



Ανάλυση Πληροφοριακών Συστημάτων

«Εισαγωγή στην UML»

Βασίλειος Καρακίδας

26 – 10 – 2005



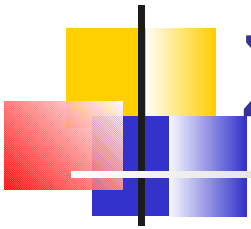
Περιεχόμενα Παρουσίασης

- Εισαγωγικά
- Προαπαιτούμενα
- Πρόγραμμα διαλέξεων
- Εργασίες
- Λογισμικό
- Εισαγωγή στην UML
- Βιβλιογραφία



Εν αρχή ...

- Βασίλειος Καρακίδας
 - e-mail: bkarak@aub.gr
 - Εκπαιδευτικό εργαστήριο ΔΕΤ 2 – 3^{ος} Δερισνύ
 - Τηλέφωνο: +210 8203 370
 - Ιστότοπος:
<http://istlab.dmst.aub.gr/~bkarak/>



Σχετικά με το φροντιστήριο

- Κάθε Τετάρτη, 17:00 – 19:00
- Ωρες Γραφείου: **θα ανακοινωθούν**
- <http://istlab.dmst.aueb.gr/~bkararak/lectures/diis-05.html>
- <http://eduportal.dmst.aueb.gr/>



Προαπαιτούμενα

- Αντικειμενοστρεφή Ανάλυση και Σχεδιασμό
- Java ή οποιαδήποτε αντικειμενοστρεφή γλώσσα προγραμματισμού



Τι θα μάθετε ...

- Τι είναι η UML
- Τι κάνει και τι δεν κάνει η UML
- Τα βασικά διαγράμματα της UML
- Βασικές αρχές ΟΟΑ



Προγραμμα Διαλέξεων

- 26-10-2005 17:00 - 19:00,
Εισαγωγική διάλεξη
- 02-11-2005 17:00 - 19:00,
Διαγράμματα Use Case, Μελέτη αποτελεσμάτων έργου Gemini
- 09-11-2005 17:00 - 19:00, Διαγράμματα Activity
- 16-11-2005 17:00 - 19:00, Αργία
- 23-11-2005 17:00 - 19:00, Παρουσίαση προόδου εργασίας
- 30-11-2005 17:00 - 19:00,
Βασικές Έννοιες Αντικειμενοστρεφούς Προγραμματισμού (OOP), Διαγράμματα κλάσεων (Μέρος Α')



Προγραμμα Διαλέξεων (2)

- 07-12-2005 17:00 - 19:00,
Διαγράμματα κλάσεων (Μέρος Β')
- 14-12-2005 17:00 - 19:00,
Διαγράμματα Sequence και Collaboration
- 21-12-2005 17:00 - 19:00,
Διάγραμμα Deployment, Μελέτη ολοκληρωμένου παραδείγματος
- 11-01-2006 17:00 - 19:00,
Επαναληπτικό Μάθημα



Εργασίες

- 1 παραδοτέο UML – 20% συνολικού βαθμού
- Προδιαγραφές και ανάλυση ενός οργανισμού με την χρήση της UML
- 11 θέματα διαθέσιμα



Εργασίες / Θέματα (1)

- Σύστημα διαχείρισης παραγγελιών, πωλήσεων και εξόδων σε ένα κατάστημα λιανικής
- Διαχείριση των βασικών λειτουργιών ενός εικονικού τραπεζικού υποκαταστήματος
- Ανάπτυξη συστήματος λογισμικού για ένα φροντιστήριο
- Σύστημα διαχείρισης κρατήσεων για ένα ξενοδοχείο
- Κατάστημα ενοικίασης DVD
- Πληροφοριακό σύστημα υποστήριξης εκδοτικού οίκου
- Πληροφοριακό σύστημα υποστήριξης συνεργείου αυτοκινήτων



Εργασίες / Θέματα (2)

- Γραμματεία πανεπιστημίου
- Ανάπτυξη συστήματος επεξεργασίας και παρακολούθησης λειτουργιών Δανειστικής Βιβλιοθήκης Πανεπιστημίου
- Στατιστική Επεξεργασία Ποδοσφαιρικής Διοργάνωσης
- Γραφείο ενοικιάσεως αυτοκινήτων



Εργασίες

1. First come, First served ...
2. Επιλογή θέματος
3. Επιλογή πεδίου ανάλυσης
4. Καταγραφή παραδοχών
5. Use Case / Activity / State Chart
6. Παρουσίαση πρώτων αποτελεσμάτων (5%)
7. Class / Sequence ή Collaboration / Deployment
8. Παράδοση τελικού κειμένου / τελική βαθμολόγηση (15%)



Παράδειγμα Εργασίας

ΘΕΜΑ:

Ανάπτυξη συστήματος επεξεργασίας και παρακολούθησης λειτουργιών *Δανειστικής Βιβλιοθήκης Πανεπιστημίου*.

Το λογισμικό θα υλοποιεί τις βασικές λειτουργίες δανεισμού και επεξεργασίας στοιχείων όπως :

- 1) Διαδικασία δανεισμού και επιστροφής βιβλίων
- 2) Έκδοση και ακύρωση καρτών χρηστών βιβλιοθήκης.
- 3) Αναζήτηση πληροφοριών μέσω ηλεκτρονικού υποσυστήματος
- 4) Συντήρηση και ανανέωση υλικού βιβλιοθήκης
- 5) Ταξινόμηση βιβλίων ανά κατηγορία.



Λογισμικό

- ArgoUML, εφαρμογή Java

<http://argouml.tigris.org/>

- UMLet, εφαρμογή Java

<http://qse.ifs.tuwien.ac.at/~auer/umlet/>

- UMLGraph, εφαρμογή Java

<http://www.spinellis.gr/sw/umlgraph/>

ArgoUML

The screenshot displays the ArgoUML application window titled "The Shapes Project.zargo - shapes - ArgoUML". The interface includes a menu bar (File, Edit, View, Create, Arrange, Generation, Critique, Tools, Help), a toolbar, and a package-centric tree on the left. The tree shows a project structure with packages like "OneDimensional", "Point", "Polygon", "Shape", and "TwoDimensional".

The main workspace shows a UML class diagram with the following elements:

- Shape**: Base class with `newOperation(): void`.
- OneDimensional**: Inherits from Shape, with `getLength(): double`.
- TwoDimensional**: Inherits from Shape, with `getArea(): double`.
- Polygon**: Inherits from both OneDimensional and TwoDimensional. It has a constructor `create Polygon(): void` and is highlighted with a red border.
- Point**: A class with attributes `x: int` and `y: int`.
- Association**: A composition relationship between Polygon and Point, labeled "+Vertices" and "+", with a multiplicity of "1..*" at the Point end.
- Note**: A pink note attached to the Shape class with the text "This is a note."

At the bottom, the "ToDo Item" panel is active, displaying a warning:

Polygon has multiple base classes, but Java does not support multiple inheritance. You must use interfaces instead. This change is required before you can generate Java code. To fix this, use the "Next>" button, or manually (1) remove one

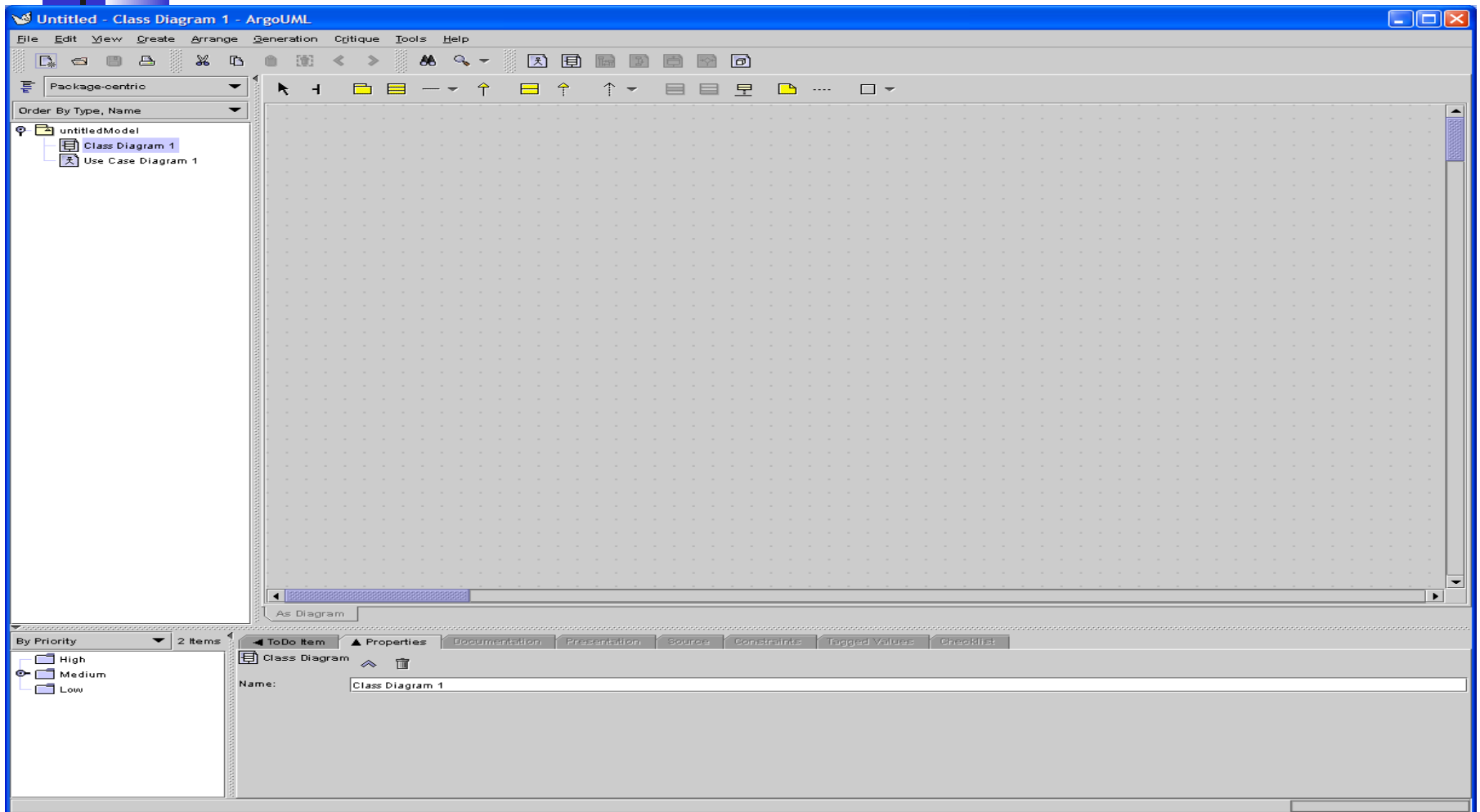
Navigation buttons "Back", "Next", "Finish", and "Help" are visible at the bottom of the panel.



ArgoUML - Εγκατάσταση

- Εγκατάσταση java 1.5.x
<http://java.sun.com/>
- Τρέχουσα έκδοση, 0.18.1
- Αποσυμπίεση αρχείου
«ArgoUML-0.18.1.zip»
- Εκτέλεση της εντολής
`java -jar argouml.jar`

ArgoUML – Κενό έργο





Εισαγωγή στην UML

- Η UML είναι μια γλώσσα για την οπτικοποίηση, καθορισμό δημιουργία και τεκμηρίωση ενός συστήματος (συνήθως λογισμικού)
- Υιοθετήθηκε ως πρότυπο από την OMG (Object Management Group) τον νοέμβριο 1997
- UML έχει γεννήτορες κώδικα για αρκετές γλώσσες προγραμματισμού όπως η C++, Java, Ada, VB κ.α.



Ιστορικό

- 1980 – 1994 → Παραπάνω από 50 ΟΟ μεθόδους όπως Fusion, Coad/Yourdon, Shlaer/Mellor, Booch, Object-Oriented Software Engineering (OOSE), Object Modeling Technique (OMT)
- 1995 → Συνένωση των μεθοδολογιών Booch και OMT καταλήγοντας στην Unified Method
- Ιανουάριος 1997 → Κατάθεση της UML 1.0 στην OMG για προτυποποίηση
- Νοέμβριος 1997 → Η OMG υιοθέτησε την UML 1.1 ως πρότυπο
- Ιούνιος 1998 → UML 1.2
- Ιούνιος 1999 → UML 1.3
- Μάιος 2001 → UML 1.4
- Μάρτιος 2003 → UML 1.5
- Φεβρουάριος 2004 → UML 2.0



Διαγράμματα της UML

- Διάγραμμα περιπτώσεων χρήσης (*use case diagram*)
- Διαγράμματα δομής
 - Διάγραμμα κλάσεων (*class diagram*)
 - Διάγραμμα αντικειμένων (*object diagram*)
- Διαγράμματα συμπεριφοράς
 - Διάγραμμα καταστάσεων (*statechart diagram*)
 - Διάγραμμα δραστηριοτήτων (*activity diagram*)
 - Διαγράμματα αλληλεπίδρασης
 - Διάγραμμα ακολουθίας (*sequence diagram*)
 - Διάγραμμα συνεργασίας (*collaboration diagram*)
- Διαγράμματα δομής υλοποίησης
 - Διάγραμμα εξαρτημάτων (*component diagram*)
 - Διάγραμμα ανάπτυξης (*deployment diagram*)

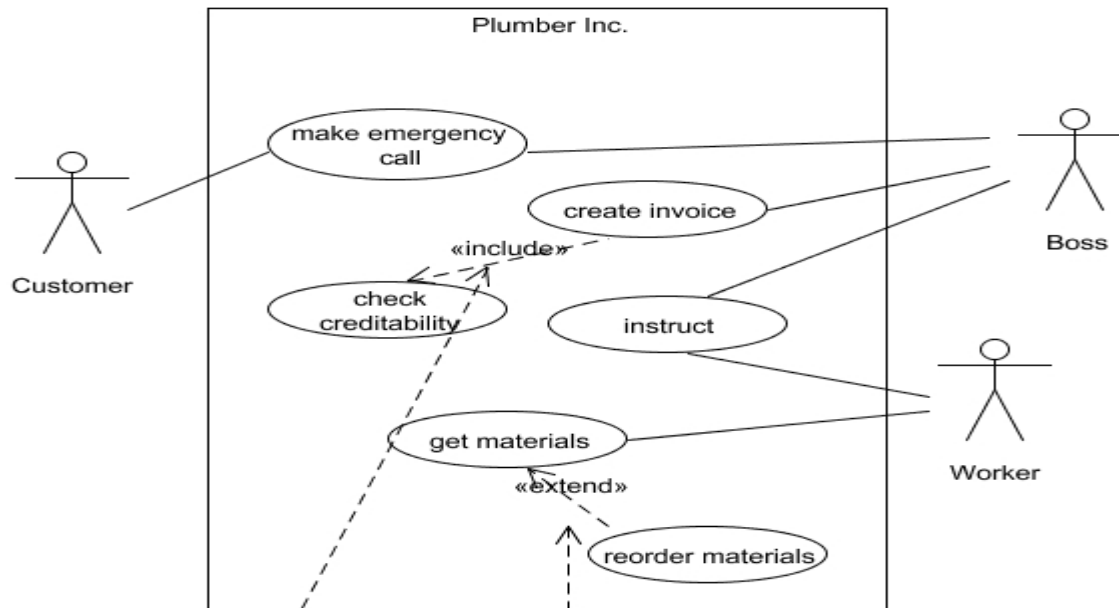


Χρήση Διαγραμμάτων

<i>Διεργασία</i>	<i>Διάγραμμα</i>	<i>Απεικονίζει</i>
Ανάλυση απαιτήσεων	Διάγραμμα περιπτώσεων χρήσης (use case)	Οπτική χρήση στην συμπεριφορά του συστήματος
	Διάγραμμα δραστηριοτήτων (activity)	Δραστηριότητες χρήστη – αλληλεπίδραση με το σύστημα
	Διάγραμμα κλάσεων (class)	Βασικές οντότητες συστήματος και σχέσεις τους
	Διάγραμμα καταστάσεων (statechart)	Κύκλος ζωής σημαντικών δραστηριοτήτων
Σχεδιασμός	Διάγραμμα κλάσεων (class diagram)	Εσωτερική δομή συστήματος
	Διαγράμματα αλληλεπίδρασης (Sequence/Collaboration)	Αλληλεπίδραση των κλάσεων του συστήματος
	Διάγραμμα εξαρτημάτων (component)	Απεικόνιση του συστήματος σε επαναχρησιμοποιούμενα τμήματα
	Διάγραμμα ανάπτυξης (deployment)	Φυσική απεικόνιση του λογισμικού
	Διάγραμμα καταστάσεων (statechart)	Κύκλος ζωής βασικών κλάσεων
Τεκμηρίωση	Όλα τα διαγράμματα	Διαγραμματική υποστήριξη στην τεκμηρίωση του λογισμικού

Διάγραμμα περιπτώσεων χρήσης

Use case:
Message exchange between
actors without revealing the
internals of the processes.

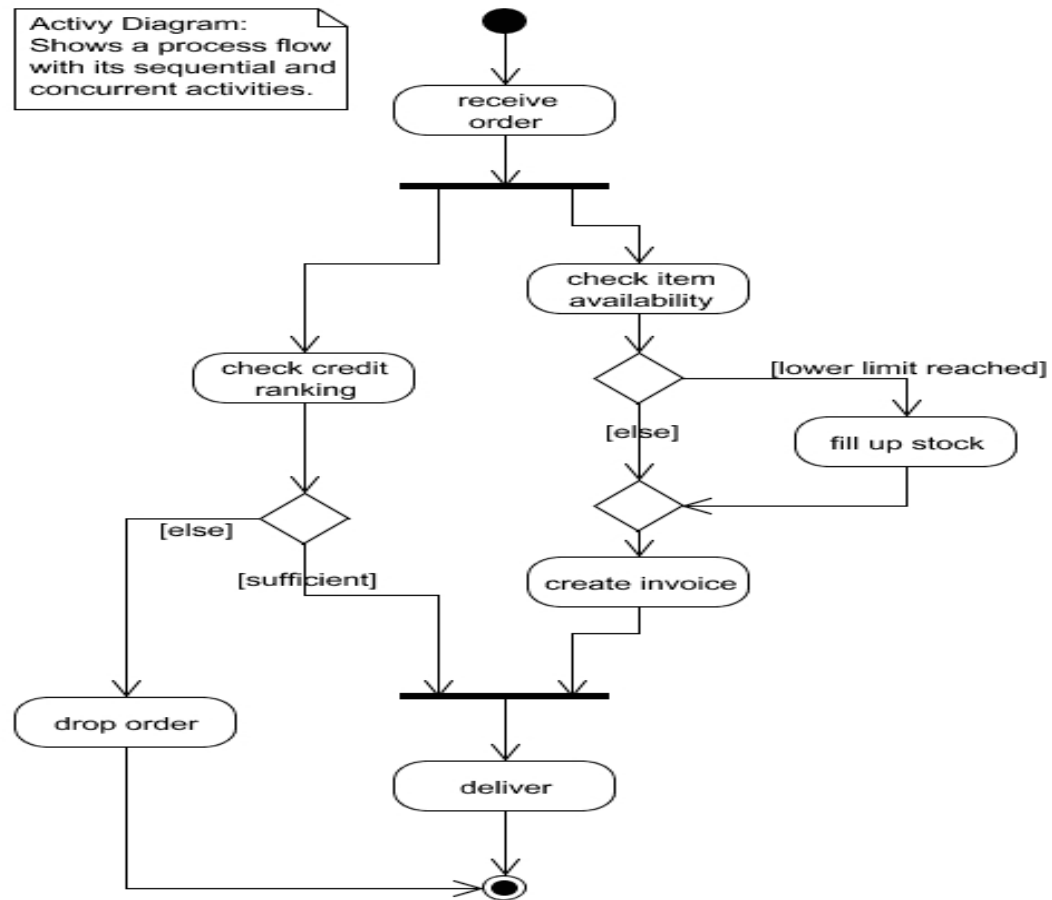


"create invoice" includes
"check creditability".
So every time the boss creates
the invoice he also checks the
customers creditability.

"reorder materials" extends
"get materials".
Only if there are too view
materials they are reordered.

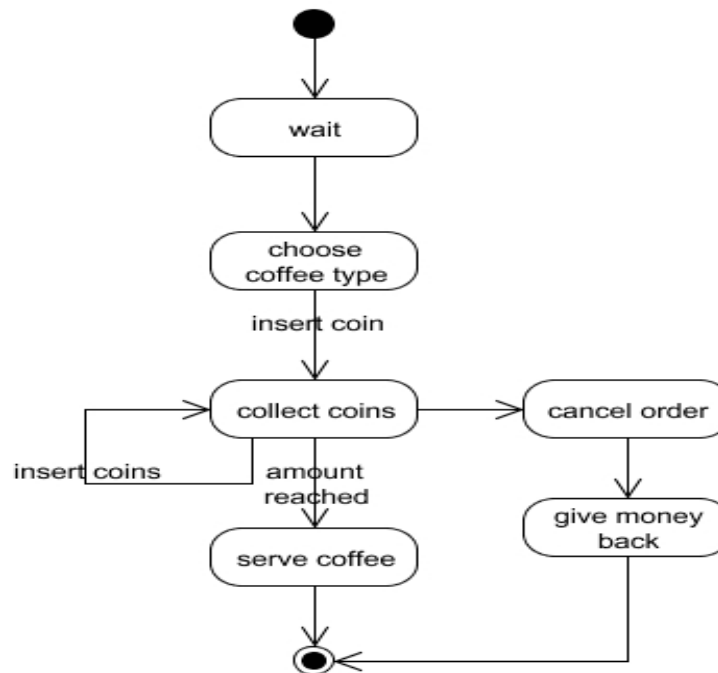
Διάγραμμα Δραστηριοτήτων

Activity Diagram:
Shows a process flow
with its sequential and
concurrent activities.



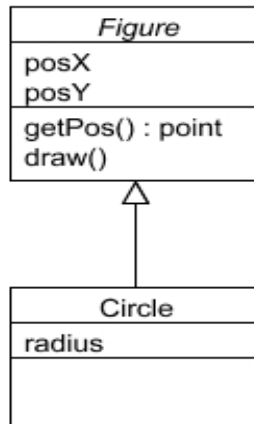
Διάγραμμα Καταστάσεων

State Machine Diagram:
Illustration of the states and state
transitions of a state machine.

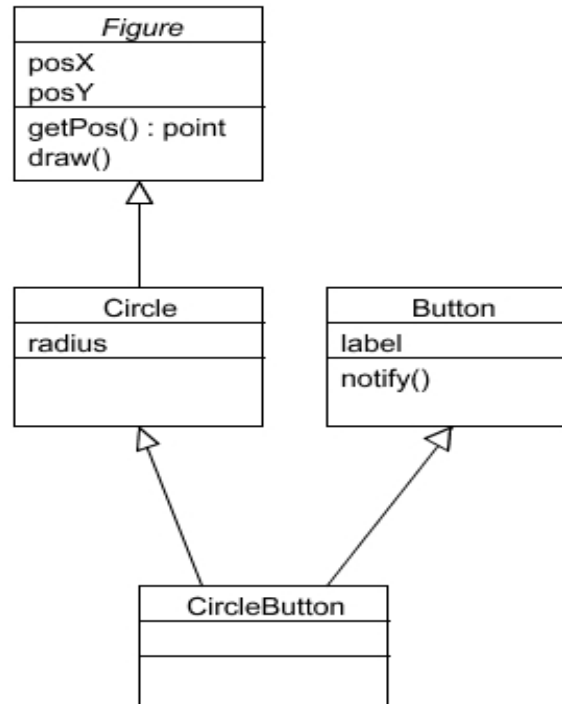


Διάγραμμα Κλάσεων

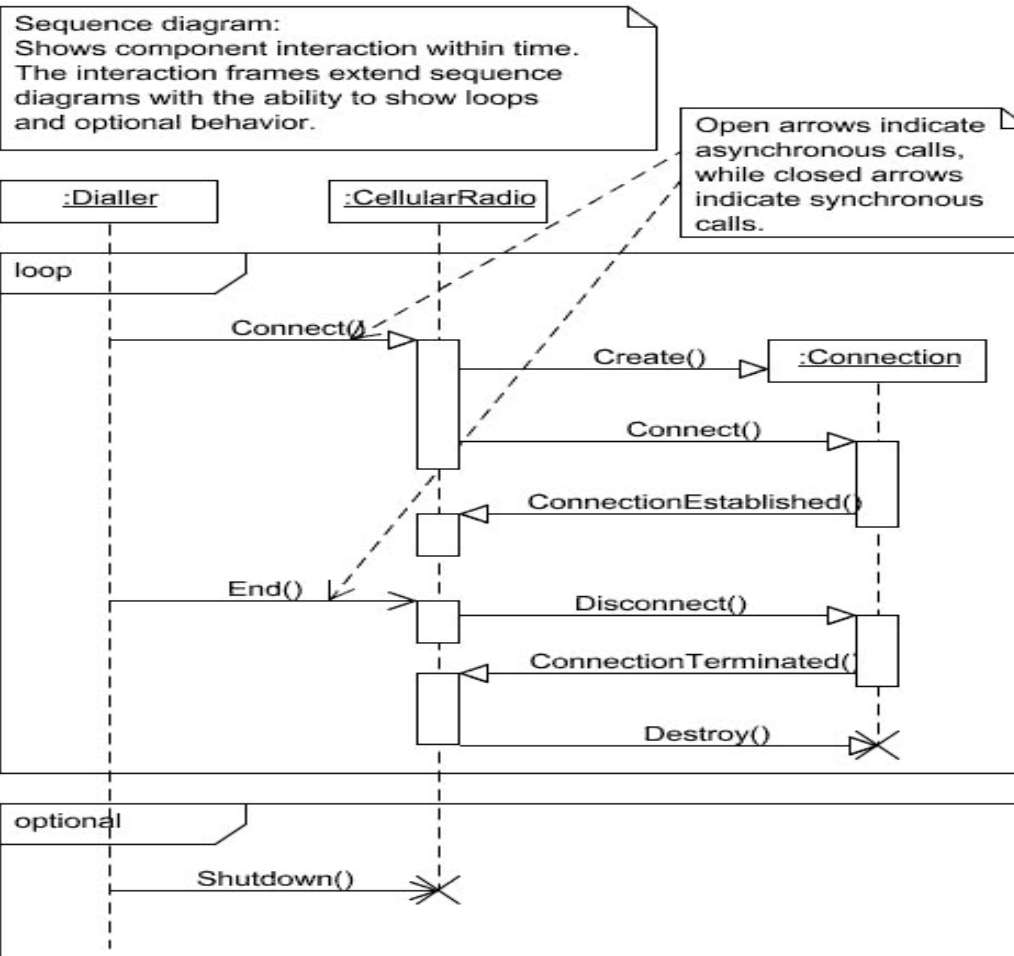
Inheritance:
Subclasses inherit
attributes and operations.



Multiple inheritance:
Subclasses inherit attributes and
operations from all parents.

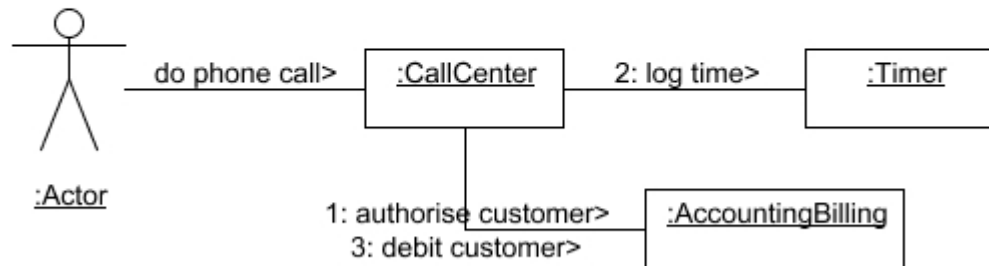


Διάγραμμα Ακολουθίας

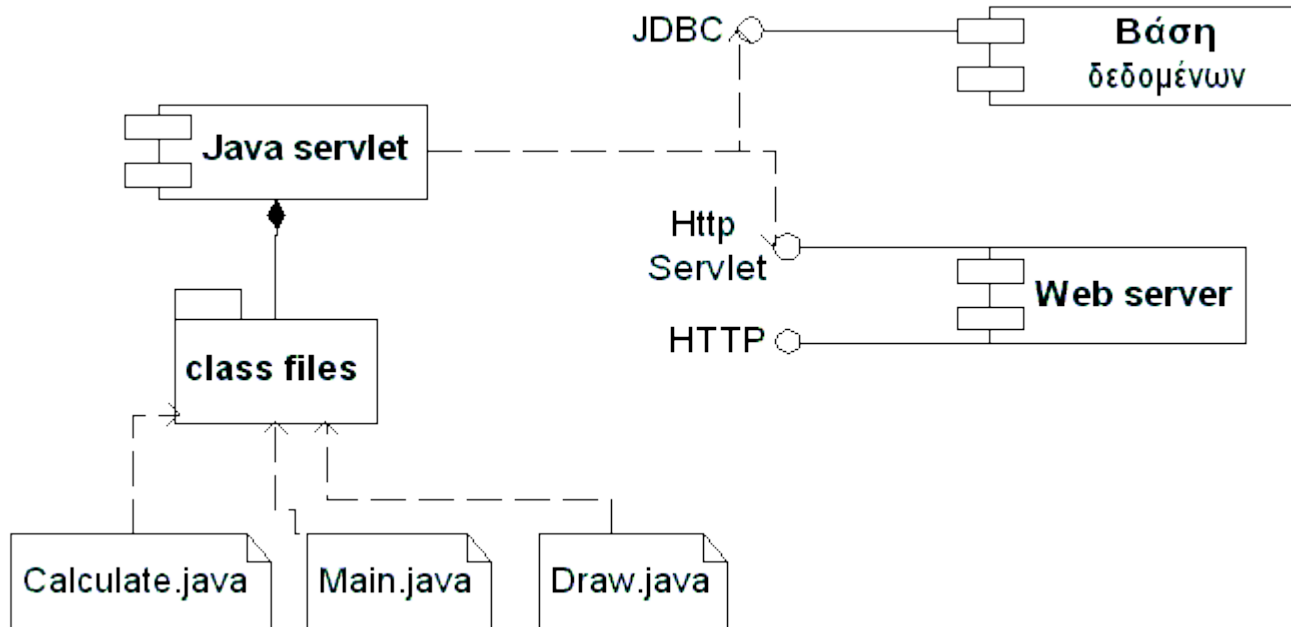


Διάγραμμα Συνεργασίας

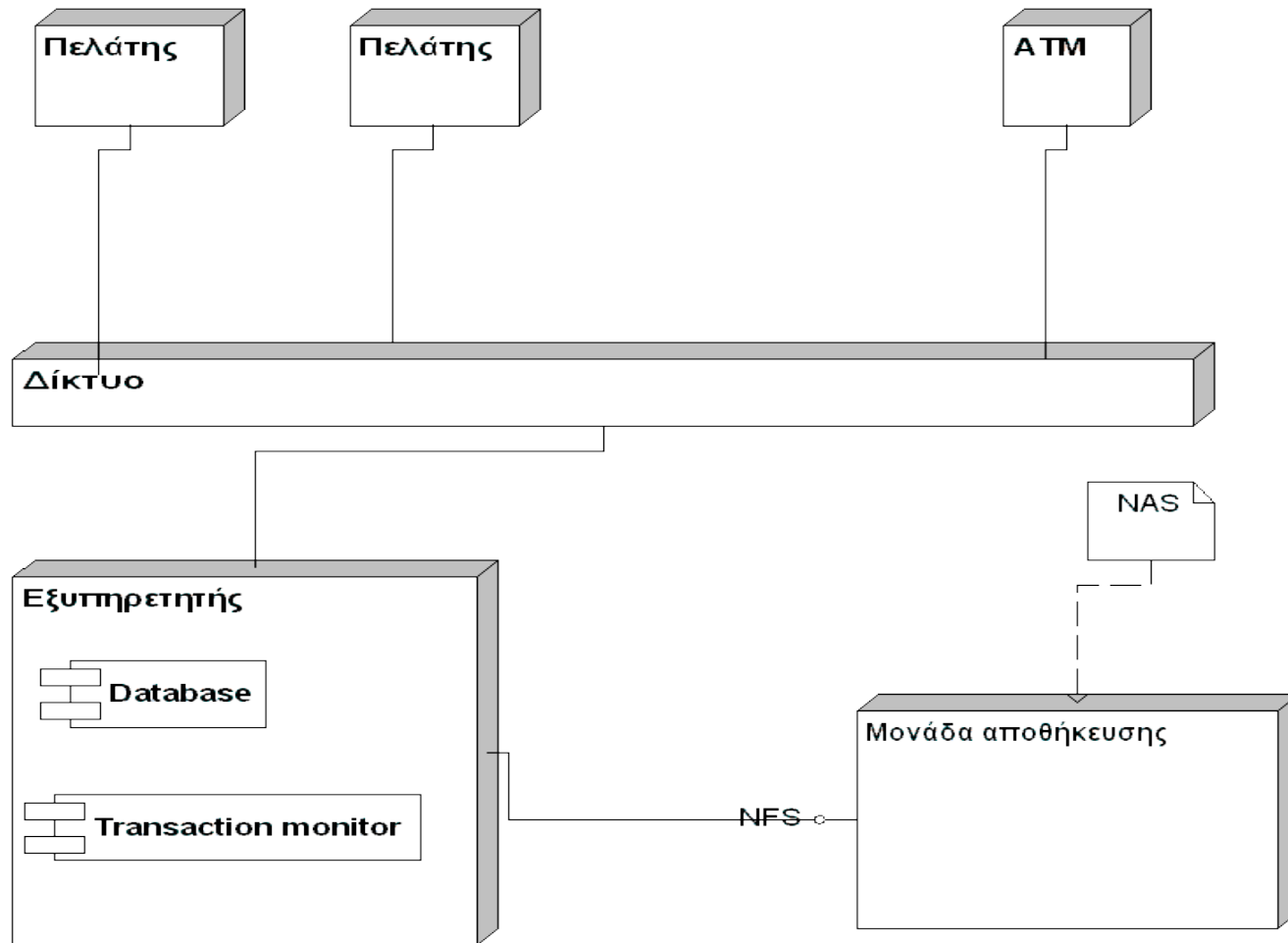
Collaboration:
Describes configuration
and interaction of objects.



Διάγραμμα Εξαρτημάτων



Διάγραμμα ανάπτυξης





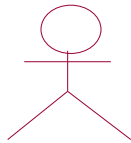
Βιβλιογραφία

- Martin Fowler και Kendall Scott. *Εισαγωγή στη UML: Συνοπτικός οδηγός της πρότυπης γλώσσας μοντελοποίησης*. Δεύτερη αμερικάνικη έκδοση. Εκδόσεις Κλειδάριθμος, 2001.
- Narasimha Bolloju. Improving the quality of business object models using collaboration patterns. *Communications of the ACM*, 47(7):81–86, July 2004.
- Grady Booch, James Rumbaugh, and Ivar Jacobson. *The Unified Modeling Language User Guide*. Addison-Wesley, 1999.
- Roger Box and Michael Whitelaw. Experiences when migrating from structured analysis to object-oriented modelling. In *Proceedings of the Australasian conference on Computing education*, pages 12–18. ACM Press, 2000. (doi:10.1145/359369.359372)
- John Daniels. Modeling with a sense of purpose. *IEEE Software*, 19(1):8–10, January/February 2002.
- Yann-Gaël Guéhéneuc and Hervé Albin-Amiot. Recovering binary class relationships: Putting icing on the UML cake. In *OOPSLA '04: Proceedings of the 19th annual ACM SIGPLAN Conference on Object-oriented programming, systems, languages, and applications*, pages 301–314. ACM Press, 2004. (doi:10.1145/1028976.1029002)
- Roger S. Pressman. *Software Engineering: A Practitioner's Approach*, pages 685–635. McGraw-Hill, fifth edition, 2000. European Adaptation. Adapted by Darrel Ince.
- James Rumbaugh, Ivar Jacobson, and Grady Booch. *The Unified Modeling Language Reference Manual*. Addison-Wesley, 1999.
- Graeme Shanks, Elizabeth Tansley, and Ron Weber. Representing composites in conceptual modeling. *Communications of the ACM*, 47(7):77–80, July 2004.
- Ian Sommerville. *Software Engineering*, pages 42–69. Addison-Wesley, sixth edition, 2001.
- Roel Wieringa. A survey of structured and object-oriented software specification methods and techniques. *ACM Computing Surveys*, 30(4):459–527, December 1998.

Διάγραμμα περιπτώσεων χρήσης - Εισαγωγή

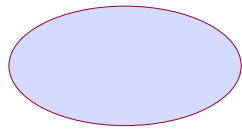
- Αποτυπώνει την συμπεριφορά ενός συστήματος, υποσυστήματος ή κλάσης όπως αυτή εμφανίζεται σε ένα εξωτερικό χρήστη
- Διαμερίζει την λειτουργικότητα σε λειτουργίες του συστήματος οι οποίες εκκινούνται από Δρώντες (actors)
- Οι διαμερίσεις ονομάζονται Περιπτώσεις Χρήσης (use cases)

Διάγραμμα περιπτώσεων χρήσης – Σημειολογία (1)



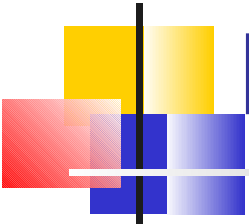
Actor

Δρώντες

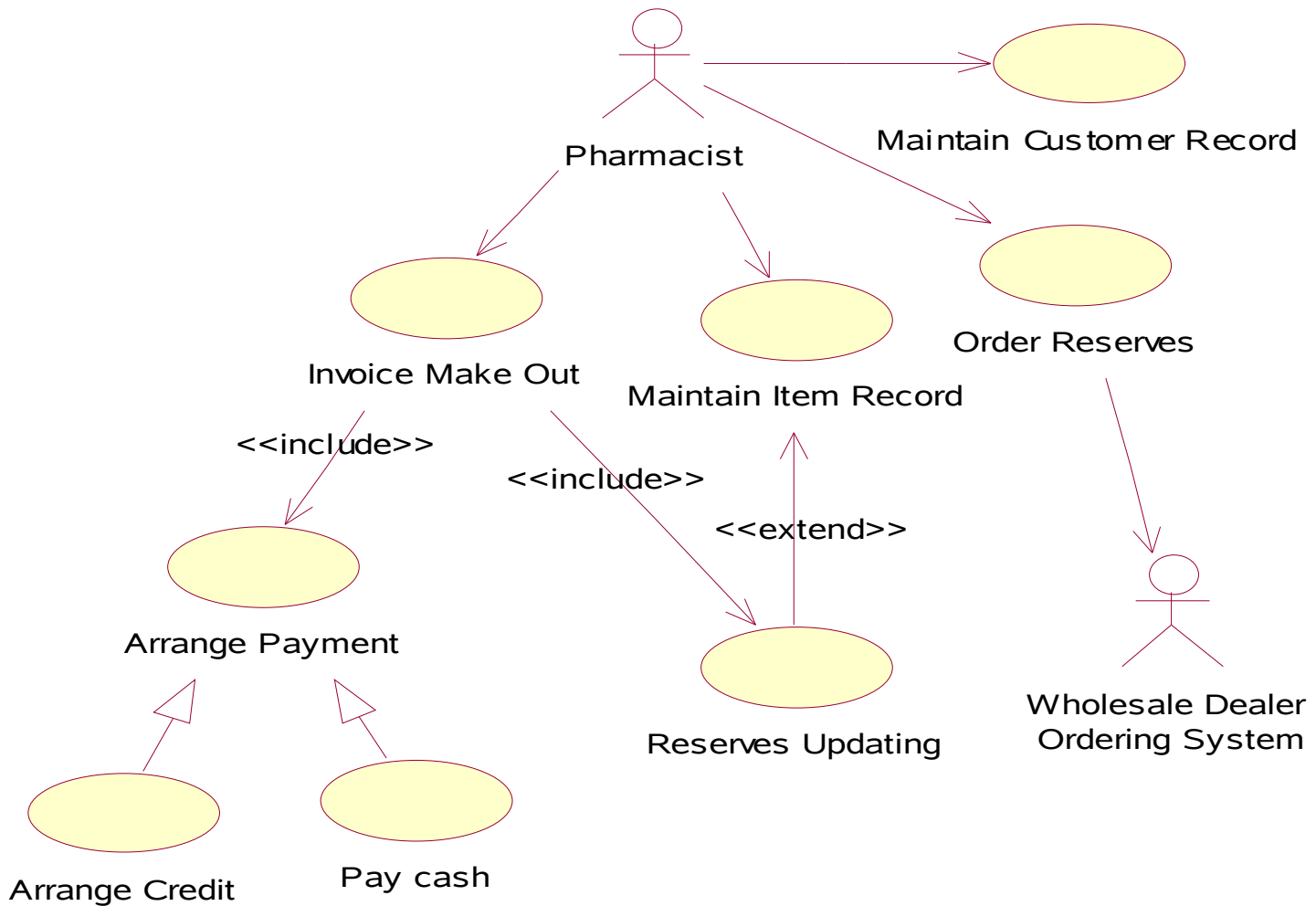


Use Case

Περίπτωση
Χρήσης



Παράδειγμα





Την επόμενη φορά ...

- Διαγράμματα Use Case
- Μελέτη αποτελεσμάτων έργου Gemini