

Εισαγωγή στη UML

Περιεχόμενα

- Γενικά χαρακτηριστικά
- Στοιχεία μοντέλων
- Συσχετίσεις
- Διαγράμματα
- Παραδείγματα

Τι είναι μοντέλο?

- Αποτελεί μια *αφηρημένη περιγραφή* ενός φυσικού συστήματος.
- Αποτελεί ένα σχέδιο για την *κατασκευή* ενός συστήματος.
- Βοηθάει στην *κατανόηση* ενός μεγάλου μεγέθους συστήματος.
- Βοηθάει στην *επικοινωνία* των μελών της ομάδας που αναπτύσσει το σύστημα.

Τι είναι η UML?

Η UML είναι μια οπτική αντικειμενοστρεφής γλώσσα μοντελοποίησης που χρησιμοποιείται για:

- Απεικόνιση (visualization)
- Προδιαγραφή (specification)
- Τεκμηρίωση (documentation)
- Κατασκευή (construction)

των δομικών συστατικών ενός συστήματος (λογισμικού ή όχι).

Χαρακτηριστικά της UML (I)

- Είναι ιδιαίτερα εκφραστική.
- Σημασιολογικά, είναι εκτενής.
 - Υποστηρίζει τη σημασιολογία τύπων και μοντέλων για όλα τα μοντέλα ενός συστήματος.
- Σε επίπεδο βασικών αρχών, είναι “μικρή” και απλή.
 - Διακρίνουμε πέντε βασικούς άξονες.
- Είναι επεκτάσιμη.
 - Υπάρχει η δυνατότητα εμπλουτισμού του μετα-μοντέλου με κλάσεις, ιδιότητες και σημασιολογία.
- Είναι επακριβώς ορισμένη με βάση τα δομικά συστατικά ενός αντικειμενοστρεφούς συστήματος.
- Προήλθε από την ενοποίηση των συμβολισμών που χρησιμοποιούσαν οι μεθοδολογίες Booch, OMT, OOSE κ.λπ.

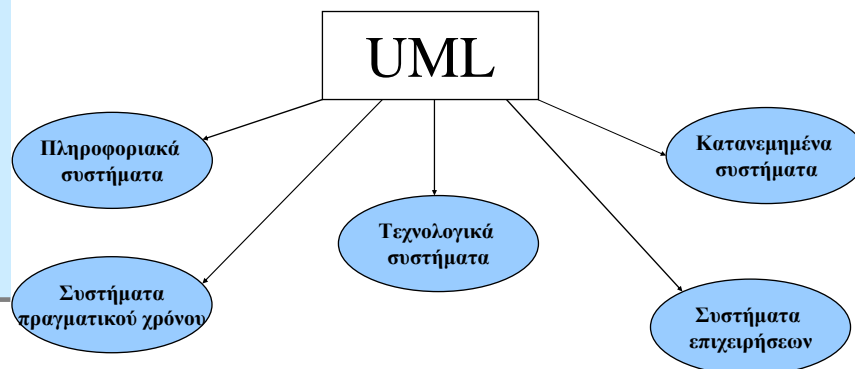
Χαρακτηριστικά της UML (II)

- Ενσωματώνει τις ιδέες “καλής πρακτικής” από τη βιομηχανία λογισμικού.
- Είναι σήμερα βιομηχανικό πρότυπο και αναπτύχθηκε στο πλαίσιο του οργανισμού OMG.
- Υλοποιεί την ανάγκη της βιομηχανίας λογισμικού για μια ενιαία γλώσσα μοντελοποίησης.
- Αντιμετώπιση σημερινών και βραχυπρόθεσμων προβλημάτων στην ανάπτυξη λογισμικού:
 - Κλίμακα
 - Γλώσσες: Java, C++, Smalltalk, Ada, Visual Basic
 - Πολυεπεξεργασία και παραλληλία
 - Πρότυπα Λογισμικού (Patterns)
 - Ψηφίδες λογισμικού (Componentware)
 - Μοντελοποίηση επιχειρησιακής πρακτικής

Είδη συστημάτων που μοντελοποιούνται με τη UML

- Συστήματα με έμφαση στο λογισμικό (software-intensive systems)
- Επιχειρησιακά συστήματα (business systems)
- Συστήματα που δεν περιέχουν λογισμικό (non-software systems)

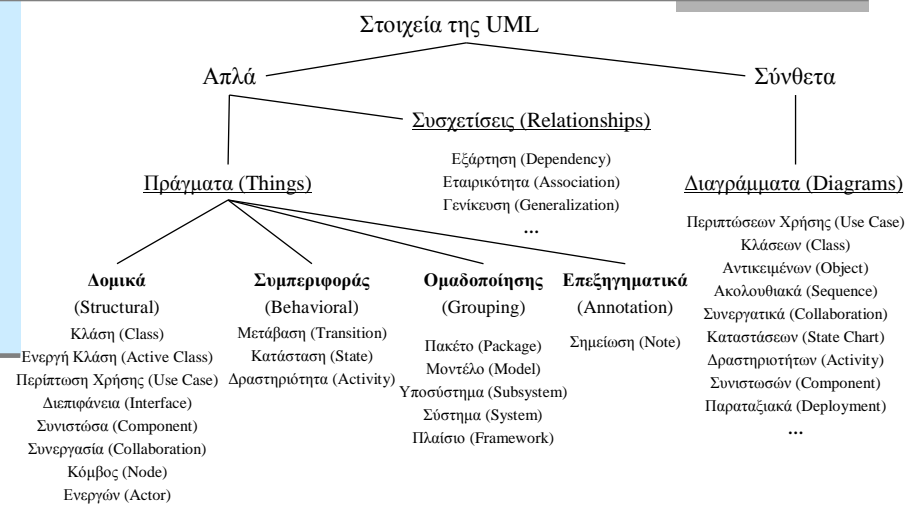
Χρήση της UML



Οι πέντε βασικοί άξονες της UML

- Στοιχεία του μοντέλου (model elements)
- Συσχετίσεις (relationships)
- Μηχανισμοί (mechanisms)
- Διαγράμματα (diagrams)
- Αρχιτεκτονικές όψεις (architectural views)

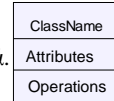
Μια κατηγοριοποίηση των στοιχείων της UML



Δομικά στοιχεία της UML (I)

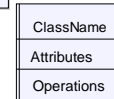
- Κλάση (class)

- Ένα σύνολο αντικειμένων με κοινή δομή και συμπεριφορά.



- Ενεργή κλάση (active class)

- Μια κλάση που περιγράφει μια διεργασία ή ένα νήμα εκτέλεσης και αλληλεπιδρά με άλλες.



- Περίπτωση χρήσης (use case)

- Μια λειτουργία που επιτελεί ένα σύστημα και είναι διαθέσιμη στο χρήστη. Είναι συμπεριφορά του συστήματος που συνεπάγεται τη συνεργασία ενός συνόλου αντικειμένων.



- Διεπιφάνεια (interface)

- Ένα σύνολο από λειτουργίες που ορίζουν την εξωτερική συμπεριφορά ενός αντικειμένου.



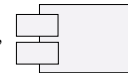
Γλώσσες Προδιαγραφής

11

Δομικά στοιχεία στη UML (II)

- Συνιστώσα (component)

- Ένα φυσικό και επαναρησιμοποιήσιμο τμήμα ενός συστήματος, με λογική και φυσική υπόσταση που συνήθως υλοποιεί κάποιες διεπιφάνειες (interfaces).



- Συνεργασία (collaboration)

- Η περιγραφή μιας διάδρασης μεταξύ ενός συνόλου αντικειμένων.



- Κόμβος (node)

- Ένας υπολογιστικός πόρος που έχει κάποια μνήμη και υπολογιστική ικανότητα, οπότε εκεί αποθηκεύεται ή/και εκτελείται το λογισμικό.



- Ενεργών (actor)

- Εξωτερική του συστήματος οντότητα που χρησιμοποιεί τη λειτουργικότητα και τις διεπιφάνειές του.



Γλώσσες Προδιαγραφής

12

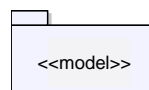
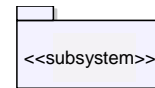
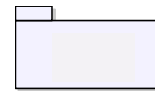
Στοιχεία συμπεριφοράς στη UML

- **Κατάσταση (state)**
 - Μια συνθήκη ή περίπτωση στο χρόνο ζωής ενός αντικειμένου, όπου ικανοποιεί κάποιους περιορισμούς, εκτελεί κάποια δραστηριότητα ή αναμένει κάποιο γεγονός.
- **Μετάβαση (transition)**
 - Μια σχέση μεταξύ δύο καταστάσεων ενός αντικειμένου που υποδηλώνει αλλαγή στην κατάσταση του αντικειμένου με την εμφάνιση ενός γεγονότος.
- **Δραστηριότητα (activity)**
 - Μια εκτέλεση λειτουργίας κατά τη διάρκεια ζωής ενός αντικείμενου.



Στοιχεία ομαδοποίησης στη UML

- **Πακέτο (package)**
 - Ένα δομικό στοιχείο γενικής χρήσης για την οργάνωση άλλων δομικών στοιχείων, διαγραμμάτων ή και άλλων πακέτων της UML σε ομάδες.
- **Υποσύστημα (subsystem)**
 - Μια μονάδα στην ιεραρχική αποσύνθεση ενός μεγάλου συστήματος. Επικοινωνεί με το περιβάλλον του μέσω διαπροσωπειών.
- **Μοντέλο (model)**
 - Μια όψη του συστήματος.



Επεξηγηματικά στοιχεία της UML

- Σημείωση (note)

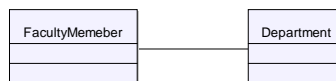
- Ένα δομικό στοιχείο κειμενικού σχολιασμού για την περιγραφή ή επεξήγηση ενός άλλου δομικού στοιχείου ή μιας ομάδας δομικών στοιχείων. Αποτελεί σχόλιο ή επεξήγηση ή κείμενο αναφοράς.



Συσχετίσεις στη UML (I)

- Συσχέτιση (Association)

- Μια δομική σχέση που περιγράφει ένα σύνολο συνδέσεων μεταξύ αντικειμένων.



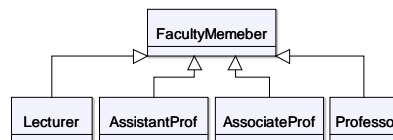
- Εξάρτηση (Dependency)

- Μια σχέση μεταξύ δυο δομικών στοιχείων, όπου μια αλλαγή στο πρώτο επιδρά στο δεύτερο.



- Γενίκευση (Generalization)

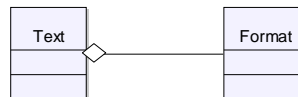
- Μια σχέση μεταξύ ενός δομικού στοιχείου (πατέρας) και ενός δεύτερου (παιδί) που εξειδικεύει το πρώτο.



Συσχετίσεις στη UML (II)

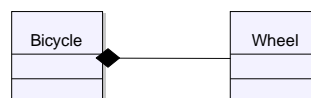
- **Συσσωμάτωση (Aggregation)**

- Μια σχέση μεταξύ δυο δομικών στοιχείων, όπου το πρώτο μπορεί να περιέχει το δεύτερο.



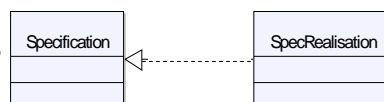
- **Σύνθεση (Composition)**

- Μια σχέση μεταξύ δυο δομικών στοιχείων, όπου το πρώτο εντάσσεται αναπόσπαστα στο δεύτερο και δεν μπορεί να ανήκει σε κανένα άλλο του ίδιου τύπου.



- **Πραγματοποίηση (Realization)**

- Μια σχέση μεταξύ δυο δομικών στοιχείων, όπου το πρώτο προδιαγράφει μια συμπεριφορά και το δεύτερο την υλοποιεί.



Τύποι διαγραμμάτων στη UML

- **Δομικά Διαγράμματα (Structural Diagrams):** Περιγράφουν την εσωτερική λογική δομή ενός συστήματος, δηλαδή τα συστατικά του και τις σχέσεις μεταξύ τους.

- Διάγραμμα Κλάσεων (**Class Diagram**)
- Διάγραμμα Αντικειμένων (**Object Diagram**)
- Διάγραμμα Συνιστώσων (**Component Diagram**)
- Παραταξιακό Διάγραμμα (**Deployment Diagram**)

- **Διαγράμματα Συμπεριφοράς (Behavior Diagrams):** Περιγράφουν τη δυναμική συμπεριφορά ενός συστήματος, δηλαδή την απόκρισή του σε γεγονότα του περιβάλλοντός του.

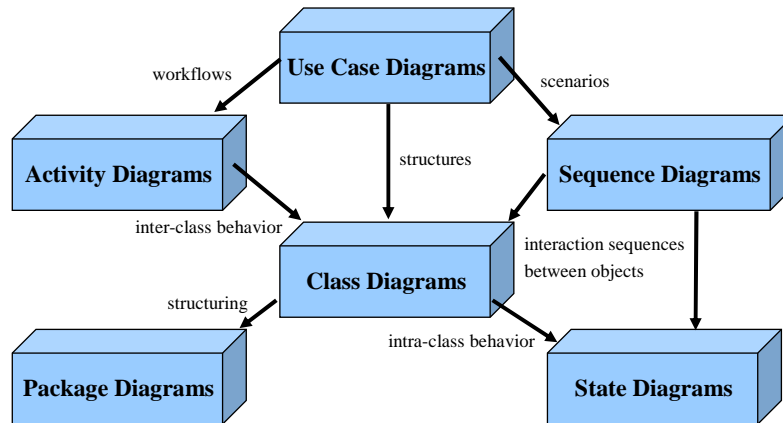
- Διάγραμμα Περιπτώσεων Χρήσης (**Use Case Diagram**)
- Διάγραμμα Αλληλουχίας (**Sequence Diagram**)
- Διάγραμμα Δραστηριοτήτων (**Activity Diagram**)
- Διάγραμμα Συνεργασίας (**Collaboration Diagram**)
- Διάγραμμα Καταστάσεων (**Statechart Diagram**)

- **Διαγράμματα Διαχείρισης Μοντέλου (Model Management Diagrams):**

Περιγράφουν τη φυσική δομή ενός συστήματος, δηλαδή τις μονάδες λογισμικού που το αποτελούν, σε όρους περιβάλλοντος υλοποίησης.

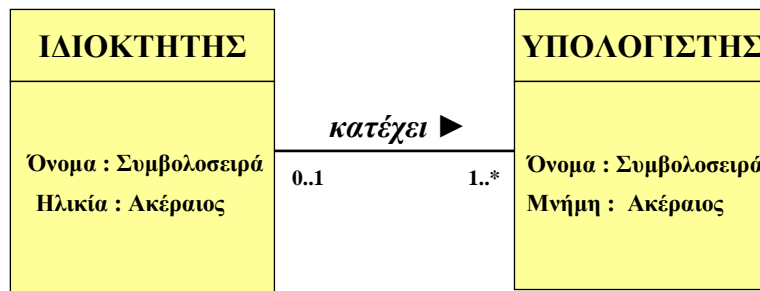
- Διάγραμμα Πακέτων (**Package Diagram**)
- Διάγραμμα Υποσυστημάτων (**Subsystem Diagram**)
- Διάγραμμα Μοντέλων (**Model Diagram**)

Σχέσεις διαγραμμάτων στη UML



Class Diagrams

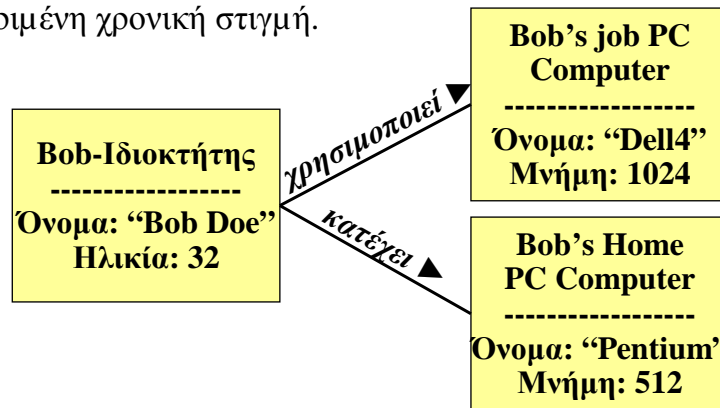
Τα *Class Diagrams* παρουσιάζουν κάποια στατικά στοιχεία μοντελοποίησης και τις σχέσεις μεταξύ τους.



Όπως έχει ήδη αναφερθεί, οι κλάσεις μπορεί να συσχετίζονται με σχέσεις εξάρτησης, συσχέτισης, γενίκευσης, συσσωμάτωσης, σύνθεσης, πραγματοποίησης, ...

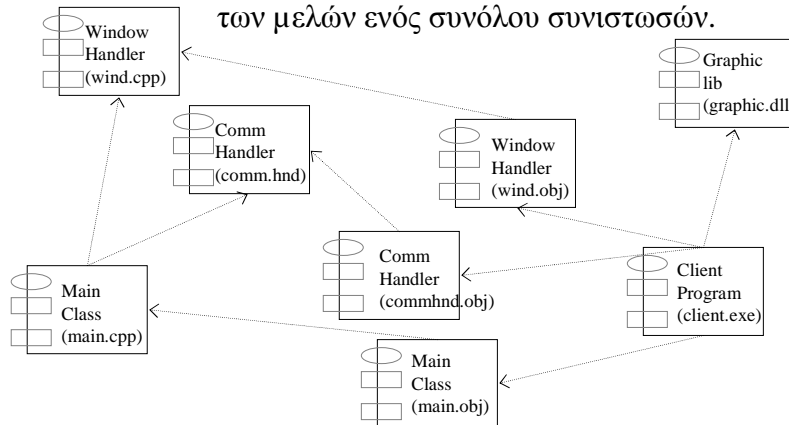
Object Diagrams

Τα *Object Diagrams* δίνουν παράδειγμα εμφάνισης στιγμιότυπων (instances) των κλάσεων και τις μεταξύ τους σχέσεις σε μια συγκεκριμένη χρονική στιγμή.



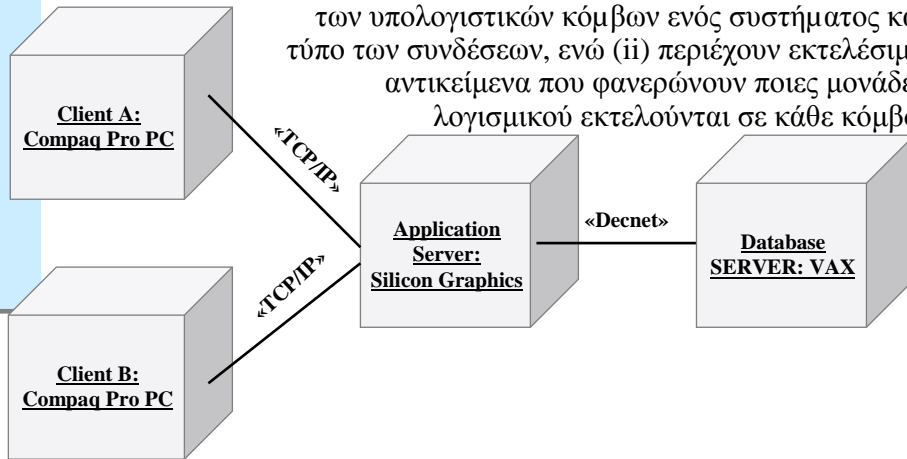
Component Diagrams

Τα *Component Diagrams* δείχνουν τα συστατικά μέρη του κώδικα και την φυσική τους δομή. Παρουσιάζουν την οργάνωση και τις εξαρτήσεις των μελών ενός συνόλου συνιστωσών.



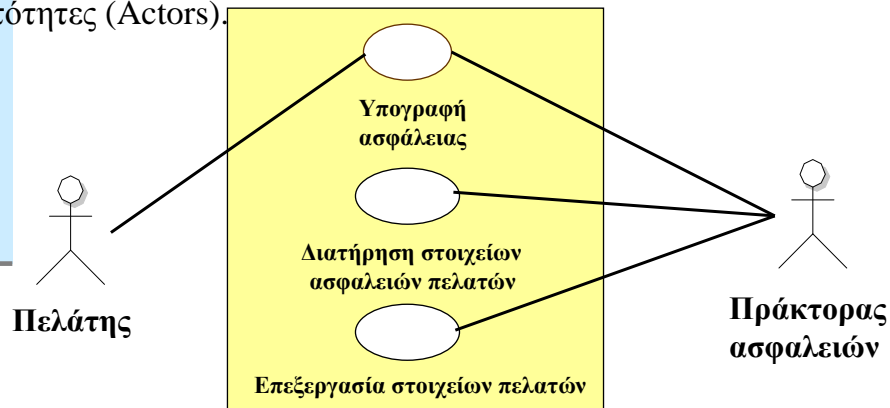
Deployment Diagrams

Τα *Deployment Diagrams* (i) παρουσιάζουν την τοπολογία των υπολογιστικών κόμβων ενός συστήματος και τύπο των συνδέσεων, ενώ (ii) περιέχουν εκτελέσιμα αντικείμενα που φανερώνουν ποιες μονάδες λογισμικού εκτελούνται σε κάθε κόμβο.



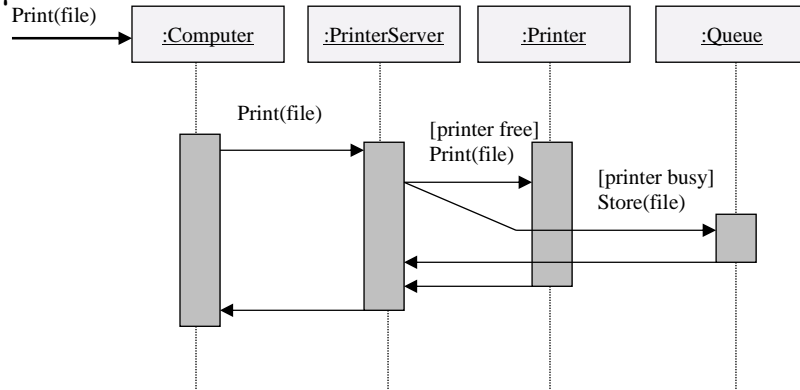
Use Case Diagrams

Τα *Use Case Diagrams* περιγράφουν τη λειτουργικότητα του συστήματος όπως αυτή γίνεται αντιληπτή από εξωτερικές οντότητες (Actors).



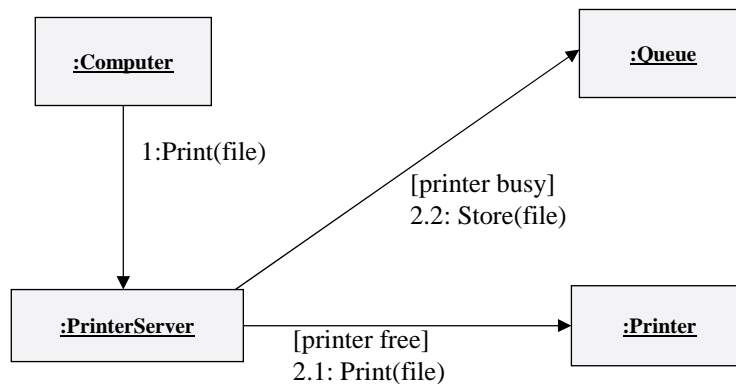
Sequence Diagrams

Τα *Sequence Diagrams* παρουσιάζουν μια αλληλεπίδραση αντικειμένων με έμφαση στην χρονική σειρά ανταλλαγής μηνυμάτων.



Collaboration Diagrams

Τα *Collaboration Diagrams* παρουσιάζουν μια αλληλεπίδραση αντικειμένων με έμφαση στη δομική τους οργάνωση.

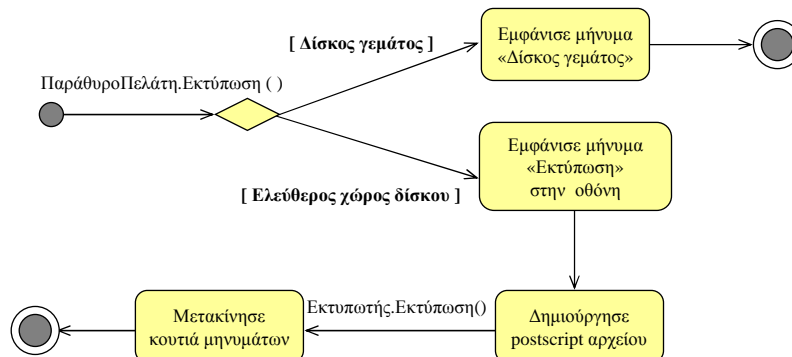


Διαφορές Sequence Diagrams & Collaboration Diagrams

- Sequence diagram : Δείχνει την ακολουθία μηνυμάτων ανάμεσα στα αντικείμενα, με την πάροδο του χρόνου.
- Collaboration diagram: Δίνει μεγαλύτερη έμφαση στο περιεχόμενο και στις σχέσεις μεταξύ των αντικειμένων.

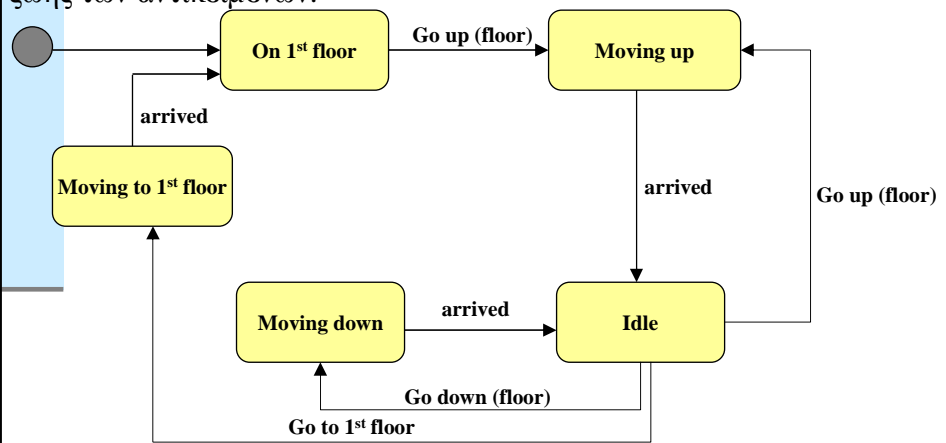
Activity Diagrams

Τα *Activity Diagrams* (i) παρουσιάζουν την ακολουθιακή ροή των δραστηριοτήτων και (ii) περιέχουν προσδιορισμούς των μηνυμάτων που στέλνονται.



Statechart Diagrams

Τα *Statechart Diagrams* παρουσιάζουν τις καταστάσεις του κύκλου ζωής των αντικειμένων.

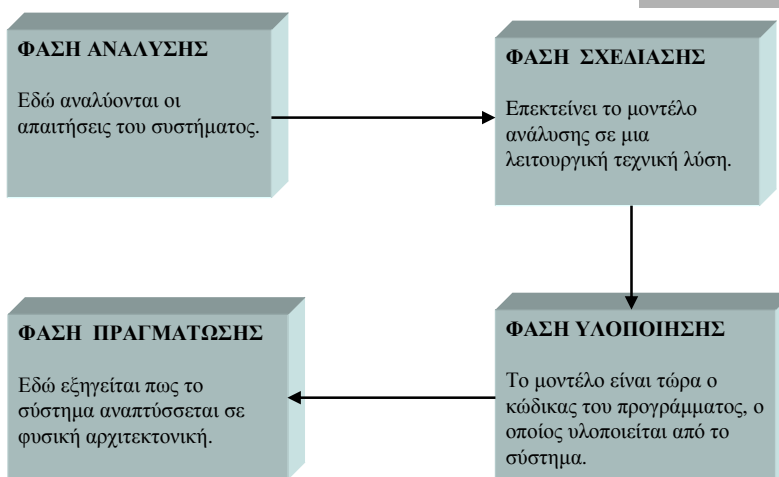


Γλώσσες Προδιαγραφής

Time out

29

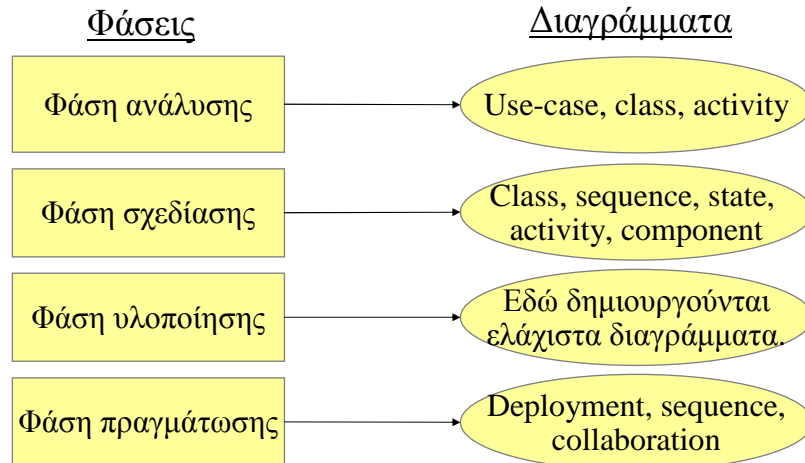
Μοντελοποιώντας με τη UML



Γλώσσες Προδιαγραφής

30

Αντιστοιχία διαγραμμάτων στις διάφορες φάσεις ανάπτυξης συστήματος



Γλώσσες Προδιαγραφής

31

Λειτουργίες των εργαλείων μοντελοποίησης (*Case Tools*) (I)

- ❖ Σχεδιασμός διαγραμμάτων που εξασφαλίζουν την σωστή χρήση των στοιχείων του μοντέλου.
- ❖ Αποθήκευση δεδομένων. Αν το όνομα μιας κλάσης αλλάζει σ' ένα διάγραμμα, η αλλαγή αυτή διαδίδεται και στα υπόλοιπα.
- ❖ Υποστήριξη διάδοσης ενός στοιχείου από ένα διάγραμμα σε ένα άλλο κι επέκταση της περιγραφής του στοιχείου.
- ❖ Υποστήριξη πολλών χρηστών. Δίνει την δυνατότητα να δουλεύουν πολλοί χρήστες ταυτόχρονα.
- ❖ Δημιουργία κώδικα

Γλώσσες Προδιαγραφής

32

Λειτουργίες των εργαλείων μοντελοποίησης (*Case Tools*) (II)

- ❖ **Αντίστροφη παραγωγή**, δηλαδή δυνατότητα παραγωγής μοντέλων από τον κώδικα (reverse engineering).
- ❖ **Συμβατότητα με άλλα εργαλεία** όπως editors, μεταγλωττιστές κι άλλα επιχειρησιακά εργαλεία.
- ❖ **Κάλυψη όλων των επιπέδων ανάπτυξης** του συστήματος, από το επίπεδο περιγραφής του συστήματος στο επίπεδο κώδικα.
- ❖ **Επικοινωνία ανάμεσα στα μοντέλα**. Ένα διάγραμμα σε κάποιο μοντέλο θα πρέπει να έχει τη δυνατότητα να εισάγεται από το ένα μοντέλο στο άλλο.