HY351:
Ανάλυση και Σχεδίαση Πληροφοριακών Συστημάτων
Information Systems Analysis and Design

Από την Ανάλυση στη Σχεδίαση
(Moving on Design)

Στρατηγικές Σχεδίασης (Design Strategies)
Διαστρωμάτωση και Πακετοποίηση (Layering and Packaging)

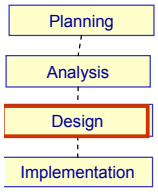
Γιάννης Τζίτζικας

Διάλεξη : 12b
Ημερομηνία :

Διάρθρωση

- Φάση Σχεδίασης (Design Phase)
- Στρατηγικές Σχεδίασης (Design Strategies)
  - Custom Development, Packaged Software, Outsourcing
  - Επιλέγοντας στρατηγική σχεδίασης (Selecting a Design Strategy)
- Από την Ανάλυση στη Σχεδίαση
  - Από τα Μοντέλα Ανάλυσης στα Μοντέλα Σχεδίασης
  - Διαμερίζοντας το Πρόβλημα – Διαστρωμάτωση (Layering)
  - Πακέτα και Διαγράμματα Πακέτων (Packages and Package Diagrams)
  - Πακετοποίηση και Ομαδοποίηση (Packaging and Clustering)

Υπενθυμιστικό: Η Φάση της Σχεδίασης (Design)



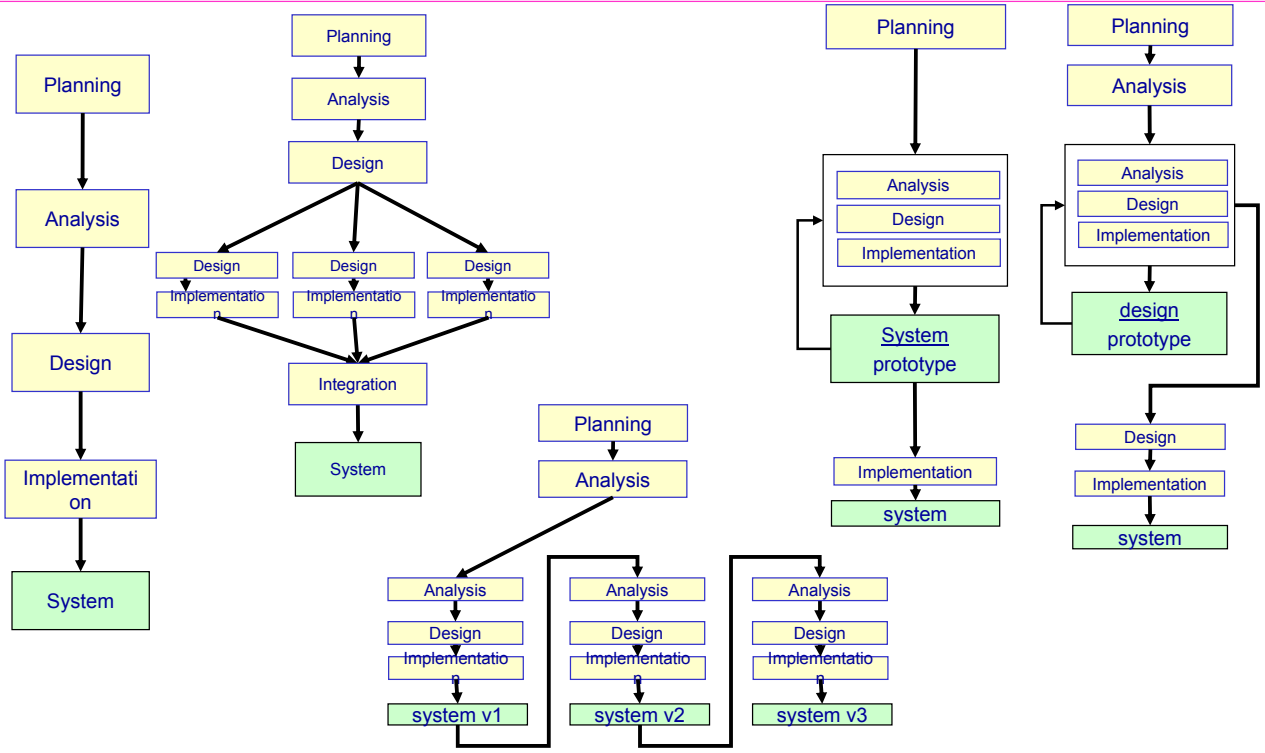
Βήματα:

1. Στρατηγική Σχεδίασης (Design Strategy). Ανάπτυξη από τον **ίδιο τον οργανισμό**, ή **εξωτερική ανάθεση** (outsourcing) σε μια άλλη εταιρεία, ή **αγορά ενός υπάρχοντος πακέτου λογισμικού**;
2. Αρχιτεκτονική Σχεδίαση (Architecture Design): υλικό, λογισμικό, δικτυακή υποδομή, διεπαφή χρήστη, φόρμες, αναφορές, βάσεις δεδομένων, κλπ.
3. Περιγραφή Αρχείων και Βάσεων Δεδομένων : ποια ακριβώς δεδομένα θα αποθηκεύονται και πού;
4. Σχεδίαση των Προγραμμάτων : ποια προγράμματα πρέπει να γραφτούν και τι θα κάνει το κάθε ένα;

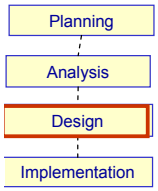
Αποτέλεσμα/Παραδοτέα:

Προδιαγραφή Συστήματος (System Specification) η οποία δίδεται στην ομάδα των προγραμματιστών προς υλοποίηση

Υπενθυμιστικό: Η Φάση Σχεδίασης στις Μεθοδολογίες Ανάπτυξης



Υπενθυμιστικό: Η Φάση της Σχεδίασης (Design)



Βήματα:

1. Στρατηγική Σχεδίασης (Design Strategy). Ανάπτυξη από τον **ίδιο τον οργανισμό**, ή **εξωτερική ανάθεση** (outsourcing) σε μια άλλη εταιρεία, ή **αγορά ενός υπάρχοντος πακέτου λογισμικού**;
2. Αρχιτεκτονική Σχεδίαση (Architecture Design): υλικό, λογισμικό, δικτυακή υποδομή, διεπαφή χρήστη, φόρμες, αναφορές, βάσεις δεδομένων, κλπ.
3. Περιγραφή Αρχείων και Βάσεων Δεδομένων : ποια ακριβώς δεδομένα θα αποθηκεύονται και πού;
4. Σχεδίαση των Προγραμμάτων : ποια προγράμματα πρέπει να γραφτούν και τι θα κάνει το κάθε ένα;

Αποτέλεσμα/Παραδοτέα:

Προδιαγραφή Συστήματος (System Specification) η οποία δίδεται στην ομάδα των προγραμματιστών προς υλοποίηση

Στρατηγικές Σχεδίασης
Design Strategies

Οι κυριότερες στρατηγικές σχεδίασης
(Three Main Design Strategies)

- Custom Development → Building the system from scratch

  Ανάπτυξη από τον ίδιο τον οργανισμό

- Packaged Software → Buying a packaged software and customizing it

  Έτοιμο Πακέτο Λογισμικού

- Outsourcing → Hiring an external person/company to build the system

  Εξωτερική ανάθεση

Ανάπτυξη κατά παραγγελία
Πακέτο Λογισμικού
Εξωτερική ανάθεση

Design Strategies
Custom Development

Custom development means building the system from scratch

Advantages
- Complete control (and freedom) over the functionality of the system
- Extensibility of the system (future needs may be taken into account)
- The skills of the staff improve (beneficial for future projects)

Weaknesses
- Takes time, requires hard work (the staff may be already overcommitted)
- Appropriately skilled staff is needed
- High risk

Design Strategies
Packaged Software

Buying a packaged system and customizing it
(commercial off-the-shelf (COTS) software product)

- many business needs are common to a large number of organizations
- there are thousands of commercially available programs
  - that support basic and ubiquitous needs (e.g. payrolls, orders, inventory, sales, etc)
- these programs range from small components to complete ERP (enterprise resource planning) systems
  - complete ERP systems have taken years and have cost millions of Euros to develop (SAP, PeopleSoft, Oracle, Baan)

Design Strategies
Packaged Software (II)

This decision strategy introduces an additional phase, the "procurement phase" which is a special decision analysis phase comprising the following steps:

1) Identify products that could fit the needs of the project
   - sources: special magazines, journals, Web sites, other organizations with similar needs
2) solicit, evaluate and rank vendor proposals
3) select and recommend the best of them
4) contract with the awarded vendor to obtain the product

Procurement phase: φάση προμήθειας

Design Strategies
Packaged Software (III)

2) solicit, evaluate and rank vendor proposals

Commonly, this step involves preparing an RFQ or an RFP document:

RFQ (Request for Quotations) (αίτηση προσφοράς)

used when you have selected the product but the product is available from various distributions. Objective: negotiate configuration, prices, maintenance contract.

RFP (Request for Proposals) (αίτηση προτάσεων)

used when there are several different vendors and products that are candidate. RFP is a superset of RFQ because the former describes in detail the requirements of the project (it is like the Requirements Specification Document)

Design Strategies
Packaged Software (IV)

Advantages

- It takes a short time to buy and install it (no extra staff is needed)
- Low risk (packaged software has already been written, tested and proved to be effective)

Weaknesses

- Not full control over the functionality of the system (we cannot customize everything)
  - sometimes the business processes of the organization have to adapt to the software!
- Extensibility of the system for the future needs of the organization is not ensured
- The skills of your staff do not improve
- The price of the packages software is sometimes high
- The organization depends on the vendor
  - maintenance (plus cost of maintenance)
  - Risk if the vendor bankrupts

Design Strategies
Outsourcing

Hiring an external person/company to build the system

Types of contracts:

- ***time and arrangements***
  - payment based on the time and the expenses
  - [-] the bill may climb high
- ***fixed-price contract***
  - requirements should be very clear
  - [-] usually outsourcers are not willing to accept changes in the requirements
- ***value-added contract***
  - the outsourcer will have some share of the benefits from the system
  - becomes more and more popular

Design Strategies
Outsourcing (II)

Advantages

- You have the full control over the requirements
- The external party may be more skilled than you

Weaknesses

- No full control over the internal architecture of the system
- The skills of your staff do not improve
- You may have to share confidential information
- You need a person to manage the outsourcing
- Risk if you haven't specified well the requirements
  - ***never outsource something you don't understand***
- Risk if the external party fails

Επιλέγοντας Στρατηγική Σχεδίασης (Selecting a Design Strategy)

Επιλέγοντας Στρατηγική Σχεδίασης

Custom Development	**Packaged System**	**Outsourcing**
Unique business need	Very common business need	The business need is not crucial for the organization
Your staff has technical skills and you want to enhance them	Technical skills are lacking and it is not strategically important to build them	Technical skills are lacking and it is not strategically important to build them
Loose and flexible deadline	Short timeframe with strict deadline	A "good" outsourcer is available
	You want to avoid risk	

What about your project?

Επιλέγοντας Στρατηγική Σχεδίασης
Πίνακας Εναλλακτικών (Matrix of Alternatives)

We can use a matrix of alternatives (like the one we used in feasibility analysis) for organizing the pros and cons of each alternative. We could also employ a weighting scheme or any other decision theoretic method for selecting the best strategy for the project at hand

Description	Custom Development	Packaged Software	Outsourcing
Operational feasibility			
Technical feasibility			
Economic feasibility Cost Payback period ROI			
Schedule feasibility			

Plus whatever other criteria are appropriate

Hereafter we assume that we have selected the custom development design strategy

Από τα Μοντέλα Ανάλυσης στα Μοντέλα Σχεδίασης
(From Analysis Models to Design Models)

Από τα Μοντέλα Ανάλυσης στα Μοντέλα Σχεδίασης

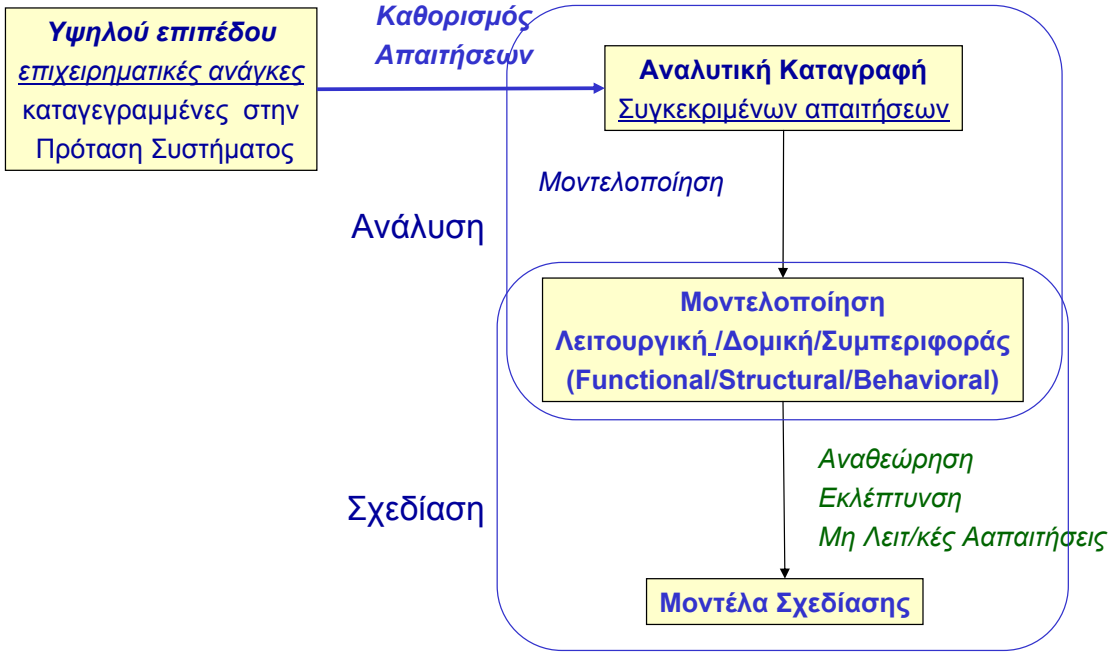
Σκοπός

- Τα μοντέλα ανάλυσης που έχουμε δημιουργήσει μέχρι τώρα έχουν κυρίως εστιάσει στις Λειτουργικές Απαιτήσεις (ΛΑ)
- Τώρα πρέπει να λάβουμε υπόψη και τις **Μη Λειουργικές Απαιτήσεις (ΜΛΑ)**
- Στην Αντικειμενοστρεφή Ανάλυση και Σχεδίαση: πρέπει να αναθεωρήσουμε και εκλεπτύνουμε τα μοντέλα που έχουμε δημιουργήσει μέχρι τώρα
  - Προσθέτοντας λεπτομέρειες που αφορούν το περιβάλλον του συστήματος

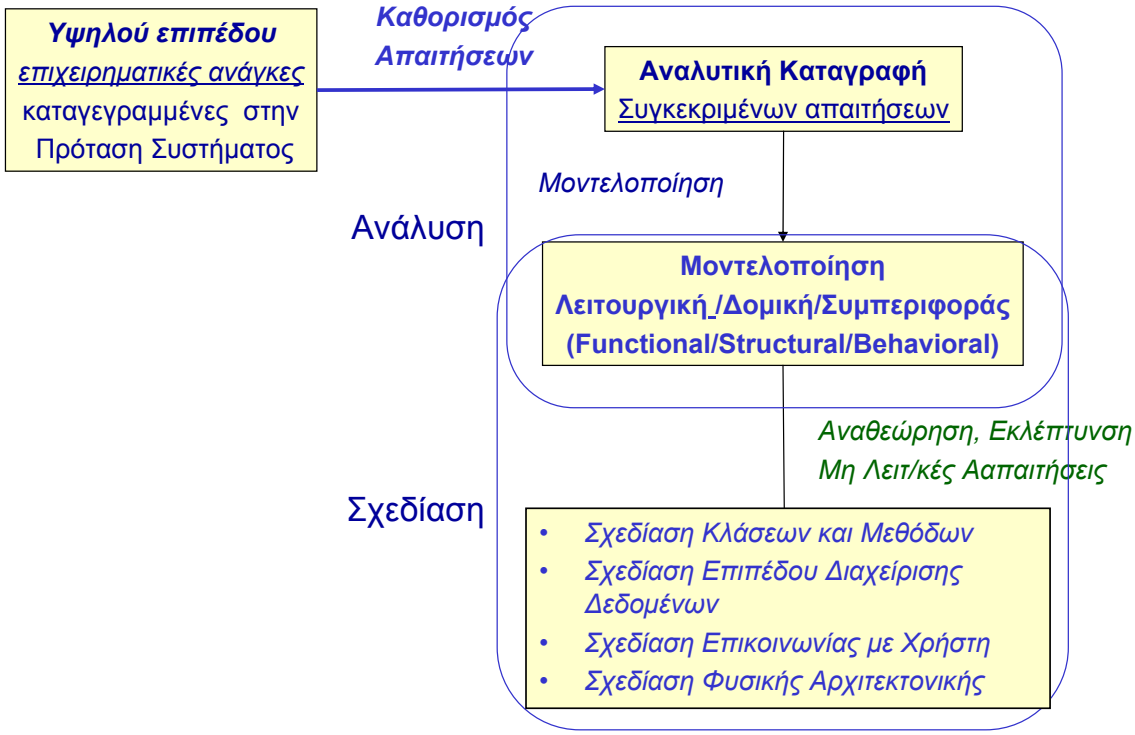
Μη-Λειτουργικές Απαιτήσεις (ΜΛΑ)

- Περιγράφουν **ιδιότητες** του συστήματος που συνήθως εκφράζονται βάσει χαρακτηριστικών της μορφής: Απόδοση (performance), Χρηστικότητα (usability), Ασφάλεια (security), Νομιμότητα (legislative), Ιδιωτικότητα (privacy)
- Με άλλα λόγια: περιγράφουν το **πώς** (ή το **πόσο καλά**) το σύστημα θα υποστηρίξει τις λειτουργικές απαιτήσεις. Μπορούμε να τις θεωρήσουμε ως «**περιορισμούς**» που περιορίζουν τους τρόπους με τους οποίους θα μπορούσαμε να πραγματώσουμε τις λειτουργικές απαιτήσεις.

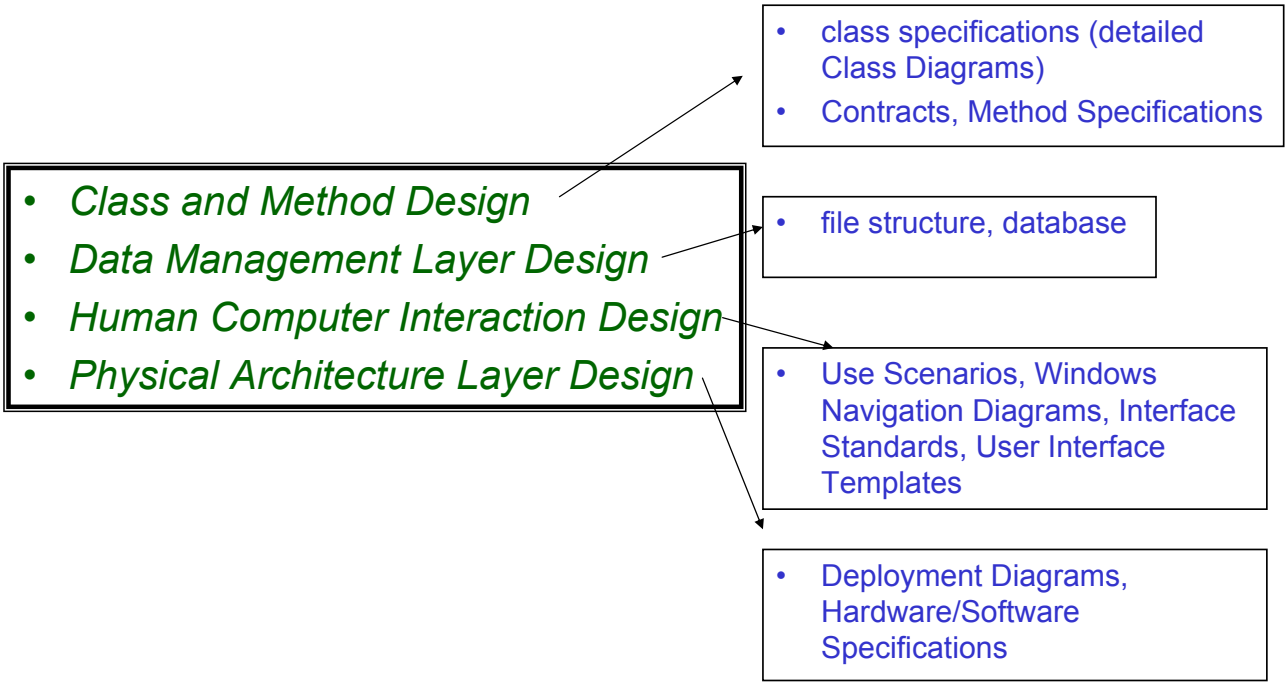
Από τα Μοντέλα Ανάλυσης στα Μοντέλα Σχεδίασης



Από τα Μοντέλα Ανάλυσης στα Μοντέλα Σχεδίασης
Τεχνικά, η φάση Σχεδίασης περιλαμβάνει:



Από την Ανάλυση στη Σχεδίαση
Τεχνικά, η φάση Σχεδίασης περιλαμβάνει:



Each of the above topics will be covered in the subsequent lectures

Ένα από παλαιότερα ερωτήματα
(One of the oldest questions in software methods)

Πώς να σπάσουμε ένα μεγάλο σύστημα;

Λόγος: **Αν είναι μεγάλο τότε έχουμε δυσκολίες κατανόησης, επέκτασης, αλλαγής**

Φάση Ανάλυσης:

Για να σπάσουμε ένα μεγάλο σύστημα στο παρελθόν κάναμε Λειτουργική Αποσύνθεση (Functional Decomposition). Χρησιμοποιούνταν τις μέρες που οι διαδικασίες (processes) και τα δεδομένα ήταν διαχωρισμένα

Σήμερα: Περιπτώσεις Χρήσης (Use Cases)

Φάση Σχεδίασης:

Εντοπισμός στρωμάτων (layers) και πακέτων (packages)

– (packages are sometimes called subsystems)

Τα πακέτα είναι ζωτικής σημασίας εργαλεία για τα μεγάλα έργα

Τα πακέτα είναι επίσης καλά για δοκιμές (testing in package basis)

Εντοπισμός Στρωμάτων
Identification of *Layers*

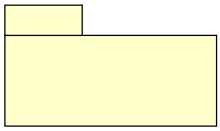
- Έως τώρα έχουμε κυρίως εστιάσει στην «εννοιολογική προοπτική» (ή αλλιώς μοντέλο προβλήματος, μοντέλο πεδίου εφαρμογής, **problem domain layer**)
- Τώρα πρέπει να προσθέσουμε λεπτομέρειες που αφορούν στο περιβάλλον του συστήματος
- Ένας τρόπος για να **αποφύγουμε την υπερφόρτωση του σχεδιαστή** είναι να χρησιμοποιήσουμε πολλά ***layers (στρώματα)***

Παράδειγμα διαστρωμάτωσης:

Problem Domain	Η εννοιολογική προοπτική
Data Management	Αφορά τα αντικείμενα που απαιτούν μόνιμη αποθήκευση
Human Computer Interaction	Κλάσεις Διεπαφής Χρήστη
Physical Architecture	Επικοινωνία, Αρχιτεκτονική Λογισμικού
Foundation Layer	Π.χ. θεμελιώδεις τύποι δεδομένων (συνήθως περιλαμβάνονται στο αντικειμενοστρεφές περιβάλλον ανάπτυξης)

layers can be represented using the higher level (UML) concept of PACKAGE

Πακέτα και Διαγράμματα Πακέτων
Packages and Package Diagrams



Divida et impera (divide and rule)

Πακέτα της UML
UML Packages

Τα Πακέτα είναι ένας γενικός μηχανισμός ομαδοποίησης ο οποίος επιτρέπει να ομαδοποιήσουμε οποιαδήποτε στοιχεία της UML σε υψηλότερου επιπέδου μονάδες

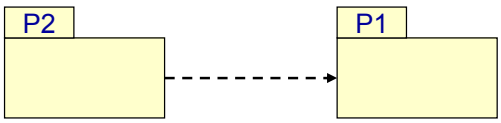
– Συνήθως χρησιμοποιείται για την ομαδοποίηση κλάσεων

Ένα πακέτο μπορεί να ομαδοποιήσει

- κλάσεις
- άλλα μοντέλα σχεδίασης
- …….

Διαγράμματα Πακέτων
Package Diagrams

Απεικονίζουν τα πακέτα και τις μεταξύ τους εξαρτήσεις (dependencies)



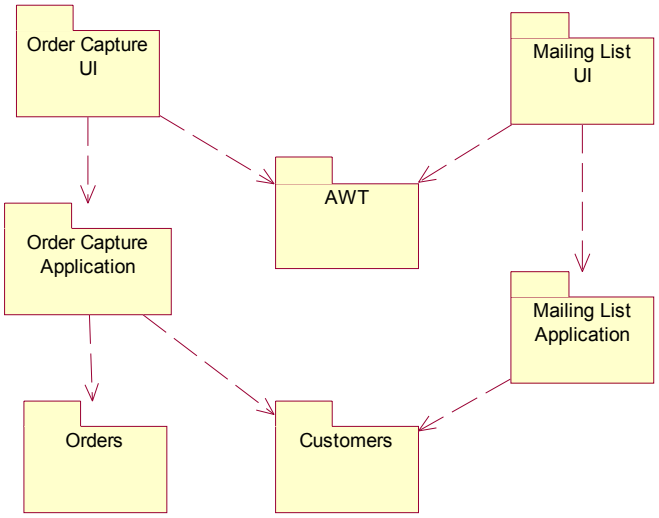
Το P2 εξαρτάται από το P1

Σημαίνει ότι μια αλλαγή στο P1 μπορεί να απαιτήσει αλλαγή του P2

Εξαρτήσεις (Dependencies):

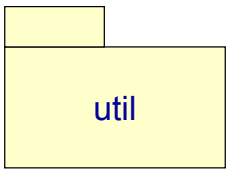
- Οι εξαρτήσεις δεν είναι μεταβατικές
- Μια καλή πρακτική είναι η απαλοιφή των κυκλικών εξαρτήσεων (τουλάχιστον η μείωση τους)
- Η τέχνη της σχεδίασης μεγάλης κλίμακας: ***ελαχιστοποίηση εξαρτήσεων***

Ένα διάγραμμα πακέτων (που ομαδοποιούν κλάσεις)

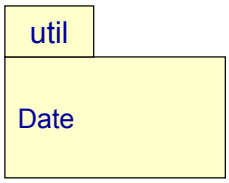


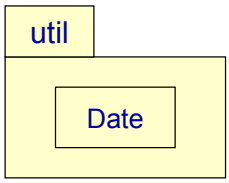
Τρόποι Διαγραμματικής Απεικόνισης Πακέτων

package name



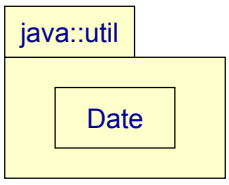
package name + contents

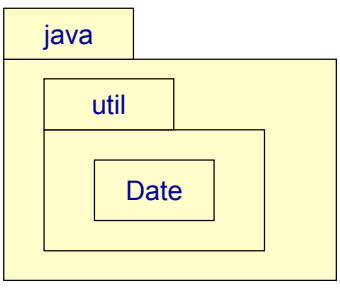




nested packages

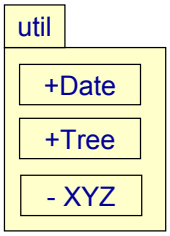
fully qualified package name





Πακέτα, Ορατότητα, και Εισαγωγή
Packages, Visibility and Importing

- We can define the visibility of the elements owned by a package as we defined the visibility of the attributes and operations of a class.

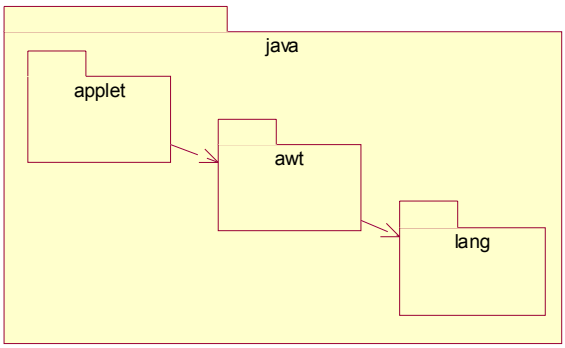


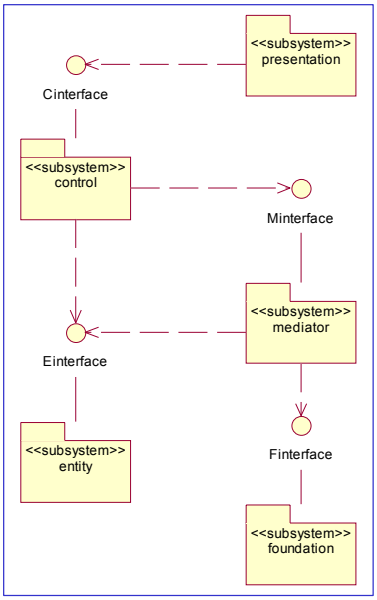
  - +: public
  - -: private: seen only the elements of the same package
  - #: protected: seen only by the elements of the same package and its children

- To refer to the class of another package you need to use the qualified name of the class (package name::class name)
- ***import*** allows using direct names (not qualified ones)
- importing adds the public elements from the target package to the public namespace of the importing package. Notation in UML: <<import>>

"Import" vs "Access": "Access" is like "import" but you cannot reexport. So it is not transitive (import is transitive)

Παραδείγματα





Στρώματα ως Πακέτα
Layers as Packages

Μοντελοποιώντας Στρώματα ως Πακέτα
Modeling Layers using Packages

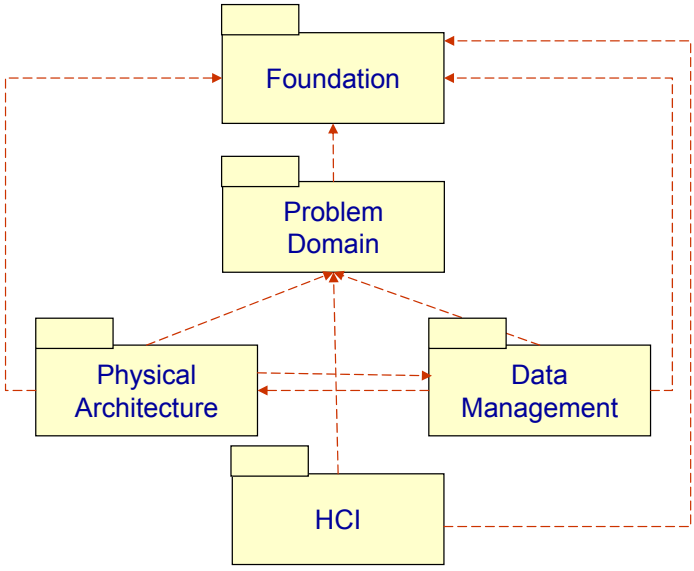
Problem Domain

Data Management

Human Computer Interaction

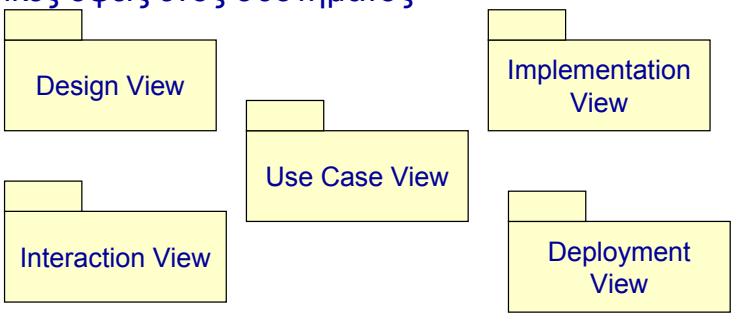
Physical Architecture

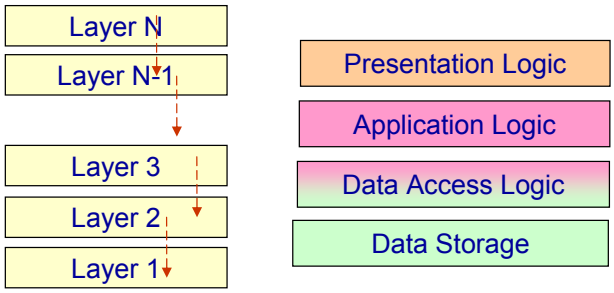
Foundation Layer



Πακέτα για Μοντελοποίηση διάφορων Αρχιτεκτονικών Όψεων
Packages for Modeling the various Architectural Views

Μπορούμε να χρησιμοποιήσουμε πακέτα για να μοντελοποιήσουμε τις διαφορετικές όψεις ενός συστήματος

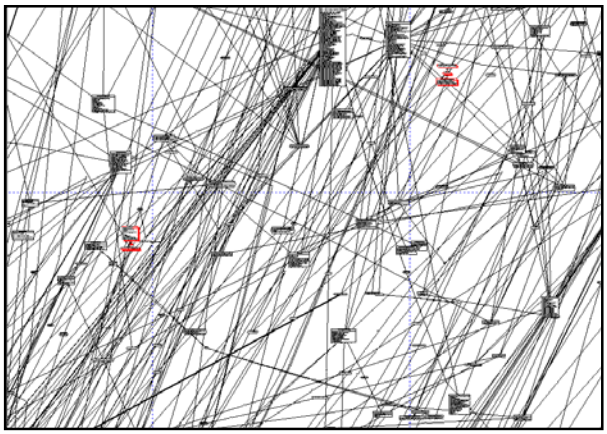


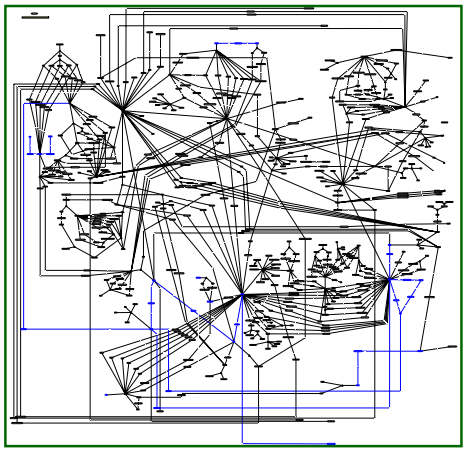


Πακετοποίηση Κλάσεων
Packaging Classes

Πακετοποίηση Κλάσεων: Κίνητρο
Packaging classes: Motivation

- Οι κλάσεις αποτελούν το βασικό μέσο δόμησης ενός αντικειμενοστρεφούς συστήματος
- *Είναι όμως αυτή η δόμηση αρκετή αν έχουμε εκατοντάδες κλάσεων;?*





Συμβουλή: Χρησιμοποίηστε Πακέτα όταν ένα διάγραμμα κλάσεων δεν χωράει σε μια σελίδα Α4

Πακετοποίηση Κλάσεων: Ιδιότητες
Packaging classes: Properties

Ιδιότητες
- Μια κλάση μπορεί να ανήκει σε ένα μόνο πακέτο
- Μια κλάση πρέπει να έχει μοναδικό όνομα μέσα στο πακέτο
- Ένα πακέτο μπορεί να περιέχει κλάσεις και άλλα πακέτα

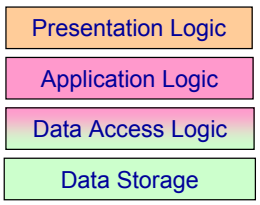
Άρα μπορούμε να ορίσουμε μια ιεραρχική δομή

Σημείωση:
- Τα UML packages αντιστοιχούν στα packages της Java

Πακετοποίηση Κλάσεων: Κριτήρια

Η πακετοποίηση των κλάσεων μπορεί να γίνει βάσει διαφόρων κριτηρίων, π.χ.:

- βάσει συγγραφέα (who wrote what)
- βάσει λειτουργίας (e.g. Storage Manager, UI, MVC, …)
- βάσει αρχιτεκτονικού στρώματος (architectural tier), π.χ. θα μπορούσαμε να διακρίνουμε τις εξής κατηγορίες κλάσεων:

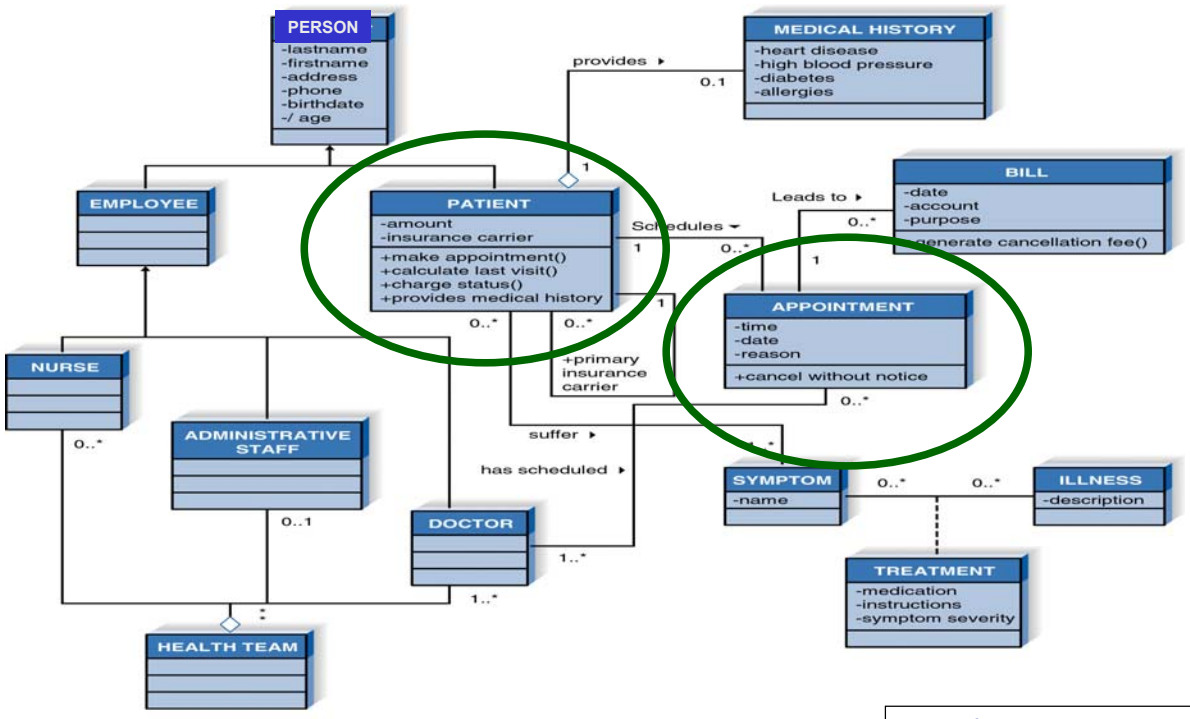


Πακετοποίηση Κλάσεων: Κριτήρια

Ένας άλλος τρόπος διαμέρισης των κλάσεων:

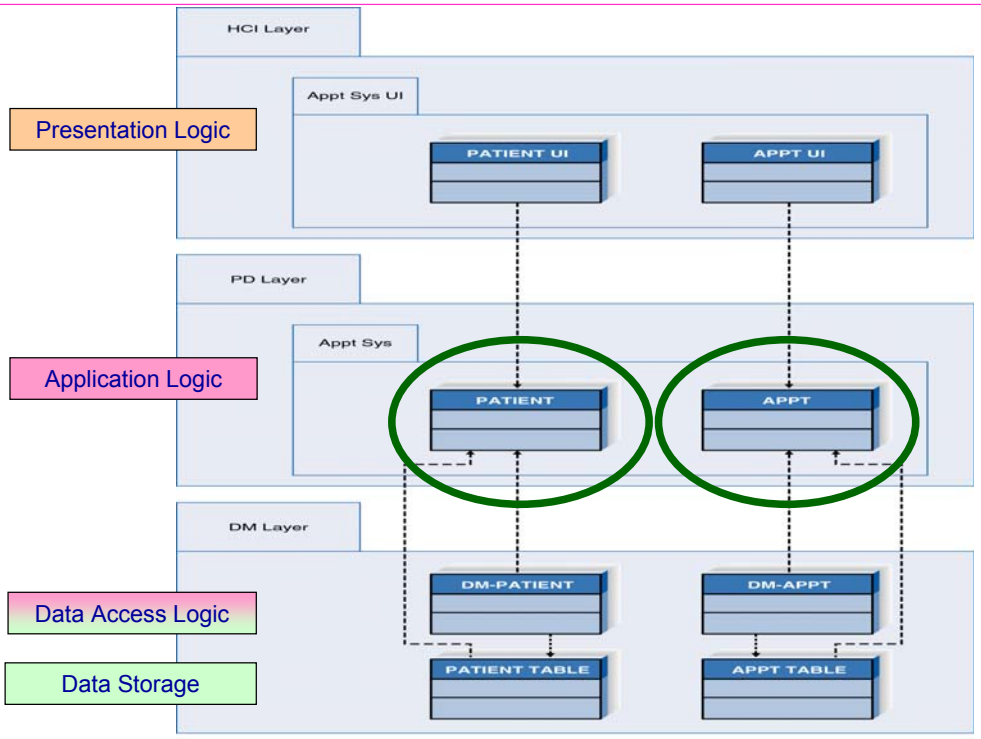
- **persistent database classes**
  - Οι κλάσεις που απαιτούν μόνιμη αποθήκευση
- **entity classes**: those that will reside in the main memory
  - Οι κλάσεις που θα βρίσκονται στην κύρια μνήμη
- **boundary classes** (interfaces)
  - Οι διεπαφές
- **control classes**: specify business logic function
  - Επιχειρηματική λειτουργίες

Παράδειγμα: Διάγραμμα Κλάσεων Συστήματος Κράτησης Ραντεβού (εννοιολογικό επίπεδο)



Taken from Dennis et al. 2005

Παράδειγμα: Διάγραμμα Πακέτων του Συστήματος Κράτησης Ραντεβού



Ποιες κλάσεις να βάλουμε σε ένα πακέτο; Which classes to put in which packages ?

Δυο γενικές αρχές:

- **Common Closure Principle**
  - the classes in a package should need **changing for similar reasons**
- **Common Reuse Principle**
  - the classes in a package should all be **reused together** (semantically close)

These principles drive us to the notion of ***cohesion***

Πακετοποίηση Κλάσεων: Συμβουλές

A well structured package

- is **cohesive** (providing a crispy boundary around a set of related elements)
- is **loosely coupled** (exporting only those elements other packages really need to see and importing only those elements necessary and sufficient for the elements in the package)
- is **not deeply nested** (human understanding is limited)

Cohesion =συνεκτικότητα/συνοχή, Cohesive = συνεκτικό
coupling = σύζευξη, loosely coupled = χαλαρά ζευγμένα

Packaging relates to clustering

Ομαδοποίηση
Clustering

- ***Clustering*** is the process of grouping similar objects into naturally associated subclasses.
- This process results in a set of "clusters" which somehow describe the underlying objects at a more abstract or approximate level.
- The process of clustering is typically based on a "similarity measure" which allows the objects to be classified into separate natural groupings.
- A *cluster* is then simply a collection of objects that are grouped together because they collectively have a strong internal similarity based on such a measure.
- A *similarity measure* (or *dissimilarity measure*) quantifies the conceptual distance between two objects, that is, how alike or disalike a pair of objects are.
  - Determining exactly what type of similarity measure to use is typically a domain dependent problem.

Ομαδοποίηση
Clustering

A clustering of a set N is a partition of N, i.e. a set $C_1,\ldots, C_k$ of subsets of N, such that:

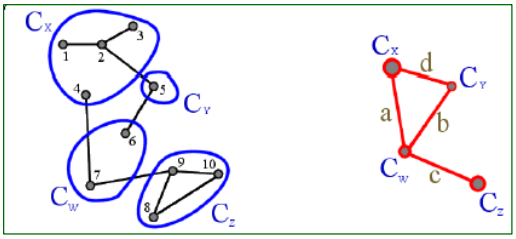
$C_1 \cup \ldots \cup C_k = N$ and $C_i \cap C_j = \varnothing$, for all $i \neq j$.

- Clustering is used in areas such as:
  - medicine, anthropology, economics, soil analysis, data mining
  - software engineering (reverse engineering, program comprehension, software maintenance)
  - information retrieval
- In general, any field of endeavor that necessitates the analysis and comprehension of large amounts of data may use clustering.

Packaging relates to graph clustering

Ομαδοποίηση Γράφου
Graph Clustering

- Graph clustering deals with the problem of clustering a graph
  - grouping similar nodes of a graph into a set of subgraphs

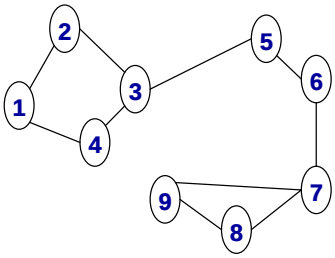


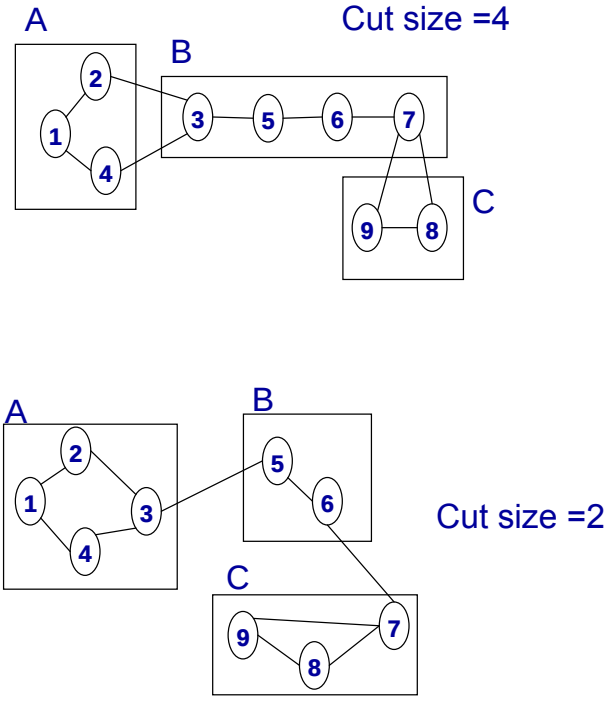
Ποιοτικά κριτήρια μεθόδων ομαδοποίησης γράφων
Quality criteria for graph clustering methods

Graph clustering methods should produce clusters with
high cohesion and low coupling

- high cohesion:
  - there should be many internal edges
- low "cut size":
  - The cut size (else called *external cost)* of a clustering measures how many edges are external to all sub-graphs, that is, how many edges cross cluster boundaries.
- Uniformity of cluster size is also often desirable.
  - A uniform graph clustering is where $|C_i|$ is close to $|C_j|$ for all i,j in {1..k}

Παράδειγμα





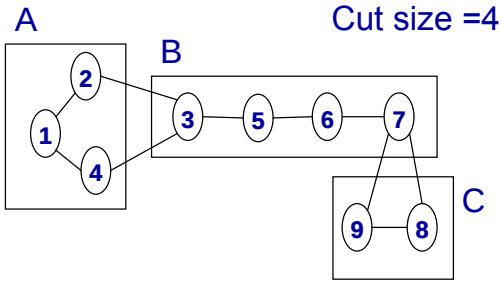
Μέτρα Ποιότητας για ομαδοποίηση γράφων
Quality Measures for Graph Clustering

- There are several. One well known is the **CC measure (Coupling-Cohesion measure)**

$$CC = \frac{|E^{in}| - |E^{ex}|}{|E|}$$

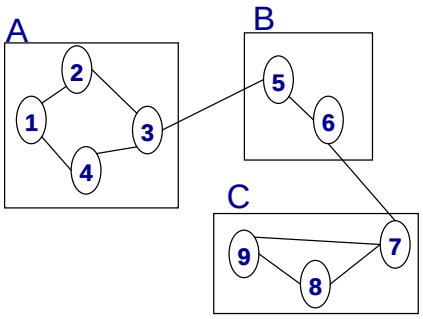
- E^{in}: the "internal" edges: those that connect nodes of the same cluster
- E^{ex}: the "external" edges: those that cross cluster boundaries
- maximum value of CC: 1
  - when all edges are internal
- minimum value of CC: -1
  - when all edges are external

Παράδειγμα



Cut size =4

$$CC = \frac{6-4}{10} = 0.2$$

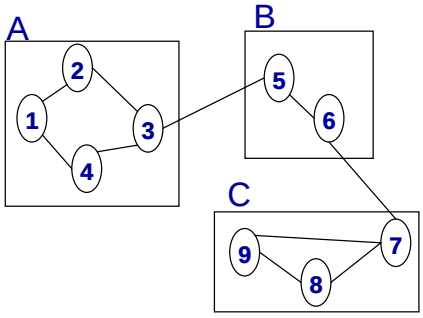


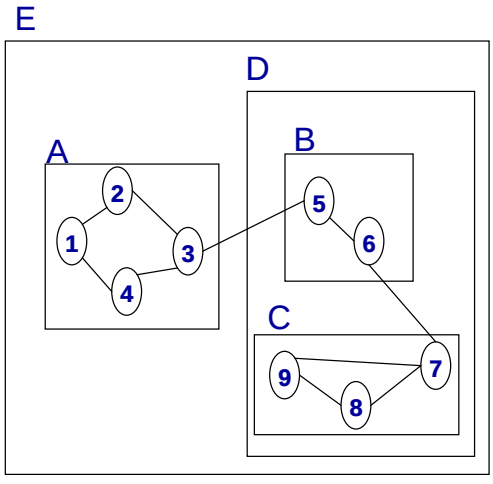
Cut size =2

$$CC = \frac{8-2}{10} = 0.6$$

Ιεραρχική Ομαδοποίηση Γράφων
Hierarchical Graph Clustering

- The clusters of the graph can be clustered themselves to form a higher level clustering, and so on.
- A hierarchical clustering is a collection of clusters where any two clusters are either disjoint **or** nested.





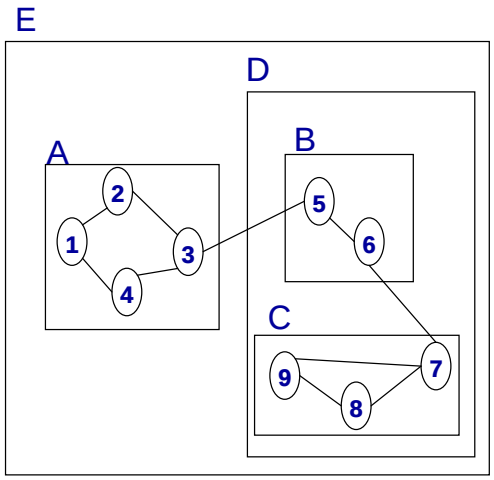
Ιεραρχικά Ομαδοποιημένος Γράφος
Hierarchical Clustered Graph

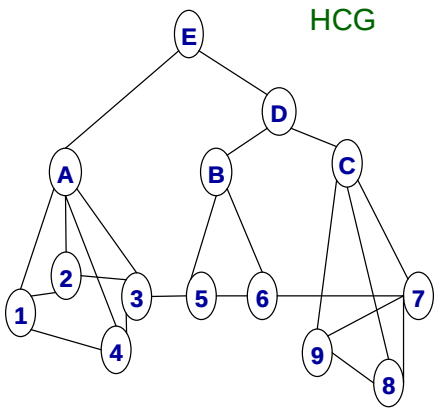
A Hierarchical Clustered Graph (HCG) is a pair (G,T) where

G is the underlying graph, and

T is a rooted tree such that the leaves of T are the nodes of G.

(the tree T represents an inclusion relation: the leaves of T are nodes of G, the internal nodes of T represent a set of graph nodes, i.e. a cluster)



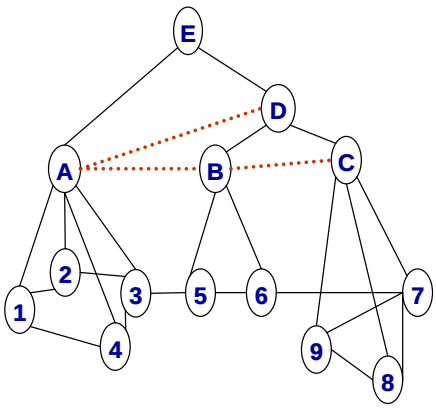


Συνεπαγόμενες Ακμές
Implied Edges

Implied edges: edges between the internal nodes.

Two clusters are connected iff the nodes that they contain are related.

Multiple implied edges (between the same pair of clusters) can be ignored or summed up to form weighted implied edges. Thresholding can applied in order to filter out some implied edges



A Hierarchical Compound Graph is a triad (G,T, I) where (G,T) is a hierarchical clustered graph (HCG), and I the set of implied edges set.

Πακέτα και Ομαδοποίηση Γράφων
Packages and Graph Clustering

Ποιος είναι ο γράφος G=(N,E) στην περίπτωση μας?

- N: Κλάσεις
- E: οι διάφορες σχέσεις εξάρτησης
  - δομικές, π.χ. συσχετίσεις κλάσεων και οι σχέσεις γενίκευσης/εξειδίκευσης
  - συμπεριφορικές, π.χ. τα βέλη στα διαγράμματα επικοινωνίας
  - Οι απλές εξαρτήσεις (π.χ. λόγω παραμέτρων)
  - "semantical" (similarity of use/purpose: hard to infer automatically]

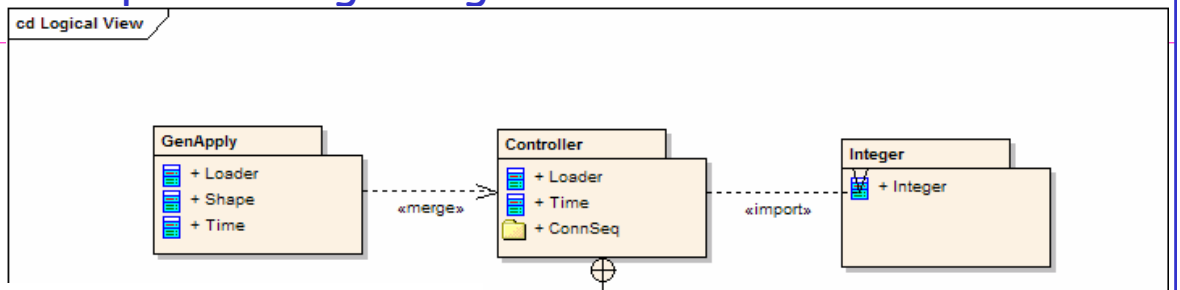
Ποιος είναι ο ιεραρχικά ομαδοποιημένος γράφος (G,T, I) στην περίπτωση μας?

- G (όπως πριν)
- T
  - Φύλλα του T: οι κόμβοι του N (δηλαδή οι κλάσεις)
  - Εσωτερικοί κόμβοι αμέσως πάνω από τα φύλλα: πακέτα
  - Εσωτερικοί κόμβοι υψηλότερου επιπέδου: πακέτα πακέτων
- I : οι συνεπαγόμενες ακμές εδώ είναι εξαρτήσεις μεταξύ των πακέτων

Packaging and Clustering: *Homework*

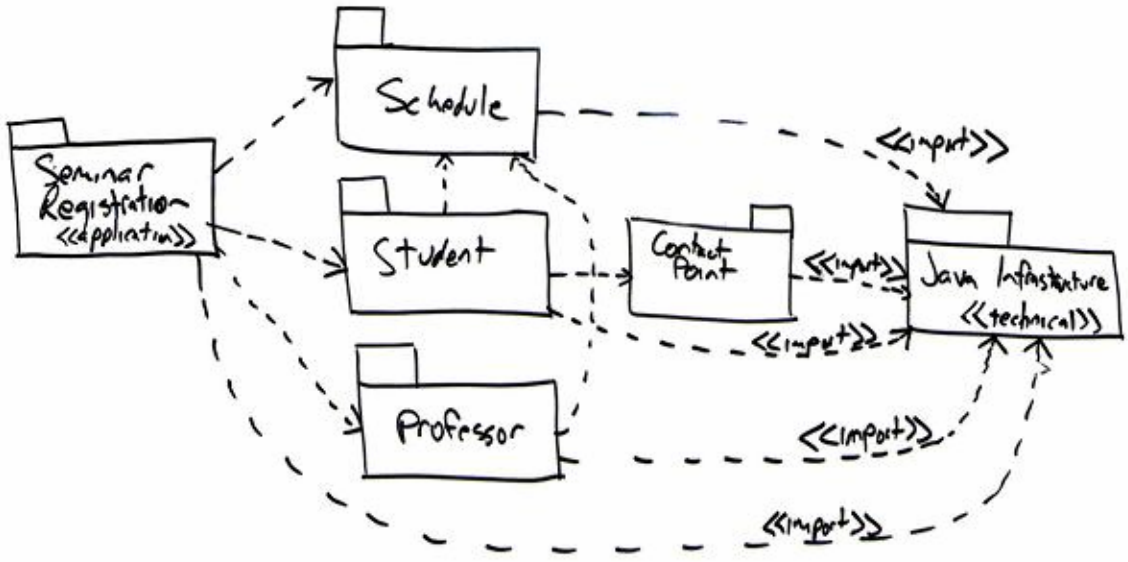
- Assume we have an implemented system whose classes are not packaged.
- *How could you partition them to packages?*

Example: Package diagrams in EA



A «**merge**» connector between two packages defines an implicit generalization between elements in the source package, and elements with the same name in the target package. The source element definitions are expanded to include the element definitions contained in the target. The target element definitions are unaffected, as are the definitions of source package elements that don't match names with any element in the target package.

«**import**» connector: indicates that the elements within the target package, which in this example is a single class, use unqualified names when being referred to from the source package. The source package's namespace gains access to the target classes; the target's namespace is not affected.



Περίληψη
Summary

- Alternative strategies to design and create the new system: *Custom building, packaged software, outsourcing*
- *The object-oriented approach* to analysis and design is based on use-case modeling, is architecture centric, and supports functional, static and dynamic views of the system.
- When evolving analysis into design models, it is important to review the analysis models then add system environment information.
- *Packages and package diagrams* help provide structure and less complex views of the new system.
  - Set the context
  - Cluster classes together based on shared relationships
  - Model clustered classes as a package
  - Identify dependency relationships among packages
  - Place dependency relationships between packages