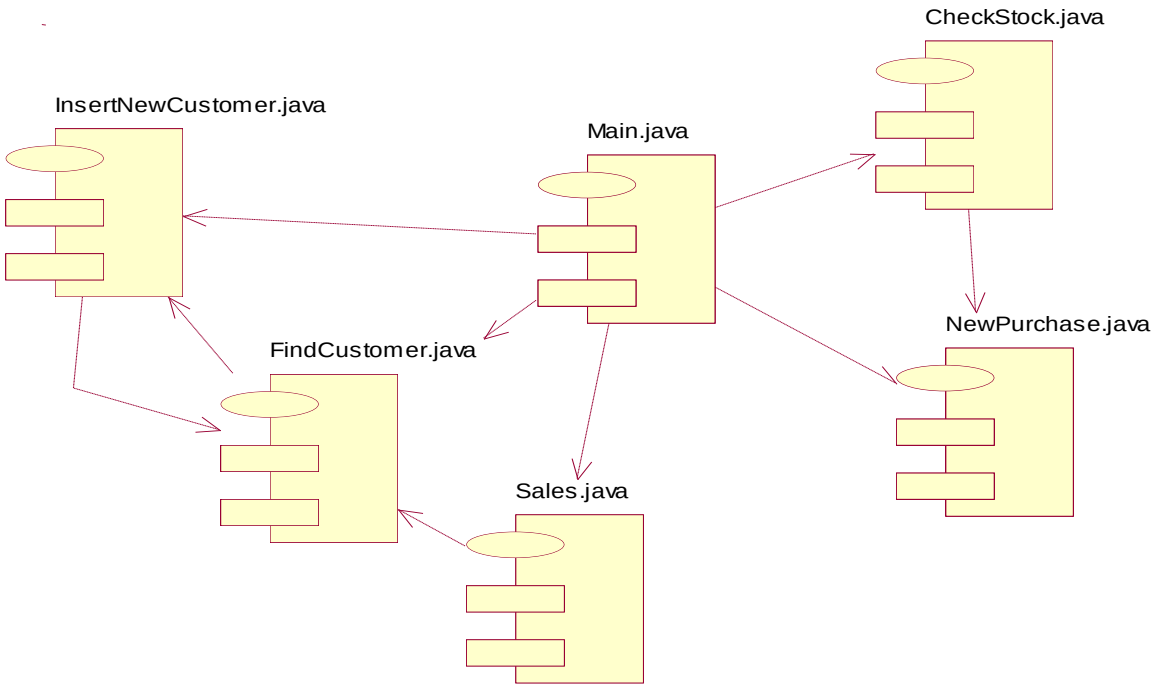
Διαγράμματα καταστάσεων (Statechart diagrams)



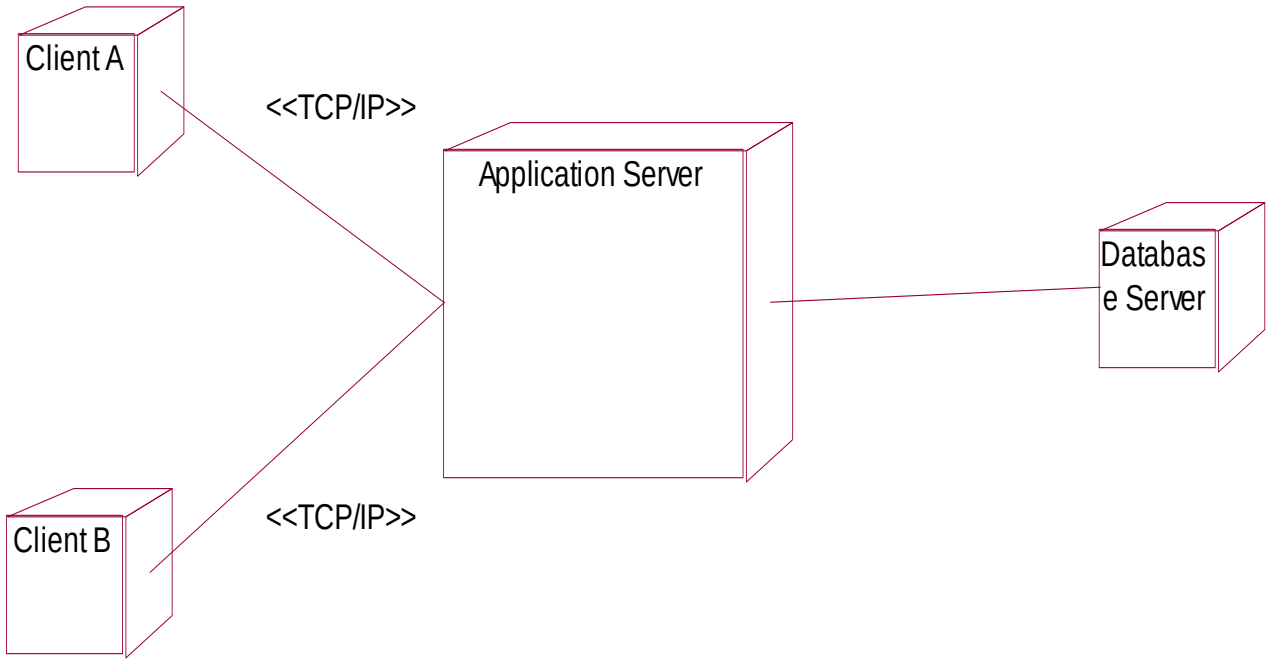
Διαγράμματα δραστηριοτήτων (Activity diagrams)



Διαγράμματα εξαρτημάτων (Component diagrams)



Διαγράμματα διανομής (Deployment diagrams)



Διαγράμματα περιπτώσεων χρήσης (Use case diagrams)