

Ανάλυση Πληροφοριακών Συστημάτων

Εαρινό Εξάμηνο 2020-2021

Lec01

Εισαγωγικό Μάθημα

23/02/2021

Διδάσκων: Δρ. Γεώργιος Χρ. Μακρής

Ανάλυση Πληροφοριακών Συστημάτων

Α.Π.Θ Τμήμα Πληροφορικής Μάθημα 6ου εξαμήνου

Διδάσκων: Γεώργιος Χρ. Μακρής

Οι Διαφάνειες και το Υλικό Μαθήματος στηρίχθηκαν
στο υλικό διδασκαλίας του κ. Ι. Σταμέλου

Η ιστοσελίδα του μαθήματος

■ <https://gmakris.wordpress.com/>

■ AUTH e-learning

<https://elearning.auth.gr/course/view.php?id=8170>

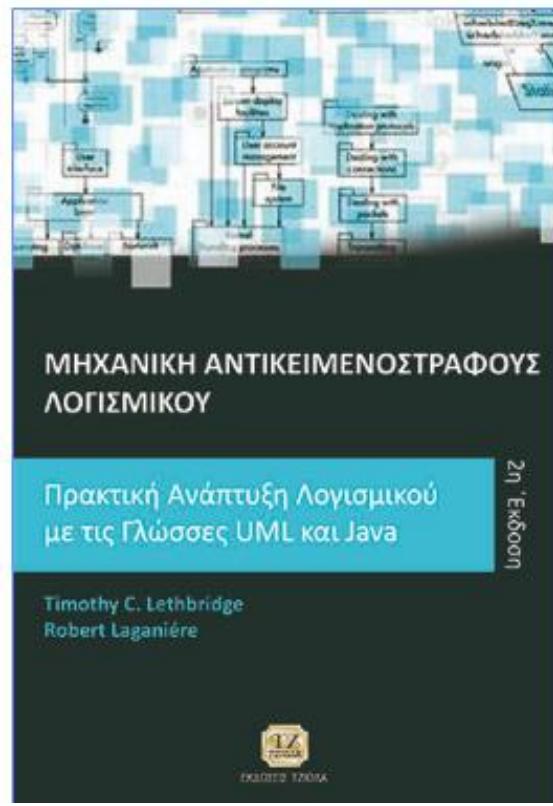
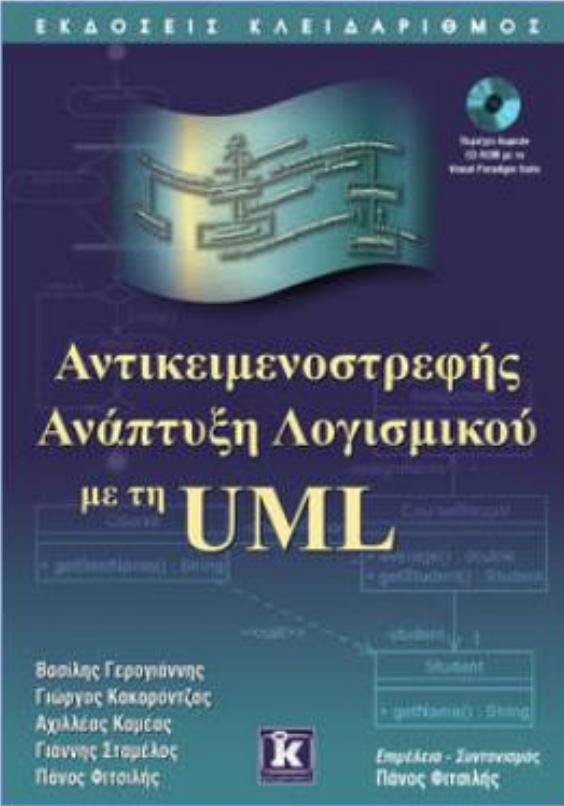
■ Ανακοινώσεις

■ Διαφάνειες μαθημάτων θεωρίας

■ Ασκήσεις – απαντήσεις εργαστηρίου

■ Υπερσύνδεσμοι, ενδιαφέροντα άρθρα και νέα

Βιβλία



Πάνος Φιτσιλής

**Σύγχρονα
Πληροφοριακά
Συστήματα
Επιχειρήσεων**

ERP-CRM-BPR

Επίκουρη Καθηγητής
Επίκουρη Καθηγητής
Επίκουρη Καθηγητής

HEALINK

Επίκουρη Καθηγητής
Επίκουρη Καθηγητής
Επίκουρη Καθηγητής

Επιλογή από Εύδοξο

Ανάλυση Πληρ. Συστημάτων

Δωρεάν από Κάλλιπο

Εισαγωγή

Περιεχόμενο μαθήματος

- 1 Εισαγωγή στην Ανάλυση Πληροφοριακών Συστημάτων
- 2 Πληροφοριακά Συστήματα Οργανισμών/Επιχειρήσεων – Τύποι
- 3 Πληροφοριακά Συστήματα Οργανισμών/Επιχειρήσεων – Παραδείγματα
- 4 Επιχειρηματική Αρχιτεκτονική – Εργαλείο ArchiMate*
- 5 Επιχειρηματικές διεργασίες, ανασχεδιασμός Ε.Δ. (BPR)*
- 6 Rational Unified Process / Περιπτώσεις Χρήσης / Μοντέλα Π.Χ.
- 7 Μοντέλα πεδίου*
- 8 Διαγράμματα Συνεργασίας*
- 9 Διαγράμματα Ακολουθίας*
- 10 Διεργασία ICONIX
- 11 Ολοκλήρωση Εφαρμογής*
- 12 Μοντέλα Ποιότητας - Μετρικές*
- 13 Πρότυπα Σχεδίασης - Επανάληψη

Εργαστηριακές Ασκήσεις

- Η παρακολούθηση είναι προαιρετική
- Γίνεται ανάλογα με τη διάλεξη
- Δεύτερη-τρίτη ώρα του μαθήματος

Λογισμικό που θα χρησιμοποιηθεί (Εργαλεία):

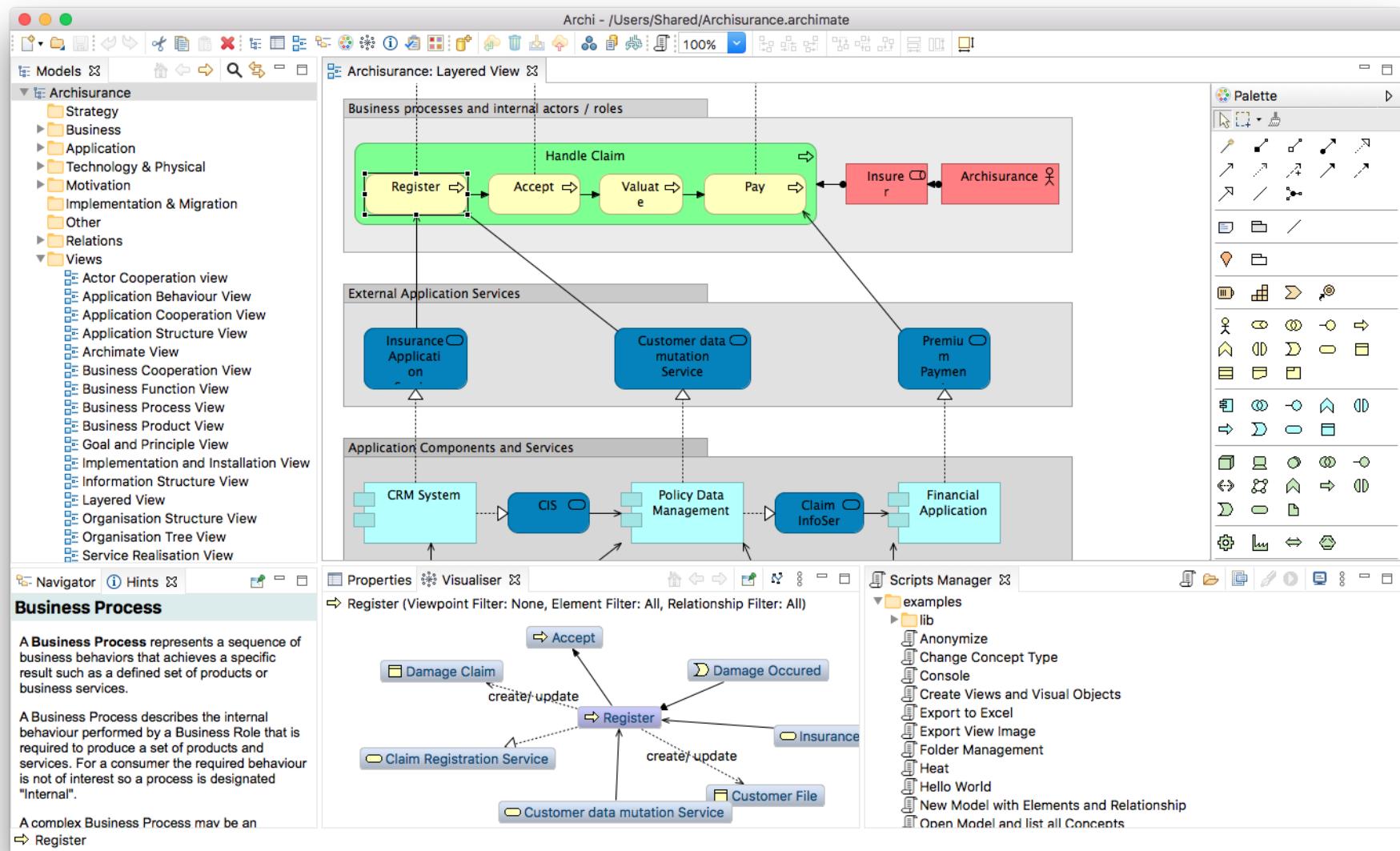
Archi (Πλ. Συστ.): <https://www.archimatetool.com/>

UMLet: <https://www.umlet.com/>

MODELIO: <https://www.modelio.org/>

ArgoUML, VisualParadigm, NetBeans

Archi (Πλ. Συστ.): <https://www.archimatetool.com/>



UMLet: <https://www.umlet.com/>

eclipse-workspace - PerpetuumMobile/doc/new_diagram.uxf - Eclipse IDE

File Edit Navigate Search Project Run Window Help UMLet

Package Explorer *new_diagram.uxf

PerpetuumMobile

JRE System Library [JavaSE-1.8]

resources.jar - C:\Program Files

rt.jar - C:\Program Files (x86)\Java

jse.jar - C:\Program Files (x86)\

jce.jar - C:\Program Files (x86)\

charsets.jar - C:\Program Files (x86)\

jfr.jar - C:\Program Files (x86)\

access-bridge-32.jar - C:\Program

clldrdata.jar - C:\Program Files (

dnsns.jar - C:\Program Files (x8

jaccess.jar - C:\Program Files (x

jfxrt.jar - C:\Program Files (x86)

localdatajar - C:\Program File

nashorn.jar - C:\Program Files (

sunec.jar - C:\Program Files (x8

sunje_providerjar - C:\Program

sumscapijar - C:\Program File

sunpkcs11.jar - C:\Program File

zipsfs.jar - C:\Program Files (x86,

src

doc

new_diagram.uxf

*new_diagram.uxf

-> CLASSATTRIBUTE: Long
#Operation(i: int): int
+AbstractOperation()

Responsibilities
-- Resp1
-- Resp2

Component

A complex combined fragments

Lifeline A Lifeline B Lifeline C Lifeline D

execute() d=duration

combined fragment

{x=0}

loop [1..10]

Message

[x>0]

alt [else]

Note..

Use case 1

Use case 3

Use case 2

Collaboration

EmptyPackage

Package 1

Actor

Properties

<<Stereotype>>
Package::FatClass
(Some Properties)
--
-id: Long
-ClassAttribute: Long_

#Operation(i: int): int
/+AbstractOperation()
--
Responsibilities
-- Resp1
-- Resp2*

SimpleClass AbstractClass

Interface Operation1 Operation2

teaches to

0..n «some Stereotype» 0..1

«instanceOf»

object: Class id: Long="36548" [waiting for message]

Qualification 1..5,6

Use case 1

Use case 3

Use case 2

Collaboration

EmptyPackage

Package 1

Actor

role A role B multiple lines are possible and label positions can be customized role C role D

msg A msg B msg C msg D

msg B

msg C

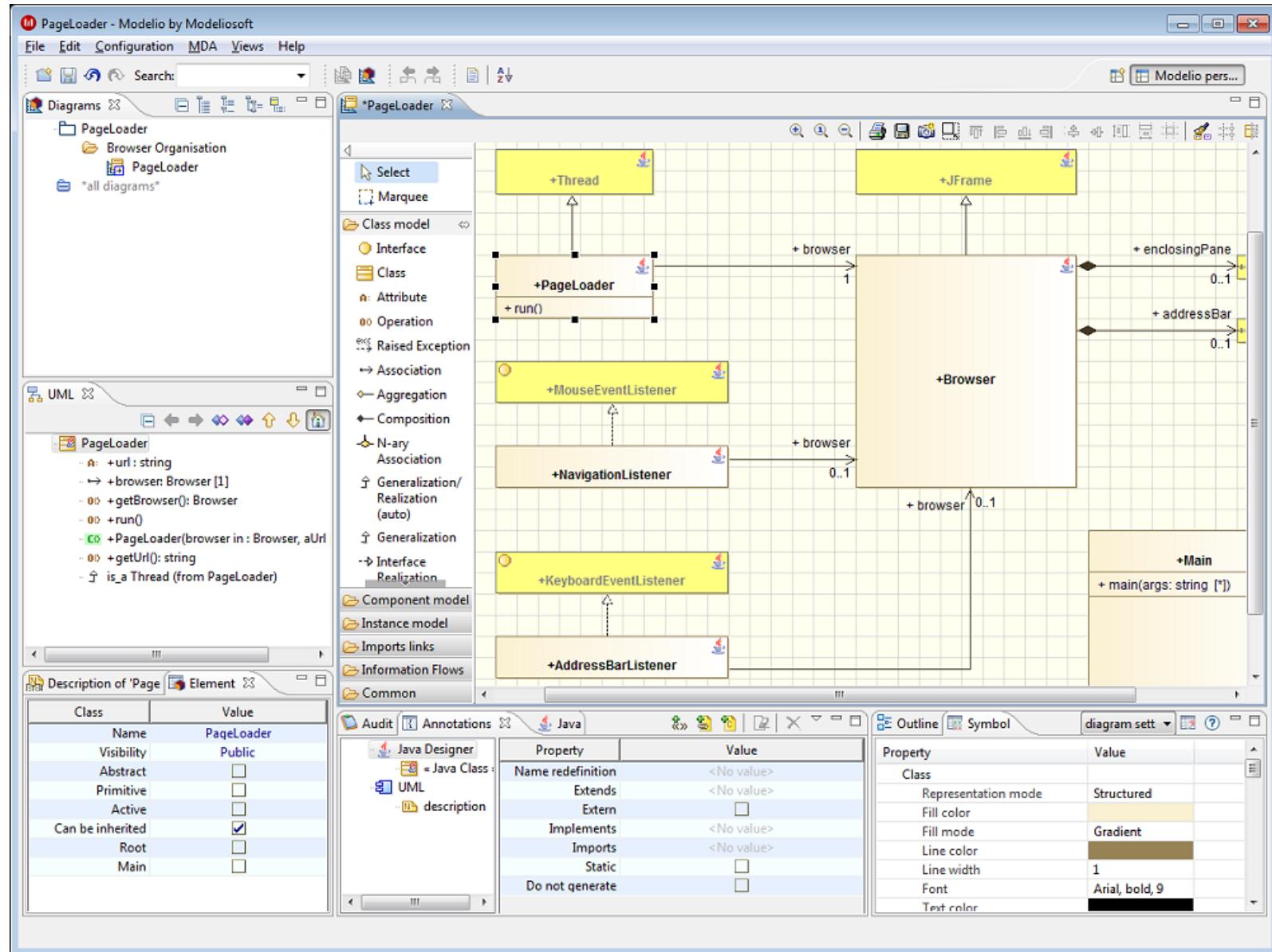
msg D

«Deployment Descriptor» Main

Quick Access

Diagram showing a UML sequence diagram titled "A complex combined fragments". The diagram features four lifelines: A, B, C, and D. Lifeline A sends a "execute()" message to itself with duration "d". Lifeline B receives this message and enters a "combined fragment". Inside the fragment, a loop labeled "loop" iterates from 1 to 10. An "alt" block contains two parallel regions: one for "x=0" and one for "x>0". The "x=0" region has an "else" branch. Lifeline C sends a "Message" to the "x>0" region. Lifeline D receives a message from the "x>0" region. The "Properties" panel shows class and package stereotypes, responsibilities, and deployment descriptor details.

MODELIO: <https://www.modelio.org/>



Αξιολόγηση / Εξέταση

- 40% Εργασία (συνυπολογίζεται μόνο αν στις γραπτές εξετάσεις η βαθμολογία είναι ≥ 5)
- 60% Γραπτές Εξετάσεις

Η εργασία (1)

- Σε ένα έργο ΕΛΛΑΚ (πιθανά στο έργο στο οποίο έγινε η εργασία της Τεχνολογίας Λογισμικού):
- Εντοπίστε μία λειτουργία που είναι επιθυμητό να προστεθεί στο έργο
- Δημιουργήστε το τμήμα του διαγράμματος κλάσεων του έργου που ενδιαφέρει (reverse engineering)
- Προδιαγράψτε την Περίπτωση Χρήσης που αντιστοιχεί στη λειτουργία
- Δώστε τα διαγράμματα συνεργασίας, ακολουθίας για την Π.Χ., ενημερώστε το διάγραμμα κλάσεων
- Προαιρετικά, υλοποιήστε τη λειτουργία

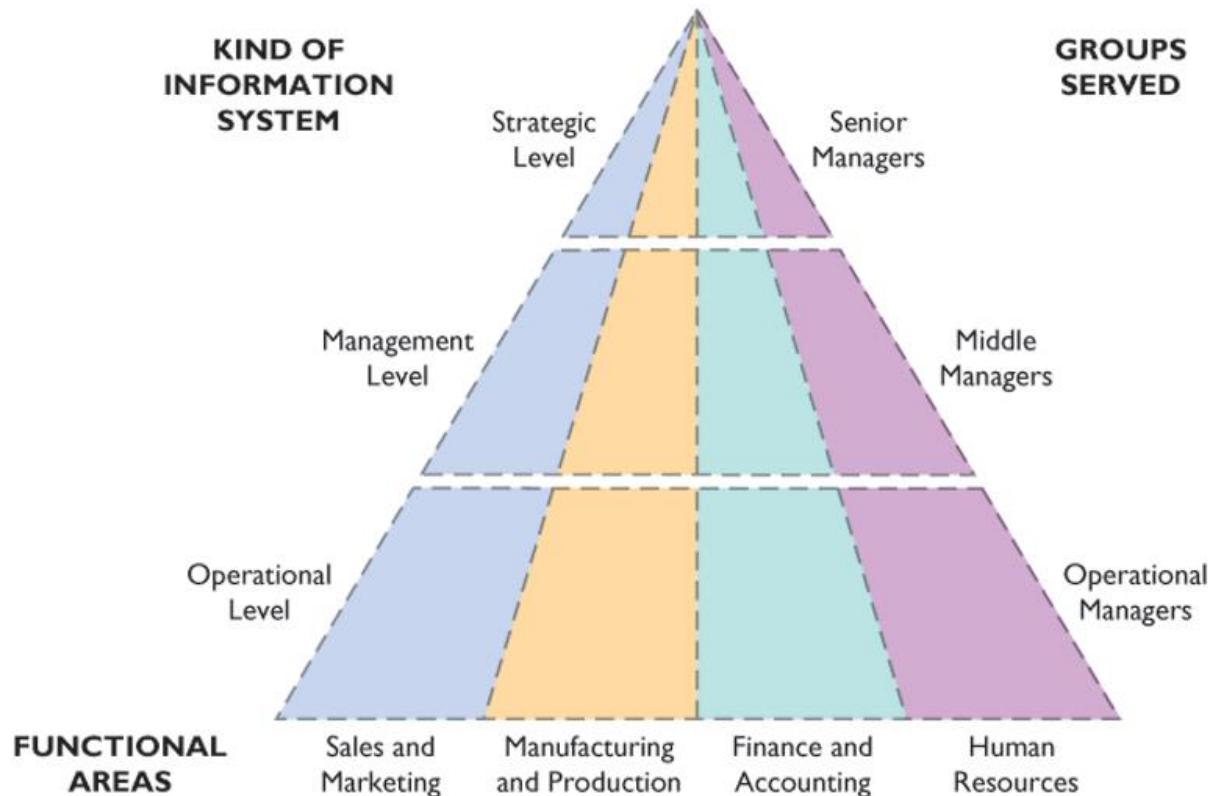
Εργασία: εναλλακτικά (2)

- Σε ένα έργο ΕΛΛΑΚ (πιθανά στο έργο στο οποίο έγινε η εργασία της Τεχνολογίας Λογισμικού):
 - Εφαρμόζετε ένα εργαλείο εντοπισμού προβλημάτων σχεδίασης (code smell detection, π.χ. JDeodorant) και στη συνέχεια κάνετε αναδόμηση (refactoring) είτε με χρήση του εργαλείου είτε “με το χέρι”
 - Εφαρμόζετε ένα εργαλείο μέτρησης του κώδικα και συγκρίνετε με πρότυπα ποιότητας (benchmarking). Μπορεί να γίνει με διάφορα ανεξάρτητα εργαλεία ή με ένα plugin στο IntelliJ

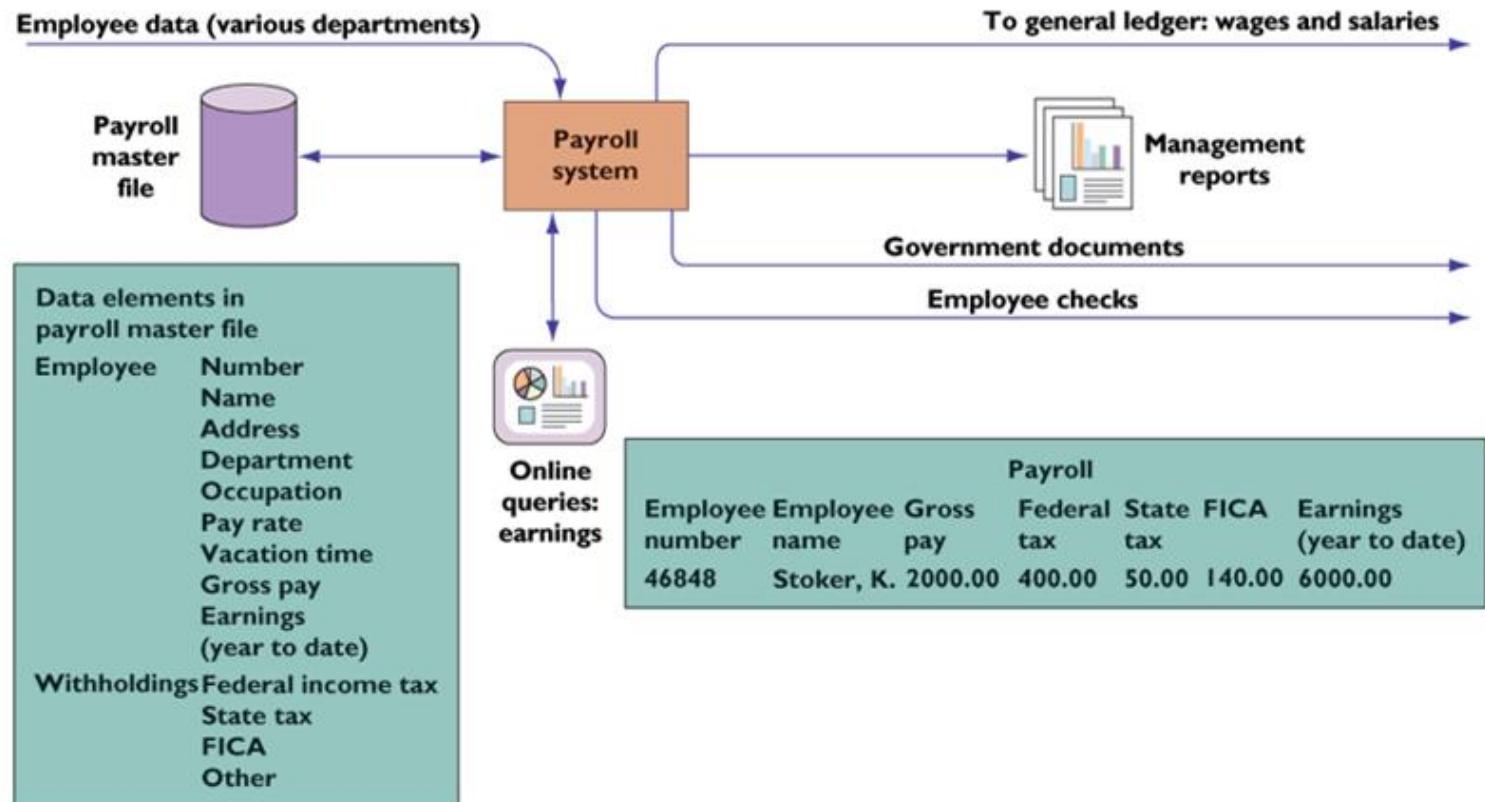
Εργασία: εναλλακτικά (3)

- Συμμετοχή στο εαρινό/θερινό σχολείο κώδικα της Μονάδας Αριστείας ΕΛΛΑΚ Α.Π.Θ.
- Θα διεξαχθεί αμέσως μετά το Πάσχα και η συμμετοχή θα περιοριστεί σε όσους είχαν συμμετάσχει στο σχολείο κώδικα της Τεχνολογίας Λογισμικού

Πληροφοριακά Συστήματα Επιχειρήσεων



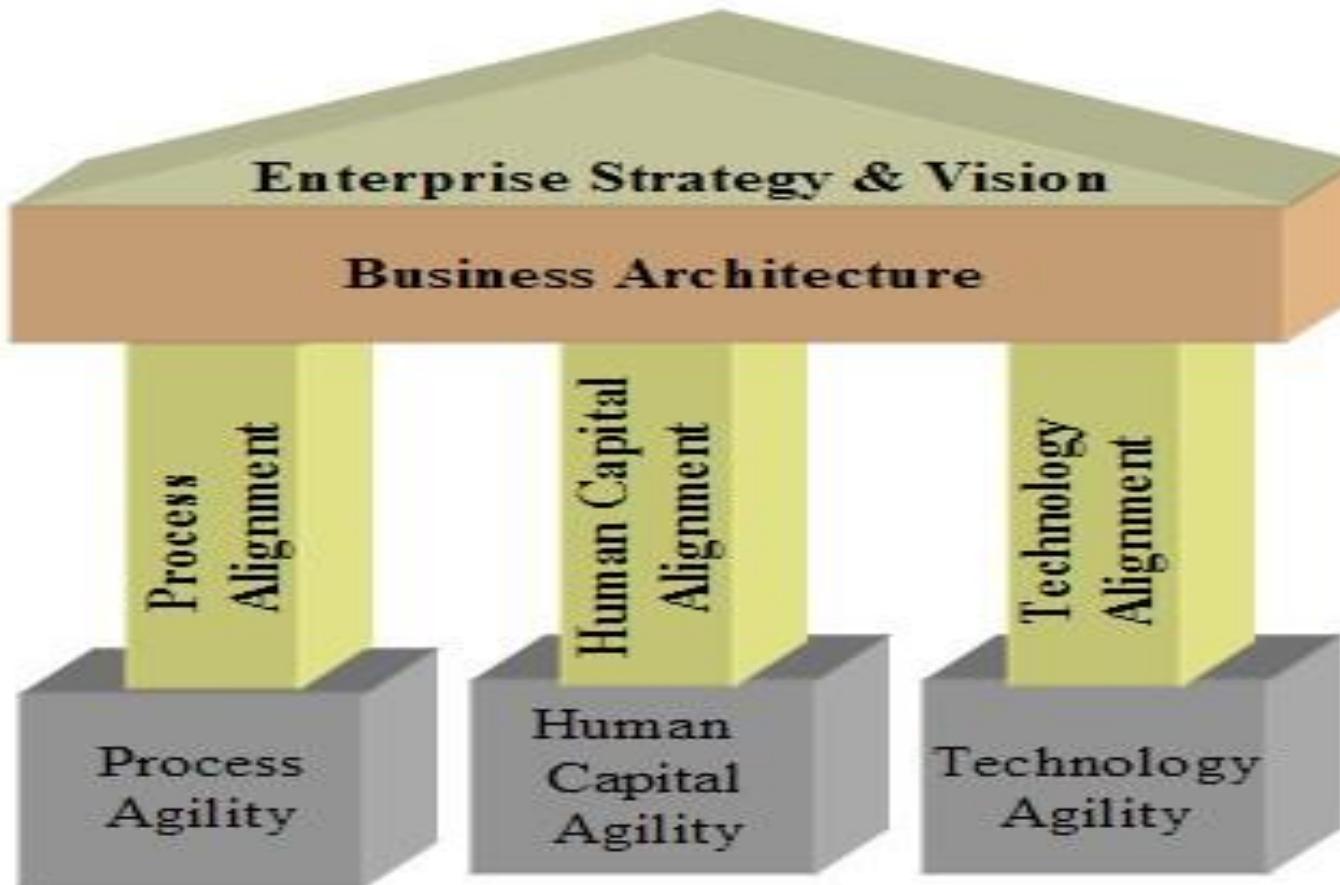
Παράδειγμα Πληροφοριακού Συστήματος



Επιχειρηματική Αρχιτεκτονική

- Έχει στόχο να αντιμετωπίσει τα προβλήματα της πολυπλοκότητας πληροφοριακών συστημάτων και της ελλιπούς ευθυγράμμισης των αναγκών της επιχείρησης με τα πληροφοριακά συστήματα
- Με την επιλογή της κατάλληλης αρχιτεκτονικής μπορεί μια επιχείρηση όχι μόνο να εξασφαλίσει τη βιωσιμότητά της σε ένα διαρκώς μεταβαλλόμενο περιβάλλον αλλά και να αυξήσει την ανταγωνιστικότητα και την ευελιξία της (ανταγωνιστικό πλεονέκτημα)
- Τέσσερα πρότυπα E.A. (Zachman, TOGAF, ...)
- Εργαλεία E.A. (ARIS, ADONIS, Archi,...)

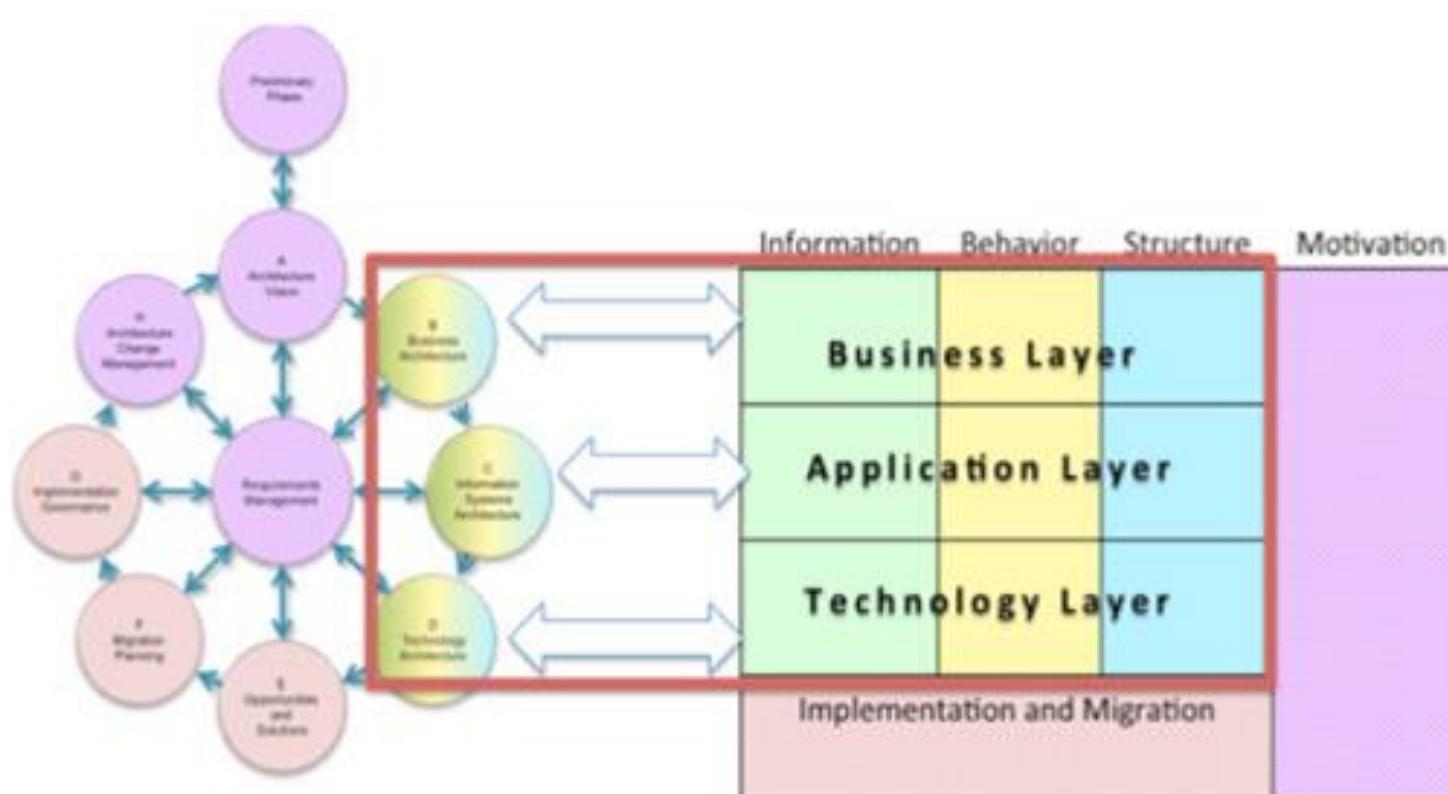
Αρχιτεκτονική Επιχειρήσεων



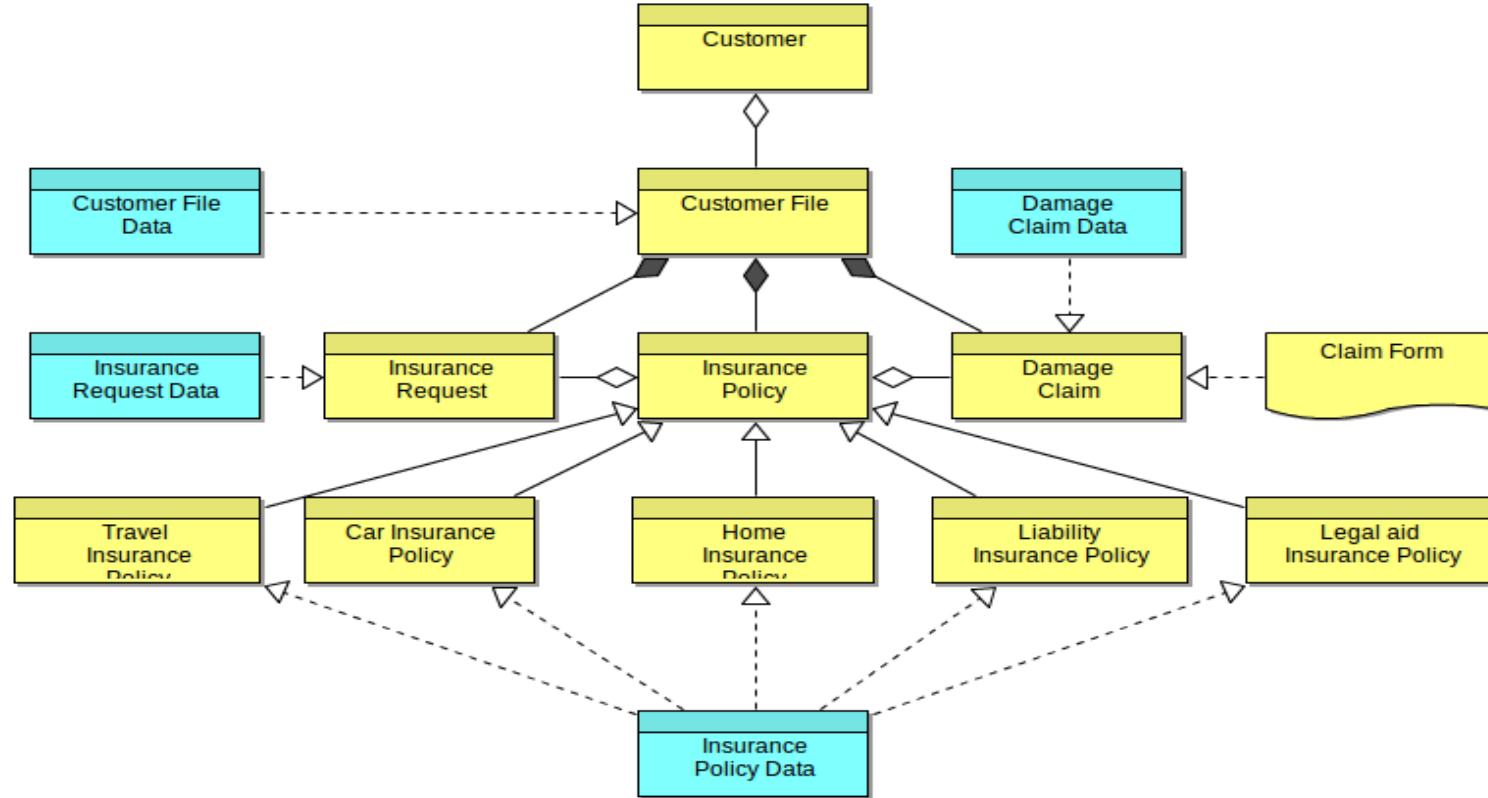
Πλαίσιο Zachman

	WHAT	HOW	WHERE	WHO	WHEN	WHY
	DATA	FUNCTION	NETWORK	PEOPLE	TIME	MOTIVATION
SCOPE (contextual)	List of things important to the business 	List of processes the business performs 	List of locations in which the business operates 	List of organisations important to the business 	List of event cycles significant to the business 	List of business goals/strategies
Planner	Entity = Class of business things	Process = Class of business process	Node = Major business locations	People = Major business unit	Time = Major Business Event Cycle	End/Means = Major Business Goal/Strategy
BUSINESS MODEL (Conceptual)	e.g., Semantic Model 	e.g., Business Process Model 	e.g., Business Logistics System 	e.g., Workflow Model 	e.g., Master Schedule 	Business Plan
Owner	Entity = Business Entity Relationship = Business	Process = Business I/O = Business Resource	Node = Business Location Link = Business Linkage	People = Organisation unit Work = Work Product	Time = Business Event Cycle Cycle = Business Cycle	End = Business Objective Means = Business Strategy
SYSTEM MODEL (Logical)	e.g., Logical Data Model 	e.g., Application Architecture 	e.g., Distributed System Model 	e.g., Human Interface Architecture 	e.g., Processing Structure 	e.g., Business Rule Model
Designer	Entity = Data Entity Relationship = Data Relationship	Process = Application Function I/O = User Views	Node = IS Function Relationship = Line Characteristics	People = Role Work = Deliverable	Time = System Event Cycle Cycle = Processing Cycle	End = Structural Assertion Means = Action Assertion
TECHNOLOGY MODEL (Physical)	e.g., Physical Data Model 	e.g., System Design 	e.g., Technology Architecture 	e.g., Presentation Architecture 	e.g., Control Structure 	e.g., Rule Design
Builder	Entity = Segment/Table Relationship = Pointer/key	Process = Computer Function I/O = Data Elements/sets	Node = H/w /System s/w Relationship = Line Specifications	People = User Work = Screen Formats	Time = Execute Cycle = Component Cycle	End = Condition Means = Action
DETAILED REPRESENTATIONS (Out-of-context)	e.g., Data Definition 	e.g., Program 	e.g., Network Architecture 	e.g., Security Architecture 	e.g., Timing Definition 	e.g., Rule Specification
Subcontractor	Entity = Field Relationship = Address	Process = Language Statement I/O = Control Block	Node = Address Link = Protocol	People = Identity Work = Job	Time = Interrupt Cycle = Machine Cycle	End = Sub-condition Means = step
FUNCTIONING ENTERPRISE	e.g. DATA	e.g. FUNCTION	e.g. NETWORK	e.g. ORGANISATION	e.g. SCHEDULE	e.g. STRATEGY

Εργαλείο Archi



Διάγραμμα Application Behaviour



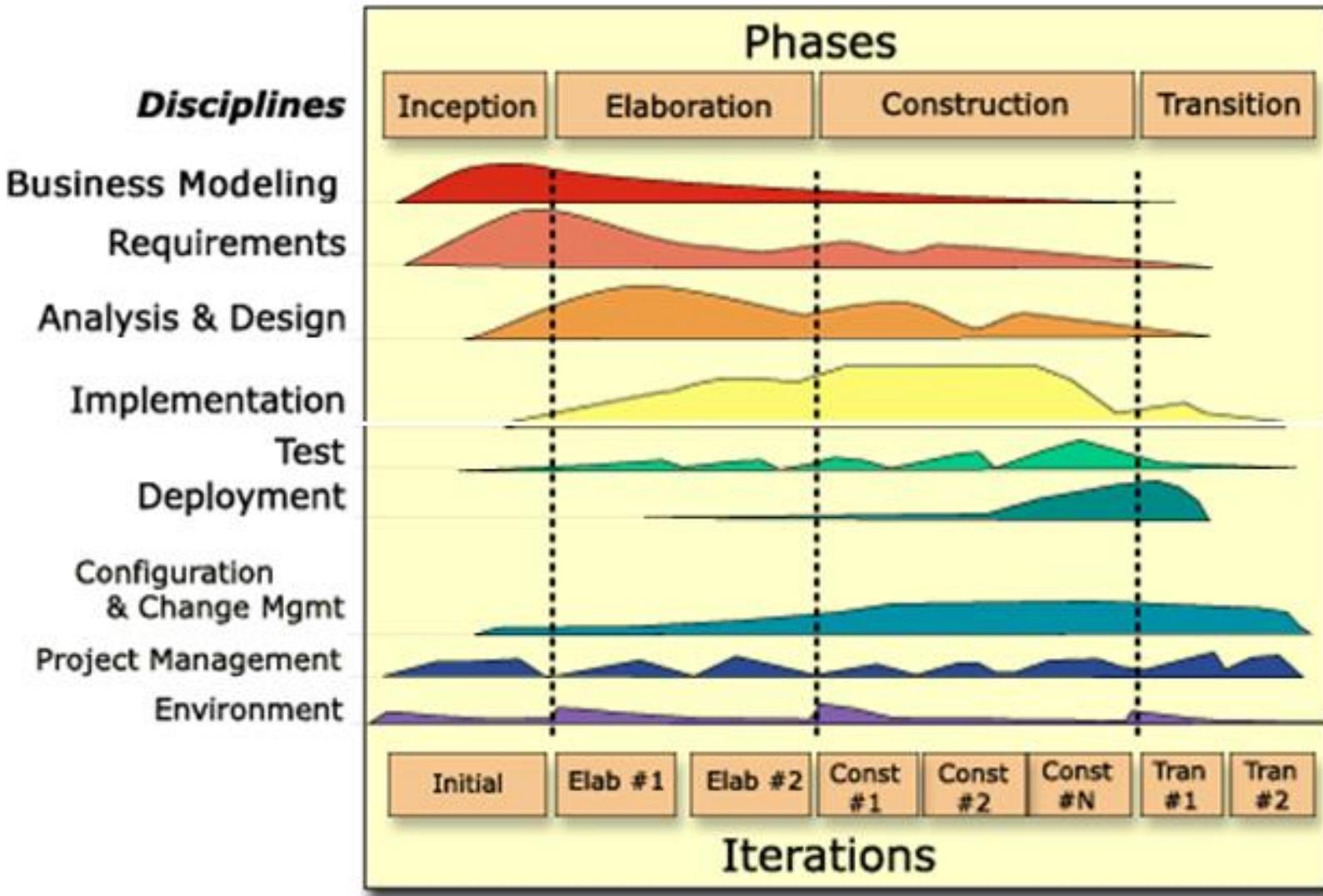
Αντικειμενοστρεφής Ανάλυση με τη UML

Ανάλυση Πληρ. Συστημάτων

Εισαγωγή

Τι είναι η UML

- Η UML αποτελεί μία γλώσσα απεικόνισης ή μοντελοποίησης ενός πληροφοριακού συστήματος βασισμένου σε αντικείμενα (αντικειμενοστραφούς συστήματος).
- Η UML αποτελεί την πρότυπη (standard) γλώσσα μοντελοποίησης αντικειμενοστραφών συστημάτων.
- Η UML αποτελεί πρότυπο του OMG (Object Management Group – www.omg.org).



Γιατί μοντελοποιούμε πληροφοριακά συστήματα;

- Η μοντελοποίηση ενός συστήματος παρέχει
- τη δυνατότητα της αφαίρεσης των ασήμαντων με αυτό λεπτομερειών και της εστίασης στις σημαντικές λεπτομέρειες του συστήματος
- τη δυνατότητα του πειραματισμού με διαφορετικές λύσεις ή προσεγγίσεις για το ίδιο πρόβλημα
- τη δυνατότητα ανάλυσης, σχεδιασμού, καταγραφής και παρακολούθησης της προόδου ενός έργου πληροφορικής
- μία κοινή γλώσσα για την επικοινωνία όσων εμπλέκονται στην κατασκευή του συστήματος
- Χωρίς ένα μοντέλο δεν είναι δυνατόν να προσεγγίσει κανείς την πολυπλοκότητα των σύγχρονων πληροφοριακών συστημάτων

Γιατί Αντικειμενοστρεφές Λογισμικό;

- Το αντικειμενοστρεφές λογισμικό είναι ευκολότερο στην αρχική του σύλληψη μια και τα αντικείμενα είναι – εν μέρει – οντότητες του υπαρκτού κόσμου
- Το αντικειμενοστρεφές λογισμικό είναι ευκολότερο στην εξέλιξή του.
- Η αντικειμενοστρεφής προσέγγιση επιτρέπει τη δημιουργία λογισμικού με βάση τα συστατικά (components)
- Σύγχρονες τεχνολογίες κατασκευής κατανεμημένων συστημάτων προσανατολισμένων στην σύνδεση επιχειρήσεων (Business-to-business – B2B) και στην σύνδεση επιχειρηματικών εφαρμογών (Enterprise Application Integration – EAI), όπως οι Υπηρεσίες του παγκόσμιου ιστού (Web Services), έχουν σαν τεχνολογικό υπόβαθρο αντικειμενοστρεφείς γλώσσες προγραμματισμού (π.χ. Java, C#).

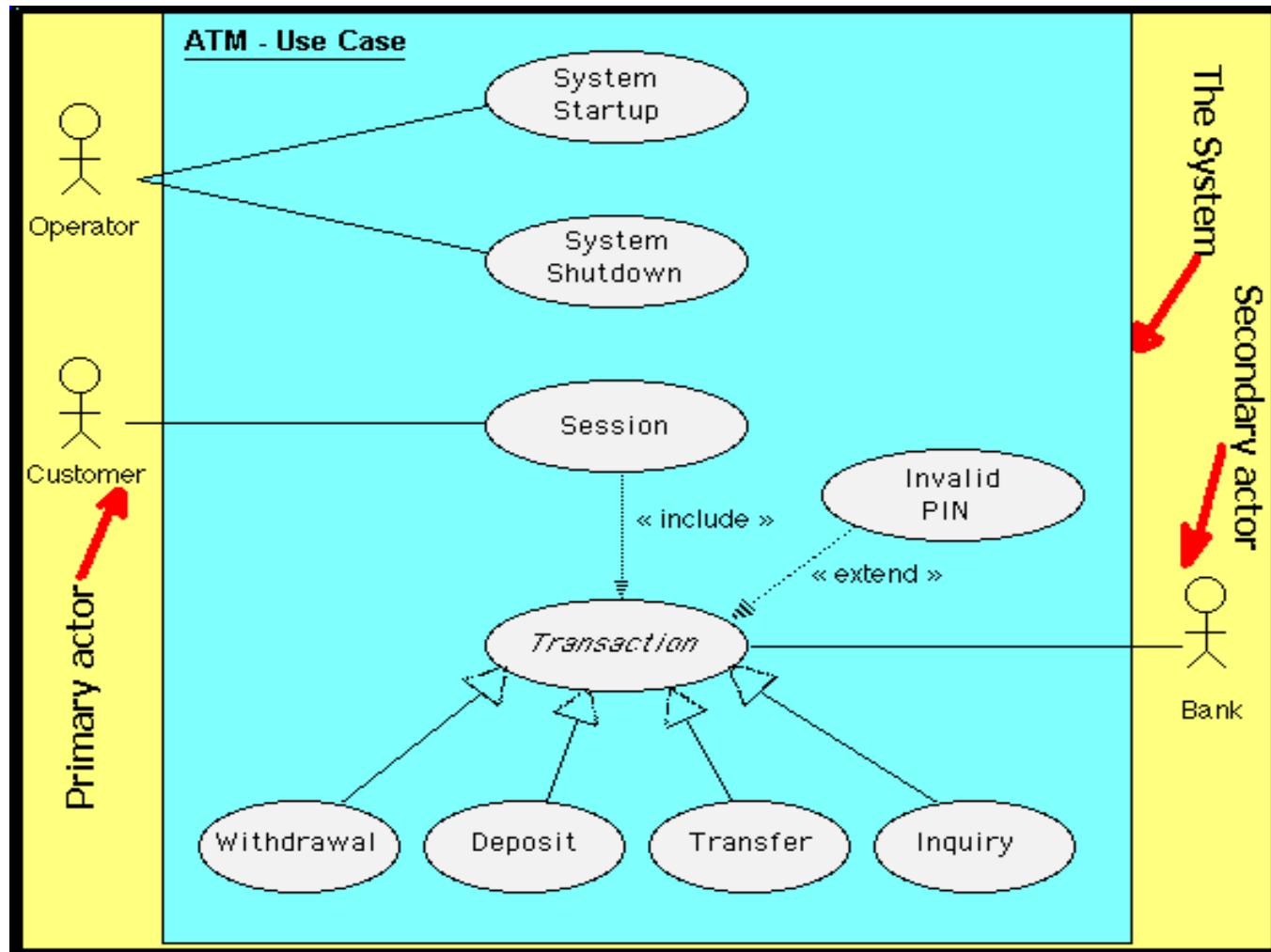
UML και Μεθοδολογίες Ανάπτυξης Λογισμικού

- Μία μεθοδολογία ανάπτυξης λογισμικού παρέχει μία συστηματική προσέγγιση στην διαδικασία ανάλυσης, σχεδίασης, κατασκευής και εξέλιξης ενός έργου πληροφορικής.
- Είναι μία σειρά σταδίων τα οποία περιγράφουν συγκεκριμένες εργασίες.
- **Η UML είναι ουδέτερη σε σχέση με τις μεθοδολογίες χωρίς να επιβάλλει κάποια συγκεκριμένη μεθοδολογία**

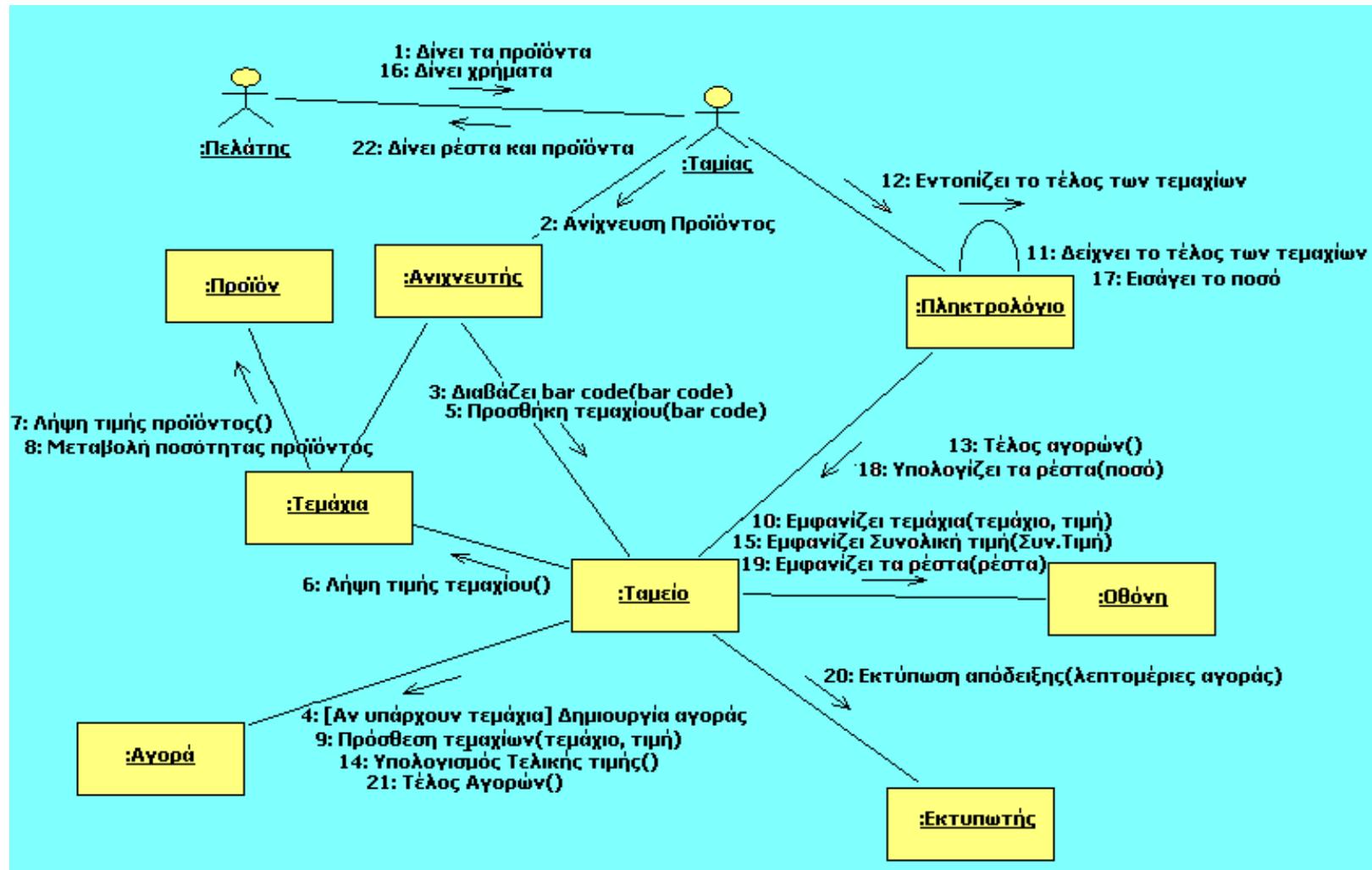
Τύποι Διαγραμμάτων της UML

- Διάγραμμα Περιπτώσεων Χρήσης (Use Case Diagram)
- Διάγραμμα Κλάσεων (Class Diagram)
- Διαγράμματα Συμπεριφοράς (Behavior Diagrams):
 - Διάγραμμα Καταστάσεων (Statechart Diagram)
 - Διάγραμμα Δραστηριοτήτων (Activity Diagram)
 - Διαγράμματα Αλληλεπίδρασης
 - Διάγραμμα Ακολουθίας (Sequence Diagram)
 - Διάγραμμα Συνεργασίας (Collaboration Diagram)
- Διαγράμματα Υλοποίησης (Implementation Diagrams):
 - Διάγραμμα Συστατικών (Component Diagram)
 - Διάγραμμα Διάταξης (Deployment Diagram)

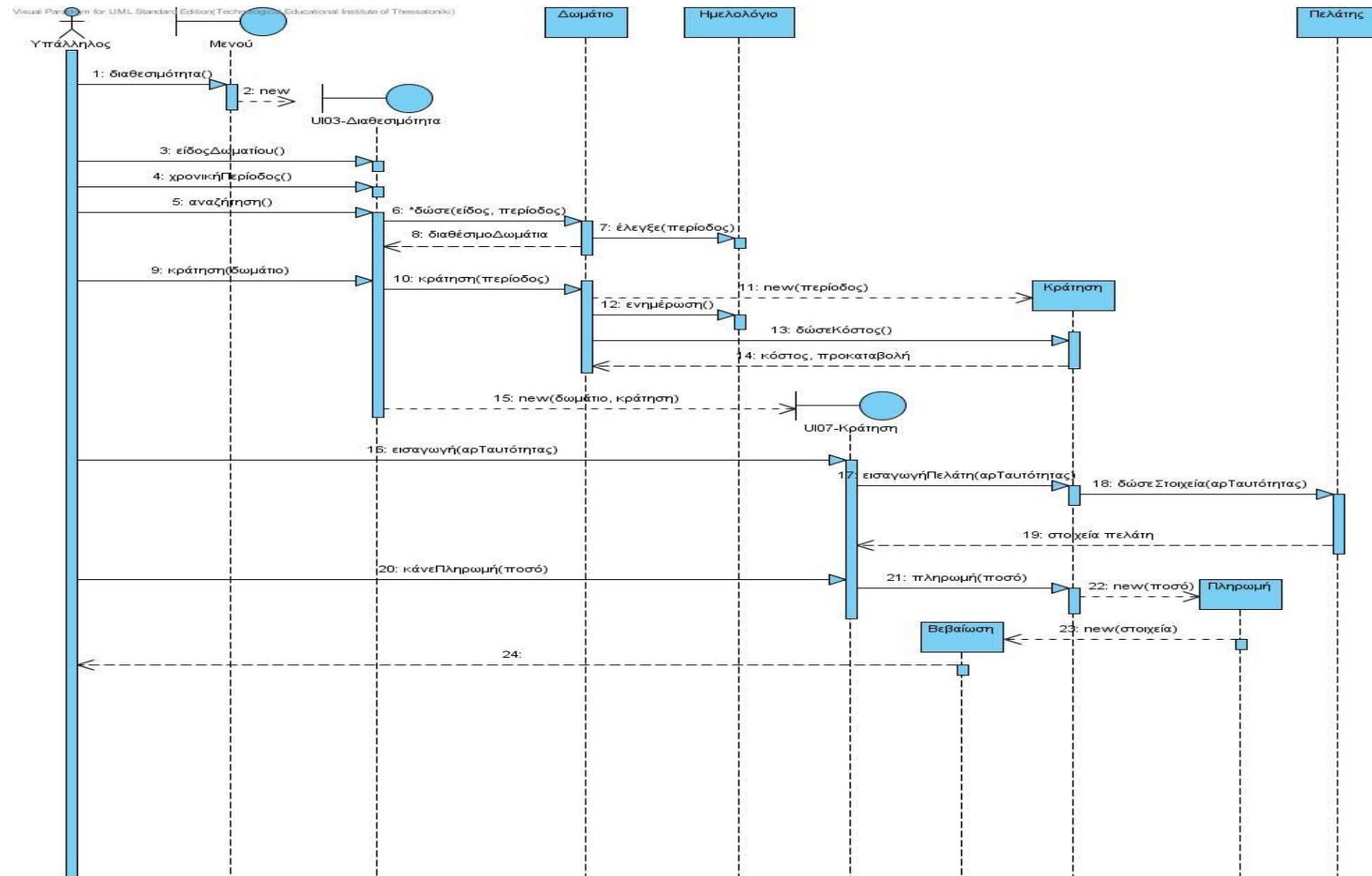
Μοντέλο Περιπτώσεων Χρήσης



Διάγραμμα Συνεργασίας



Διάγραμμα Ακολουθίας



Ανάλυση Πληρ. Συστημάτων

Εισαγωγή

Γιατί τόσοι διαφορετικοί τύποι διαγραμμάτων;

- Το σύστημα έχει στατικά στοιχεία, δυναμικά στοιχεία, στοιχεία υλοποίησης κοκ.
- Δεν είναι δυνατόν ένας και μόνο τύπος διαγράμματος να περιγράψει όλες αυτές τις διαφορετικές οπτικές γωνίες ενός συστήματος.
- Έτσι τυπικά χρησιμοποιούμε
- διαγράμματα κλάσεων για την περιγραφή των στατικών σχέσεων μεταξύ των κλάσεων,
- διαγράμματα συμπεριφοράς (κατάστασης, ακολουθίας κοκ) για την περιγραφή της δυναμικής συμπεριφοράς του συστήματος,
- διαγράμματα υλοποίησης (συστατικών, διάταξης) για την καταγραφή των λεπτομερειών υλοποίησης του συστήματος.
- Τα διαγράμματα περιπτώσεων χρήσης είναι μία ειδική περίπτωση υπό την έννοια πως δεν έχουν κάποια σχέση με αντικείμενα ή αντικείμενοστρεφή συστήματα.

Ανάλυση Πληροφοριακών Συστημάτων

Εαρινό Εξάμηνο 2020-2021

Lec02

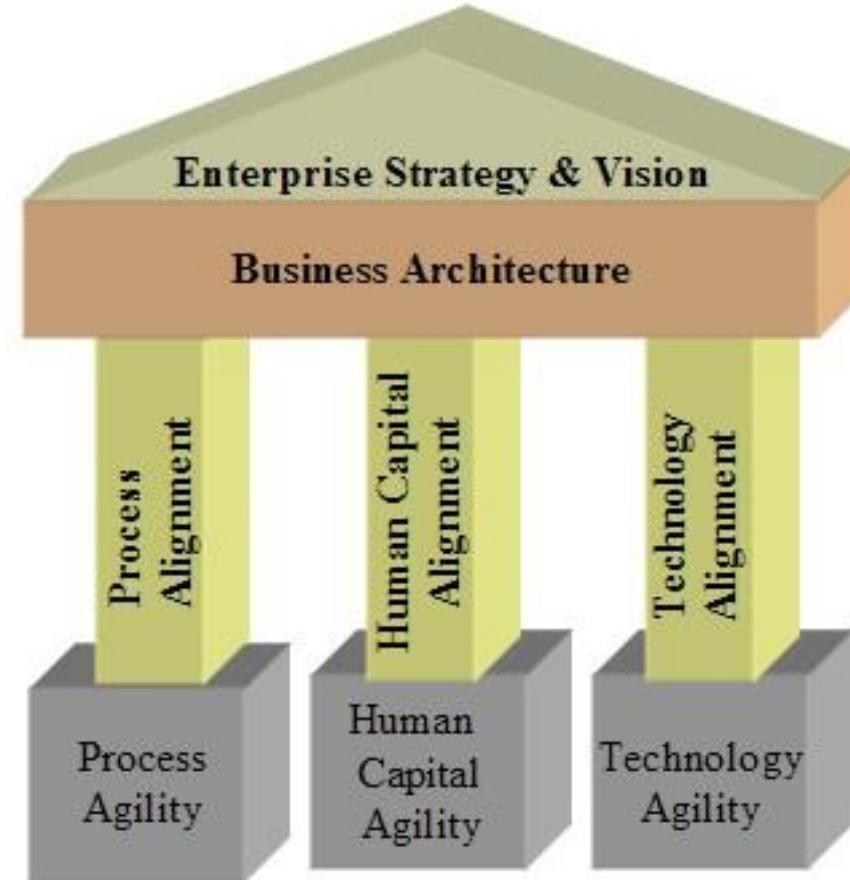
Επιχειρησιακές Αρχιτεκτονικές & Πλαίσια ΕΑ

02/03/2021

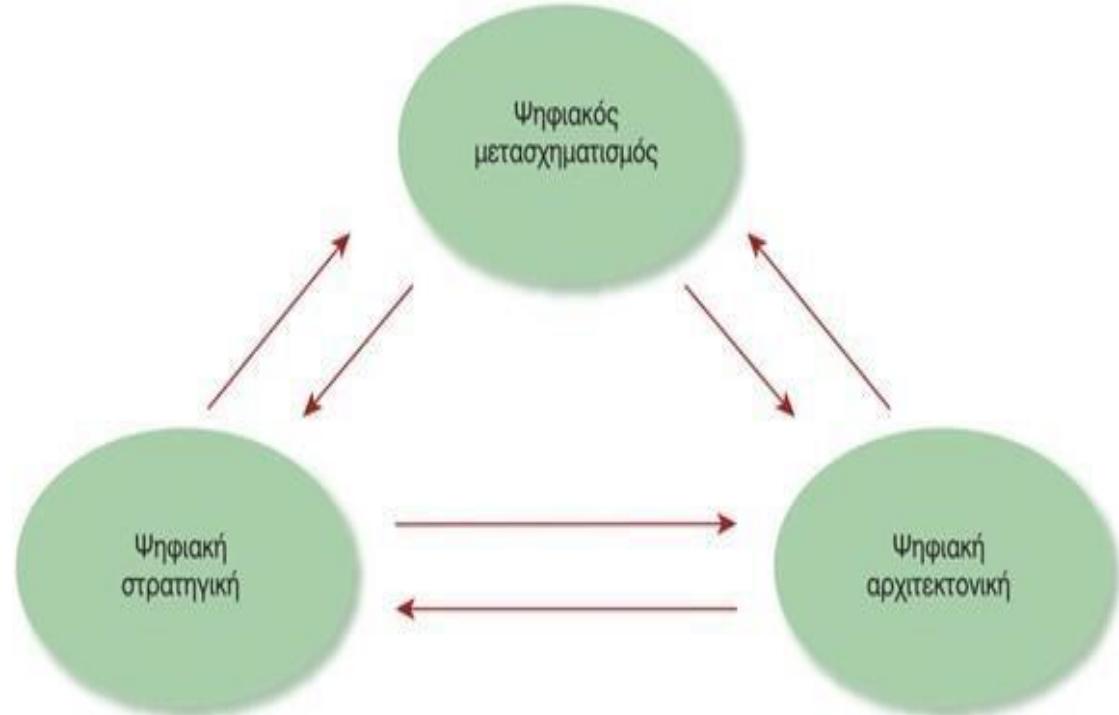
Διδάσκων: Δρ. Γεώργιος Χρ. Μακρής

Αρχιτεκτονική Επιχειρήσεων

- Πλαίσια



Ψηφιακή στρατηγική και πληροφοριακά συστήματα



ΕΙΚΟΝΑ 2.1

Σχέση ψηφιακής στρατηγικής με την αρχιτεκτονική των πληροφοριακών συστημάτων.

Η **Ψηφιακή στρατηγική** της επιχείρησης συνδέεται πλέον στενά με την αρχιτεκτονική των πληροφοριακών συστημάτων και επηρεάζει άμεσα το επιχειρηματικό και λειτουργικό μοντέλο της επιχείρησης, τον τρόπο αλληλεπίδρασης με τους πελάτες και τους συνεργάτες, ενώ επιτρέπει στην επιχείρηση να καινοτομεί. Επομένως ο ρόλος της αρχιτεκτονικής των πληροφοριακών συστημάτων είναι σημαντικότερος από ποτέ και είναι αλληλένδετος με την Ψηφιακή στρατηγική, αλλά και τον Ψηφιακό μετασχηματισμό της επιχείρησης.

Η αρχιτεκτονική ενός πληροφοριακού συστήματος περιλαμβάνει

Πιο συγκεκριμένα, η αρχιτεκτονική είναι:

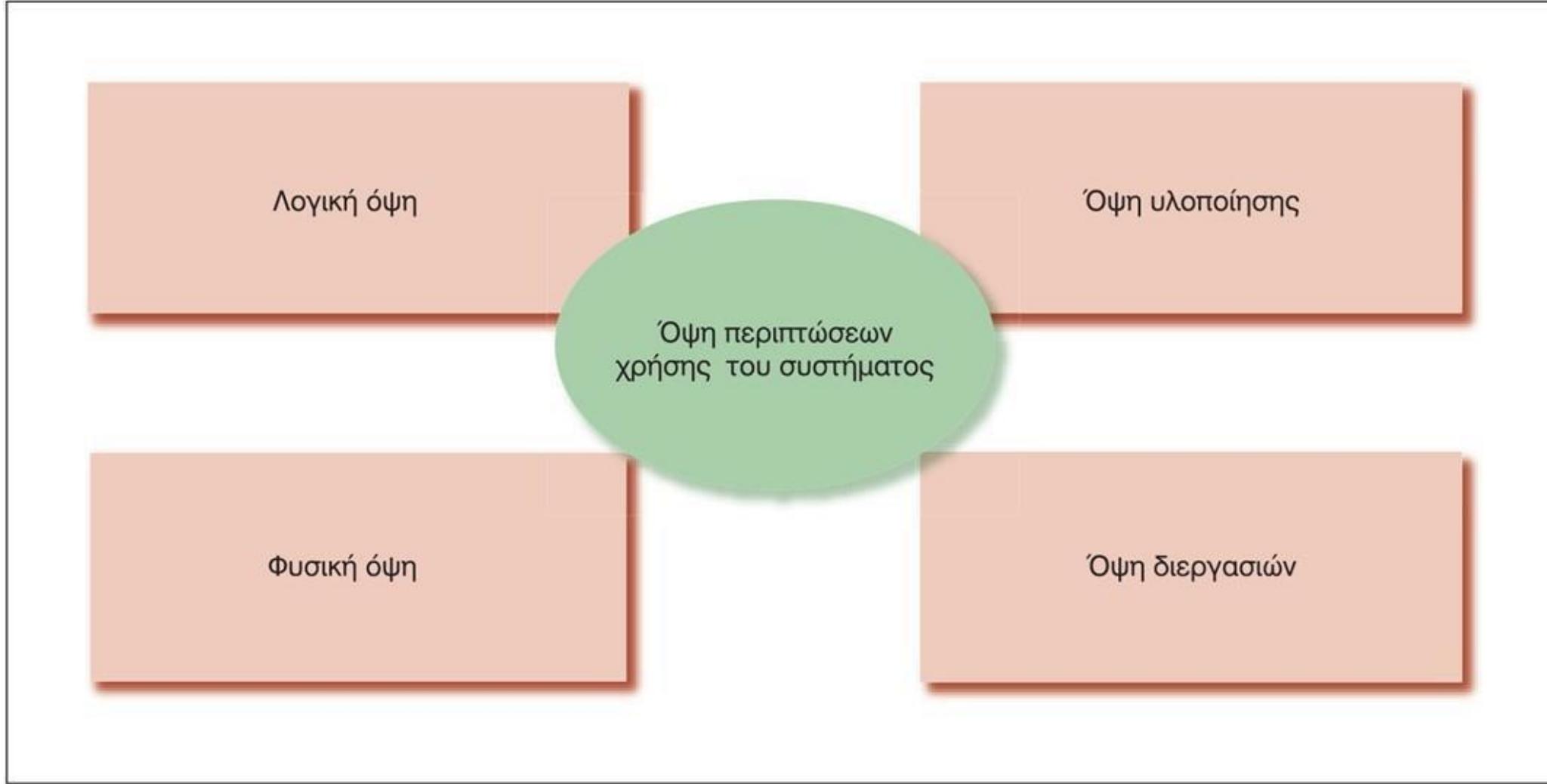
- ένα σύνολο αποφάσεων με σκοπό την οργάνωση ενός πληροφοριακού συστήματος,
- η επιλογή των δομικών στοιχείων ενός πληροφοριακού συστήματος καθώς και των διεπαφών τους,
- η συμπεριφορά των δομικών στοιχείων όπως αυτή καθορίζεται από την αλληλεπίδρασή τους,
- η σύνθεση των δομικών στοιχείων σε προοδευτικά μεγαλύτερα υποσυστήματα,
- το αρχιτεκτονικό στυλ που καθοδηγεί την οργάνωση των δομικών στοιχείων (δηλαδή αυτά τα ίδια τα στοιχεία, οι διεπαφές τους, οι συνεργασίες τους και η σύνθεσή τους).

Οι όψεις ενός πληροφοριακού συστήματος

Οι πέντε αυτές όψεις είναι:

- **Λογική όψη.** Περιγράφει με αφηρημένο τρόπο τα συστατικά ενός πληροφοριακού συστήματος.
 - Χρησιμοποιείται για να καταδείξει από τι αποτελείται ένα σύστημα και πώς τα συστατικά αυτά αλληλεπιδρούν μεταξύ τους.
- **Όψη διεργασιών.** Περιγράφει τις διεργασίες στο εσωτερικό του συστήματος. Είναι ιδιαίτερα
 - χρήσιμη στο να κατανοήσουμε την εσωτερική λειτουργία του συστήματος.
- **Όψη υλοποίησης.** Περιγράφει τον τρόπο με τον οποίο τα συστατικά του συστήματος είναι οργανωμένα σε ομάδες. Είναι χρήσιμη για τη διαχείριση των επιπέδων της αρχιτεκτονικής του συστήματος.
- **Φυσική όψη.** Περιγράφει τον τρόπο με τον οποίο ο σχεδιασμός του συστήματος έχει υλοποιηθεί
 - σε υλικό (hardware) και λογισμικό.
- **Όψη περιπτώσεων χρήσης του συστήματος.** Περιγράφει τη λειτουργικότητα του συστήματος όπως αυτή είναι κατανοητή από τους χρήστες του συστήματος. Η όψη αυτή είναι πάντα αναγκαία καθότι περιγράφει τι θα κάνει το πληροφοριακό σύστημα και αποτελεί την βάση για τις άλλες όψεις. Άλλωστε αυτός είναι ο λόγος που το μοντέλο αυτό ονομάζεται $4 + 1$.

Το μοντέλο του Kruchten



ΕΙΚΟΝΑ 2.2

Το μοντέλο του Kruchten των 4+1 όψεων.

Η αρχιτεκτονική σε τρία διαφορετικά επίπεδα

- **Επίπεδο της επιχειρησιακής αρχιτεκτονικής** (Enterprise Architecture – EA), δηλαδή το πώς το πληροφοριακό σύστημα εντάσσεται γενικότερα στην επιχείρηση.
- **Επίπεδο της αρχιτεκτονικής συστημάτων** (system architecture), δηλαδή το πώς το υλικό (hardware) σε συνεργασία με το λογισμικό (software) συντίθενται ώστε να αποτελέσουν ένα ενιαίο πληροφοριακό σύστημα.
- **Επίπεδο της αρχιτεκτονικής λογισμικού** (software architecture), που παρουσιάζει το πώς τα υποσυστήματα λογισμικού οργανώνονται σε ένα σύστημα λογισμικού.

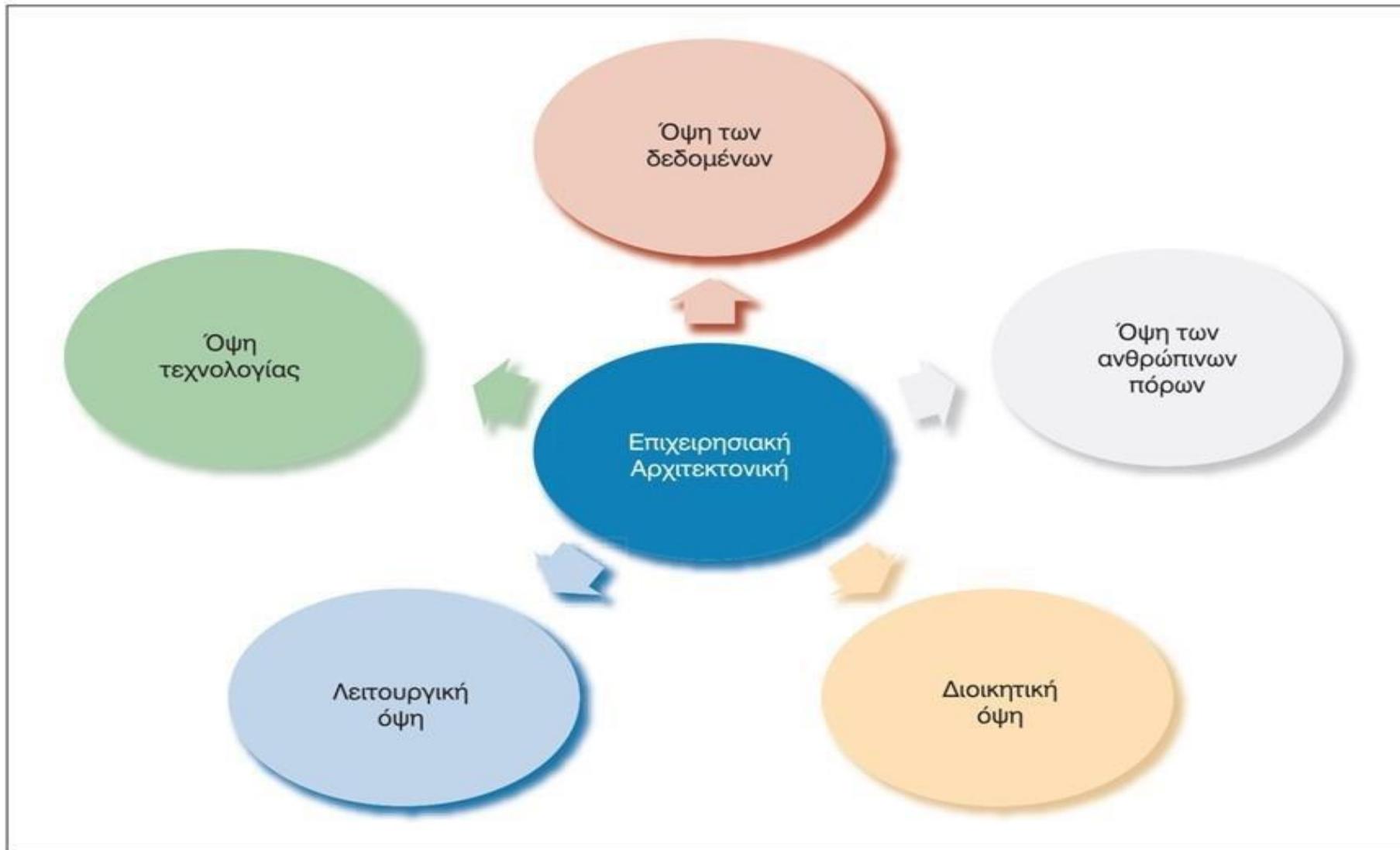
Τα επίπεδα της αρχιτεκτονικής

Επίπεδο	Εύρος (scope)	Λεπτομέρεια αρχιτεκτονικής	Αντίκτυπος	Συμμετέχοντες
Επιχειρησιακή αρχιτεκτονική	Όλη η επιχείρηση	Μικρή	Επιχειρηματική στρατηγική	Ανώτατη διοίκηση
Αρχιτεκτονική πληροφοριακού συστήματος	Επιχειρηματικός τομέας	Μεσαία	Στρατηγική επιχειρηματικού τομέα	Διοίκηση τομέα
Αρχιτεκτονική λογισμικού	Επιχειρηματική διεργασία	Μεγάλη	Λειτουργική στρατηγική	Χρήστες

ΕΙΚΟΝΑ 2.3

Τα επίπεδα της αρχιτεκτονικής.

Πληροφοριακό σύστημα και δομικά στοιχεία



ΕΙΚΟΝΑ 2.4

Οι όψεις της επιχειρησιακής αρχιτεκτονικής.

Εισαγωγή

Η Αρχιτεκτονική Επιχειρήσεων

- Έχει στόχο να αντιμετωπίσει τα προβλήματα της πολυπλοκότητας πληροφοριακών συστημάτων και της φτωχής ευθυγράμμισης των αναγκών της επιχείρησης με τα συστήματα πληροφορικών τεχνολογιών ΠΤ.
- Με την επιλογή της κατάλληλης αρχιτεκτονικής μπορεί μια επιχείρηση όχι μόνο να εξασφαλίσει τη βιωσιμότητά της σε ένα διαρκώς μεταβαλλόμενο περιβάλλον αλλά και να αυξήσει την ανταγωνιστικότητα και την ευελιξία της (ανταγωνιστικό πλεονέκτημα).

Ιστορική Αναδρομή (1/3)

- Η Επιχειρησιακή Αρχιτεκτονική (ΕΑ) θεωρείται ξεχωριστός κλάδος, που έχει μικρή ιστορία.
- Το 1987 ο J. A. **Zachman** δημοσιεύει το άρθρο “A Framework for Information Systems Architecture” στο IBM Journal. Θεωρείται το πρώτο άρθρο σχετικό με την ΕΑ, στο οποίο προτείνεται ένα πλαίσιο εργασίας για την αρχιτεκτονική πληροφοριακών συστημάτων.
- Το 1992 ο J. A. Zachman δημοσιεύει το άρθρο “Extending and formalizing the framework for information systems architecture”, στο οποίο αναθεωρεί το πλαίσιο εργασίας.
- Το πλαίσιο εργασίας του J. A. Zachman έχει υπάρξει η **Βάση** για τα μετέπειτα πλαίσια εργασίας.

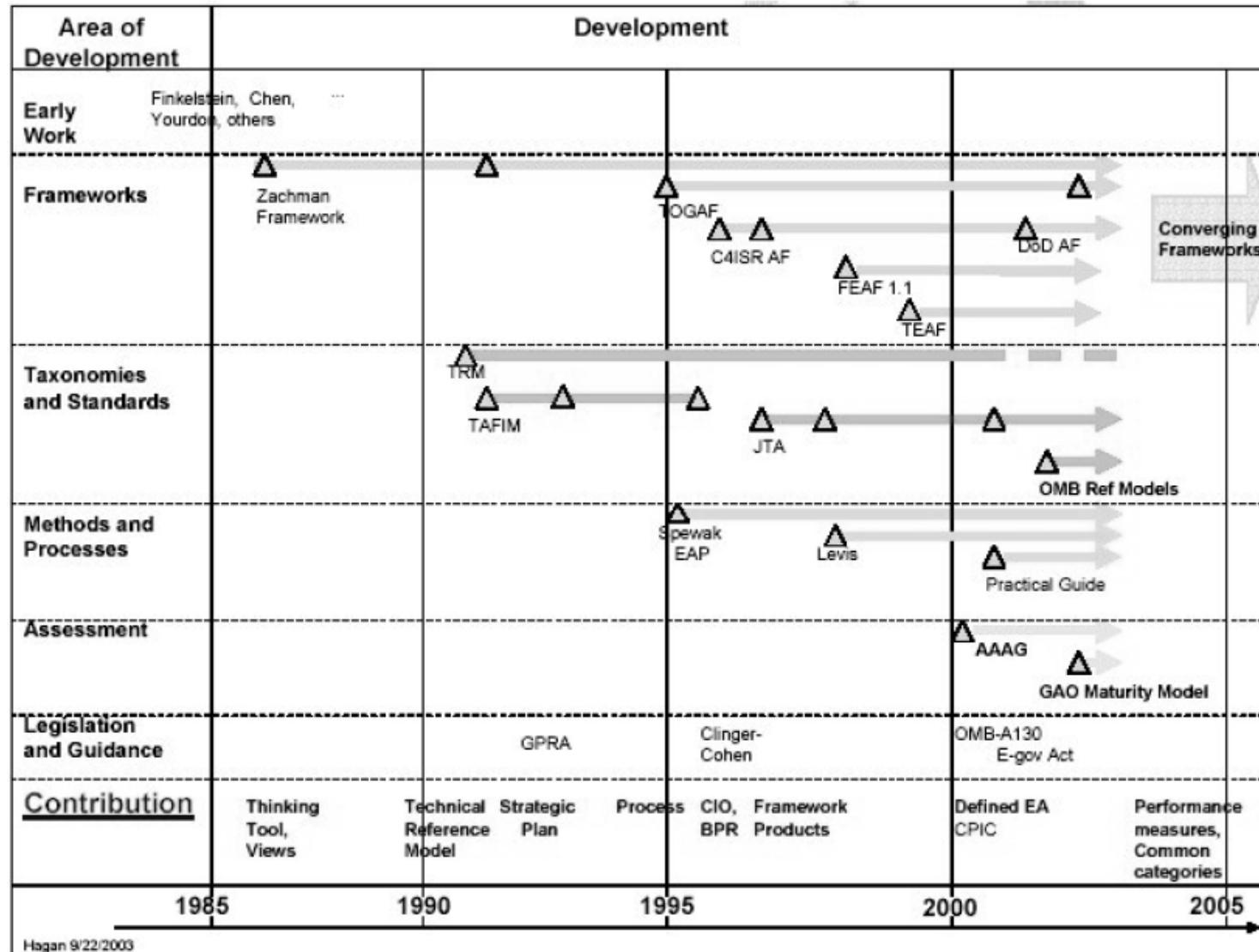
Ιστορική Αναδρομή (2/3)

- Το 1992 ο Steven H. Spewak δημοσιεύει το βιβλίο “Enterprise Architecture Planning”, το οποίο βασίζεται στο πλαίσιο εργασίας του J. A. Zachman και προτείνει μια μεθοδολογία για την ανάπτυξη της EA. Θεωρείται το πρώτο βιβλίο σχετικό με την EA.
- Στις αρχές του 90, το Υπουργείο Εθνικής Άμυνας των Η.Π.Α. (DoD), για να αντιμετωπίσει προβλήματα διαλειτουργικότητας, εξέδωσε ένα πλαίσιο εργασίας, το “Technical Architecture for Information Management” (**TAFIM**), για την ανάπτυξη τεχνικής υποδομής.
- Το 1995, το Open Group (κοινοπραξία βιομηχανιών), βασίστηκε στο TAFIM και ανέπτυξε ένα πλαίσιο εργασίας για EA, το **TOGAF** (The Open Group Architecture Framework).

Ιστορική Αναδρομή (3/3)

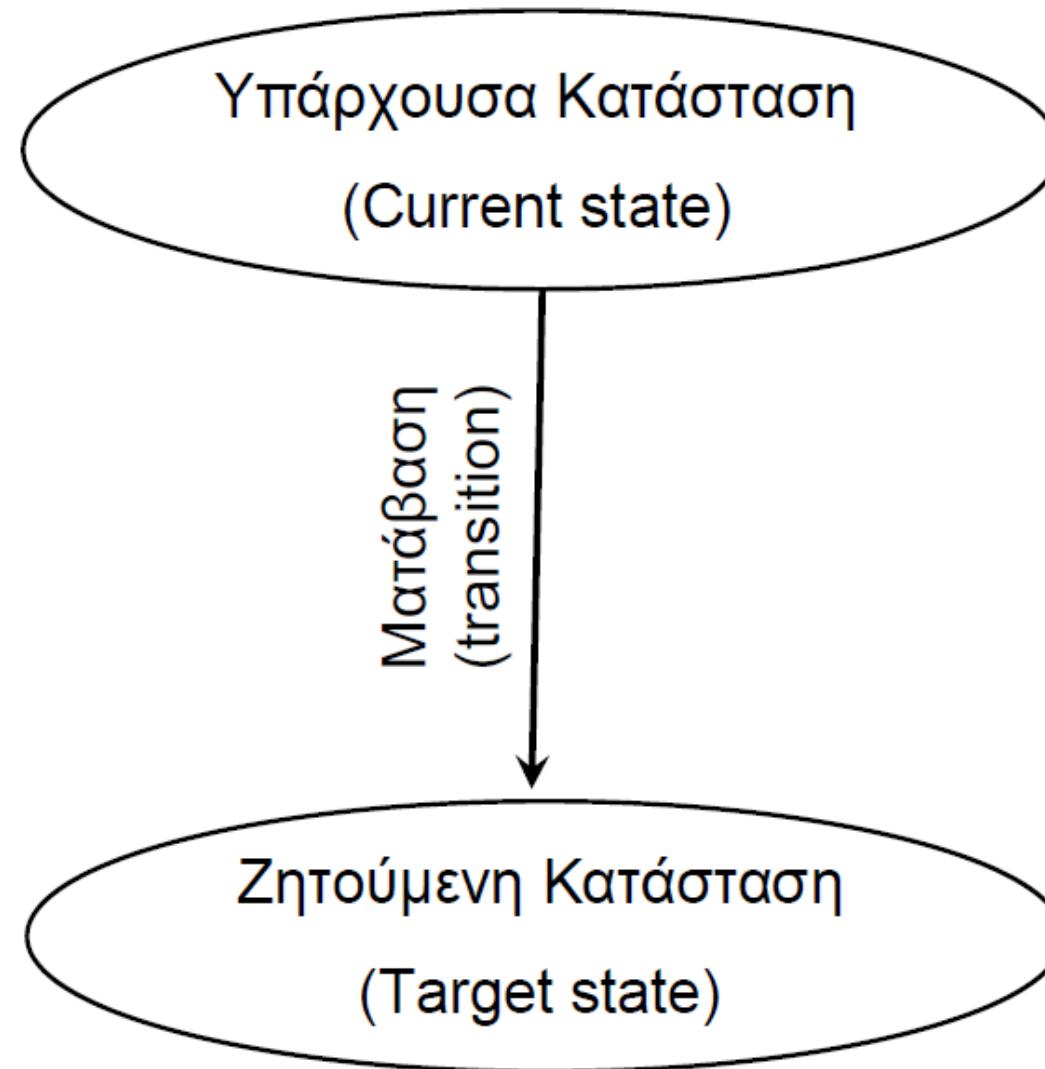
- Το 1996 το DoD ανέπτυξε το πλαίσιο εργασίας **C4ISR** (Command, Control, Communications, Computers Intelligence, Surveillance, and Reconnaissance) Architecture Framework, στη συνέχεια αναθεωρήθηκε και αποκαλείται σήμερα DoD Architecture Framework (**DODAF**). Χρησιμοποιείται ευρέως για την ανάπτυξη EA και την προώθηση της διαλειτουργικότητας.
- Το 1999, το CIO (Chief Information Officer) των Η.Π.Α. ανέπτυξε ένα πλαίσιο εργασίας EA για τις ομοσπονδιακές υπηρεσίες, το **FEAF** (Federal Enterprise Architecture Framework). Το FEAF είναι αρκετά δημοφιλές, καθώς χρησιμοποιείται συχνά και από επιχειρήσεις.
- Μετά το 1999, πολλές κρατικές υπηρεσίες των Η.Π.Α. αλλά και της Ευρωπαϊκής Ένωσης, έχουν αναπτύξει δικά τους πλαίσια εργασίας για ανάπτυξη EA. Ένα από τα πιο γνωστά είναι το **TEAF** (Treasury Enterprise Architecture) του Department of Treasury των Η.Π.Α.

Ορόσημα στην ιστορία ανάπτυξης της EA

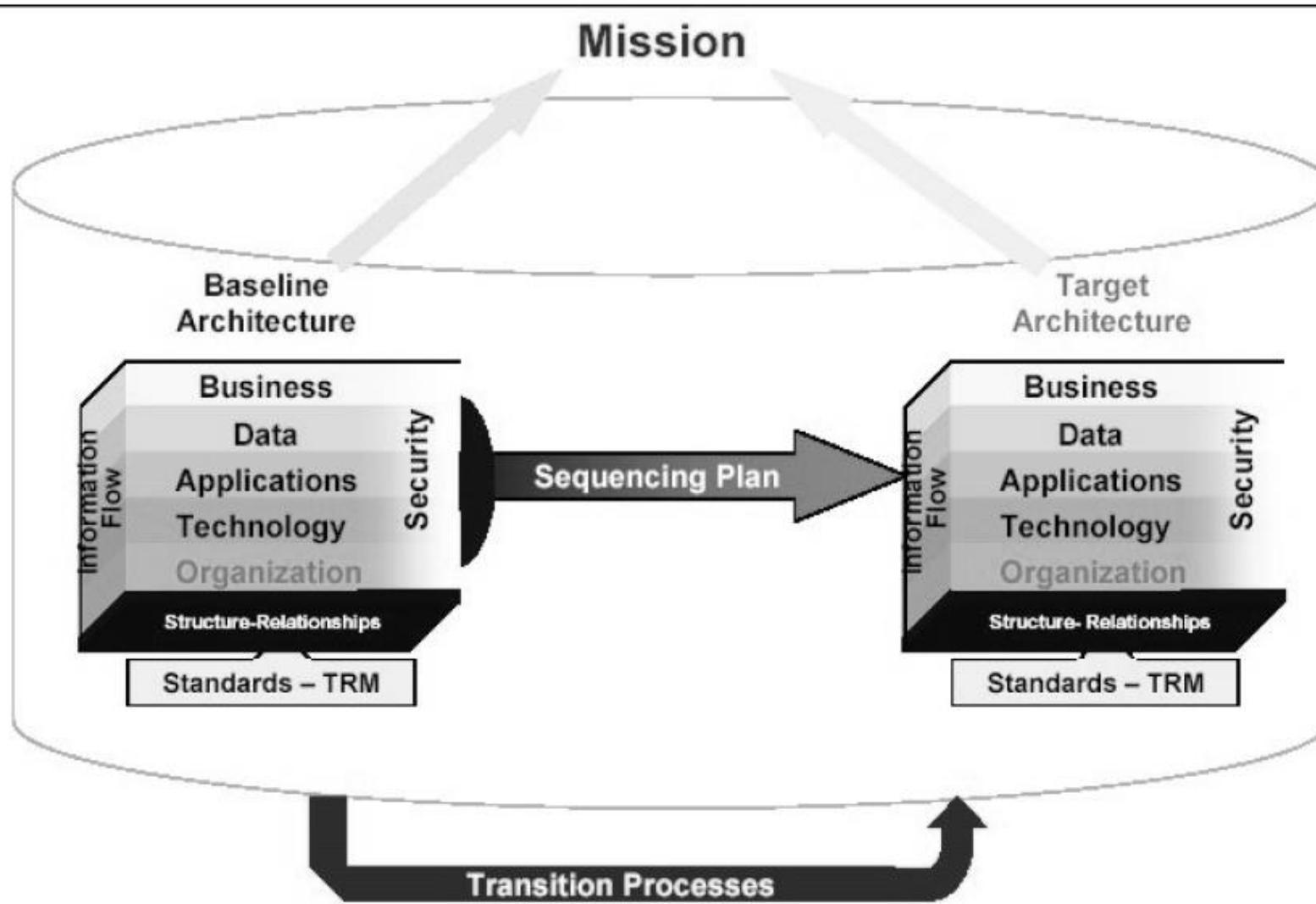


Πηγή: EA Book, A project by the MITRE corporation, 2004, DRAFT

Βασικά Συστατικά μιας ΕΑ



Βασικά Συστατικά μιας EA



Πηγή: *EA Book*, A project by the MITRE corporation, 2004, DRAFT

Πρακτικές Οδηγίες για EA

- Μια EA περιέχει:
 - Μια διαδικασία εφαρμογής EA (**EA process**) που δίνει λεπτομέρειες πώς θα πρέπει να υλοποιηθεί η EA
 - Ένα πλαίσιο EA (**EA framework**) που δείχνει πώς αντιλαμβανόμαστε την EA
- Από την αρχή ανάπτυξης της EA έχουν προταθεί πολλά πλαίσια και διαδικασίες για την ανάπτυξη της EA

Γιατί πολλά Πλαίσια και Διαδικασίες;

- Κάποιοι ίσως αναρωτιούνται για πιο λόγο χρειαζόμαστε πολλά πλαίσια και πολλές διαδικασίες;
- Το ζητούμενο θα ήταν να βρούμε ένα **βέλτιστο** πλαίσιο και μια βέλτιστη διαδικασία και να την εφαρμόζουμε σε κάθε περίπτωση υλοποίησης ΕΑ. Κάτι τέτοιο δεν έχει βρεθεί ακόμη!
- Ο λόγος είναι η **πολυπλοκότητα** του όλου εγχειρήματος αφού ακόμη και ο ορισμός της έννοιας της ΕΑ μπορεί να διαφοροποιείται ανάλογα με αυτόν που την υλοποιεί (θυμηθείτε η ΕΑ οριοθετείται από τους υποστηρικτές της αλλά όχι πάντα με τον ίδιο τρόπο από όλους!)
- Κάποιοι ισχυρίζονται πως υπάρχει **ανάγκη ύπαρξης εναλλακτικών** πλαισίων και διαδικασιών ανάλογα με τους επιμέρους στόχους του φορέα, τις επιδιώξεις, τα μέσα που διαθέτει κλπ.
- Κάποιοι θεωρούν πως τα **πλαίσια δεν χρειάζονται**, αφού οι διαδικασίες αρκούν.

Πλαίσια

- Ένα πλαίσιο EA είναι μια **δομή** μέσα στην οποία περιγράφονται τα **συστατικά** ενός συστήματος και οι **σχέσεις** που έχουν μεταξύ τους
- Ένα πλαίσιο εργασίας EA:
 - Προσδιορίζει τους διαφορετικούς τύπους πληροφοριών που απαιτούνται για την περιγραφή μιας EA.
 - Οργανώνει τους διαφορετικούς τύπους πληροφοριών σε μια λογική δομή.
 - Περιγράφει τις σχέσεις ανάμεσα στους διαφορετικούς τύπους πληροφοριών.
- Γιατί χρειαζόμαστε ένα πλαίσιο;
 - Ανεκτίμητο εργαλείο επικοινωνίας και συμφωνίας: ένα πρόβλημα που υπάρχει είναι η έλλειψη συμφωνίας στην επιχείρηση σχετικά με τι είναι EA (θυμηθείτε: η EA ορίζεται από τους υποστηρικτές της!)
 - Βοηθά στη διαχείριση της πολυπλοκότητας του όλου εγχειρήματος
 - Βοηθά στην πειθαρχεία και την ομοιόμορφη προσέγγιση στην ομάδα υλοποίησης

Διαδικασία εφαρμογής EA

- Έχουν προταθεί διάφορες διαδικασίες με διαφορετικό περιεχόμενο
- Το εύρος **ποικίλει** π.χ. εάν περιέχουν την επίτευξη διοικητικής υποστήριξης, τη συντήρηση της EA και τη χρήση της EA
- Όλες περιέχουν τη διαδικασία ανάπτυξης **προϊόντων** ή **παραδοτέων** EA (EA products)
- Κάποιες στηρίζονται σε συγκεκριμένα πλαίσια
- Οι έννοιες και οι δράσεις των διαδικασιών μπορούν να χρησιμοποιηθούν στη γενική περίπτωση οπότε ο υποστηρικτής μιας EA μπορεί να κατανοήσει καλύτερα την EA μελετώντας τις
- **Καμιά** δεν θεωρείται **απόλυτα ολοκληρωμένη** αφού δεν ασχολούνται όλες με όλα τα θέματα π.χ. προγραμματισμός (planning), επιλογή εγαλείων και εκτίμηση κόστους

Πλαίσια

- Πλαίσιο Zachman
- Πλαίσιο Open Group
- Ομοσπονδιακό Πλαίσιο
- Μεθοδολογία Gartner

Πλαίσιο Zachman

Πλαίσιο Zachman

- Είναι το πιο γνωστό πλαίσιο εργασίας.
- Είναι **ουδέτερο** ως προς την επιλογή **τεχνικών, μοντέλων και εργαλείων**.
- Είναι οργανωμένο σε μια μήτρα (**πίνακα**) που αποτελείται από 6 στήλες και 6 σειρές.
- Ο Zachman **δεν** έχει ορίσει μια **διαδικασία** εφαρμογής EA αφήνοντας έτσι ανοικτή την επιλογή της

Πλαίσιο Zachman

Ορίζει μια προσπάθεια για ταξινόμηση διαφόρων πτυχών μιας επιχείρησης απαντώντας σε ερωτήματα σχετικά με το **πότε, πώς, τί, γιατί, πού, ποιοί**, εξετάζοντας έτσι περισσότερα για το **συγχρονισμό**, τις **διαδικασίες**, τη **λειτουργικότητα**, τη **λογική και τους παράγοντες** που την αποτελούν. Λαμβάνει υπόψη τόσο τους στόχους της αρχιτεκτονικής ανάλυσης, όσο και το συγκεκριμένο θέμα το οποίο μελετάει κάθε φορά.

Το πλέγμα Zachman αναπτύσσεται σε **έξι πτυχές-στήλες και έξι επίπεδα-γραμμές** και σχηματίζει έτσι τριάντα έξι κελιά καθένα από τα οποία απευθύνεται στην οπτική γωνία ενός παράγοντα (γραμμές) σχετικά με κάποιο σημείο ενδιαφέροντος (στήλες).

Πλαίσιο Zachman

Στήλες / Έννοιες

- Οι στήλες είναι βασισμένες στις λέξεις **What, How, Where, Who, When, Why** και παίρνουν την ονομασία **Data, Function, Network, People, Time, Motivation**.
- Οι στήλες αντιπροσωπεύουν διαφορετικές έννοιες, επίπεδα αφαίρεσης ή διαφορετικούς **τρόπους περιγραφής** της επιχείρησης
- Στην πρώτη έκδοση του πλαισίου, υπήρχαν μόνο οι 3 πρώτες στήλες (Data-What, Function-How, Network-Where)

Πλαίσιο Zachman

Στήλες

- Δεδομένα (Τί):** Περιλαμβάνει τα δεδομένα που είναι απαραίτητα για τη λειτουργία μιας επιχείρησης.
- Λειτουργίες (Πώς):** Αποτελείται από τον ορισμό σειριακών εργασιών που συνθέτουν το γενικό σκοπό της επιχείρησης.
- Δίκτυο (Πού):** Αφορά τη γεωγραφική κατανομή των δραστηριοτήτων της επιχείρησης.
- Άνθρωποι (Ποιός):** Εξετάζονται οι σχέσεις και η ανάθεση εργασιών στο ανθρώπινο δυναμικό της επιχείρησης.
- Χρόνος (Πότε):** Ο πραγματικός χρόνος δε λαμβάνεται υπόψη προκειμένου να σχεδιαστούν κριτήρια και ποσοτικά επίπεδα για τις πηγές της επιχείρησης. Ο χρόνος που εξετάζεται αναφέρεται σε χρονικούς κύκλους εκτέλεσης διεργασιών που είναι σημαντικές για την επιχείρηση.
- Κίνητρο (Γιατί):** Αφορά τους στόχους και τις στρατηγικές που χρησιμοποιεί μια επιχείρηση καθώς και την περιγραφική αναπαράστασή της.

Πλαίσιο Zachman

Γραμμές

- Πεδίο δράσης (Οργανωτής):** Απεικονίζει το μέγεθος, το σχήμα, τις χωρικές σχέσεις και τον βασικό σκοπό της τελικής κατασκευής. Αντιστοιχεί σε μια εκτελεστική περίληψη που ενημερώνει κάποιον ενδιαφερόμενο σχεδιαστή ή επενδυτή για τη δράση του συστήματος, το κόστος που περιλαμβάνει και τον τρόπο λειτουργίας του.
- Μοντέλο επιχείρησης (Ιδιοκτήτης):** Περιλαμβάνει αρχιτεκτονικά σχέδια σχετικά με τη μορφή της επιχείρησης, φανερώνοντας τις επιχειρηματικές ενότητες, τις λειτουργίες και τη διάδρασή τους.
- Μοντέλο συστήματος (Σχεδιαστής):** Πρόκειται για τη μετάφραση των αρχιτεκτονικών σχεδίων από έναν αναλυτή συστημάτων σε αρχιτεκτονικά πλάνα που καθορίζουν τα δεδομένα και τις λειτουργίες της επιχείρησης.
- Τεχνολογικό μοντέλο (Εκτελεστής):** Ο εργολάβος τώρα πρέπει να μετατρέψει τα πλάνα σε μια πιο κατασκευαστική πλευρά λαμβάνοντας υπόψη περιορισμούς που προκύπτουν από την τεχνολογία και τα υλικά που χρησιμοποιούνται.
- Λεπτομερείς αναπαραστάσεις (Υπεργολάβος):** Αφορούν τις μεμονωμένες ενδείξεις που δίνονται σε προγραμματιστές οι οποίοι απλά πρέπει να ολοκληρώσουν τμήματα του έργου χωρίς να ασχοληθούν με τον τρόπο διασύνδεσής τους.
- Λειτουργική επιχείρηση:** Τέλος, το σύστημα εφαρμόζεται και γίνεται τμήμα του οργανισμού, φανερώνοντας τα δεδομένα, τις λειτουργίες, το δίκτυο, το οργανισμό, το πρόγραμμα και τη στρατηγική που έχουν ακολουθηθεί.

Πλαίσιο Zachman

- **Γραμμές / Ρόλοι**
- Οι γραμμές αντιπροσωπεύουν διαφορετικές **οπτικές** της επιχείρησης, μέσα από το πρίσμα διαφορετικών **ρόλων**.
- Οι ρόλοι είναι Planner, Owner, Designer, Builder, Subcontractor. Σε μετέπειτα έκδοση υπάρχει και 6 σειρά, με τον τίτλο **Functioning enterprise**, που αντιπροσωπεύει το τελικό προϊόν του συστήματος.
 - Ο **planner** ενεργεί όπως ο επενδυτής: καθορίζει τα βασικά χαρακτηριστικά της επένδυσης
 - Ο **ιδιοκτήτης (owner)** επεξεργάζεται την οπτική του planner σε ένα επιχειρηματικό μοντέλο
 - Ο **σχεδιαστής (designer)** μεταφράζει τις προδιαγραφές του ιδιοκτήτη σε ένα συστηματικό μοντέλο
 - Ο **κατασκευαστής (builder)** κατασκευάζει το τεχνολογικό μοντέλο των ΠΣ
 - Ο **υπεργολάβος (subcontractor)** έχει μια λεπτομερή εκτός γενικού πλαισίου (out-of-context) περιγραφή του τι θα αναπτυχθεί

Πλαίσιο Zachman

Ο στόχος είναι να συμπληρωθεί ο παρακάτω πίνακας:

ΕΠΙΠΕΔΑ ΑΦΑΙΡΕΣΗΣ (ABSTRACTIONS)

ΟΠΤΙΚΕΣ (PERSPECTIVES)	Data (What)	Function (How)	Network (Where)	People (Who)	Time (When)	Motivation (Why)
Planner						
Owner						
Designer						
Builder						
Sub-contractor						
System						



Πλαίσιο Zachman

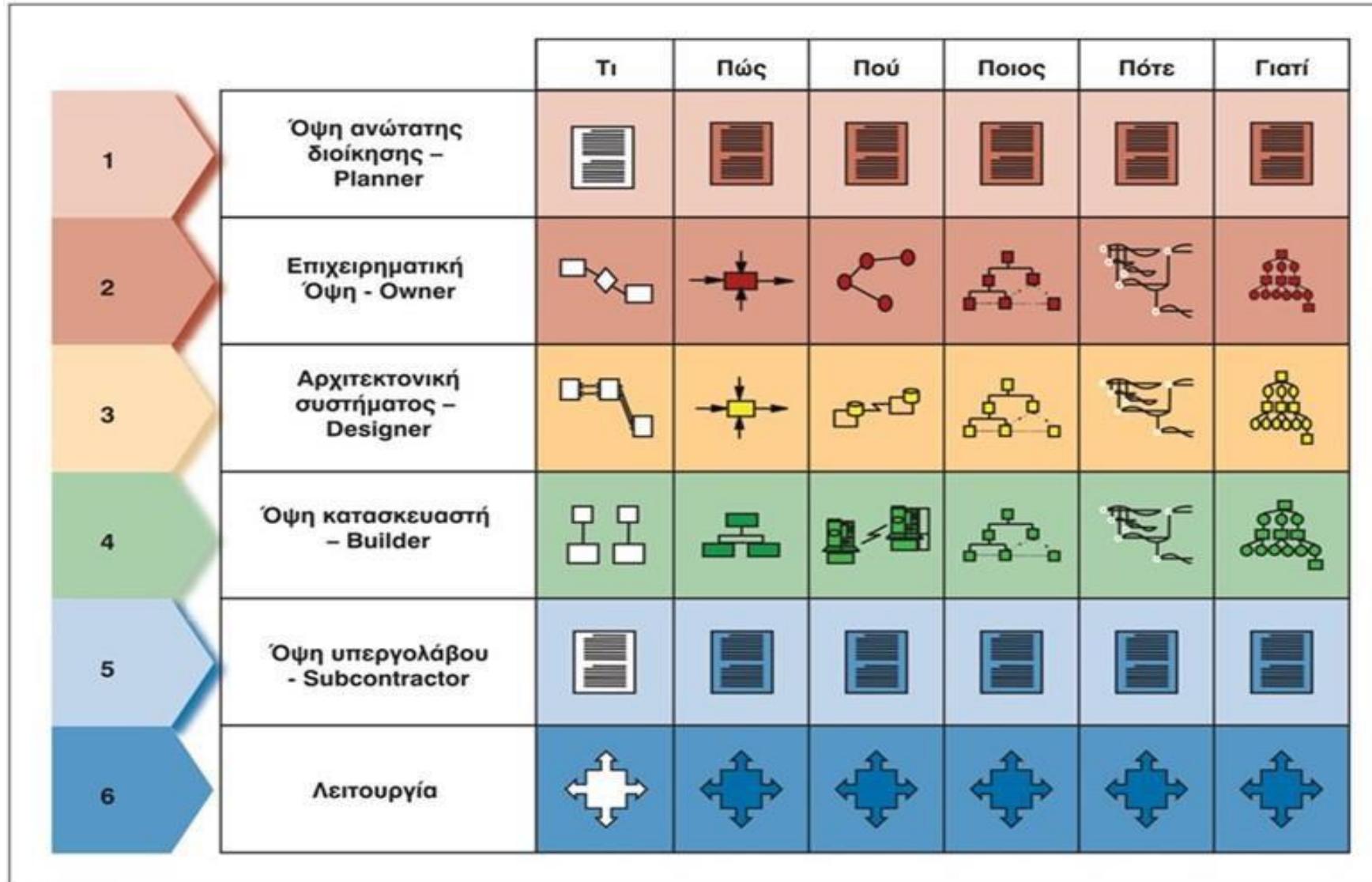
Κάθε κελί απευθύνεται στην οπτική γωνία ενός παράγοντα (γραμμές) σχετικά με κάποιο σημείο ενδιαφέροντος (στήλες).

	WHAT	HOW	WHERE	WHO	WHEN	WHY
	DATA	FUNCTION	NETWORK	PEOPLE	TIME	MOTIVATION
SCOPE {contextual}	List of things important to the business 	List of processes the business performs 	List of locations in which the business operates 	List of organisations important to the business 	List of event cycles significant to the business 	List of business goals/strategies
Planner	Entity = Class of business things	Process = Class of business process	Node = Major business locations	People = Major business unit	Time = Major Business Event Cycle	End/Means = Major Business Goal/Strategy
BUSINESS MODEL {Conceptual}	e.g., Semantic Model 	e.g., Business Process Model 	e.g., Business Logistics System 	e.g., Workflow Model 	e.g., Master Schedule 	Business Plan
Owner	Entity = Business Entity Relationship = Business	Process = Business IO = Business Resource	Node = Business Location Link = Business Linkage	People = Organisation unit Work = Work Product	Time = Business Event Cycle Cycle = Business Cycle	End = Business Objective Means = Business Strategy
SYSTEM MODEL {Logical}	e.g., Logical Data Model 	e.g., Application Architecture 	e.g., Distributed System Model 	e.g., Human Interface Architecture 	e.g., Processing Structure 	e.g., Business Rule Model
Designer	Entity = Data Entity Relationship = Data Relationship	Process = Application Function IO = User Views	Node = Function Relationship = Line Characteristics	People = Role Work = Deliverable	Time = System Event Cycle Cycle = Processing Cycle	End = Structural Assertion Means = Action Assertion
TECHNOLOGY MODEL {Physical}	e.g., Physical Data Model 	e.g., System Design 	e.g., Technology Architecture 	e.g., Presentation Architecture 	e.g., Control Structure 	e.g., Rule Design
Builder	Entity = Segment/Table Relationship = Pointer/key	Process = Computer Function IO = Data Elements/sets	Node = H/w /System s/w Relationship = Line Specifications	People = User Work = Screen Formats	Time = Execute Cycle Cycle = Component Cycle	End = Condition Means = Action
DETAILED REPRESENTATIONS {Out-of-context}	e.g., Data Definition 	e.g., Program 	e.g., Network Architecture 	e.g., Security Architecture 	e.g., Timing Definition 	e.g., Rule Specification
Subcontractor	Entity = Field Relationship = Address	Process = Language Statement IO = Control Block	Node = Address Link = Protocol	People = Identity Work = Job	Time = Interrupt Cycle = Machine Cycle	End = Sub-condition Means = step
FUNCTIONING ENTERPRISE	e.g DATA	e.g FUNCTION	e.g NETWORK	e.g ORGANISATION	e.g SCHEDULE	e.g STRATEGY

Πλαίσιο Zachman: Κελιά

- Ο Zachman αντιμετωπίζει κάθε κελί από τα 30 ως ξεχωριστή μεταβλητή, που θα πρέπει να αντιμετωπίζεται ως ανεξάρτητο από τα υπόλοιπα.
- Για κάθε κελί είναι δυνατή η χρήση **διαφορετικού μοντέλου** ή διαφορετικής σημειογραφίας (notation).
- Η πληροφορία που υπάρχει σε κάθε κελί μπορεί να βρίσκεται είτε σε **αφαιρετικό** επίπεδο είτε σε πολύ **λεπτομερές** επίπεδο.
- Τελικά το πλαίσιο Zachman πέρα από τη χρήση του στις EA αποτελεί ένα πολύτιμο εργαλείο κατάτμησης και συνεπώς καλύτερης κατανόησης πολύπλοκων καταστάσεων, έργων κλπ.

Το πλαίσιο επιχειρησιακής αρχιτεκτονικής Zachman



EIKONA 2.5

Το πλαίσιο της επιχειρησιακής αρχιτεκτονικής Zachman.

Οι όψεις του μοντέλου Zachman [1/2]

- **Η όψη της ανώτατης διοίκησης (planner).** Είναι μια υψηλού επιπέδου αρχιτεκτονική που έχει ως σκοπό να παρουσιάσει το εύρος του συστήματος, πόσο θα κοστίσει το σύστημα και πώς θα συ- σχετίζεται με το ευρύτερο περιβάλλον.
- **Η όψη της διοίκησης της επιχείρησης (owner).** Εστιάζεται στο επιχειρηματικό μοντέλο με το οποίο θα λειτουργεί η επιχείρηση και παρουσιάζει τις επιχειρηματικές οντότητες και διεργασίες και τον τρόπο με τον οποίο συνδέονται (business concepts). Είναι γνωστή και ως άποψη του ιδιοκτήτη της επιχείρησης.
- **Η αρχιτεκτονική συστήματος (designer).** Αντιστοιχεί στο μοντέλο συστήματος όπως σχεδιάζεται από έναν αναλυτή συστημάτων, ο οποίος πρέπει να καθορίσει τη δομή των δεδομένων, τις επιχειρηματικές διεργασίες, καθώς και τις λογικές ροές των διεργασιών.
- **Η όψη του κατασκευαστή (builder perspective).** Η αρχιτεκτονική δοσμένη στο επίπεδο λεπτομέρειας των γλωσσών προγραμματισμού, των συσκευών εισόδου/εξόδου ή άλλων απαιτούμενων τεχνολογιών υποστήριξης, και αφορά τη σχεδίαση του συστήματος (design).

Οι όψεις του μοντέλου Zachman [2/2]

- Η **όψη του υπεργολάβου** (subcontractor perspective). Περιγράφει τις λεπτομερείς απαιτήσεις για τα υποσυστήματα και τα συστατικά των συστημάτων. Αναφέρεται ως η όψη του υπεργολάβου, διότι αφενός κάποια τμήματα της αρχιτεκτονικής κατασκευάζονται από εξωτερικούς συνεργάτες της επιχείρησης, αφετέρου διότι στο επίπεδο αυτό δεν απαιτείται η γενικότερη γνώση της επιχείρησης. Είναι ένα καθαρά τεχνικό επίπεδο.
- Η **όψη της λειτουργίας της επιχείρησης** (functioning enterprise perspective). Είναι το σύστημα όπως αυτό λειτουργεί στην επιχείρηση.

Η χρήση του μοντέλου Zachman

	Τι (Δεδομένα)	Πώς (Διεργασίες)	Πού (Δίκτυο)	Ποιος (Ανθρώπινο Δυναμικό)	Πότε (time)	Γιατί (motive)
Planner	Απαρίθμηση σημαντικών εννοιών	Λίστα επιχειρηματικών διεργασιών	Λίστα τοποθεσιών λειτουργίας	Λίστα οργανισμών	Επιχειρηματικές απαιτήσεις χρόνου	Επιχειρηματικοί στόχοι
Owner	Σημασιολογικό μοντέλο δεδομένων	Επιχειρηματικό μοντέλο	Επιχειρηματικό μοντέλο εφοδιαστικής	Μοντέλο ροής εργασιών	Κύριο χρονοπρόγραμμα	Επιχειρηματικό πλάνο
Designer	Λογικό μοντέλο δεδομένων	Αρχιτεκτονική εφαρμογών	Αρχιτεκτονική κατανεμημένου συστήματος	Αρχιτεκτονική διεπαφής χρηστών	Κύκλοι και επιχειρηματικά γεγονότα	Επιχειρηματικοί κανόνες
Builder	Φυσικό μοντέλο δεδομένων	Σχεδιασμός συστήματος	Τεχνολογική αρχιτεκτονική	Αρχιτεκτονική παρουσίασης	Δομές ελέγχου	Σχεδιασμός κανόνων
Subcontractor	Ορισμός δεδομένων	Προδιαγραφές εφαρμογών	Αρχιτεκτονική δίκτυου	Αρχιτεκτονική ασφάλειας	Γεγονότα	Προσδιορισμός κανόνων
Σύστημα σε λειτουργία	Δεδομένα	Διεργασίες	Δίκτυο	Οργάνωση	Χρονοπρόγραμμα	Στρατηγική

EIKONA 2.6

Τα μοντέλα της επιχειρησιακής αρχιτεκτονικής Zachman.

Πλαίσιο

Open Group / TOGAF

Πλαίσιο Open Group / TOGAF

- Το TOGAF παρέχει μια μεθοδολογία ανάπτυξης αρχιτεκτονικής, σαν τμήμα του πλαισίου. Η μεθοδολογία αυτή μπορεί να χρησιμοποιηθεί και για άλλα πλαίσια.
- Αναγνωρίζει διάφορες οπτικές, όπως και το πλαίσιο Zachman, αλλά χρησιμοποιεί κυρίως τις οπτικές **Επιχείρηση (Business)**, **Δεδομένα (Data)**, **Εφαρμογές (Applications)** και **Τεχνολογία (Technology)**.
- Αναγνωρίζει την εταιρική στρατηγική ως παράγοντα στη φάση της ανάπτυξης EA

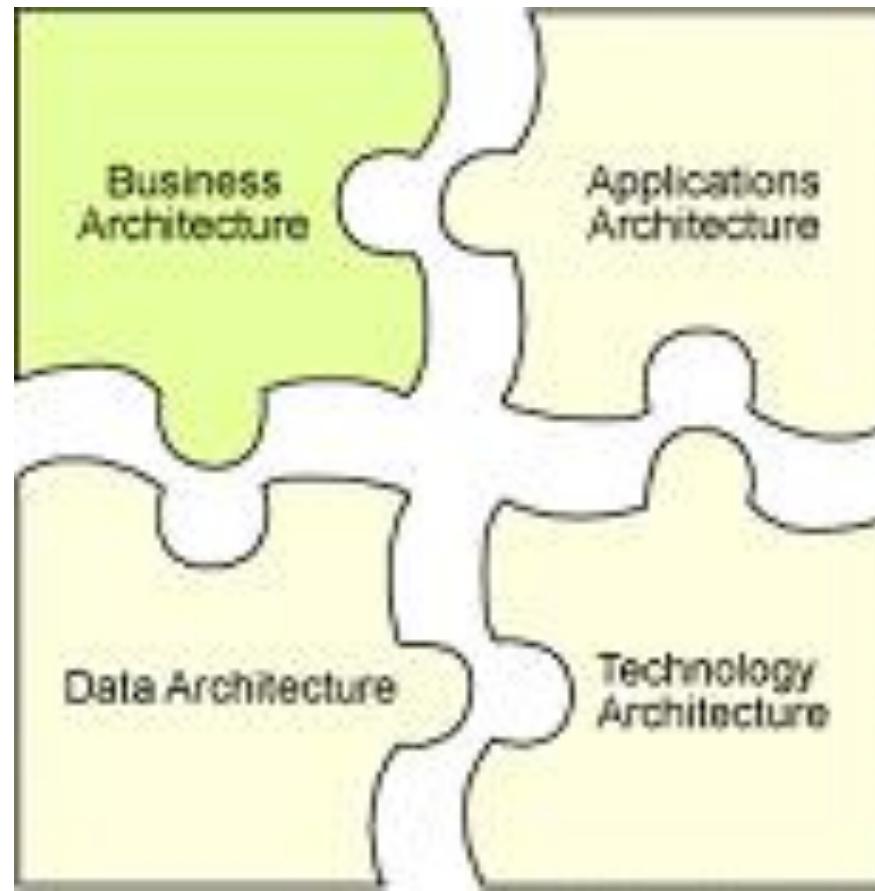
Πλαίσιο Open Group

Αναγνωρίζει τέσσερις τύπους αρχιτεκτονικής ως υποσύνολα της ολικής αρχιτεκτονικής μιας επιχείρησης.

- **Αρχιτεκτονική Εργασίας (Business architecture):** Καθορίζει τη στρατηγική της επιχείρησης, τη διακυβέρνηση, την οργάνωση και διεργασίες – κλειδιά για τη λειτουργία της.
- **Αρχιτεκτονική Δεδομένων (Data architecture):** Περιγράφει τη δομή των λογικών και φυσικών πόρων δεδομένων ενός οργανισμού και των πηγών διαχείρισης και οργάνωσής τους. Παρόλα αυτά δε συμπεριλαμβάνει και τη σχεδίαση της πραγματικής βάσης δεδομένων στην οποία αυτά αποθηκεύονται.
- **Αρχιτεκτονική Εφαρμογών (Application architecture):** Παρέχει τα σχεδιαγράμματα για το πώς ανεξάρτητα συστήματα αναπτύσσονται και αλληλεπιδρούν με τις βασικές λειτουργίες της επιχείρησης.
- **Αρχιτεκτονική Τεχνολογίας (Technology architecture):** Περιγράφει τις ικανότητες λογισμικού και υλικού που απαιτούνται για να υποστηρίξουν την ανάπτυξη της εργασίας, των δεδομένων και των εφαρμογών. Εδώ ανήκουν η τεχνολογική υποδομή, τα δίκτυα, οι επικοινωνίες και τα διάφορα πρότυπα που χρησιμοποιούνται.



Πλαίσιο Open Group



Πλαίσιο Open Group Μέθοδος ανάπτυξης αρχιτεκτονικής

Μέθοδος ανάπτυξης αρχιτεκτονικής (ADM – Architecture Development Method):

- Περιγράφει πώς να παραχθεί η σωστή αρχιτεκτονική μιας επιχείρησης προκειμένου να απευθύνεται αποτελεσματικά στις απαιτήσεις της.
- Παρέχει έναν αξιόπιστο, αποδεδειγμένο τρόπο ανάπτυξης αρχιτεκτονικής, οπτικές γωνίες που επιτρέπουν στον αρχιτέκτονα να διασφαλίσει ότι ακόμα και πολύπλοκα σύνολα προϋποθέσεων μπορούν να αντιμετωπισθούν, συνδέσεις σε πρακτικές μελέτες περιπτώσεων και αρχές και οδηγίες για εργαλεία αρχιτεκτονικής ανάπτυξης.
- Αποτελείται από εννέα αρχιτεκτονικές φάσεις οι οποίες αλληλεπιδρούν με μια φάση διαχείρισης των απαιτήσεων και κυκλικά αναπαριστούν ικανοποιητικά τη συνολική αρχιτεκτονική λειτουργία της επιχείρησης.
- Για καθεμία από αυτές παρέχει και μια περιγραφή σχετικά με τους στόχους, την προσέγγιση, τις εισόδους, τα βήματα και τις εξόδους που περιλαμβάνει.
- Τέλος, παρέχει μια συγκεντρωτική περίληψη που αφορά τη διαχείριση των απαιτήσεων και περιγραφές για τις φάσεις εισόδου-- εξόδου. Καθώς πρόκειται για ένα αμιγώς διαδικαστικό πλαίσιο, μπορεί να συνδυαστεί και με οποιαδήποτε από τις υπόλοιπες τρεις μεθοδολογίες.

Πλαίσιο Open Group Μέθοδος ανάπτυξης αρχιτεκτονικής

Οι φάσεις που συνθέτουν τη Μέθοδο Ανάπτυξης Αρχιτεκτονικής είναι:

- ® **Προκαταρκτική Φάση:** Περιγράφει την προετοιμασία και την αρχικοποίηση δραστηριοτήτων που είναι απαραίτητες για να δημιουργηθεί αρχιτεκτονική ικανότητα, περιλαμβάνοντας την εξατομίκευση του TOGAF για τη συγκεκριμένη περίπτωση και τον ορισμό αρχιτεκτονικών αρχών.
- ® **Φάση A – Architecture Vision:** Το αρχιτεκτονικό όραμα, η γέννηση της ιδέας για το μέλλον της επιχείρησης. Είναι η αρχική φάση του κύκλου που περιλαμβάνει πληροφορίες για τον ορισμό της δημιουργίας της αρχιτεκτονικής και τους μετόχους και έχει την ευθύνη για την απόκτηση της έγκρισης για τη συνέχεια της δημιουργίας της αρχιτεκτονικής.
- ® **Φάση B – Business Architecture:** Η μοντελοποίηση της πραγματοποίησης των απαιτήσεων από βασικά στοιχεία (παράγοντες, υπηρεσίες, συστατικά). Αφορά πιο συγκεκριμένα την αρχιτεκτονική του επιχειρηματικού τμήματος της επιχείρησης.

Πλαίσιο Open Group Μέθοδος ανάπτυξης αρχιτεκτονικής

Οι φάσεις που συνθέτουν τη Μέθοδο Ανάπτυξης Αρχιτεκτονικής είναι:

- ® **Φάση Γ – Information Systems Architecture:** Πρόκειται για την αρχιτεκτονική των εφαρμογών και την κατάδειξη των συνδέσεων και αλληλεξαρτήσεων μεταξύ τους.
- ® **Φάση Δ – Technology Architecture:** Αποτελεί την τεχνολογική αρχιτεκτονική και περιγράφει την ζητούμενη κατάσταση – στόχο σχετικά με την τεχνολογική υποδομή.
- ® **Φάση Ε – Opportunities & Solutions:** Διευθύνει τον αρχικό σχεδιασμό της εκτέλεσης και αναγνωρίζει τα μέσα που μπορούν να πραγματοποιήσουν την αρχιτεκτονική που έχει αποφασιστεί προηγουμένως.

Πλαίσιο Open Group Μέθοδος ανάπτυξης αρχιτεκτονικής

Οι φάσεις που συνθέτουν τη Μέθοδο Ανάπτυξης Αρχιτεκτονικής είναι:

- ® **Φάση ΣΤ – Migration Planning:** Ασχολείται με τη μετάβαση από την αρχιτεκτονική εισόδου σε αυτή του στόχου οριστικοποιώντας ένα λεπτομερές πλάνο εκτέλεσής της.
- ® **Φάση Ζ – Implementation Governance:** Παρέχει μια αρχιτεκτονικής φύσης επίβλεψη για την εκτέλεση.
- ® **Φάση Η – Architecture Change Management:** Εγκαθιστά διαδικασίες για τη διαχείριση της αλλαγής στην καινούρια αρχιτεκτονική.
- ® **Διαχείριση Απαιτήσεων -- Requirements Management:** Εξετάζει τη διαδικασία διαχείρισης των αρχιτεκτονικών απαιτήσεων σε όλη τη διάρκεια της Μεθόδου Ανάπτυξης Αρχιτεκτονικής ADM.

Πλαίσιο Open Group

Επιχειρησιακό συνεχές

Παρέχει ένα μοντέλο για τη δόμηση μιας εικονικής αρχιτεκτονικής αποθήκης, η οποία θα περιλαμβάνει όλα τα αρχιτεκτονικά στοιχεία και λύσεις - πλάνα που έχει συγκεντρώσει η επιχείρηση τόσο από την προηγούμενη φάση όσο και από τον κλάδο της πληροφορικής γενικά.

Τα αντικείμενα στο επιχειρησιακό συνεχές παρέχουν μια χρήσιμη βάση πάνω στην οποία μια επιχείρηση μπορεί να στηριχθεί ώστε να προχωρήσει μπροστά στην επιχειρηματική της προσπάθεια.

Θεωρεί τη διαδικασία δημιουργίας της αρχιτεκτονικής ως μια μετάβαση από το γενικό στο ειδικό.

Σε πολλά σημεία κατά τη διάρκεια της ADM φάσης υπάρχουν υπενθυμίσεις ώστε να καθοδηγήσουν τον αρχιτέκτονα στη χρήση των σωστών στοιχείων αρχιτεκτονικής από το επιχειρησιακό συνεχές.

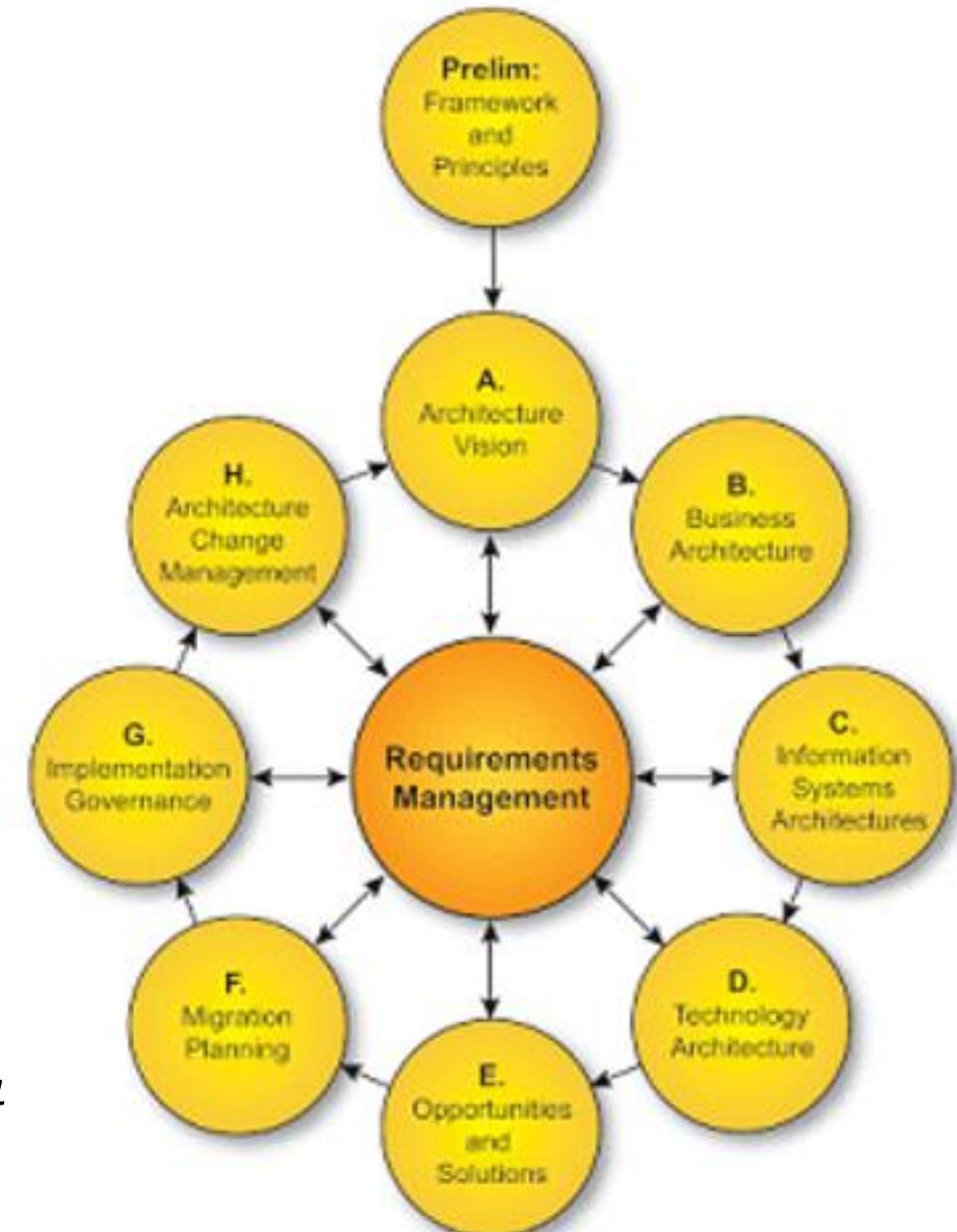
Πλαίσιο Open Group Βάση άντλησης πόρων

Είναι ένα σύνολο από πόρους, οδηγίες, πρότυπα και βασικές πληροφορίες που έχει στόχο να παρέχει βοήθεια και υποστήριξη στον αρχιτέκτονα ώστε να χρησιμοποιήσει ορθά τα αποτελέσματα της Μεθόδου Ανάπτυξης Αρχιτεκτονικής. Προκύπτει χάρη σε μελέτες περιπτώσεων (case studies), πίνακες ιδεών και απόψεων καθώς και από υλικά, εργαλεία και τεχνικές που έχουν χρησιμοποιηθεί.

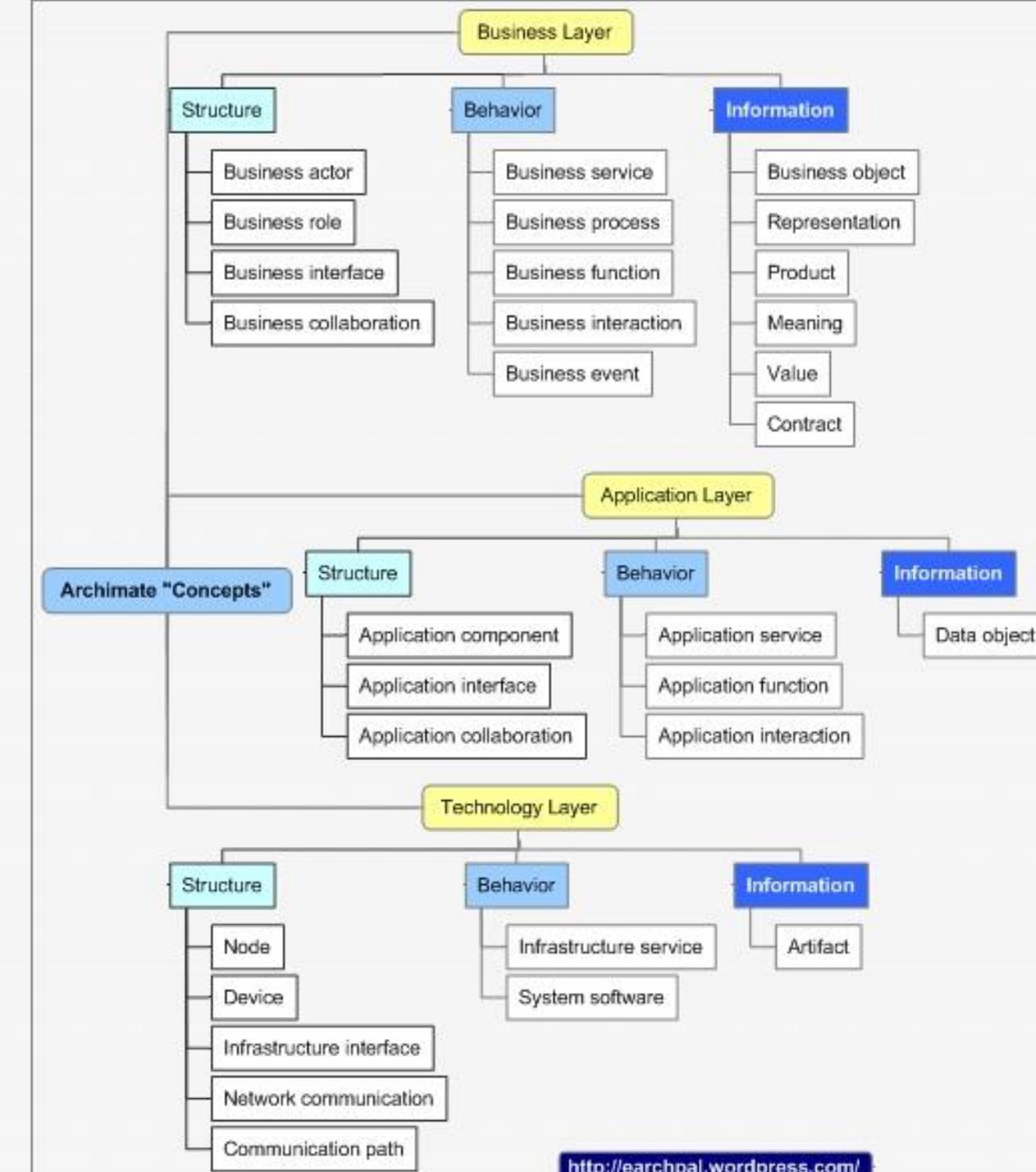
Πλαίσιο Open Group

Εμφανίζεται και ως TOGAF (The Open Group's Architecture Framework) και οργανώνεται σε 3 τομείς:

- Μέθοδος Ανάπτυξης Αρχιτεκτονικής (9 αρχιτεκτονικές φάσεις)
- Επιχειρησιακό Συνεχές (εικονική αρχιτεκτονική αποθήκη)
- Βάση άντλησης πόρων (παροχή βοήθειας στον αρχιτέκτονα για ορθή χρήση των αποτελεσμάτων της ΜΑΑ)
- Το σχήμα δείχνει τις Φάσεις
- Κάθε Φάση έχει μια **περιγραφή, αντικείμενο, βασικά βήματα, εισροές και εκροές**
- Επίσης περιέχει υλικό για ένα τεχνικό αρχιτεκτονικό πλαίσιο και πρότυπα, αρχές, τις σχέσεις του με άλλα πλαίσια και άλλα θέματα



TOGAF



Ομοσπονδιακό Πλαίσιο

Ομοσπονδιακό Πλαίσιο

Αυτό το πλαίσιο είναι γνωστό και ως **FEA** (Federal Enterprise Architecture). Είναι ακόμα σε αρχικό στάδιο και αποτελεί μια προσπάθεια για ενοποίηση των πολλών διαφορετικών προτύπων και διαδικασιών σχετικά με την επιχειρησιακή αρχιτεκτονική. Παρόλα αυτά είναι ένα αρκετά ολοκληρωμένο μοντέλο καθώς αποτελείται τόσο από μια ταξινόμηση (Zachman) όσο και από αρχιτεκτονικές διαδικασίες (TOGAF)

Ομοσπονδιακό Πλαίσιο

Μοντέλο τομέα: Μια ιδέα για το πώς θα πρέπει να αντιμετωπίζεται η αρχιτεκτονική επιχειρήσεων. Η προσέγγιση που ακολουθεί είναι ότι η επιχείρηση είναι ένα σύνολο από κομμάτια, τα οποία αντιπροσωπεύουν μια σημαντική λειτουργία της και χωρίζονται σε δύο κατηγορίες: τμήμα βασικών στόχων αποστολής και τμήμα επιχειρηματικών υπηρεσιών.

Μοντέλα αναφοράς: Περιγράφουν διαφορετικές προσεγγίσεις της αρχιτεκτονικής σχετικά με την εργασία (BRM – Business Reference Model: ορίζει τη λειτουργία των διάφορων υπηρεσιών του ομοσπονδιακού οργανισμού), τα συστατικά της (CRM – Components Reference Model: δίνει μια πιο τεχνολογική άποψη σχετικά με τα συστήματα που μπορούν να συνεισφέρουν στο σκοπό της επιχείρησης), την τεχνολογία (TRM – Technical Reference Model: ορίζει τα πρότυπα και την τεχνολογία που θα χρησιμοποιηθούν για την ανάπτυξη των συστημάτων), τα δεδομένα (DRM – Data Reference Model: ορίζει συγκεκριμένους τρόπους για περιγραφή δεδομένων) και την απόδοση (PRM – Performance Reference Model: ορίζει συγκεκριμένα μέτρα για την επιτυχία της αρχιτεκτονικής της επιχείρησης).

Ομοσπονδιακό Πλαίσιο

Μια **διαδικασία δημιουργίας** επιχειρησιακής αρχιτεκτονικής: Εστιάζει στη δημιουργία αρχιτεκτονικής για κάθε ανεξάρτητο τμήμα που συνθέτει την επιχείρηση.

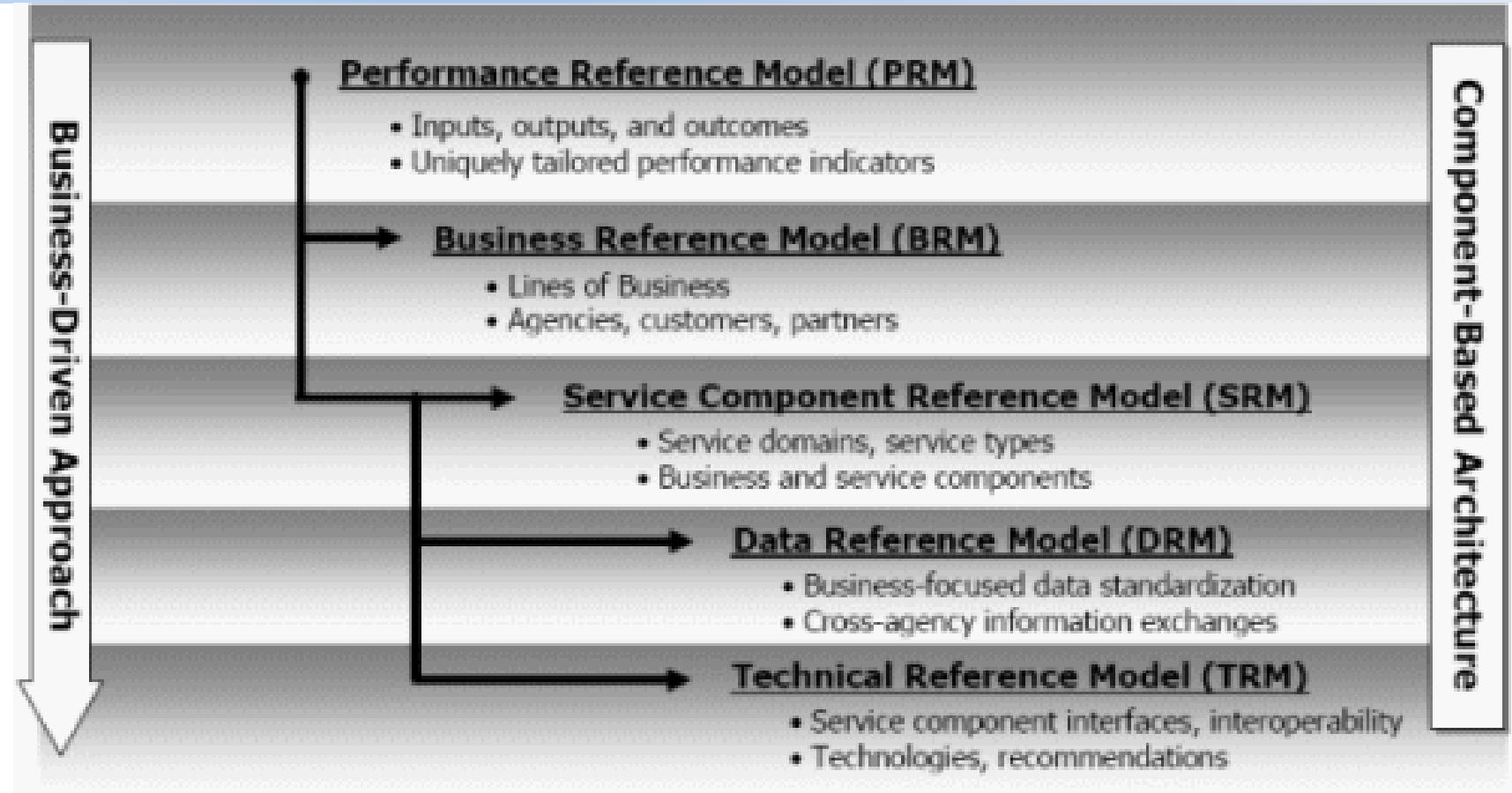
Μια **μεταβατική διαδικασία** για τη μεταφορά από την αρχική στην τελική αρχιτεκτονική.

Μια **ταξινόμηση** για τη λίστα στοιχείων που ανήκουν στις αρμοδιότητες της αρχιτεκτονικής της επιχείρησης.

Ένα **μέτρο επιτυχίας** για τα αποτελέσματα της εφαρμογής της αρχιτεκτονικής το οποίο χωρίζεται στην αξιολόγηση της ολοκλήρωσης και ωριμότητας της αρχιτεκτονικής, στο βαθμό χρήσης της σε διαδικασίες λήψης απόφασης και στα οφέλη που προκύπτουν από τη χρήση της.

Ομοσπονδιακό Πλαίσιο

Προσπάθεια για ενοποίηση των πολλών διαφορετικών προτύπων και διαδικασιών σχετικά με την ΕΑ.



- Διαδικασία δημιουργίας αρχιτεκτονικής για κάθε ανεξάρτητο τμήμα μιας επιχείρησης.
- Μεταβατική διαδικασία από αρχική σε τελική αρχιτεκτονική.

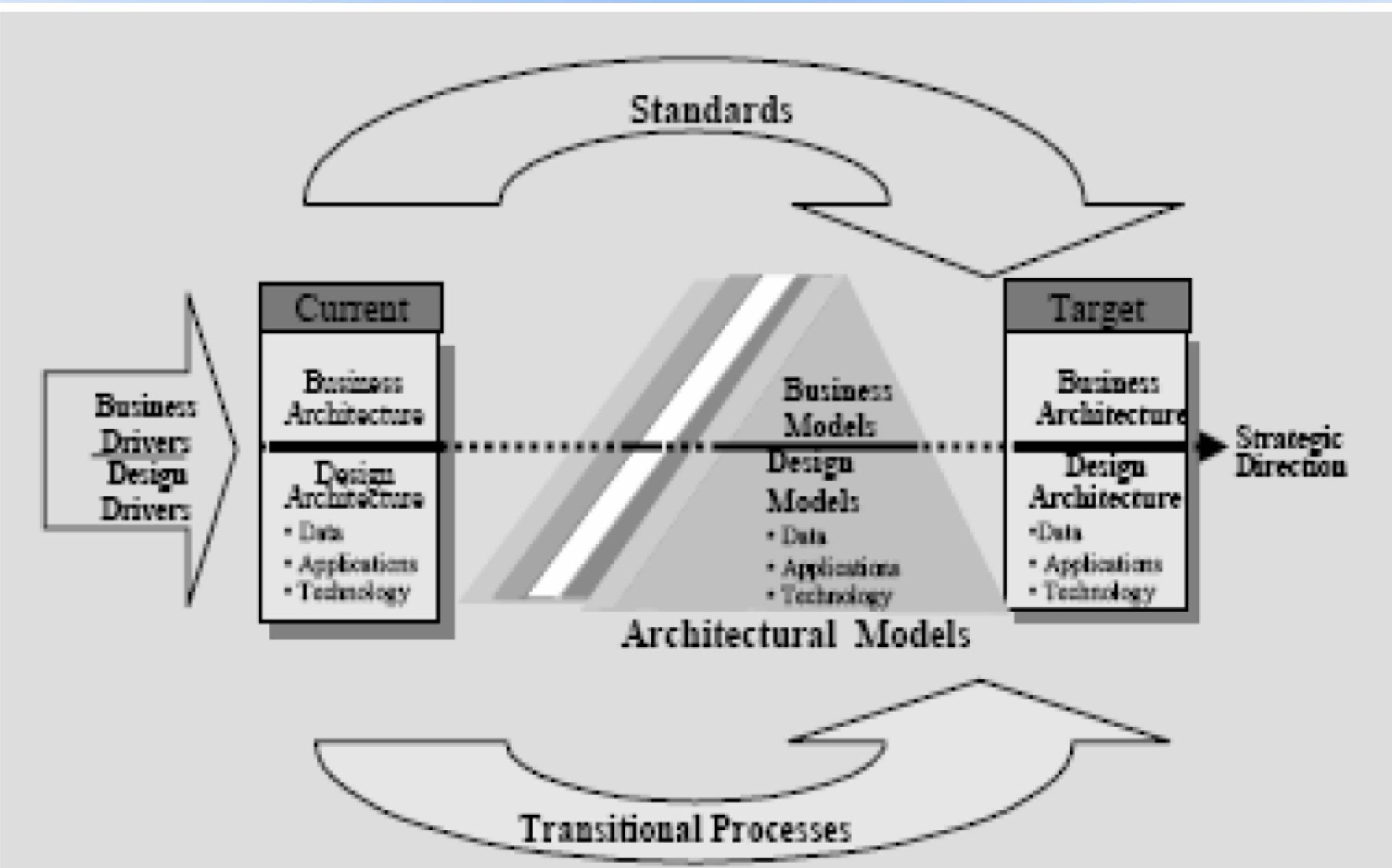
Πλαίσιο FEAf

- Το FEAf αποτελείται από 8 δομικά στοιχεία.
- Αρχιτεκτονικούς οδηγούς (Architecture drivers)
- Στρατηγική κατεύθυνση (Strategic direction)
- Παρούσα Αρχιτεκτονική (current architecture)
- Μελλοντική Αρχιτεκτονική (target architecture)
- Μεταβατικές διαδικασίες (transitional processes)
- Αρχιτεκτονικά τμήματα (Architectural segments)
- Αρχιτεκτονικά μοντέλα (Architectural Models)
- Πρότυπα (Standards)

Πλαίσιο FEAf

- Η **στρατηγική κατεύθυνση** οδηγεί στην ανάπτυξη της μελλοντικής αρχιτεκτονικής και περιλαμβάνει το όραμα της εταιρείας, τις αρχές και τους στόχους.
- Τα **αρχιτεκτονικά μοντέλα** καθορίζουν την αρχιτεκτονική της επιχείρησης, των δεδομένων, των εφαρμογών και της τεχνολογίας.
- Τα μοντέλα βασίζονται στο πλαίσιο **Zachman**. Ο Zachman παρείχε μια σύντομη περιγραφή για τι πρέπει να περιέχουν τα μοντέλα
- Στο επόμενο σχήμα (πηγή: EABOK, MITRE Corporation) φαίνεται το FEAf.

Πλαίσιο FEAF



Πλαίσιο *Gartner*

Μεθοδολογία Gartner

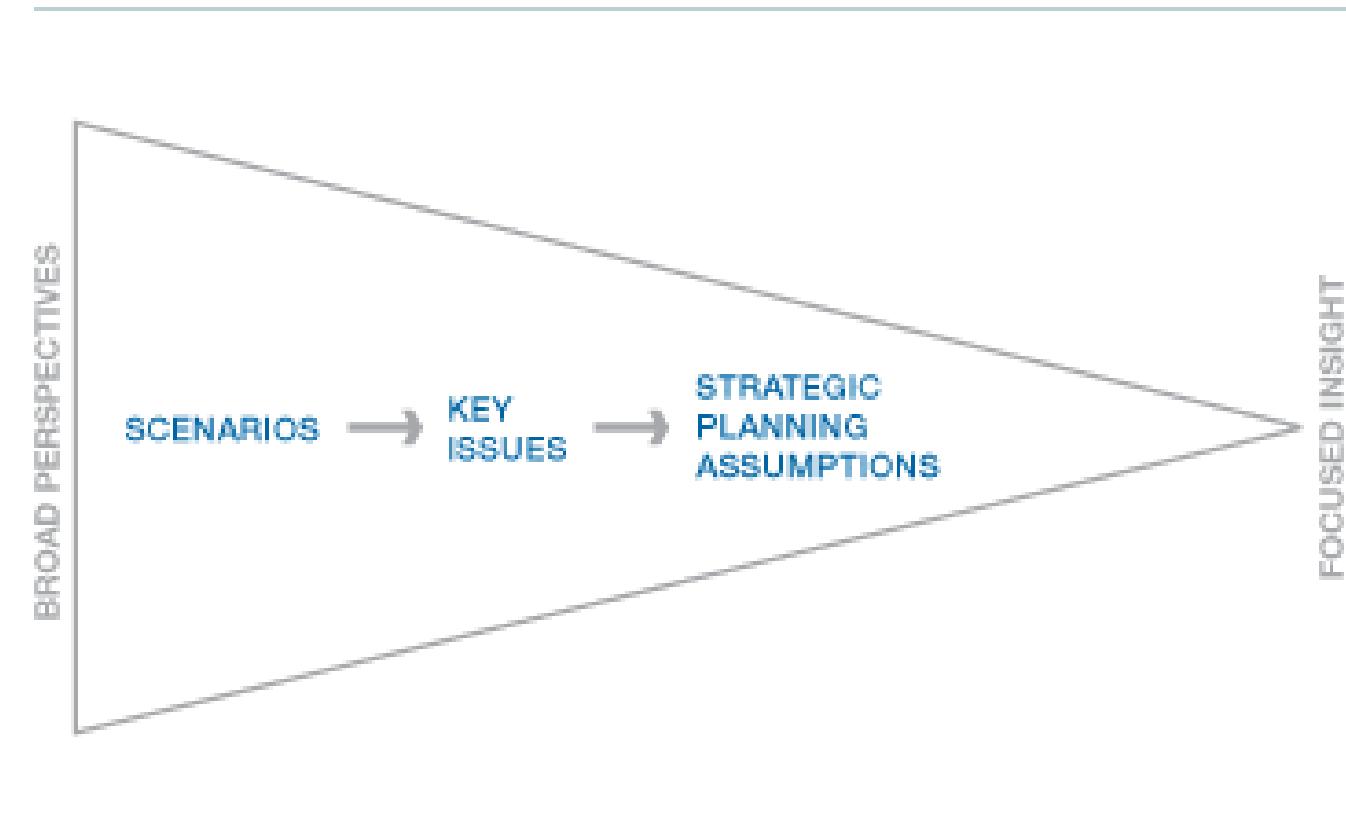
Διαφοροποιείται από τα προηγούμενα τρία πλαίσια, καθώς πρόκειται για την πρακτική που εφαρμόζει η εταιρία Gartner, η κυρίαρχη συμβουλευτική στον ιδιωτικό τομέα για θέματα τεχνολογίας και πληροφοριών. Στηρίζεται στην πεποίθηση ότι η αρχιτεκτονική είναι μια διαρκής διαδικασία δημιουργίας, συντήρησης και ειδικά αξιοποίησης η οποία προσφέρει ζωτικότητα στην επιχείρηση.

Βασικό στόχο έχει να ενισχύσει τη συνεργασία και συνύπαρξη τριών παραγόντων, **των ιδιοκτητών της επιχείρησης, των ειδικών για θέματα πληροφοριών και των υπεύθυνων για την εφαρμογή της τεχνολογίας**, καθώς πιστεύεται ότι εάν αυτοί ενωθούν πίσω από ένα κοινό όραμα η επιχείρηση έχει πετύχει σε πραγματικούς όρους κέρδους.

Μεθοδολογία Gartner

Στόχο έχει να ενισχύσει τη συνεργασία και συνύπαρξη 3 παραγόντων (ιδιοκτητών, ειδικών για θέματα πληροφοριών, υπεύθυνων για την εφαρμογή της τεχνολογίας).

Γέννηση της κατάλληλης αρχιτεκτονικής με βάση την επιθυμητή στρατηγική και τη σωστή οργάνωσή της.



Ανάλυση Πληροφοριακών Συστημάτων

Εαρινό Εξάμηνο 2020-2021

Lec11 - 08/06/2021

Διδάσκων: Γεώργιος Χρ. Μακρής

Ηλεκτρονικό Ανθοπωλείο

1. Ο κάθε πελάτης θα μπορεί να δημιουργεί προσωπικό λογαριασμό στον οποίο θα αποθηκεύονται τα προσωπικά του δεδομένα και ο οποίος θα προστατεύεται με κωδικό πρόσβασης (password).
2. Το ανθοπωλείο θα μπορεί να δέχεται παραγγελίες από το Internet. Ο χρήστης θα επιλέγει ένα-ένα τα διαφορετικά λουλούδια που θα βάλει στην ανθοδέσμη του καθώς και τον αριθμό των λουλουδιών του κάθε είδους που επιθυμεί να αγοράσει. Εναλλακτικά, ο χρήστης θα μπορεί να επιλέξει μια ανθοσύνθεση, η οποία είναι προκαθορισμένη όσον αφορά το περιεχόμενο αλλά και την τιμή της.
3. Ανάλογα με τη διεύθυνση αποστολής της ανθοδέσμης (κοντινή, μακρινή), θα επισημαίνεται στον πελάτη η ανθεκτικότητα των λουλουδιών. Επιπλέον, για κάθε λουλούδι θα παρουσιάζονται τα χαρακτηριστικά του, δηλαδή μια περιγραφή για το άρωμά του, το χρώμα του, το μέγεθος του, η οικογένειά του, η προέλευση, η ειδική περίσταση στην οποία προσφέρεται, μια φωτογραφία κ.λπ. Αντίστοιχα, για τις συνθέσεις θα δίνεται μια σύντομη περιγραφή, ο τύπος της σύνθεσης (ρομαντική, επαγγελματική, κ.λπ.) καθώς και μία φωτογραφία.



Ηλεκτρονικό Ανθοπωλείο

4. Όταν ο πελάτης ολοκληρώσει την επιλογή των λουλουδιών θα ζητούνται από το σύστημα τα στοιχεία του παραλήπτη της παραγγελίας. Τα στοιχεία αυτά θα αποθηκεύονται στο σύστημα στον προσωπικό λογαριασμό του κάθε πελάτη και θα είναι διαθέσιμα για να χρησιμοποιηθούν κάποια επόμενη φορά. Ταυτόχρονα, ο πελάτης θα γράφει το μήνυμα το οποίο θα ήθελε να συνοδεύει την ανθοδέσμη.
5. Ο πελάτης θα έχει την δυνατότητα να προσθέσει όσες ανθοδέσμες θέλει στο “ηλεκτρονικό καρότσι” του, να τις αφαιρέσει ακυρώνοντας την επιλογή του, ή να ολοκληρώσει τη συναλλαγή του με την πληρωμή αυτών που έχει επιλέξει.
6. Επιπλέον, για κάθε πελάτη, το σύστημα θα μπορεί να αποθηκεύει τα ονόματα φίλων, συνεργατών, αγαπημένων προσώπων μαζί με σημαντικές ημερομηνίες-επετείους, έτσι ώστε όταν πλησιάζει η ημερομηνία μιας επετείου να στέλνει e-mail στον πελάτη και να του υπενθυμίζει την ημέρα αυτή.
7. Το ηλεκτρονικό ανθοπωλείο θα συνδέεται με το λογιστήριο της εταιρείας όπου υπάρχει το σύστημα πληρωμής με πιστωτικές κάρτες (το υποσύστημα αυτό δεν ανήκει στο ηλεκτρονικό ανθοπωλείο αλλά συνεργάζεται με αυτό). Το σύστημα πιστωτικών καρτών επιτρέπει την ασφαλή πληρωμή των παραγγελθέντων λουλουδιών.

Ενοποιημένη προσέγγιση

Η ενοποιημένη προσέγγιση είναι μια διαδικασία αρχιτεκτονικοκεντρική (architecture centric) που εστιάζει την προσοχή στην όσο το δυνατόν νωρίτερα ανάπτυξη της αρχιτεκτονικής του λογισμικού καθώς και του καθαυτό λογισμικού. Η υιοθέτηση μιας αρχιτεκτονικής διευκολύνει την παράλληλη ανάπτυξη, ελαχιστοποιεί την επανάληψη, αυξάνει την πιθανότητα της επαναχρησιμοποίησης τμημάτων, και βελτιώνει τη συντηρησιμότητα των συστημάτων.

Τέλος, η ανάπτυξη συστημάτων με την ενοποιημένη προσέγγιση είναι βασισμένη στις περιπτώσεις χρήσης, δίνοντας έμφαση στη λεπτομερή κατανόηση για το πώς θα χρησιμοποιηθεί το σύστημα.

Συνοψίζοντας θα μπορούσαμε να πούμε ότι οι βασικές αρχές στις οποίες στηρίζεται η ενοποιημένη προσέγγιση είναι τέσσερις:

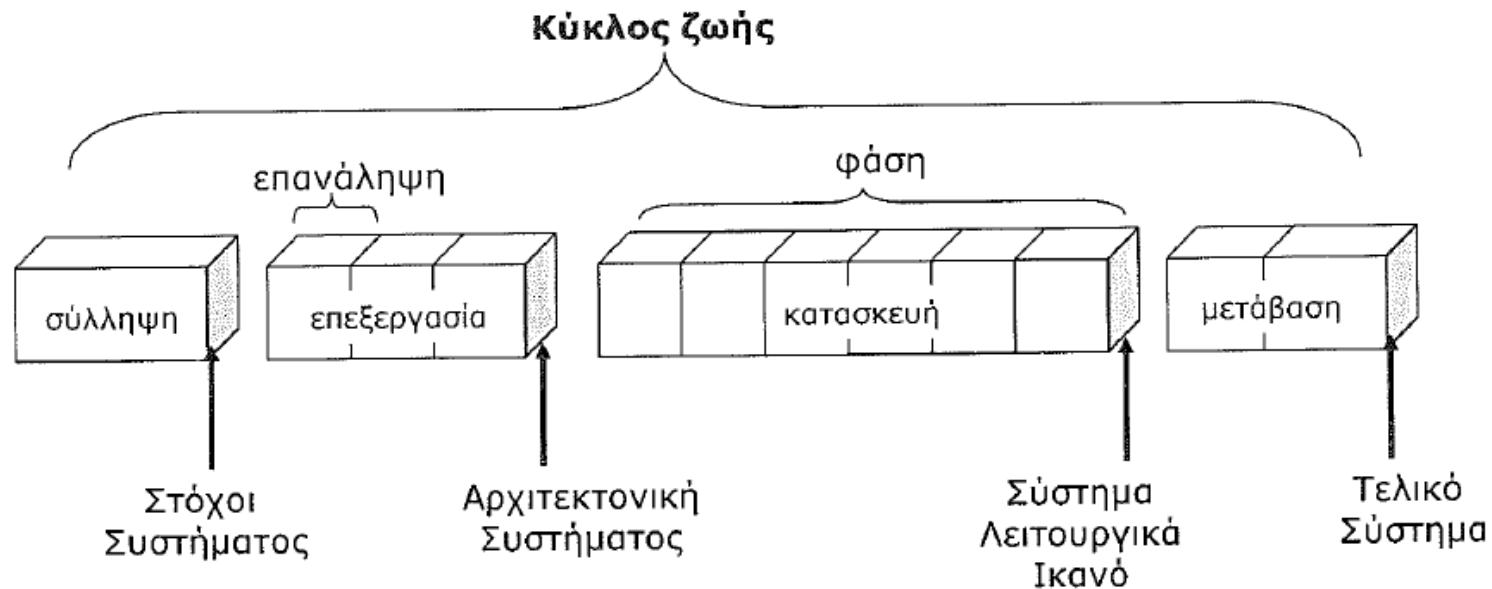
1. Εφαρμόζεται επαναληπτικά.
2. Το σύστημα χτίζεται σταδιακά.
3. Είναι αρχιτεκτονικοκεντρική και
4. Βασίζεται στις περιπτώσεις χρήσης.



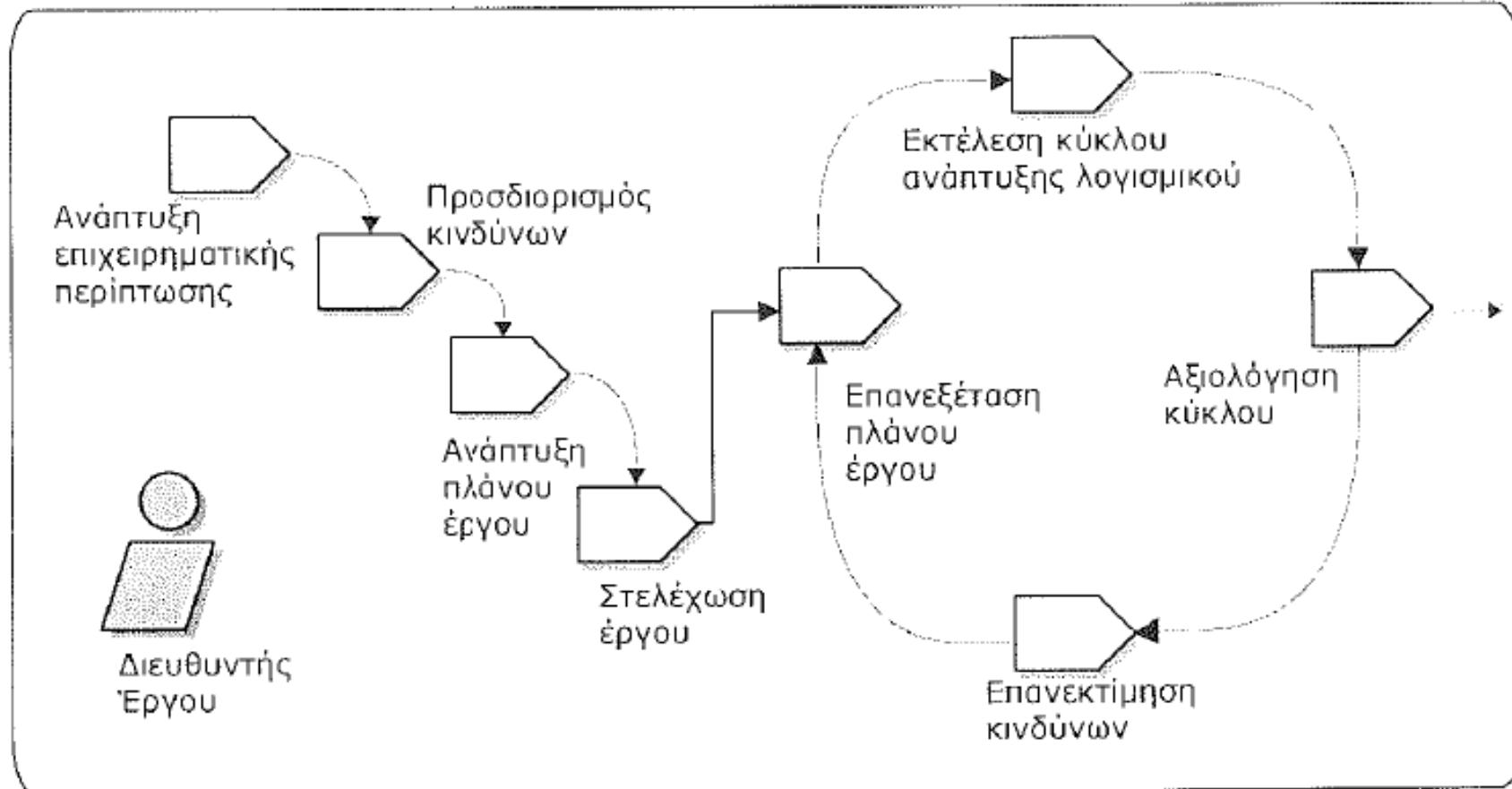
Κύκλος Ζωής

Αν προσπαθήσουμε να συνοψίσουμε σε μία φράση το περιεχόμενο της κάθε φάσης θα έχουμε:

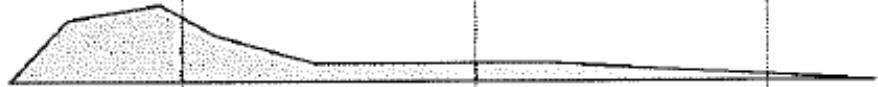
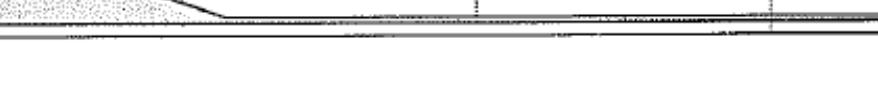
1. *Σύλληψη*: όπου τεκμηριώνουμε την επιχειρηματική ανάγκη ύπαρξης του συστήματος.
2. *Λεπτομερής επεξεργασία*: όπου προγραμματίζουμε το έργο (project planning) και δημιουργούμε την αρχιτεκτονική του συστήματος.
3. *Κατασκευή*: όπου αναπτύσσουμε το σύστημα.
4. *Μετάβαση*: όπου παραδίδουμε το σύστημα στους τελικούς χρήστες.



Κύκλος Ζωής (Ροή Εργασίας)



Κύκλος Ζωής

ΕΡΓΑΣΙΕΣ	ΦΑΣΕΙΣ			
	Σύλληψη	Επεξεργασία	Κατασκευή	Μετάβαση
Μοντελοποίηση επιχειρηματικών διαδικασιών				
Καταγραφή απαιτήσεων				
Ανάλυση και σχεδίαση				
Υλοποίηση				
Έλεγχος				
Εγκατάσταση				
Διοίκηση σχηματισμών				
Διαχείριση έργου				
Περιβάλλον				



Κατασκευή Μοντέλων από τις ροές

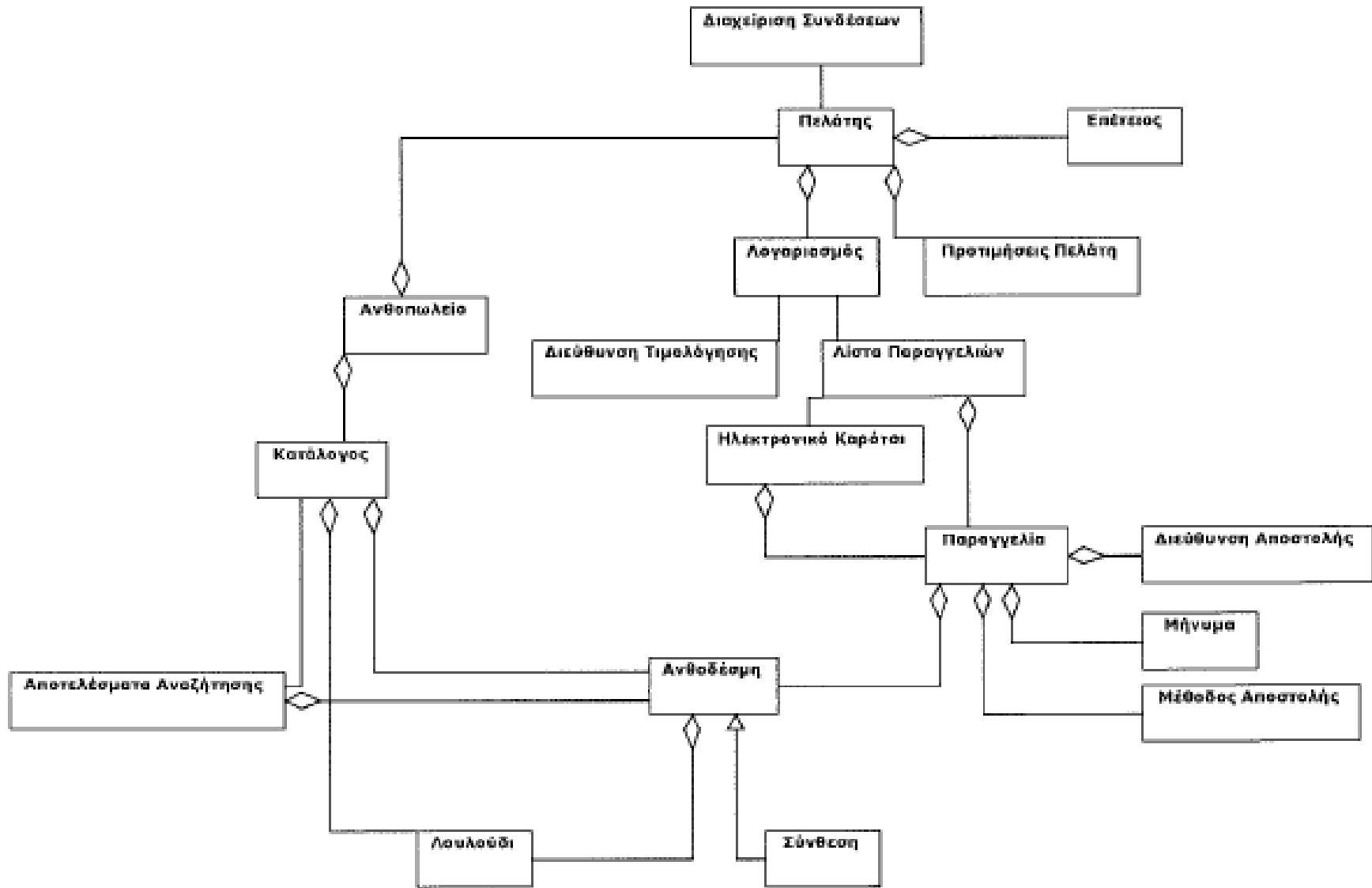
1. *Επιχειρηματικό μοντέλο* (business model): μοντελοποιεί την επιχείρηση και τις διαδικασίες της.
2. *Μοντέλο πεδίου προβλήματος* (domain model): περιγράφει το σύστημα, καθορίζει τα όρια και τις συνιστώσες του συστήματος.
3. *Μοντέλο περιπτώσεων χρήσης* (use case model): ορίζει τις προδιαγραφές του συστήματος.
4. *Μοντέλο ανάλυσης* (analysis model): περιγράφει τις βασικές ιδέες (concepts) του συστήματος.
5. *Μοντέλο σχεδίασης* (design model): περιγράφει πώς θα υλοποιηθεί το σύστημα.
6. *Μοντέλο διαδικασιών* (process model): περιγράφει τις φυσικές διαδικασίες του συστήματος και τους μηχανισμούς συγχρονισμού.
7. *Μοντέλο διάταξης* (deployment model): περιγράφει τη διάταξη-τοπολογία του φυσικού συστήματος σε σχέση με το περιβάλλον όπου εκτελείται το λογισμικό.
8. *Μοντέλο υλοποίησης* (implementation model): περιγράφει τον τρόπο με τον οποίο θα κατασκευασθεί το τελικό σύστημα από τα συστατικά του στοιχεία.
9. *Μοντέλο ελέγχου* (test model): περιγράφει τη στρατηγική και τις διαδικασίες ελέγχου, καθώς και τις περιπτώσεις ελέγχου (test cases).

Καταγραφή και ανάλυση απαιτήσεων

1. Προσδιορισμός του μοντέλου του πεδίου προβλήματος. Το μοντέλο του προβλήματος περιγράφει τις βασικές οντότητες-έννοιες που αποτελούν το σύστημα.
2. Προσδιορισμός του αντικειμένου εργασιών του συστήματος (χάρτης εμβέλειας, scope map). Προσδιορίζει με επιγραμματικό τρόπο τις απαιτήσεις του συστήματος.
3. Προσδιορισμός του μοντέλου των περιπτώσεων χρήσης. Προσδιορίζει τον τρόπο με τον οποίο χρησιμοποιείται το σύστημα.
4. Οργάνωση των περιπτώσεων χρήσης σε πακέτα. Ομαδοποιεί τις περιπτώσεις χρήσης σε λογικά πακέτα.
5. Ανάπτυξη πρωτότυπων του συστήματος με σκοπό την επίδειξη στο χρήστη βασικών παραμέτρων.



Διάγραμμα κλάσεων στο πεδίο του προβλήματος



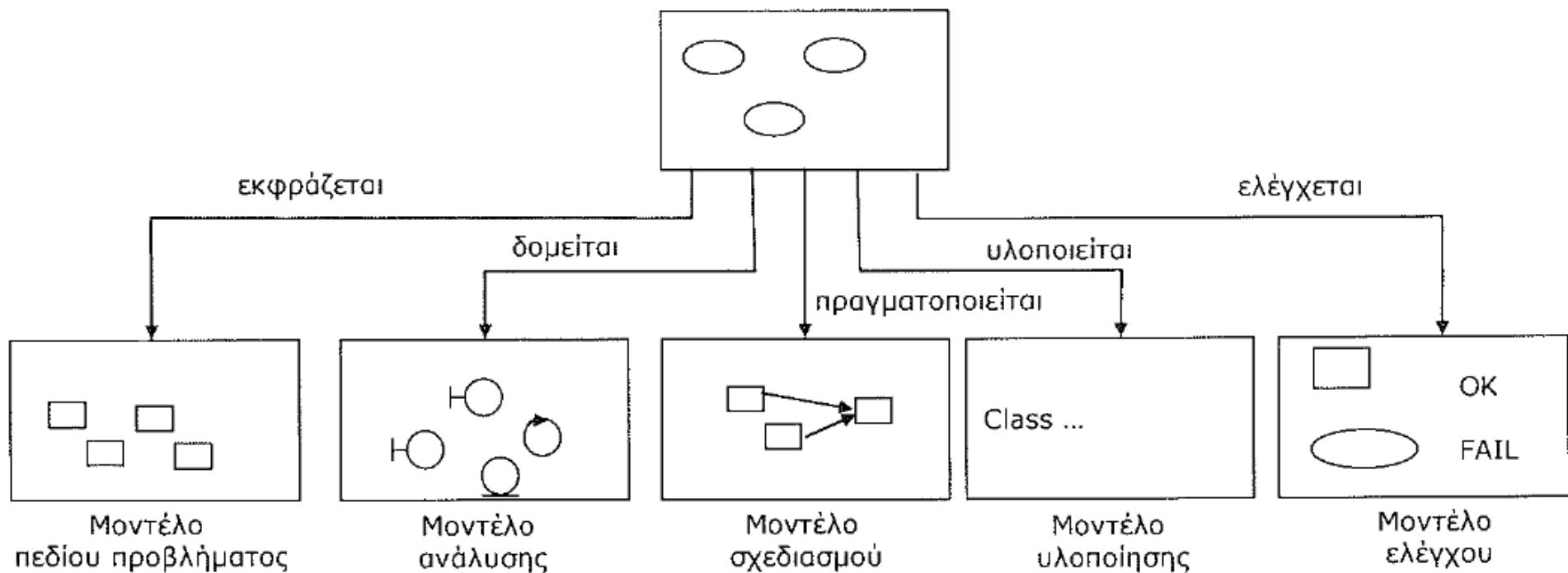
Προσδιορισμός κλάσεων και σχέσεων μεταξύ τους

Τύπος ουσιαστικού	Παραδείγματα	Τι μπορεί να είναι
Οικογένεια/κατηγορία πραγμάτων	Άτομο, λογιστής, όχημα, τιμολόγιο, προϊόν	Κλάση
Ένα όνομα, ένα ουσιαστικό με επιθετικό προσδιορισμό	Γιώργος, μεγαλύτερο όχημα, πρώτο τιμολόγιο	Αντικείμενο
Ιδιότητα αντικειμένου	Ηλικία, χρώμα	Πεδίο κλάσης
Τιμή αντικειμένου	27 ετών, κίτρινο	Τιμή πεδίου
Μια συνθήκη	Ηλικία μεγαλύτερη από	Κατάσταση αντικειμένου
Ρήμα ενεργητικής φωνής μετά από αντικείμενο	Γιώργος δίνει αναφορά Γιώργος αποθηκεύει	Μέθοδος κλάσης
Γεγονός, χρόνος, συμβάν	Χτύπημα τηλεφώνου, Πάρτι γενεθλίων	Μέθοδος κλάσης, Συνεργασία αντικειμένων



Σχέσεις του μοντέλου περιπτώσεων χρήσης με τα άλλα μοντέλα

Μοντέλο περιπτώσεων χρήσης



Περιπτώσεις χρήσης

- για την περιγραφή επιχειρηματικών διαδικασιών,
- για την περιγραφή των προδιαγραφών του συστήματος,
- για τον προσδιορισμό των απαιτήσεων,
- για την τεκμηρίωση της λειτουργικότητας του συστήματος.

- Ποιος χρησιμοποιεί το σύστημα;
- Πώς χρησιμοποιείται το σύστημα;
- Ποιο είναι το περιβάλλον και ποια είναι τα όρια του συστήματος;
- Πώς οργανώνεται το σύστημα σε πακέτα;



Περιπτώσεις χρήσης

Για το ηλεκτρονικό ανθοπωλείο μπορούμε να ορίσουμε ένα αρχικό σύνολο περιπτώσεων χρήσης (scope map), οι οποίες είναι:

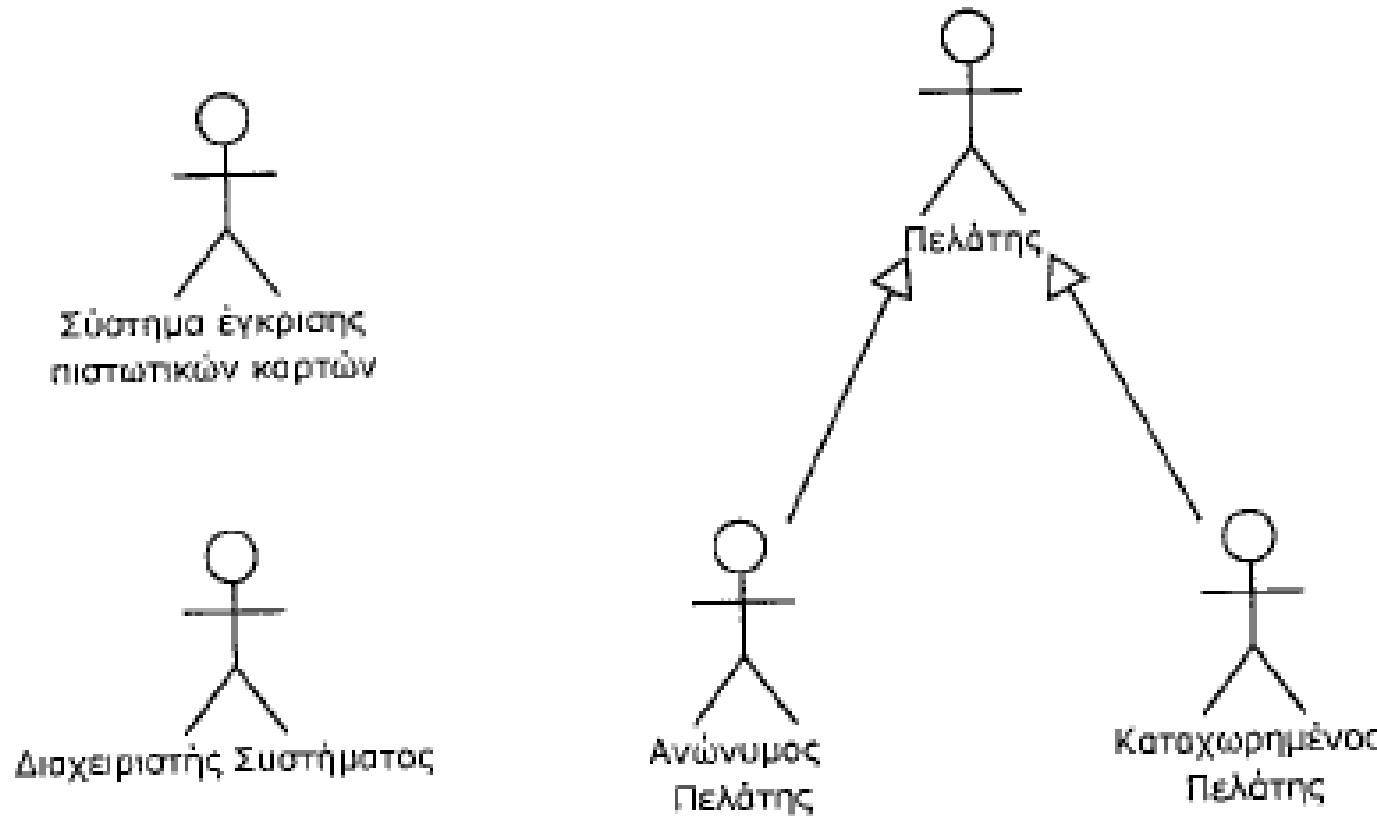
1. Η δημιουργία λογαριασμού πελάτη
2. Η σύνδεση του πελάτη με το σύστημα (Log In)
3. Η διαχείριση της παραγγελίας
4. Η δημιουργία της ανθοδέσμης
5. Η επιλογή των λουλουδιών
6. Η επιλογή της ανθοδέσμης
7. Η προβολή των στοιχείων των λουλουδιών ή της ανθοδέσμης
8. Η διαχείριση του “ηλεκτρονικού καροτσιού”
9. Η διαχείριση των στοιχείων του παραλήπτη ή του μηνύματος
10. Η πληρωμή του λογαριασμού
11. Η διαχείριση των επετείων
12. Η διαχείριση των προτιμήσεων του πελάτη

Χειριστές συστήματος

1. Ο «Πελάτης»
2. Ο «Ανώνυμος Πελάτης», ένας πελάτης που δεν έχει ακόμα ταυτοποιηθεί από το σύστημα. Το όνομα του χειριστή αυτού φαίνεται λίγο παράξενο, αλλά οι χειριστές αναπαριστούν και τους ρόλους του συστήματος. Συνεπώς, ο ανώνυμος πελάτης είναι ο πελάτης που απλώς δεν έχει καταχωρίσει ακόμη τα στοιχεία του ή δεν έχει κοιτάξει ακόμη τα προϊόντα που προσφέρει το ανθοπωλείο «Λευκός Κρίνος».
3. Ο «Καταχωρημένος Πελάτης», ο οποίος έχει καταχωρίσει τα στοιχεία του στο σύστημα και το σύστημα γνωρίζει την ταυτότητά του.
4. Ο «Διαχειριστής Συστήματος» είναι ο εργαζόμενος στην εταιρεία που διατηρεί το ηλεκτρονικό ανθοπωλείο και
5. Το «Σύστημα έγκρισης πιστωτικών καρτών», το οποίο είναι ένας χειριστής που δοα ως πληρεξούσιος του εξωτερικού συστήματος έγκρισης πιστωτικών καρτών.



Χειριστές συστήματος



Κανόνες για τον ορισμό των περιπτώσεων χρήσης

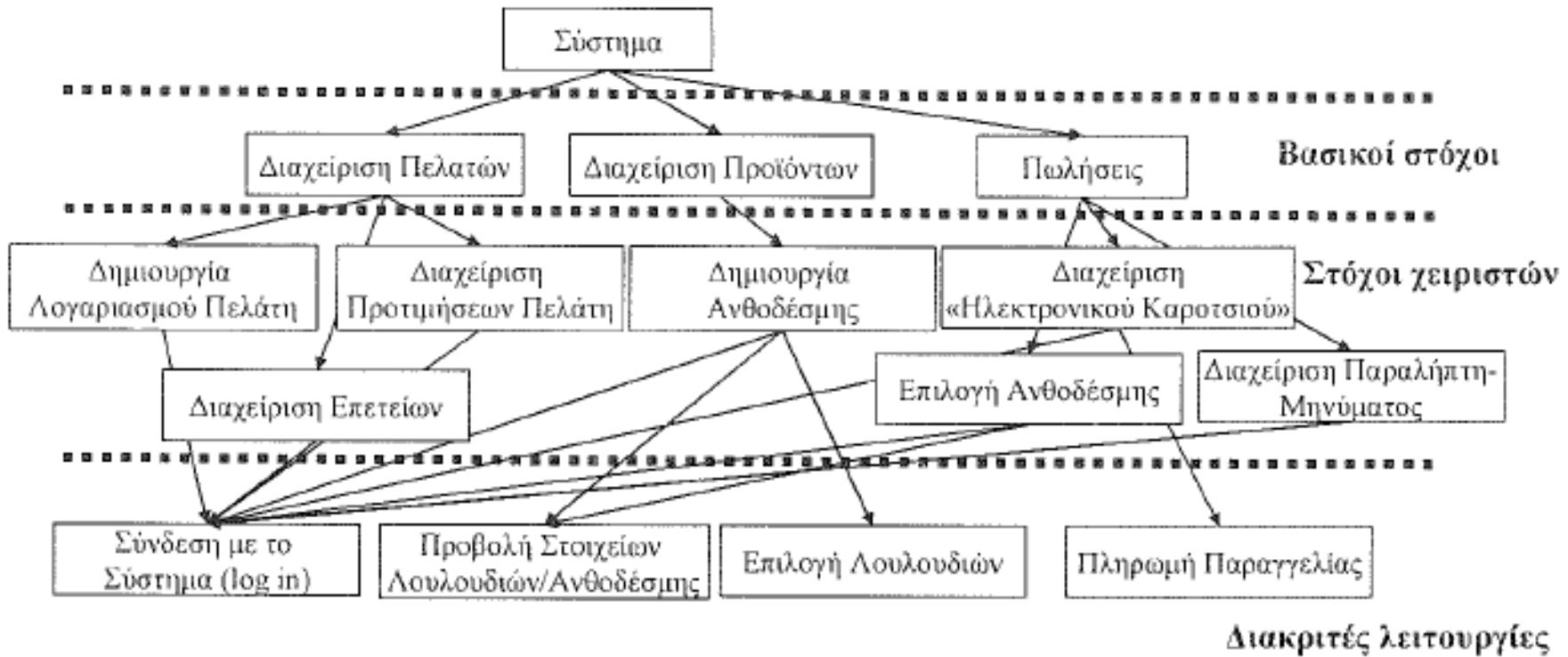
- 1. Το όνομα της περίπτωσης χρήσης δείχνει ενέργεια.** Η περίπτωση χρήσης περιγράφει μια συμπεριφορά του συστήματος και συνεπώς το όνομα της θα πρέπει να δείχνει ενέργεια (ρήμα ή ουσιαστικό που δείχνει ενέργεια).
- 2. Η περίπτωση χρήσης πρέπει να περιγράφει μια πλήρη συμπεριφορά που αρχίζει με το γεγονός έναρξης από τον αρχικό χειριστή και τελειώνει με την επιτυχημένη επίτευξη του σκοπού.** Εάν η προτεινόμενη περίπτωση χρήσης είναι απλώς ένα βήμα για την επίτευξη του τελικού στόχου του χειριστή, τότε η προτεινόμενη περίπτωση δεν θα πρέπει να θεωρηθεί ως περίπτωση χρήσης. Για παράδειγμα, η επιλογή ενός λουλουδιού είναι ένα βήμα για τη δημιουργία μιας ανθοδέσμης και δεν μπορεί να σταθεί από μόνη της ως ξεχωριστή περίπτωση χρήσης.
- 3. Η περίπτωση χρήσης θα πρέπει να μπορεί να ολοκληρωθεί.** Για να επιτευχθεί ο σκοπός της περίπτωσης χρήσης και να παραχθεί κάποιο αποτέλεσμα για το χειριστή του συστήματος θα πρέπει να υπάρχει δυνατότητα ολοκλήρωσης της περίπτωσης χρήσης. Αυτό σημαίνει ότι θα υπάρχει μια ροή εκτέλεσης που θα επιτρέπει την ολοκλήρωση της επιθυμητής συμπεριφοράς. Η ροή αυτή χαρακτηρίζεται ως το “χαρούμενο μονοπάτι” (happy path).

Κανόνες για τον ορισμό των περιπτώσεων χρήσης

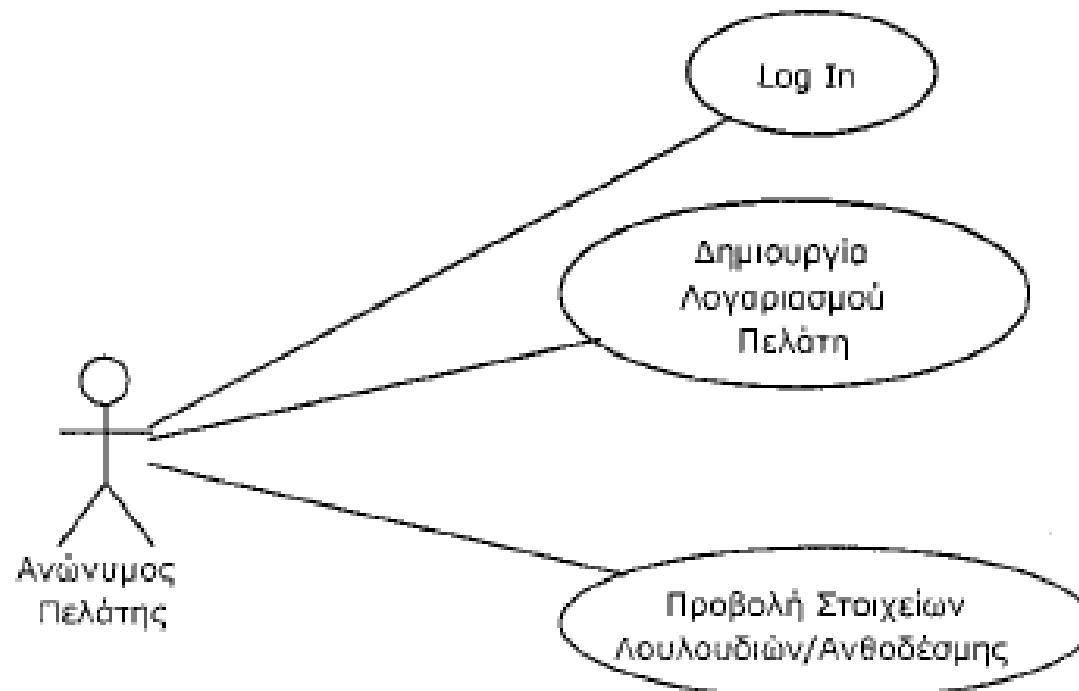
4. **Αντίστροφη περίπτωση χρήσης.** Όταν υπάρχει περίπτωση χρήσης που επιτυγχάνει ένα διακριτό αποτέλεσμα, θα πρέπει συνήθως να υπάρχει και η αντίστροφη περίπτωση χρήσης, που ακυρώνει αυτό το αποτέλεσμα. Για παράδειγμα η περίπτωση χρήσης «Παραγγελία ανθοδέσμης» καλό είναι να συνοδεύεται από την αντίστροφή της η οποία είναι η «Ακύρωση παραγγελίας».
5. **Μία περίπτωση χρήσης — μία συμπεριφορά.** Μία περίπτωση χρήσης είναι σκόπιμο να περιγράφει μία και μόνο μία συμπεριφορά. Είναι πολύ συνηθισμένο να έχουμε γενικές περιπτώσεις χρήσης που περιγράφουν περισσότερες από μία συμπεριφορές. Η λύση στο πρόβλημα αυτό είναι να εισάγουμε περιπτώσεις χρήσης που περιγράφουν μια γενική συμπεριφορά και στη συνέχεια να αναλύουμε τη γενική περίπτωση χρήσης σε υποπεριπτώσεις.
6. **Χρησιμοποιήστε την ορολογία του πεδίου προβλήματος.** Με τον τρόπο αυτό οι χρήστες και οι συμμετέχοντες στην ανάπτυξη του συστήματος μπορούν να συμμετάσχουν στην αξιολόγηση των περιπτώσεων χρήσης.



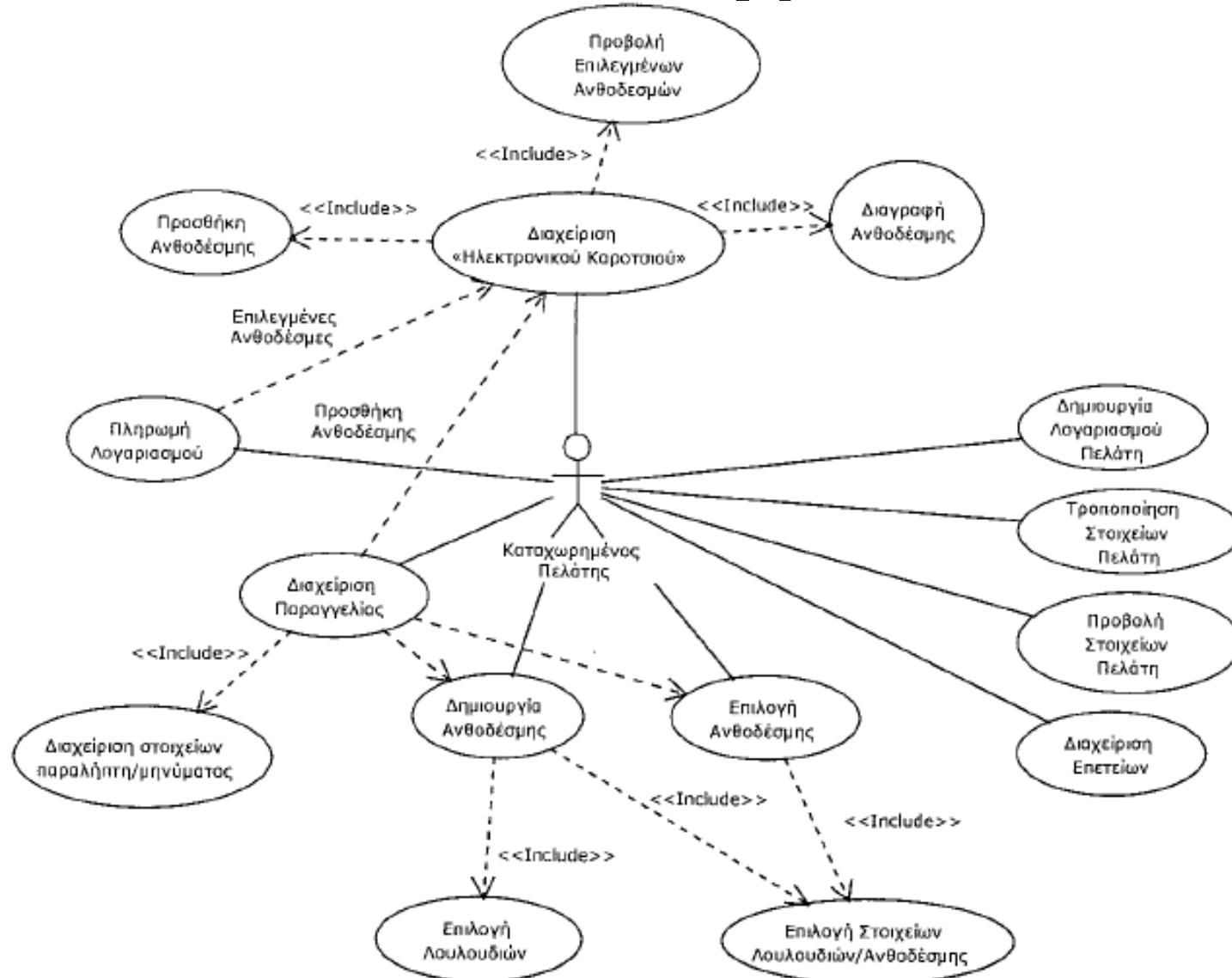
Επίπεδα περιπτώσεων χρήσης



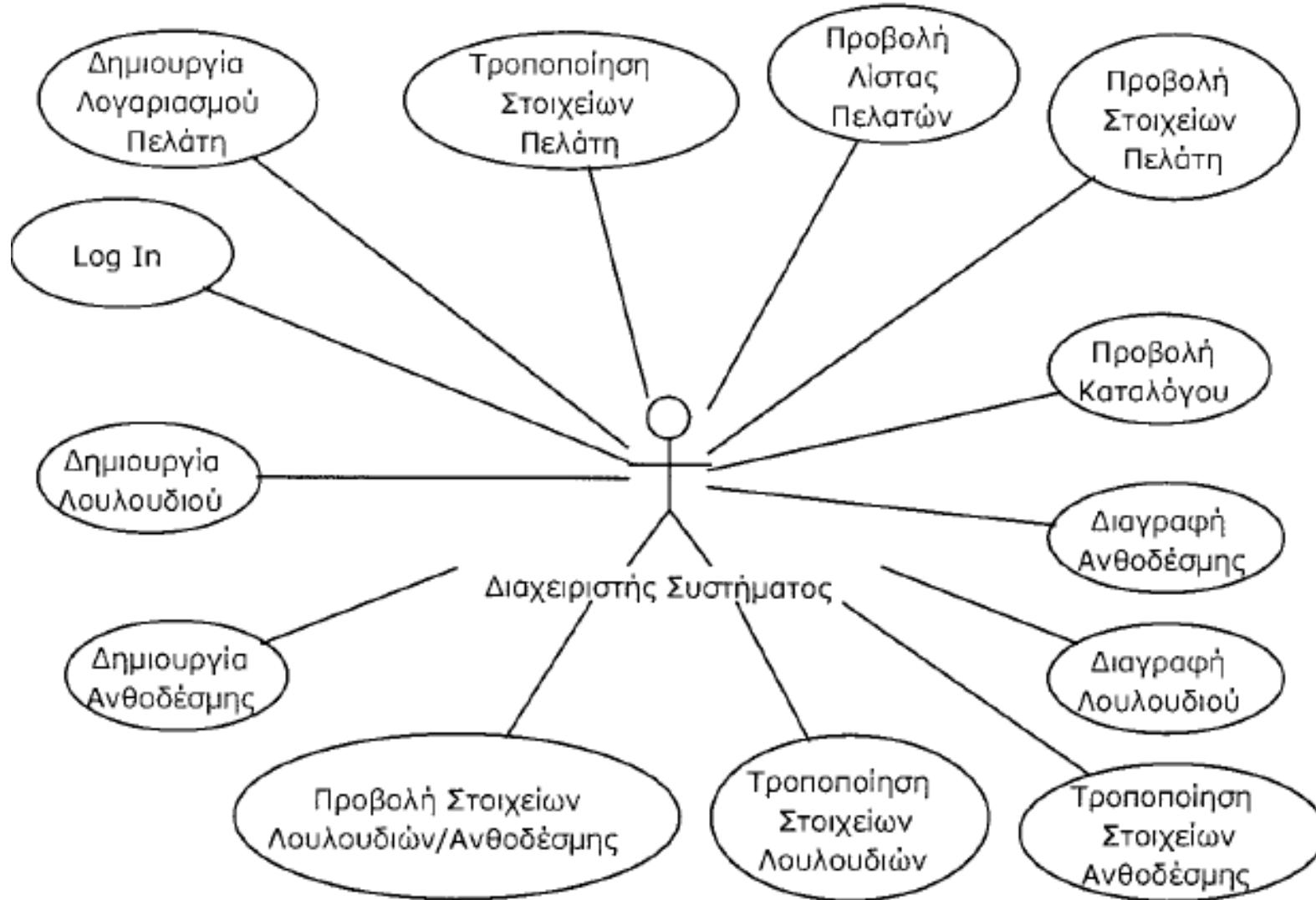
Περίπτωση χρήσης: Ανώνυμος πελάτης



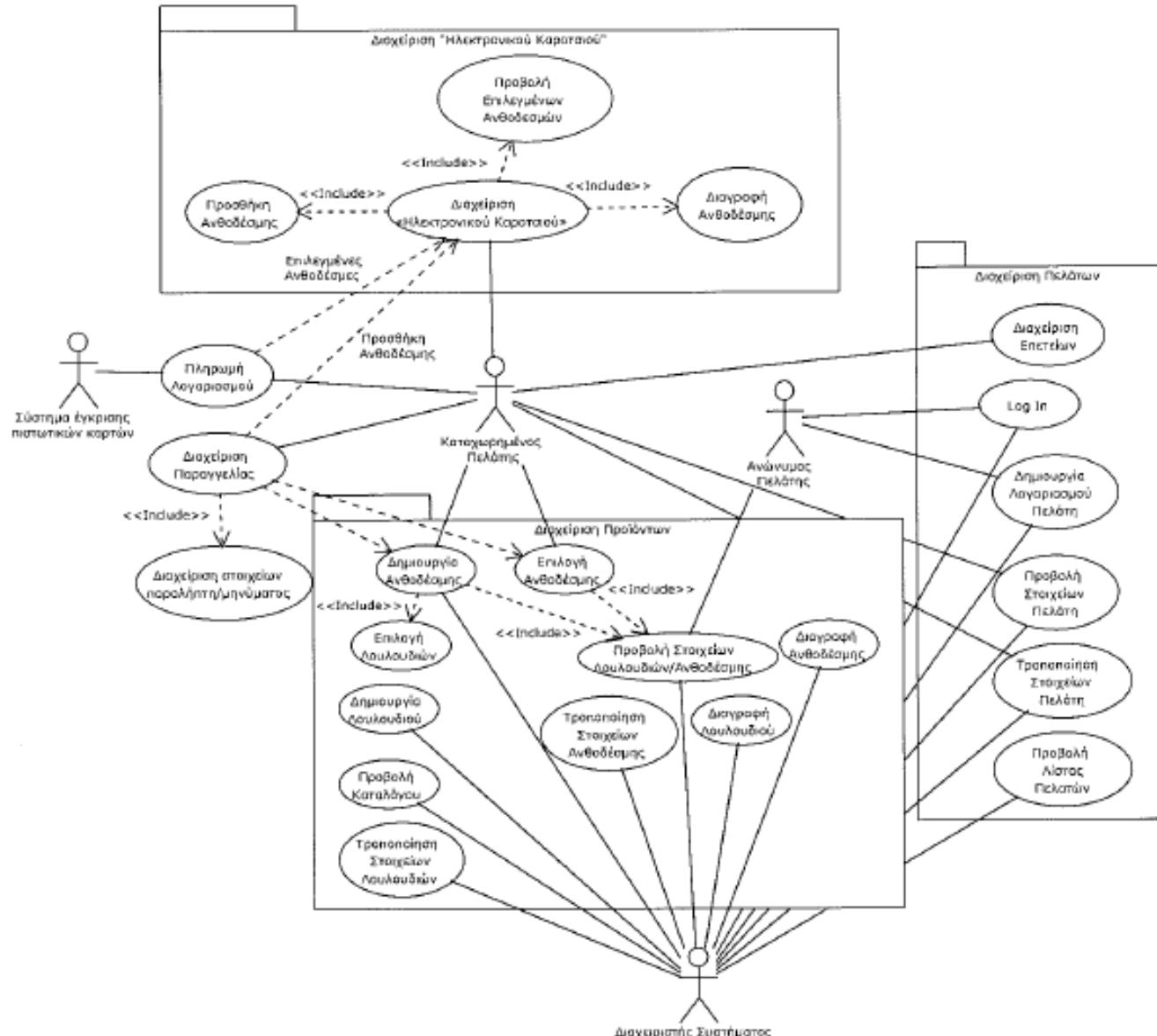
Περίπτωση χρήσης: Καταχωρημένος πελάτης



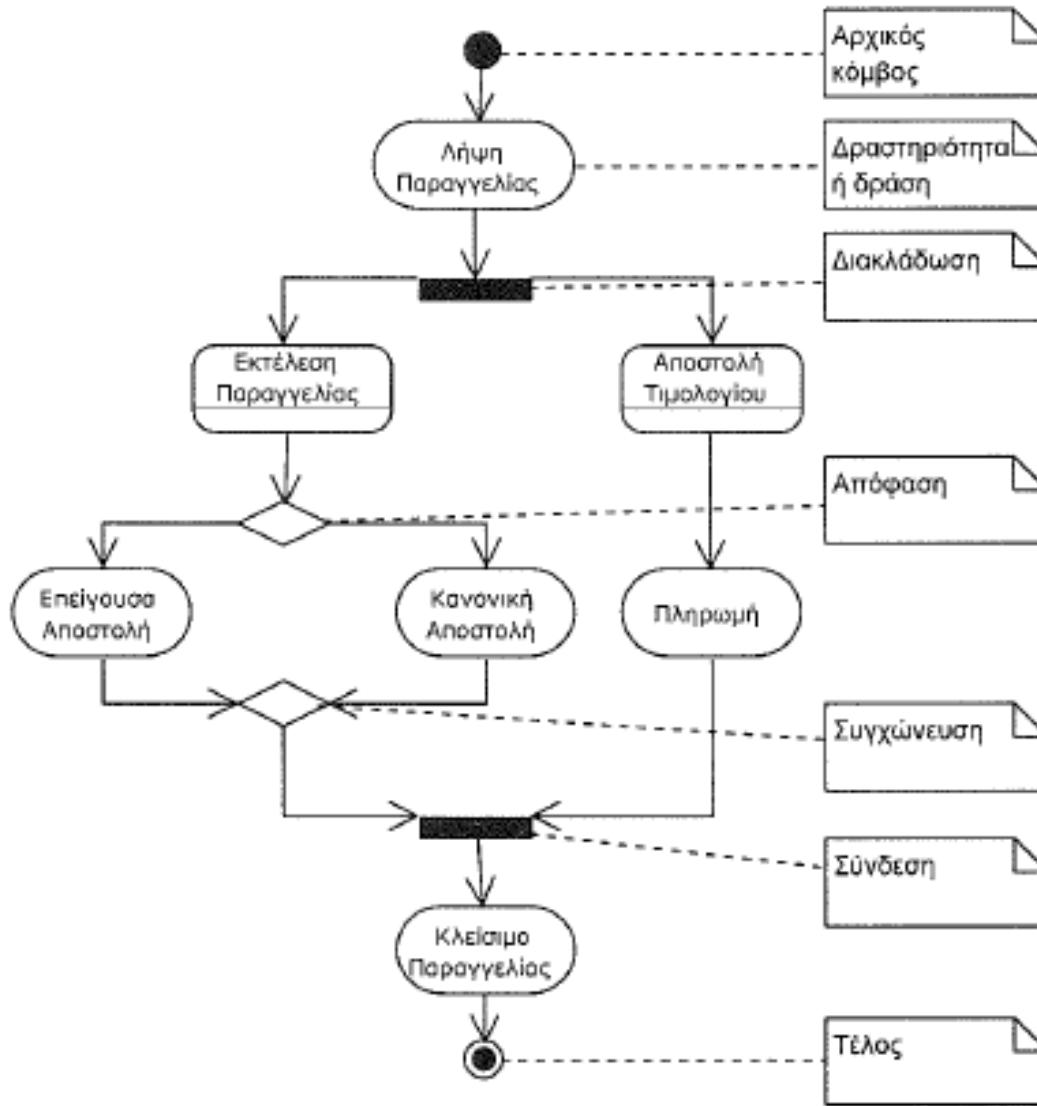
Περίπτωση χρήσης: Διαχειριστής συστήματος



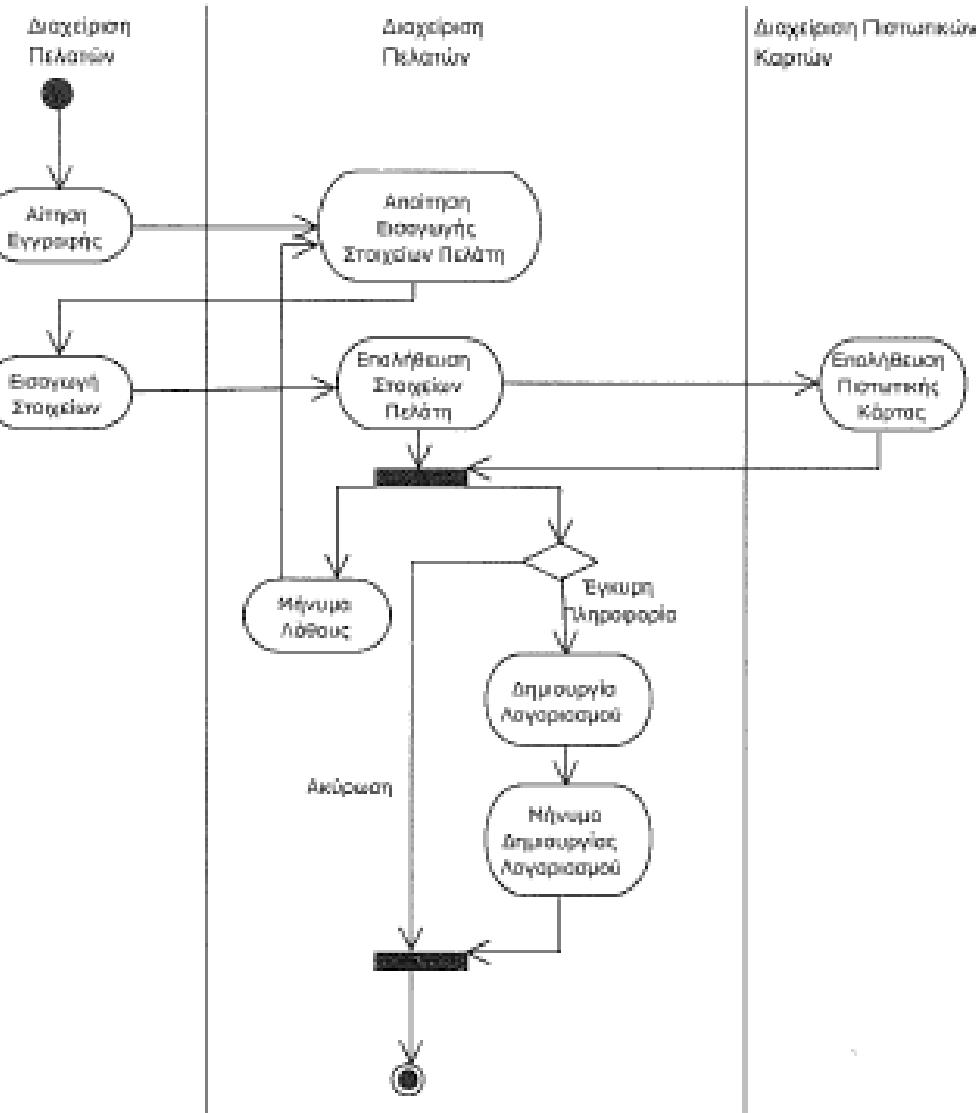
Διάγραμμα περιπτώσεων χρήσης



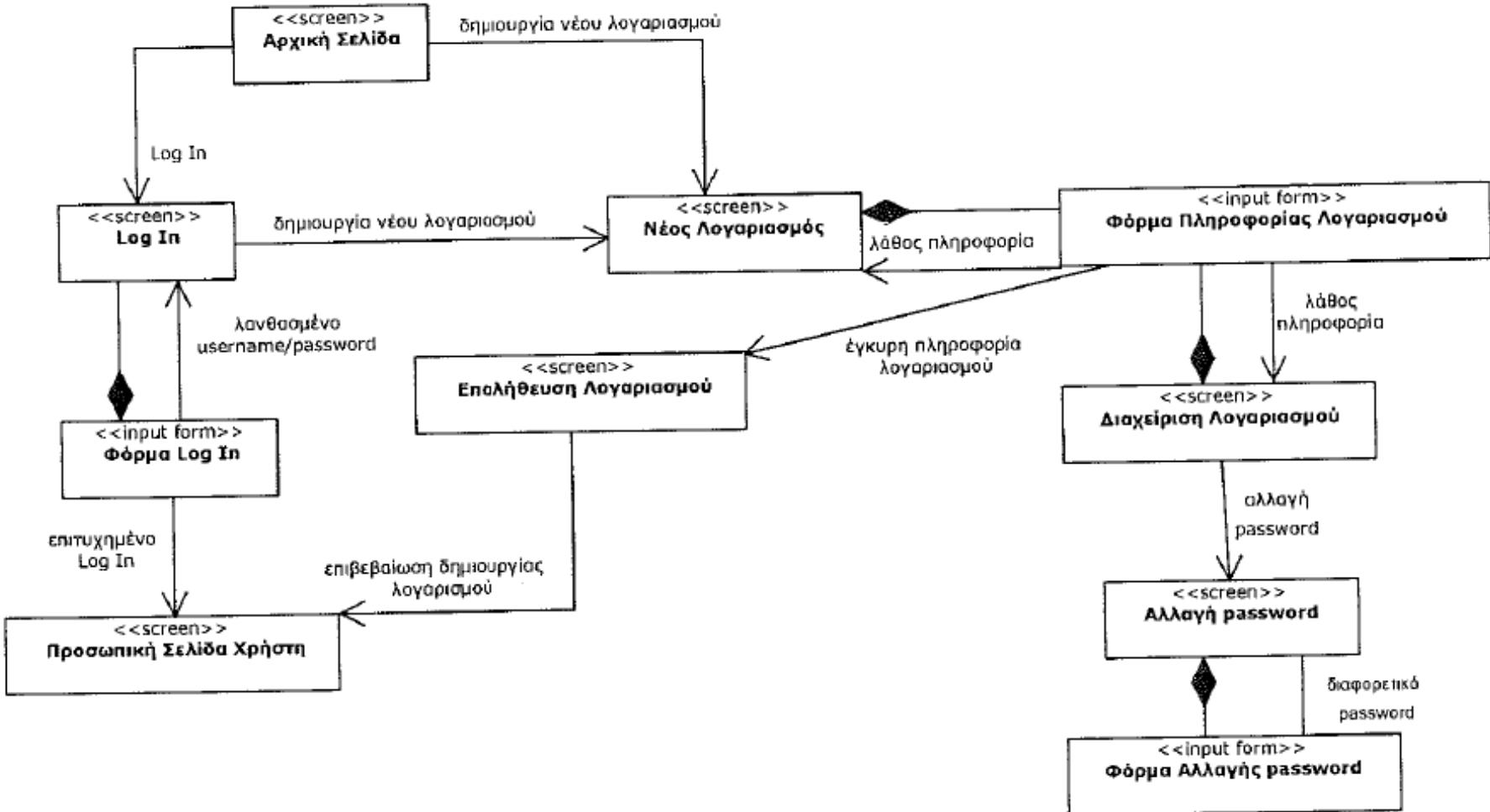
Διάγραμμα Δραστηριοτήτων



Διάγραμμα Δραστηριοτήτων (Δημιουργία λογαριασμού)



Διαχείριση Λογαριασμού πελάτη (Μοντέλο Αλληλεπίδρασης με το σύστημα)

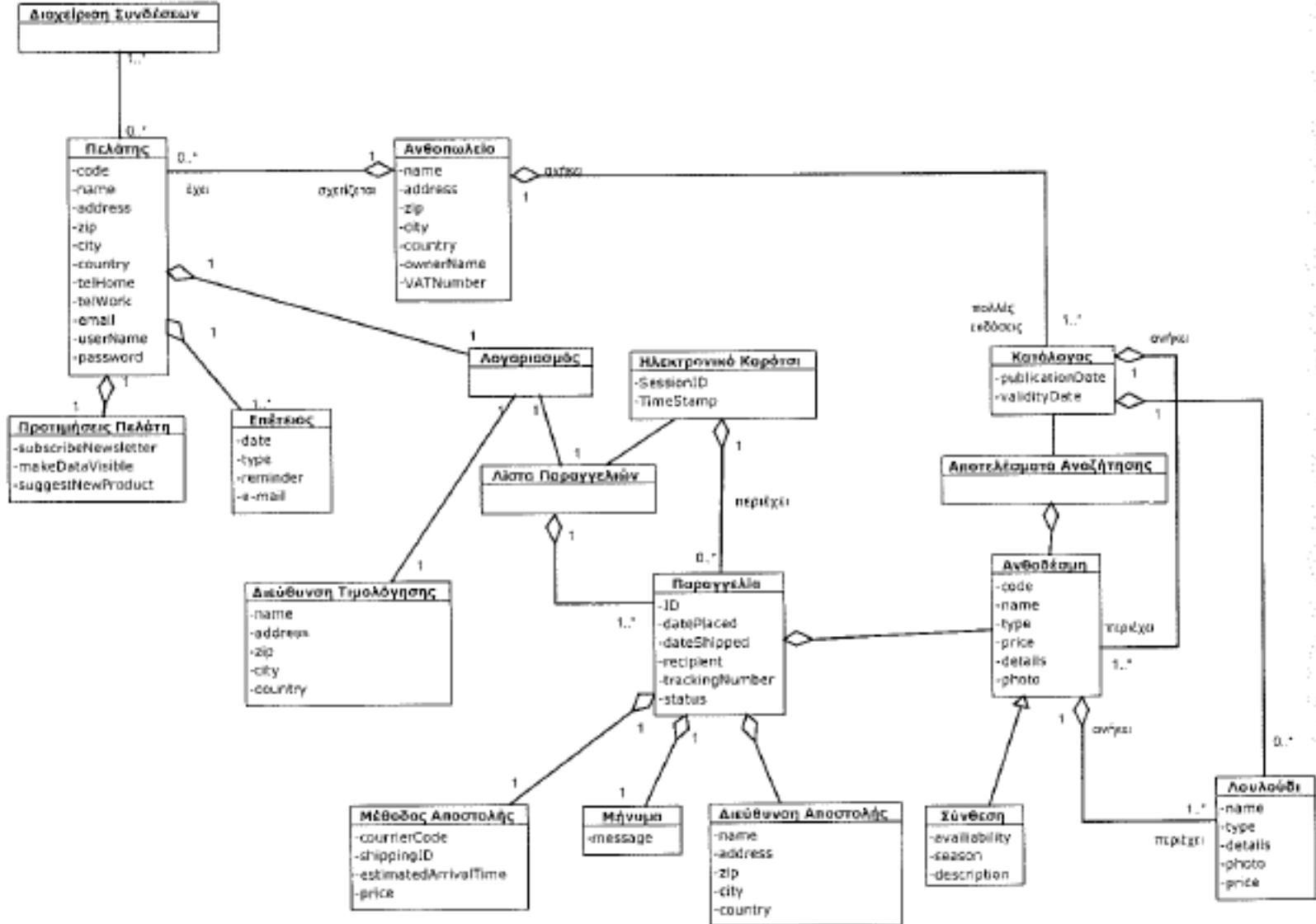


Μοντέλο Ανάλυσης

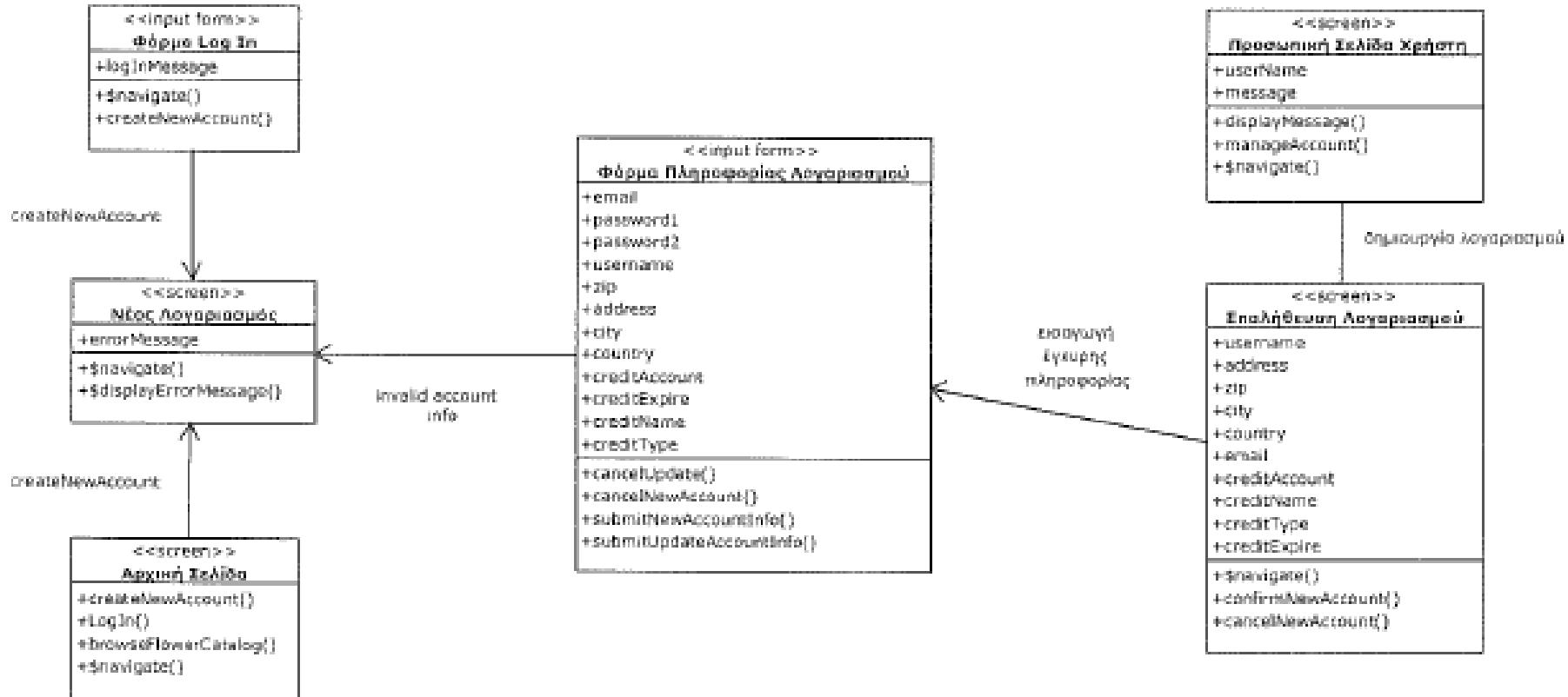
Μοντέλο απαιτήσεων	Μοντέλο ανάλυσης
Περιγράφει το σύστημα χρησιμοποιώντας τη γλώσσα του πελάτη.	Περιγράφει το σύστημα χρησιμοποιώντας τη γλώσσα του τεχνικού.
Παρουσιάζει την εξωτερική εικόνα του συστήματος (με τον τρόπο που κάποιος χρήστης βλέπει το σύστημα).	Παρουσιάζει την εικόνα του συστήματος εσωτερικά.
Είναι δομημένο με βάση τις περιπτώσεις χρήσης.	Είναι δομημένο με βάση τις κλάσεις.
Χρησιμοποιείται για την επικοινωνία μέσα στο έργο και έχει την αξία συμβολαίου μεταξύ της ομάδας ανάπτυξης και του πελάτη του έργου.	Χρησιμοποιείται κυρίως από την ομάδα του έργου για την καλύτερη κατανόηση του συστήματος.
Το μοντέλο μπορεί να περιέχει ανακρίβειες και επικαλύψεις.	Το μοντέλο πρέπει να είναι όσο το δυνατόν πιο σαφές και ακριβές.



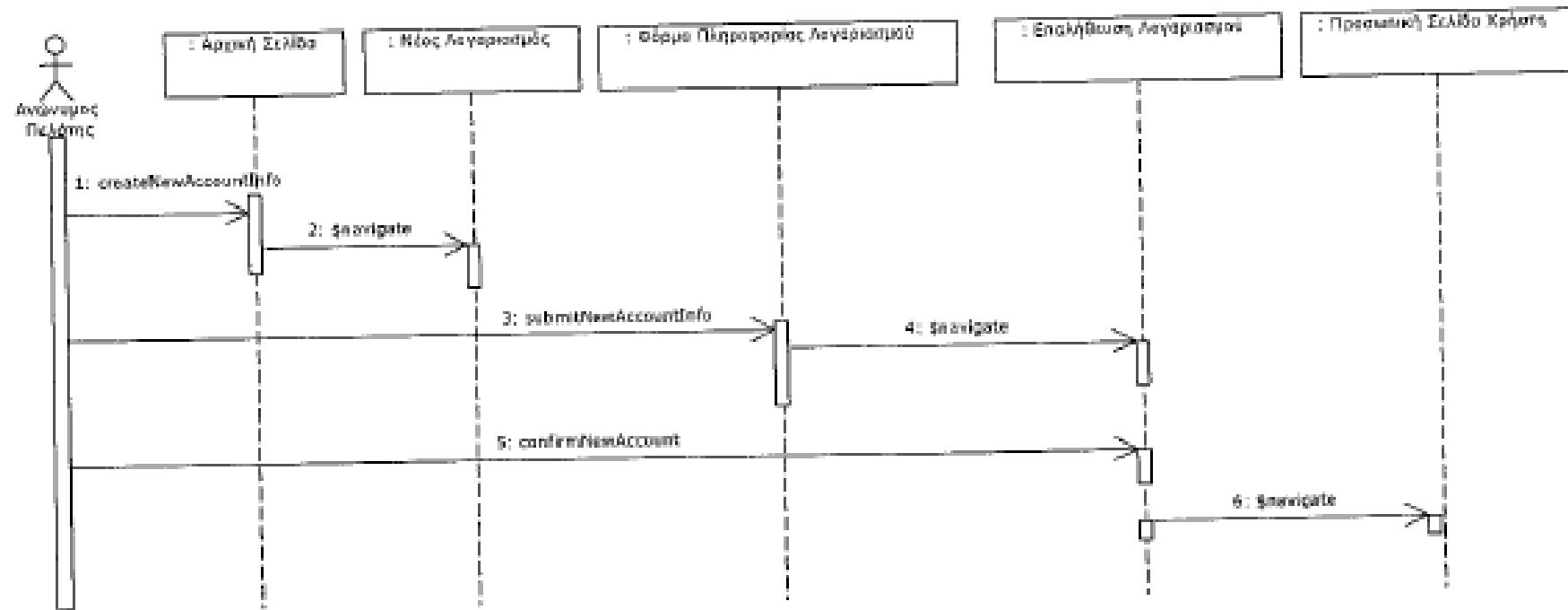
Διαγράμματα κλάσεων με πεδία



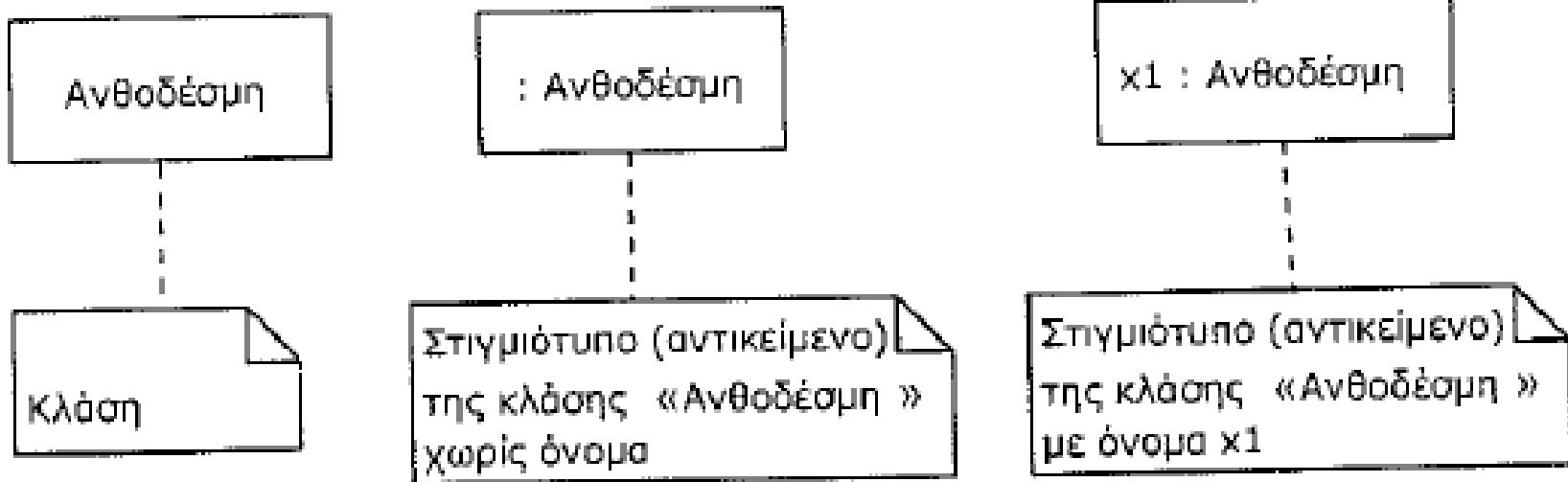
Διαγράμματα κλάσεων με πεδία



Διάγραμμα Ακολουθίας (περίπτωση χρήσης: Διαχείριση λογαριασμού πελάτη)



Συμβολισμός κλάσεις και στιγμιότυπα



Είδη μηνυμάτων Διαγράμματα Ακολουθίας



Γενική μορφή μηνύματος ή ασύγχρονο μήνυμα



Σύγχρονο μήνυμα



Επιστροφή



Αντικείμενο A

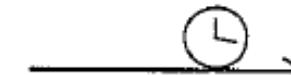
Δημιουργία αντικειμένου



Καταστροφή αντικειμένου

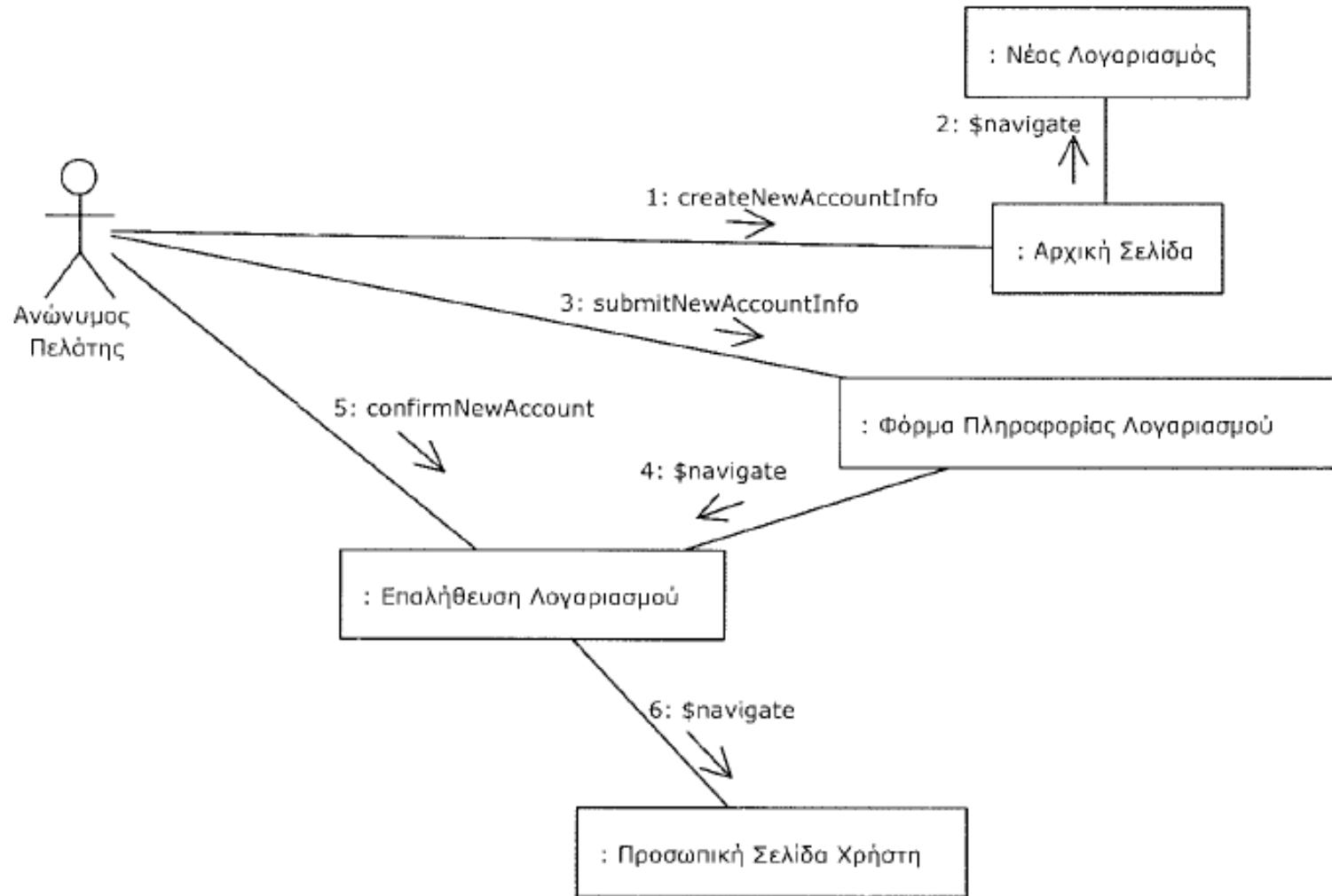


Κλήση που εγκαταλείπεται αν ο δέκτης δεν
είναι έτοιμος



Κλήση με χρόνο (ακυρώνεται αν δεν συμπληρωθεί
στον προκαθορισμένο χρόνο)

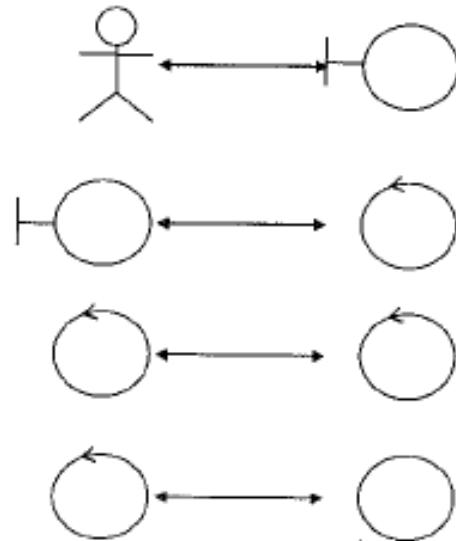
Διάγραμμα Συνεργασίας (περίπτωση χρήσης: Διαχείρισης λογαριασμού πελάτη)



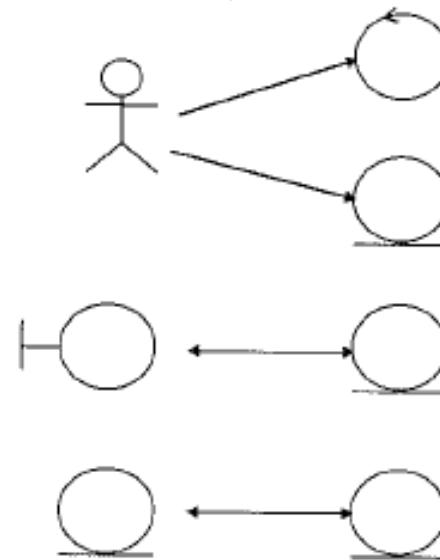
Διάγραμμα Συνεργασίας

- Οι χειριστές επικοινωνούν μόνο με συνοριακές κλάσεις.
- Οι συνοριακές κλάσεις επικοινωνούν μόνο με χειριστές και με κλάσεις ελέγχου.
- Οι κλάσεις ελέγχου επικοινωνούν με άλλες κλάσεις ελέγχου, συνοριακές κλάσεις, και κλάσεις οντοτήτων.
- Οι κλάσεις οντοτήτων επικοινωνούν μόνο με κλάσεις ελέγχου (ελεγκτή οντοτήτων).

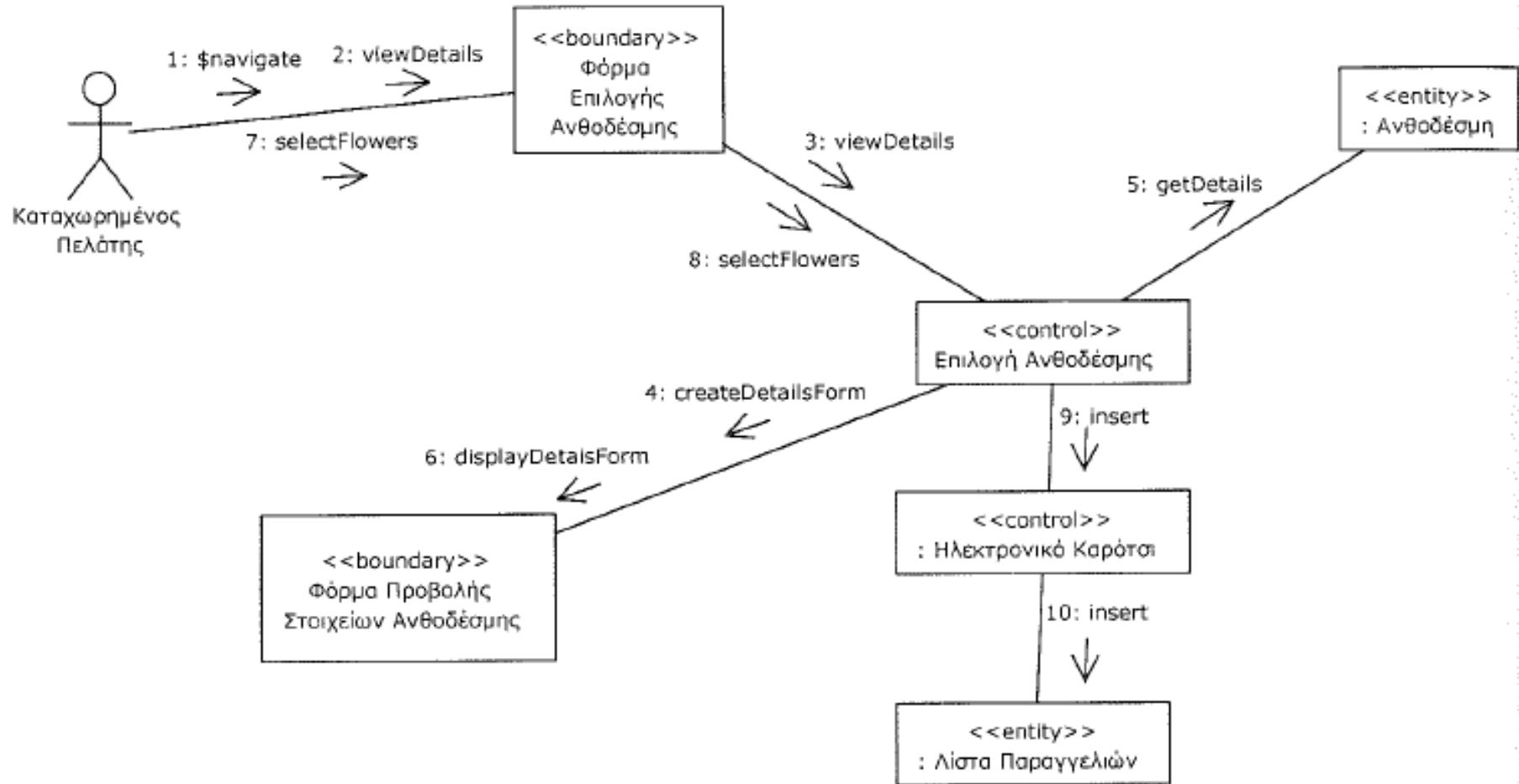
Σωστοί συνδυασμοί



Μη συνιστώμενοι συνδυασμοί



Διάγραμμα Συνεργασίας (περίπτωση χρήσης: Επιλογή Ανθοδέσμης)

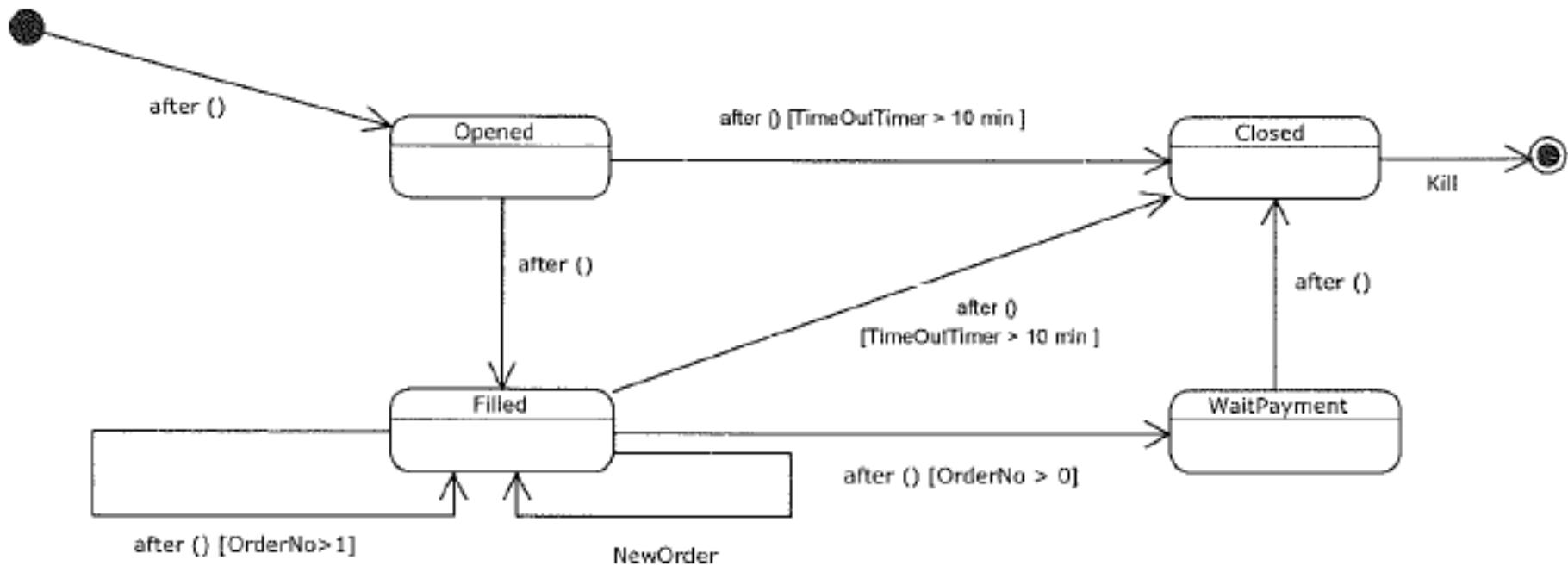


Διάγραμμα Συνεργασίας Vs Διάγραμμα Ακολουθίας

Διάγραμμα	Πλεονεκτήματα	Μειονεκτήματα
Συνεργασίας	Παρουσιάζει με καλύτερο τρόπο τις δομές επανάληψης, τις συνθήκες, την ταυτόχρονη εκτέλεση.	Παρουσιάζει δυσκολία στην κατανόηση της σειράς εκτέλεσης των μηνυμάτων. Χρησιμοποιεί πολύπλοκο συμβολισμό.
Ακολουθίας	Δείχνει καλύτερα τη χρονική ακολουθία των μηνυμάτων. Χρησιμοποιεί ευκολότερο συμβολισμό.	Όταν το διάγραμμα είναι μεγάλο και συμμετέχουν πολλά αντικείμενα επεκτείνεται προς τα δεξιά, πράγμα που δημιουργεί δυσκολίες χώρου.



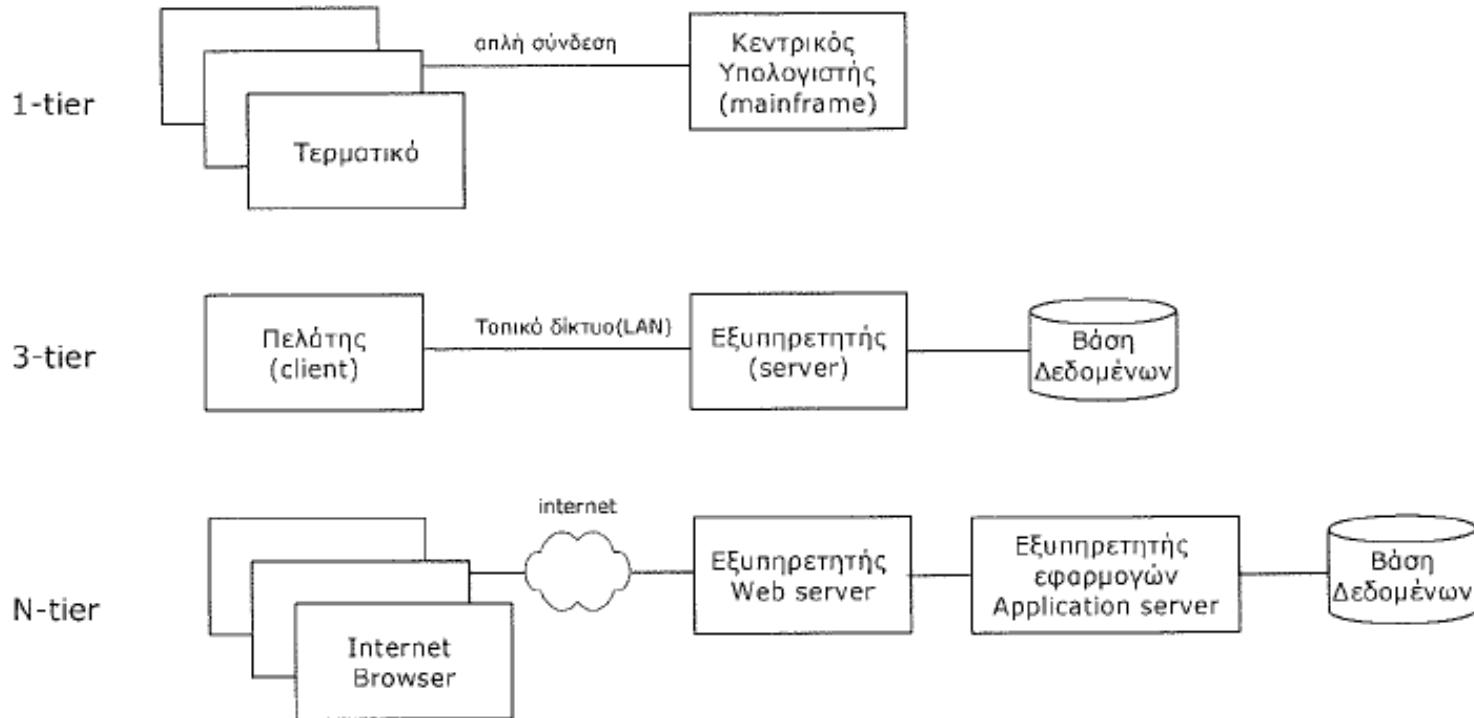
Διάγραμμα Κατάστασης (ΠΧ Ηλεκτρονικό καρότσι)



Μοντέλο Σχεδίασης

Μοντέλο ανάλυσης	Μοντέλο σχεδίασης
Παράγει το ιδεατό μοντέλο (conceptual model) όπου δεν περιγράφουμε τις λεπτομέρειες υλοποίησης.	Παράγει το φυσικό μοντέλο (physical model) που είναι λεπτομερές, αφού αυτό καθοδηγεί την ανάπτυξη.
Λόγω της γενικότητάς του, μπορεί να υλοποιηθεί από πολλά διαφορετικά και εναλλακτικά μοντέλα σχεδίασης.	Είναι ένα και συγκεκριμένο. Όλες οι εναλλακτικές λύσεις έχουν αξιολογηθεί και έχει επιλεγεί η βέλτιστη.
Είναι γενικό και βασίζεται στα τρία στερεότυπα των κλάσεων (συνοριακές, ελέγχου, και οντοτήτων).	Βασίζεται στα στερεότυπα της υλοποίησης, τα οποία εξαρτώνται από τις αποφάσεις που έχουμε λάβει.
Δεν είναι ορισμένο πολύ αυστηρά.	Είναι ορισμένο αυστηρά.
Παρουσιάζονται λίγα αρχιτεκτονικά επίπεδα.	Παρουσιάζονται πολλά αρχιτεκτονικά επίπεδα καθώς και η συγκεκριμένη αρχιτεκτονική υλοποίησης που έχει επιλεγεί.
Περιγράφει την αρχιτεκτονική του συστήματος.	Ορίζει την αρχιτεκτονική του συστήματος.
Η συντήρησή του είναι σκόπιμη σε όλο τον κύκλο ζωής ανάπτυξης του συστήματος.	Η συντήρηση, η ενημέρωση, και ο συγχρονισμός του είναι απαραίτητα σε όλο τον κύκλο ζωής ανάπτυξης του συστήματος.

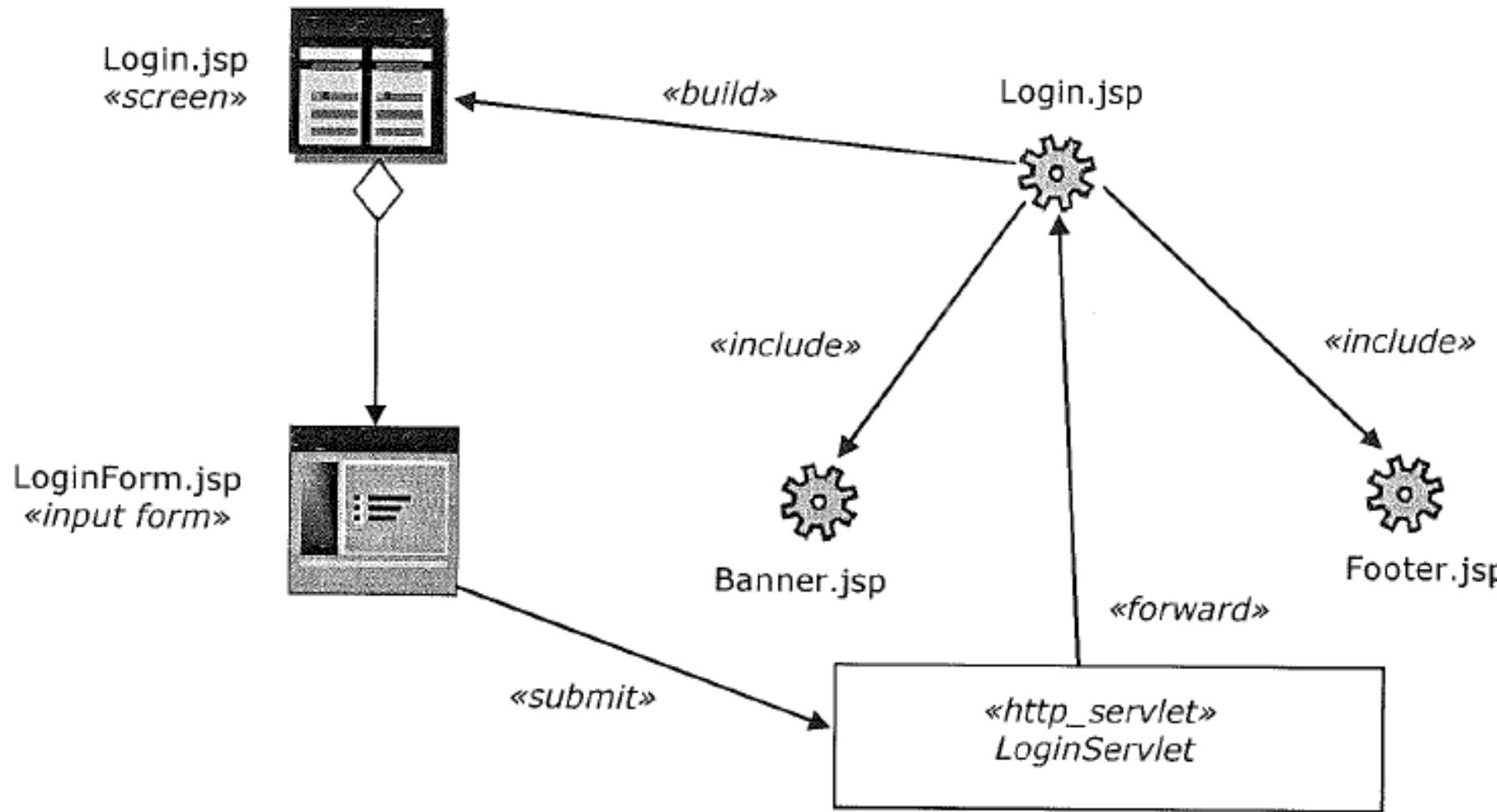
Αρχιτεκτονική πολλών επιπέδων



Συστατικό J2EE	Λογικό επίπεδο	Φυσικό επίπεδο
HTML σελίδες, μικροεφαρμογές (applets), πελάτες Java	Πελάτη	Μηχανή πελάτη
Java Server Pages Java Servlets	Web	J2EE Server
Enterprise JavaBeans	Επιχειρηματικό	J2EE Server
Database	EIS	Database Server



Σχεδιασμός της ΠΧ login



Ανάλυση Πληροφοριακών Συστημάτων

Εαρινό Εξάμηνο 2020-2021

Lec09 – 25/05/2021

Διδάσκων: Γεώργιος Χρ. Μακρής

Διαγράμματα Καταστάσεων

Διάγραμμα Καταστάσεων

- 'Ένα διάγραμμα καταστάσεων (state diagram) απεικονίζει τη δυναμική συμπεριφορά των αντικειμένων μιας κλάσης και τον τρόπο που μεταβάλλεται η κατάστασή τους ως ανάδραση σε συμβάντα.
- Μπορείτε να χρησιμοποιήσετε τα διαγράμματα κατάστασης για να μοντελοποιήσετε τη συμπεριφορά και άλλων στοιχείων της UML, αλλά τυπικά χρησιμοποιούνται για τα στιγμιότυπα μιας κλάσης.
- Συνήθως ΔΕΝ σχεδιάζουμε διαγράμματα κατάστασης για όλες τις κλάσεις σε ένα σύστημα, παρά μόνο για εκείνες που παρουσιάζουν μία έντονα δυναμική συμπεριφορά. Το διάγραμμα κατάστασης μας βοηθάει να αντιληφθούμε αυτήν τη συμπεριφορά.

Διάγραμμα Καταστάσεων

- Τα διαγράμματα καταστάσεων (State diagrams) βοηθούν:
 - στην ανάπτυξη ενός συστήματος
 - στην κατανόηση πολύπλοκων χαρακτηριστικών ή ροών εργασίας εξειδικευμένων περιοχών του συστήματος
- Τα διαγράμματα αυτά περιγράφουν όλες τις πιθανές καταστάσεις στις οποίες μπορεί να βρεθεί ένα αντικείμενο καθώς πραγματοποιούνται διάφορα γεγονότα
- Κάθε διάγραμμα συνήθως αναπαριστά τα αντικείμενα μιας κλάσης και παρακολουθεί τα διαφορετικά στάδια (καταστάσεις) των αντικειμένων καθ' όλη τη διάρκεια της ζωής τους μέσα στο σύστημα



Διάγραμμα Καταστάσεων

- Μία κατάσταση είναι μια συνθήκη ή μια περίπτωση στην οποία βρίσκεται ένα αντικείμενο κατά τη διάρκεια ζωής του
- Το αντικείμενο σε αυτήν την κατάσταση:
 - ικανοποιεί κάποια συνθήκη,
 - εκτελεί κάποια δραστηριότητα ή
 - περιμένει να πραγματοποιηθεί κάποιο γεγονός
- 'Ένα αντικείμενο παραμένει σε μια κατάσταση για κάποιο χρονικό διάστημα



Χαρακτηριστικά Κατάστασης

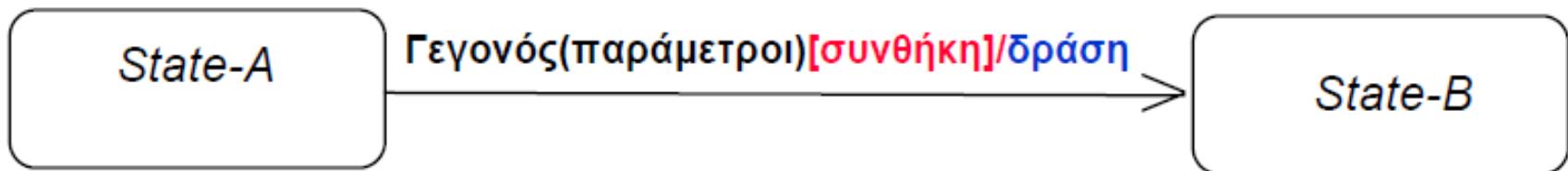
- Μια **κατάσταση** αποτελείται από τα παρακάτω μέρη:
 - **Όνομα:** Το όνομα της κατάστασης
 - **Ενέργειες Εισόδου/Εξόδου:** Ενέργειες που εκτελούνται κατά την είσοδο ή κατά την έξοδο από την κατάσταση
 - **Δραστηριότητες:** Ενέργειες που πραγματοποιούνται όσο ένα αντικείμενο είναι σε μια κατάσταση μεταξύ των ενεργειών εισόδου και εξόδου
 - **Υποκαταστάσεις:**
Οι υποκαταστάσεις μιας κατάστασης

Όνομα κατάστασης
Πεδία κατάστασης
entry / δράση εισόδου do / δραστηριότητα-A event ₁ (a:T)[exp]: action, ⋮
event _n (a:T)[exp]: action _n exit / δράση εξόδου

Χαρακτηριστικά μετάβασης

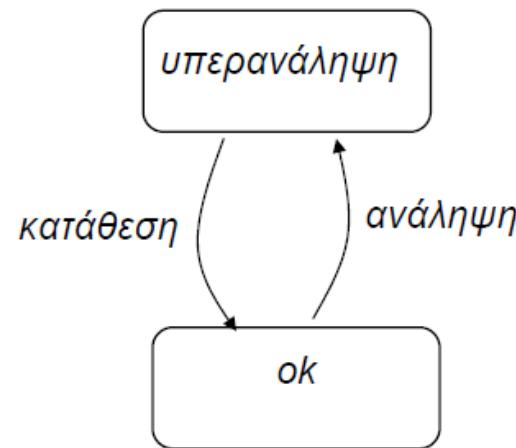
- Μια **μετάβαση** είναι μια σχέση μεταξύ δύο καταστάσεων, η οποία δείχνει ότι ένα αντικείμενο σε μια κατάσταση, αφού εκτελέσει συγκεκριμένες ενέργειες, θα μεταβεί στην επόμενη κατάσταση, όταν πραγματοποιηθεί ένα συγκεκριμένο γεγονός και ικανοποιηθούν συγκεκριμένες συνθήκες. Μέρη:

1. **Κατάσταση προέλευσης**
2. **Γεγονός ενεργοποίησης**: ενεργοποιεί τη μετάβαση από τη μια κατάσταση στην άλλη
3. **Συνθήκη ελέγχου**: μια συνθήκη, η οποία ελέγχεται για να ενεργοποιηθεί το γεγονός που οδηγεί στη μετάβαση
4. **Επίδραση**: Μια ενέργεια που μπορεί να πραγματοποιηθεί στο αντικείμενο του διαγράμματος κατάστασης ή σε κάποιο άλλο αντικείμενο που σχετίζεται με το πρώτο
5. **Κατάσταση στόχου**



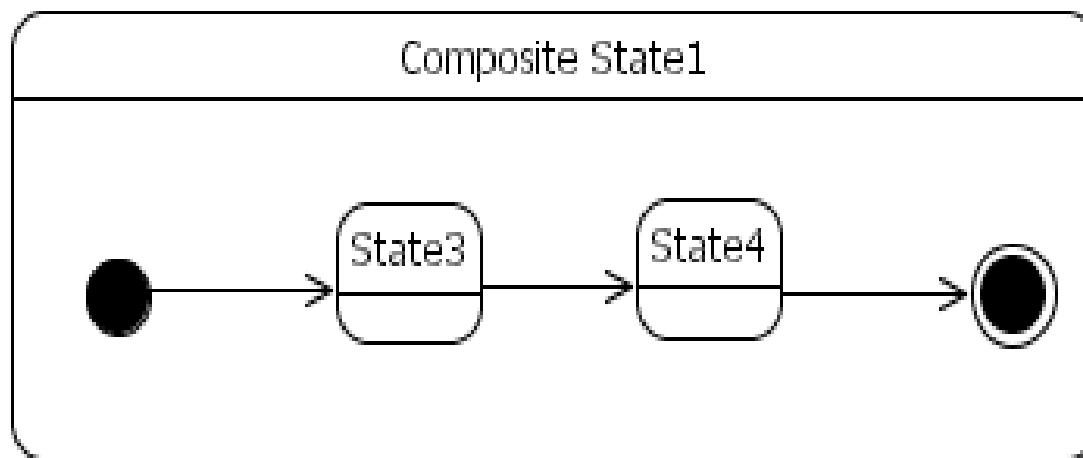
Αλλαγές καταστάσεων

- Η αλλαγή κατάστασης (statetransition) συμβαίνει ως αποτέλεσμα ενός γεγονότος (event)
- Αλλαγή κατάστασης σημαίνει ότι άλλαξε η κατάσταση του αντικειμένου
- Τα γεγονότα μπορεί να είναι: Μηνύματα (messages) ή σήματα (signals)
- Ένα γεγονός δε σηματοδοτεί υποχρεωτική αλλαγή κατάστασης. Το αντικείμενο μπορεί να παραμείνει στην ίδια κατάσταση



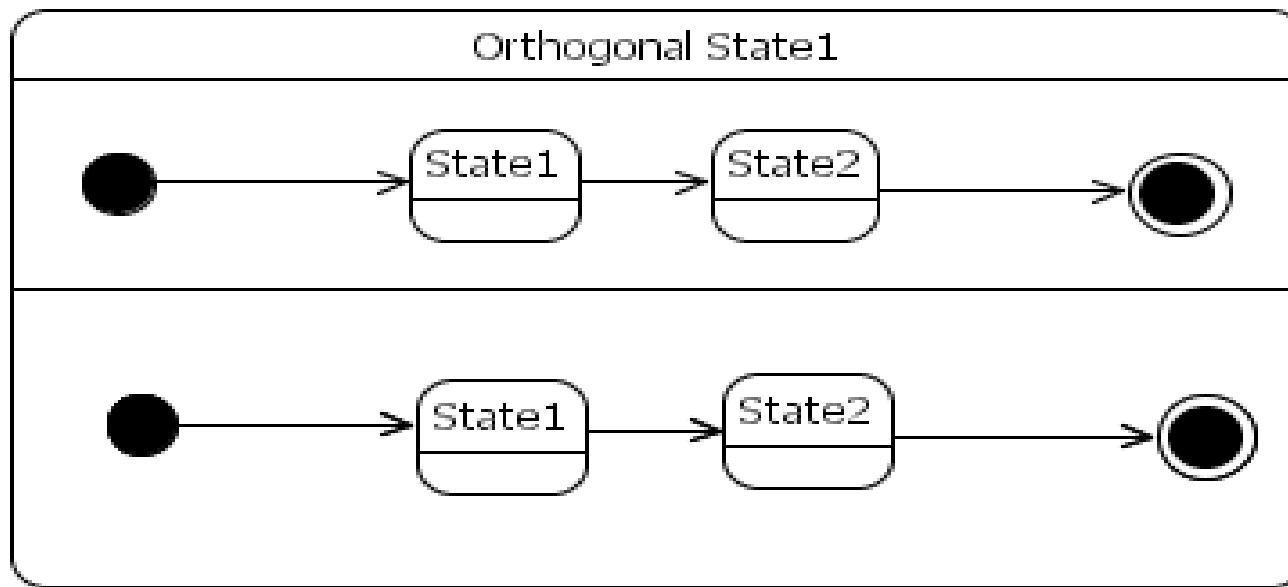
Σύνθετη κατάσταση

Μια **σύνθετη κατάσταση** χρησιμοποιείται όταν θέλουμε να μοντελοποιήσουμε διάφορες καταστάσεις (υποκαταστάσεις) που λαμβάνει ένα αντικείμενο όταν αυτό ήδη βρίσκεται σε κάποια κατάσταση



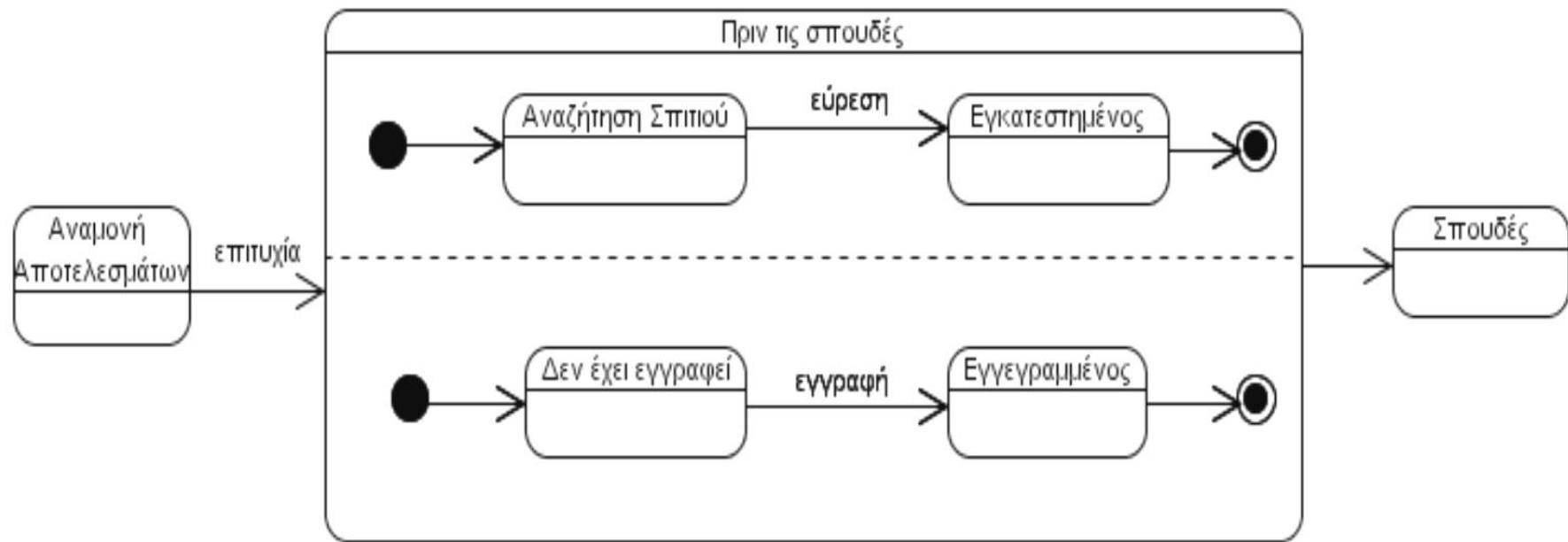
Ορθογώνια κατάσταση

Μια ορθογώνια κατάσταση χρησιμοποιείται όταν θέλουμε να μοντελοποιήσουμε την πιθανότητα να βρεθεί ένα αντικείμενο σε κάποια από διάφορες άλλες καταστάσεις (υποκαταστάσεις) μετά από την μετάβαση από μία κατάσταση στην οποία ήδη βρίσκεται



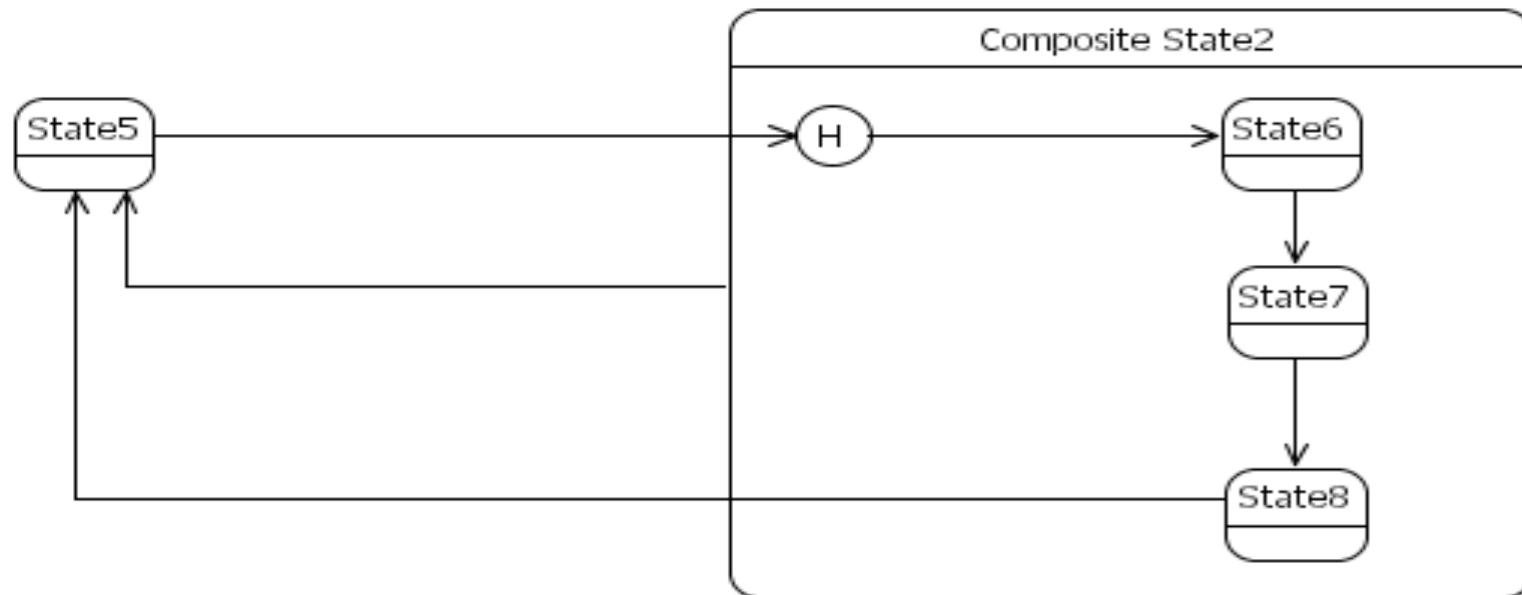
Ταυτόχρονες σύνθετες καταστάσεις

- Πολλές φορές θέλουμε να απεικονίσουμε ταυτόχρονα διαγράμματα κατάστασης στα οποία ένα αντικείμενο βρίσκεται ταυτόχρονα σε δύο ή και περισσότερες καταστάσεις.
- Για το σκοπό αυτόν μπορούμε να χωρίσουμε μία κατάσταση σε δύο ή και περισσότερα μέρη με μια διακεκομένη γραμμή και να δείξουμε τα ταυτόχρονα διαγράμματα κατάστασης στα διαφορετικά αυτά μέρη.



Ιστορική κατάσταση

Μια ιστορική κατάσταση χρησιμοποιείται σε συνδυασμό με μια σύνθετη κατάσταση για να κρατάει την τελευταία κατάσταση που ήταν ενεργή μέσα στη σύνθετη κατάσταση, σε περίπτωση που κάποιο γεγονός διακόψει την εκτέλεση της σύνθετης κατάστασης



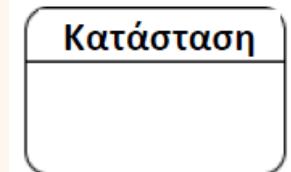
Στοιχεία Διαγράμματος Καταστάσεων

Στοιχείο Διαγράμματος Καταστάσεων

Αρχική Κατάσταση: αναπαριστά το αρχικό σημείο του διαγράμματος. Επίσης, αυτή η κατάσταση συχνά καλείται ψευδο-κατάσταση, γιατί ουσιαστικά δεν είναι μία πραγματική κατάσταση



Κατάσταση: Αναπαριστά την κατάσταση ενός αντικειμένου σε μια χρονική στιγμή



Μετάβαση: 'Ένα βέλος που αναπαριστά τη μετάβαση του αντικειμένου από τη μια κατάσταση σε μια άλλη. Το γεγονός και η ενέργεια που προκαλεί τη μετάβαση της κατάστασης του αντικειμένου αναγράφονται πάνω από το βέλος'



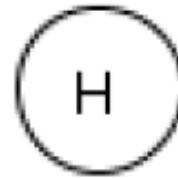
Στοιχεία Διαγράμματος Καταστάσεων

Στοιχείο Διαγράμματος Καταστάσεων

Ιστορικές Καταστάσεις: Κάποιες φορές μπορεί να απαιτείται για ένα αντικείμενο να μεταβεί σε μια κατάσταση αναμονής και με την ενεργοποίηση ενός συγκεκριμένου γεγονότος να μεταβεί στην κατάσταση που βρίσκονταν πριν από την κατάσταση αναμονής, δηλαδή την τελευταία ενεργή κατάστασή του.

Γεγονός και Ενέργεια: Η ενεργοποίηση μιας μετάβασης ονομάζεται Γεγονός ή Ενέργεια. Κάθε μετάβαση πραγματοποιείται εξαιτίας της ενεργοποίησης ενός γεγονότος ή μιας ενέργειας.

Τελική Κατάσταση: Το τέλος ενός διαγράμματος κατάστασης αναπαρίσταται με το σύμβολο της τελικής κατάστασης. Η τελική κατάσταση αποτελεί επίσης μια ψευδο-κατάσταση.

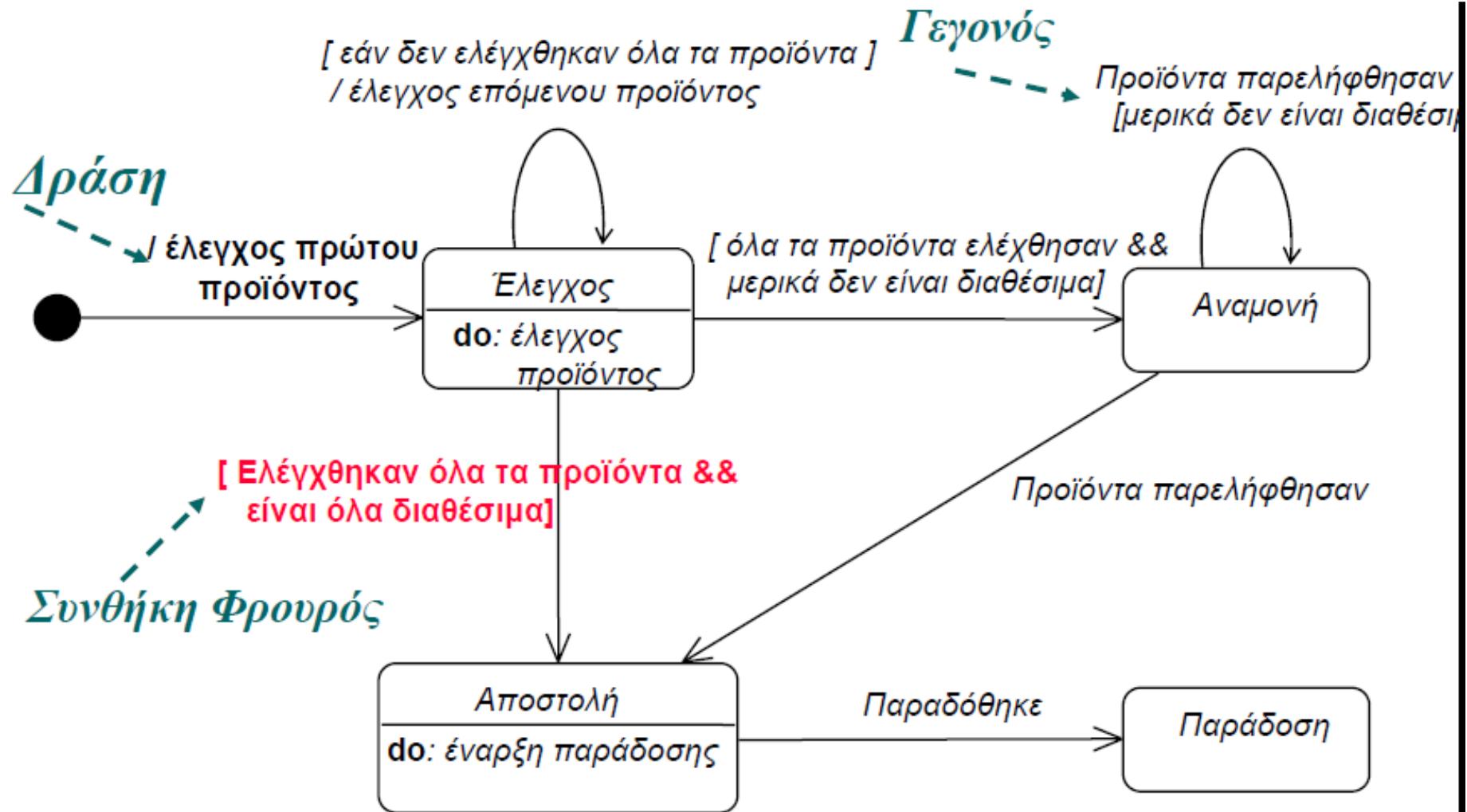


Διάγραμμα Καταστάσεων

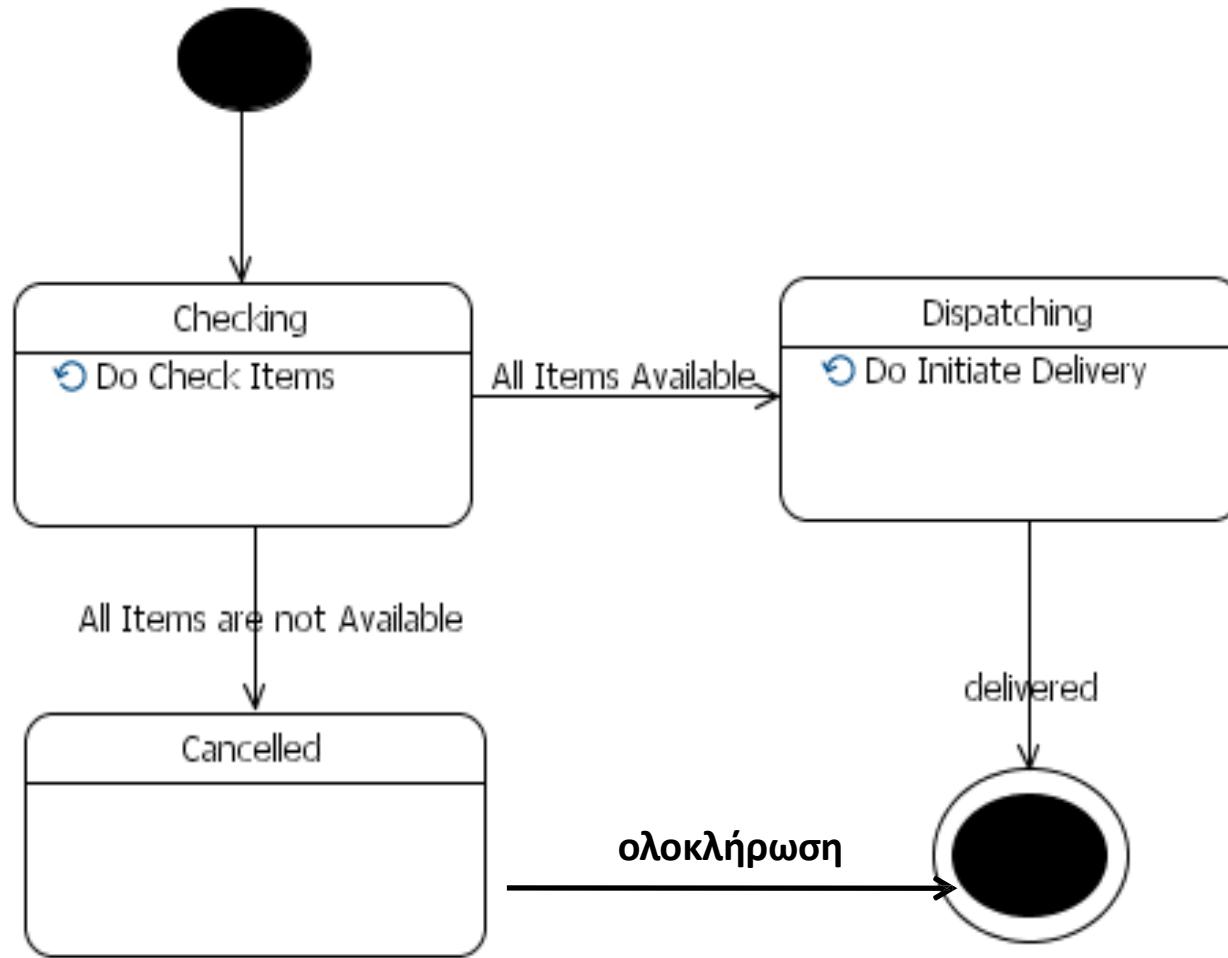
- 'Όλα τα διαγράμματα κατάστασης αρχίζουν
 - Με μια αρχική κατάσταση εκκίνησης του αντικειμένου που μοντελοποιείται
 - Αυτή είναι η κατάσταση του αντικειμένου όταν αυτό δημιουργείται
 - Μετά την αρχική κατάσταση του αντικειμένου το αντικείμενο αλλάζει καταστάσεις
 - Οι συνθήκες που βασίζονται στις δραστηριότητες μιας κατάστασης του αντικειμένου μπορούν να επηρεάσουν την επόμενη κατάσταση του αντικειμένου



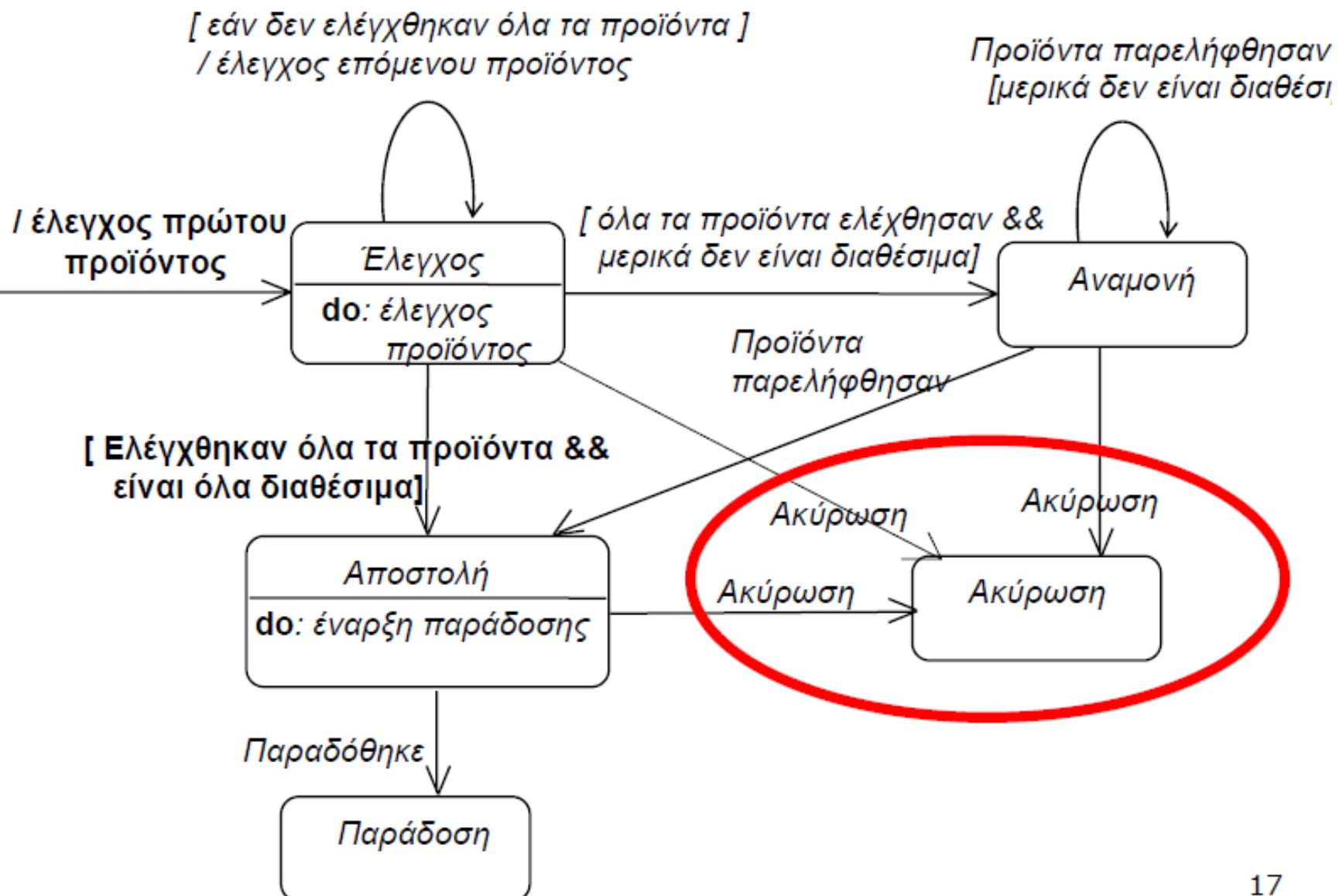
Παράδειγμα: Παραγγελία Προϊόντων



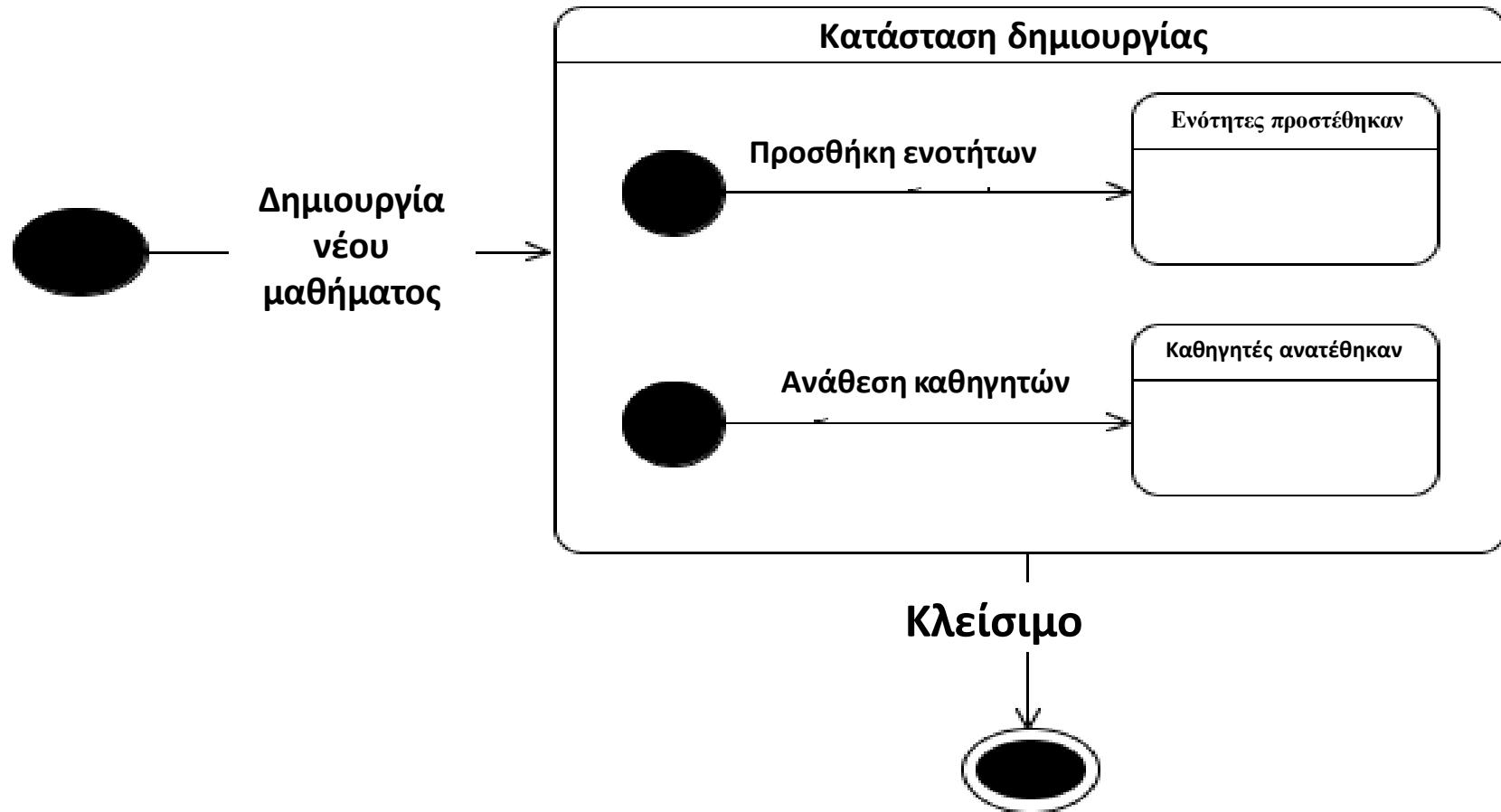
Παράδειγμα: Εκτέλεση Παραγγελίας



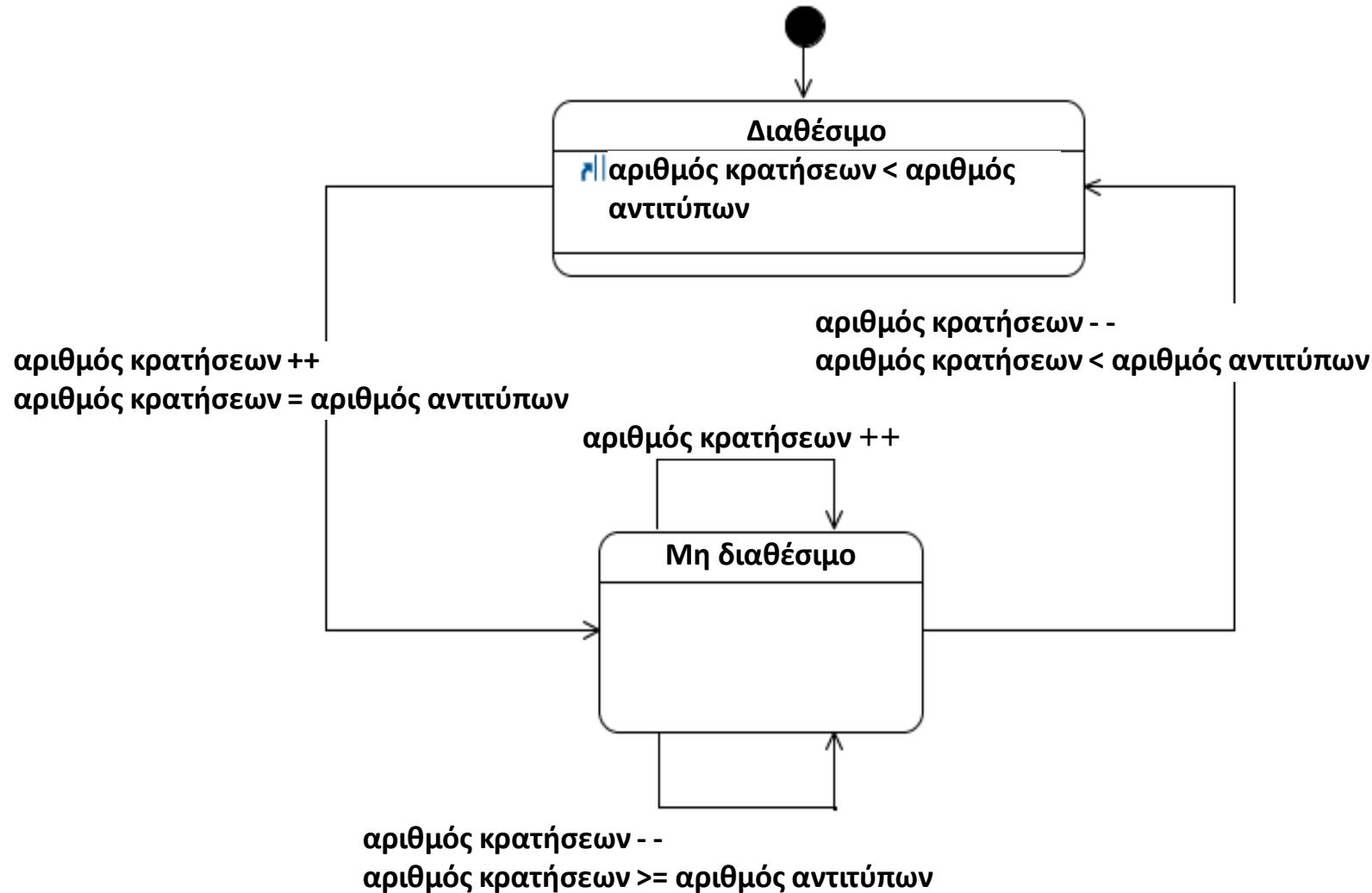
Παράδειγμα: Ακύρωση παραγγελίας



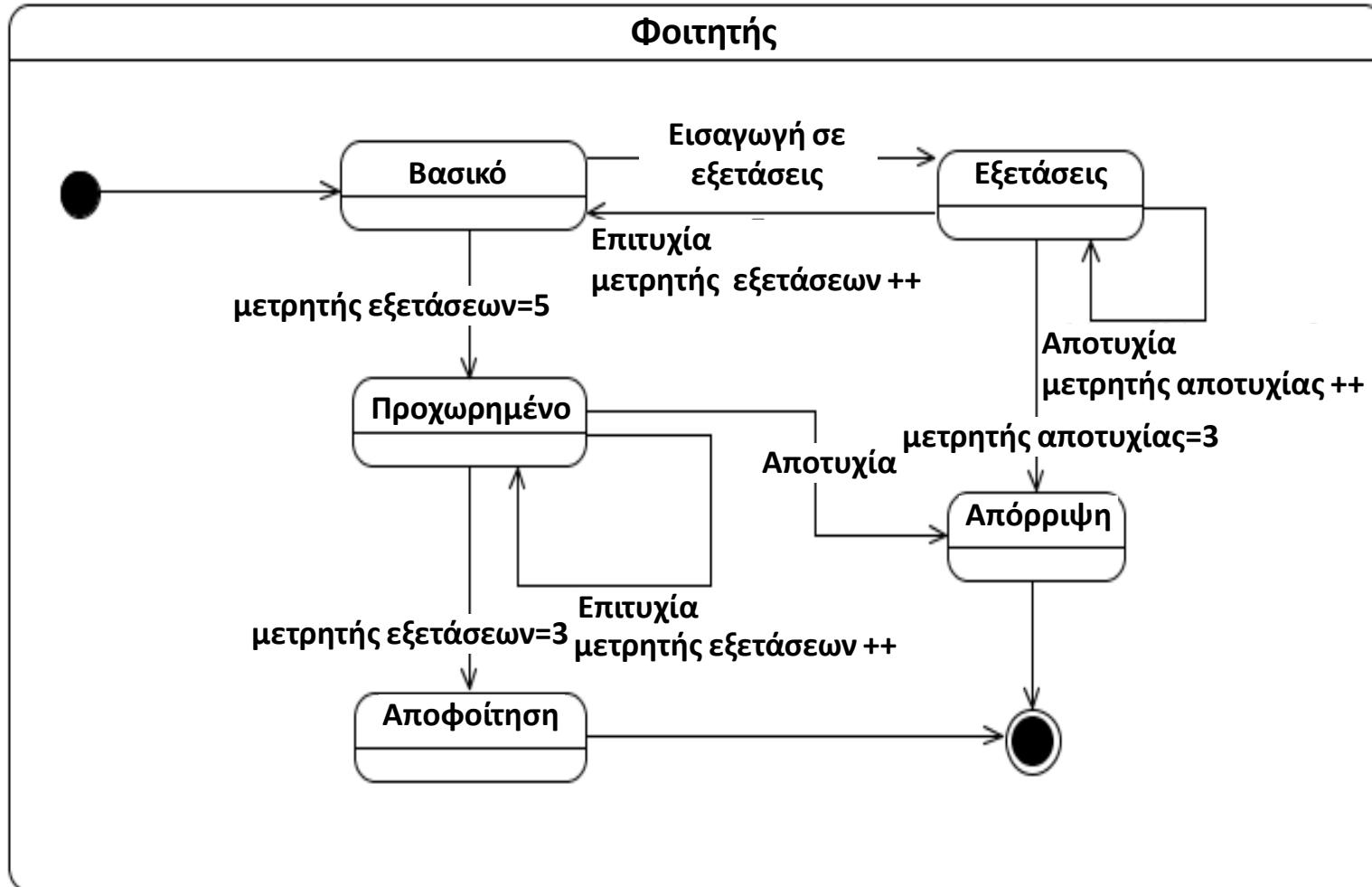
Παράδειγμα: Προσθήκη Μαθήματος



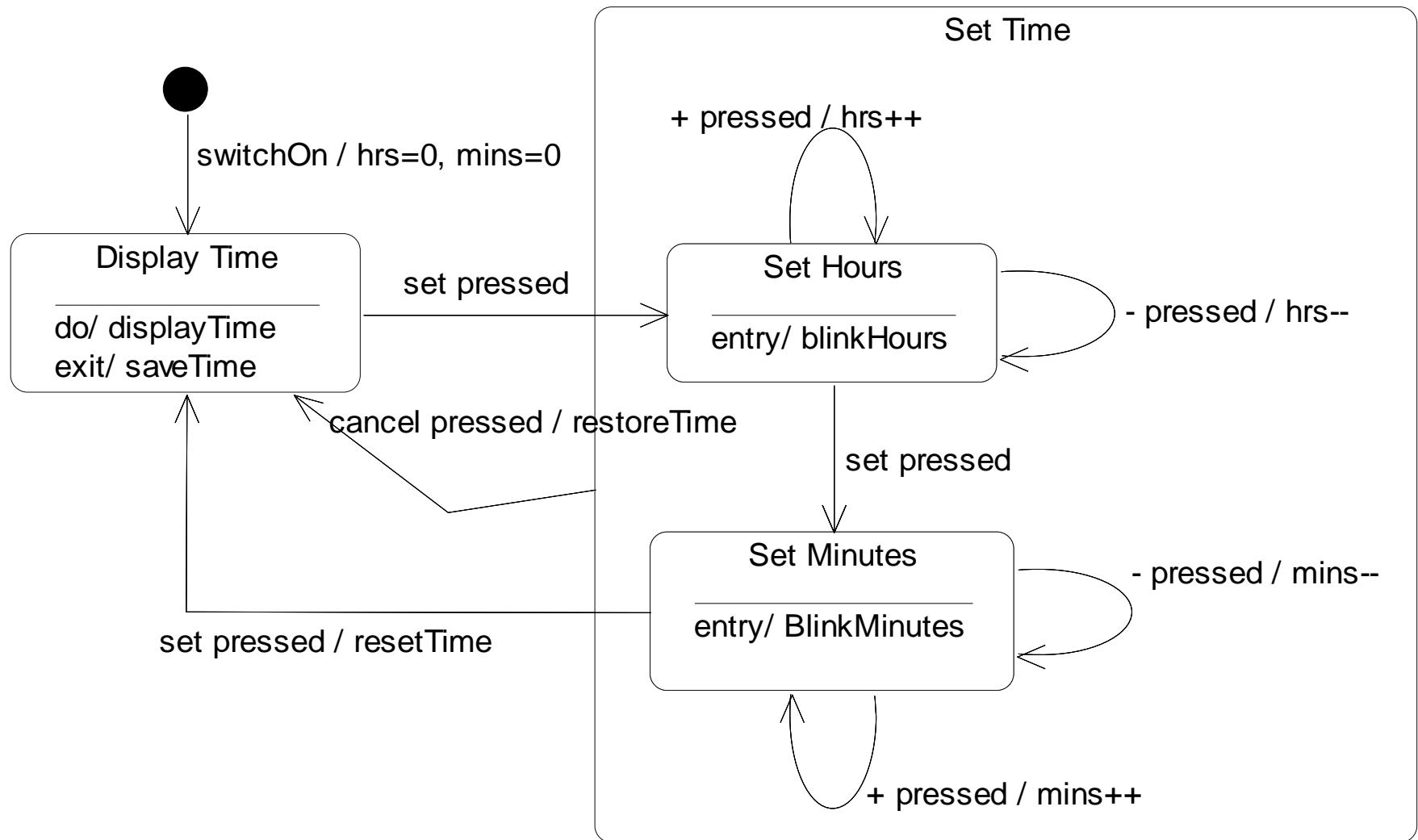
Παράδειγμα: Βιβλιοθήκη ΑΕΙ



Παράδειγμα: Φοίτηση ΑΕΙ



Παράδειγμα διαγρ. καταστάσεων



Διαγράμματα Δραστηριοτήτων

Διάγραμμα Δραστηριοτήτων

- Τα διαγράμματα δραστηριοτήτων αναπαριστούν τη ροή εργασίας ενός συστήματος
- Το *διάγραμμα δραστηριοτήτων* (activity diagram) είναι μια παραλλαγή του διαγράμματος καταστάσεων στο οποίο οι κόμβοι αναπαριστούν καταστάσεις ενεργειών και οι μεταβάσεις λαμβάνουν χώρα με την ολοκλήρωση αυτών των ενεργειών και όχι ως συνέπεια ενός συμβάντος όπως συμβαίνει με το διάγραμμα καταστάσεων.
- Τυπικά χρησιμοποιούμε τα διαγράμματα δραστηριοτήτων για την περιγραφή υπολογισμών ή *ροών εργασιών* (workflows).
- Η επισήμανση για το ποιο τμήμα ή αντικείμενο είναι υπεύθυνο για την εκτέλεση κάποιων ενεργειών σε μια δραστηριότητα γίνεται με τη χρήση διαδρόμων (swimlanes). Επίσης, είναι δυνατή η απεικόνιση της δημιουργίας και των ροών αντικειμένων μεταξύ των διαφόρων ενεργειών.



Διάγραμμα Δραστηριοτήτων

Πως και πότε χρησιμοποιούνται τα διαγράμματα δραστηριοτήτων;

- Τα διαγράμματα δραστηριοτήτων πρέπει να χρησιμοποιούνται σε συνδυασμό με άλλες τεχνικές μοντελοποίησης, όπως είναι τα διαγράμματα καταστάσεων
- Ο κυριότερος λόγος χρήσης των διαγραμμάτων δραστηριοτήτων είναι για τη μοντελοποίηση της ροής εργασιών του συστήματος που σχεδιάζεται

Τα διαγράμματα δραστηριοτήτων είναι επίσης χρήσιμα για τα ακόλουθα:

- Για την περιγραφή μιας λειτουργίας του συστήματος που περιγράφεται από μια περίπτωσης χρήσης
- Για την περιγραφή ενός πολύπλοκου αλγορίθμου ή μιας πολύπλοκης ανθρώπινης εργασίας
- Για την περιγραφή της λογικής μιας λειτουργίας



Διάγραμμα Δραστηριοτήτων

Τα διαγράμματα δραστηριοτήτων δεν αντικαθιστούν τα διαγράμματα κατάστασης

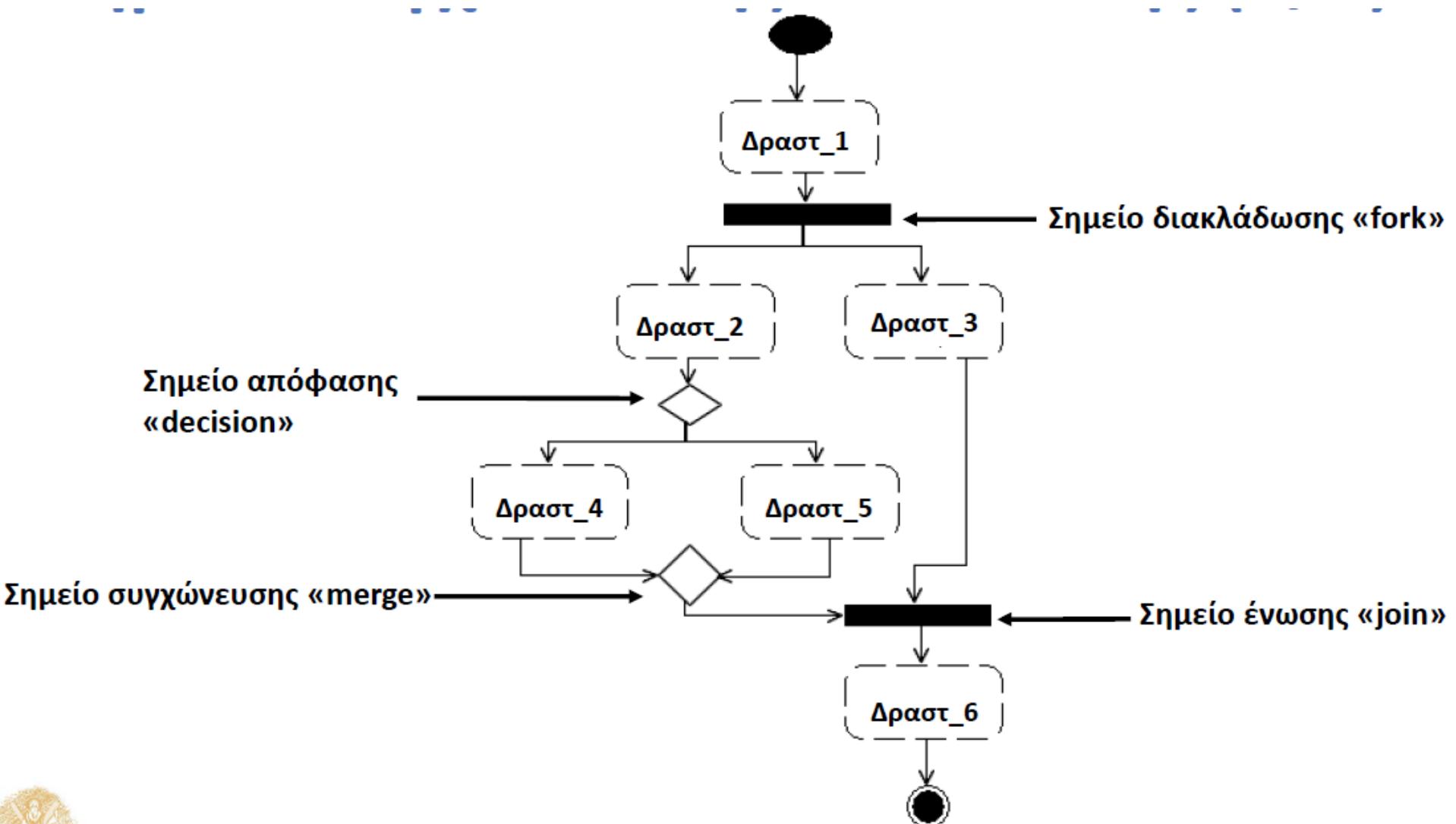
Τα διαγράμματα δραστηριοτήτων δεν αναπαριστούν λεπτομέρειες σχετικά με το πως τα αντικείμενα συμπεριφέρονται ή πως τα αντικείμενα συνεργάζονται

Τα διαγράμματα δραστηριοτήτων χρησιμοποιούν **διακλαδώσεις, σημεία αποφάσεων** και **σημεία συγχώνευσης** και **ένωσης**

Μία διακλάδωση χρησιμοποιείται όταν πολλές δραστηριότητες μπορούν να πραγματοποιούνται παράλληλα



Παράδειγμα Διαγρ. Δραστηριοτήτων



Διάγραμμα Δραστηριοτήτων

Στο παραπάνω διάγραμμα παρουσιάζεται μια διακλάδωση μετά τη Δραστηριότητα 1.

Αυτή δείχνει ότι οι Δραστηριότητες 2 και 3 πραγματοποιούνται ταυτόχρονα.

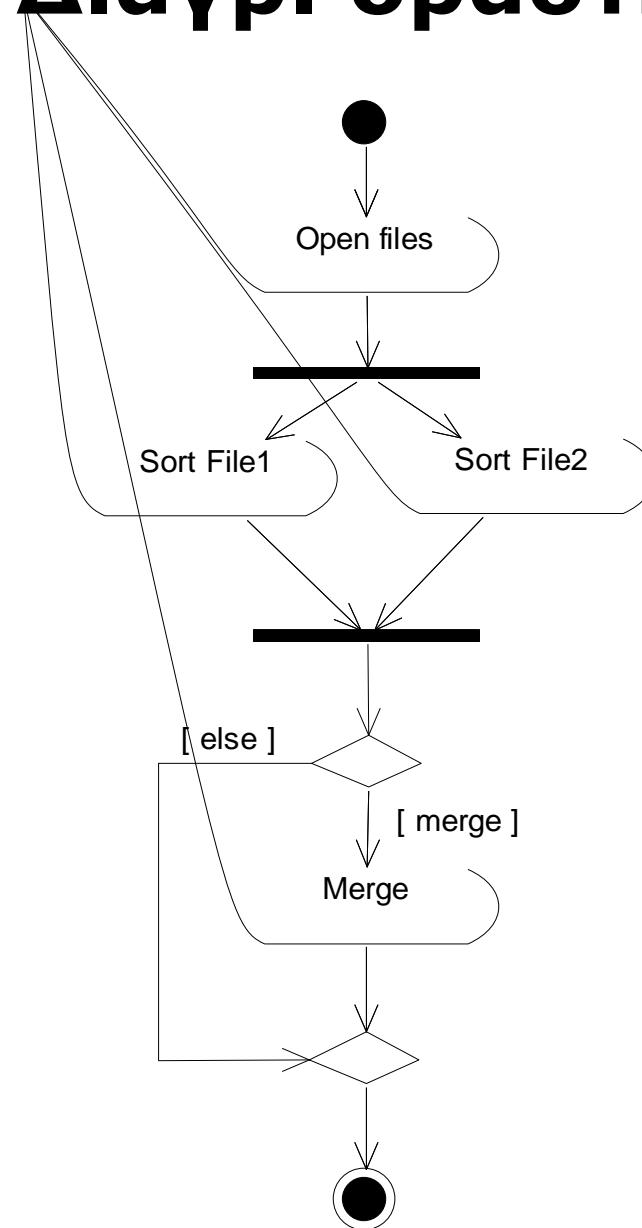
Μετά από μία διακλάδωση, όλες οι παράλληλες δραστηριότητες πρέπει να ενωθούν με ένα αντικείμενο ένωσης «join» πριν τη μετάβαση στην τελική κατάσταση.

Μετά τη Δραστηριότητα 2 υπάρχει ένα σημείο απόφασης. Το σημείο απόφασης περιγράφει ποιες δραστηριότητες θα πραγματοποιηθούν σύμφωνα με τις συνθήκες που ικανοποιούνται.

Τα σημεία απόφασης συνοδεύονται από ένα αντικείμενο συγχώνευσης «merge» που δηλώνει το τέλος της συνθήκης.



Παράδειγμα Διαγρ. δραστηριοτήτων



Διαγράμματα Δραστηριοτήτων και Καταστάσεων

Διαφορές:

- 'Ενα **διάγραμμα καταστάσεων** αναπαριστά τις διαφορετικές καταστάσεις ενός αντικειμένου κατά τη διάρκεια της ζωής του στο σύστημα και τις μεταβάσεις στις καταστάσεις των αντικειμένων
- 'Ένα **διάγραμμα δραστηριοτήτων** αναπαριστά κατά κύριο λόγο τις μεταβάσεις και τις δραστηριότητες που προκαλούν τις αλλαγές στις καταστάσεις των αντικειμένων

Στοιχεία Διαγράμματος Δραστηριοτήτων

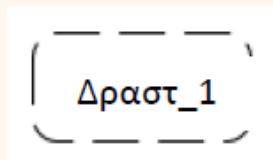
Στοιχείο Διαγράμματος Δραστηριοτήτων

Αρχική Δραστηριότητα: Αναπαριστά το σημείο εκκίνησης ή την πρώτη δραστηριότητα μιας ροής



Δραστηριότητα

Αποφάσεις: Σημείο απόφασης που οδηγεί σε δύο ή περισσότερους δρόμους ανάλογα με την απόφαση



Τελική Δραστηριότητα: Το τέλος ενός διαγράμματος δραστηριοτήτων



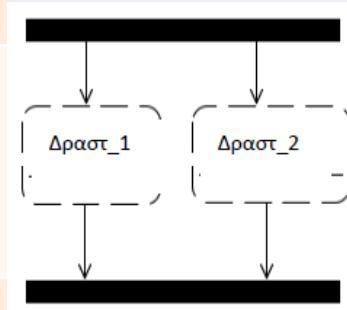
Στοιχεία Διαγράμματος Δραστηριοτήτων

Στοιχείο Διαγράμματος Δραστηριοτήτων

Συγχωνεύσεις: Σημείο συγχώνευσης χρησιμοποιείται συνήθως μετά από ένα σημείο απόφασης



Ταυτόχρονες Δραστηριότητες: Μερικές δραστηριότητες εκτελούνται παράλληλα



Διακλάδωση/Ένωση



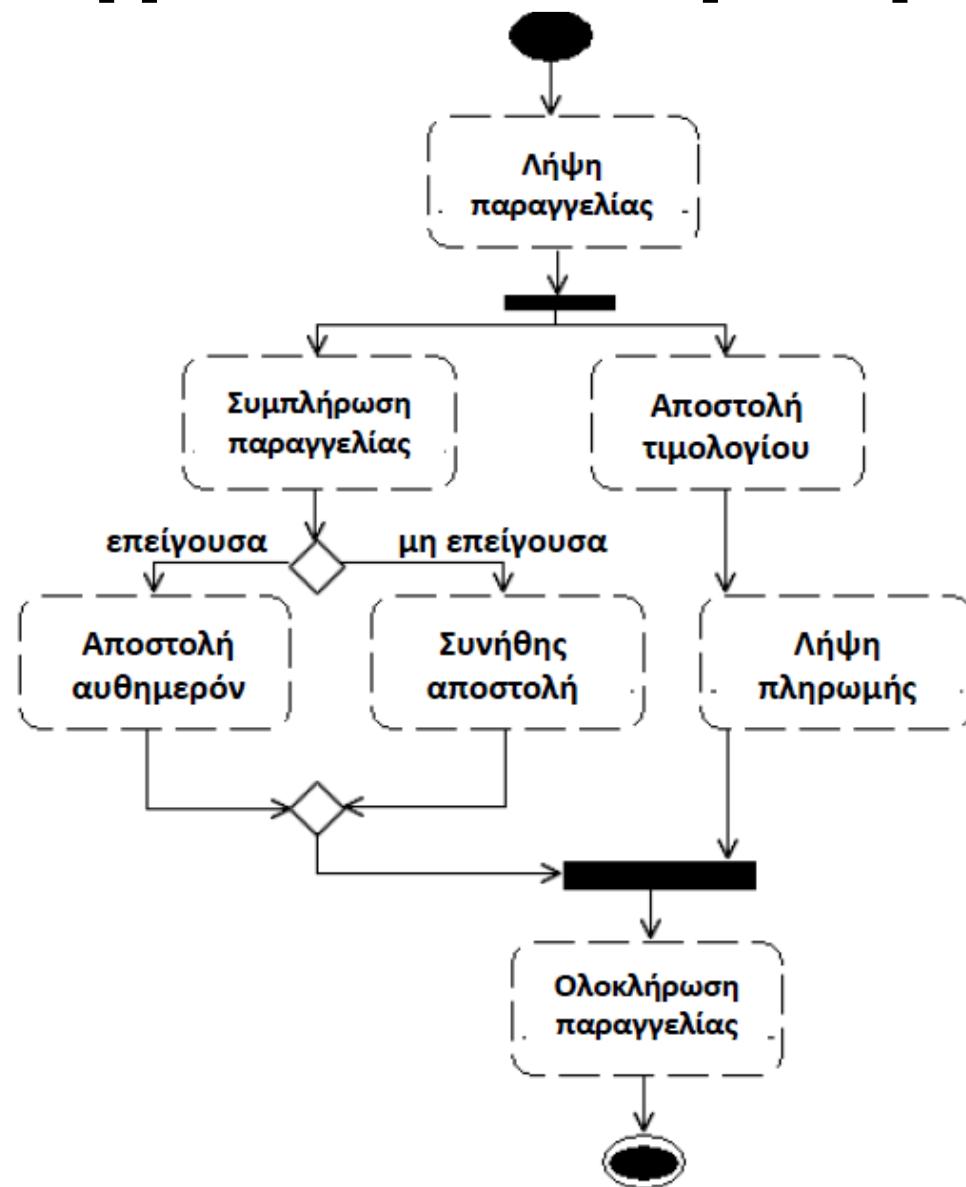
Παράδειγμα Εκτέλεση Παραγγελίας

Το διάγραμμα θα πρέπει αναπαριστά τη ροή των ενεργειών σε ένα σύστημα διαχείρισης παραγγελιών

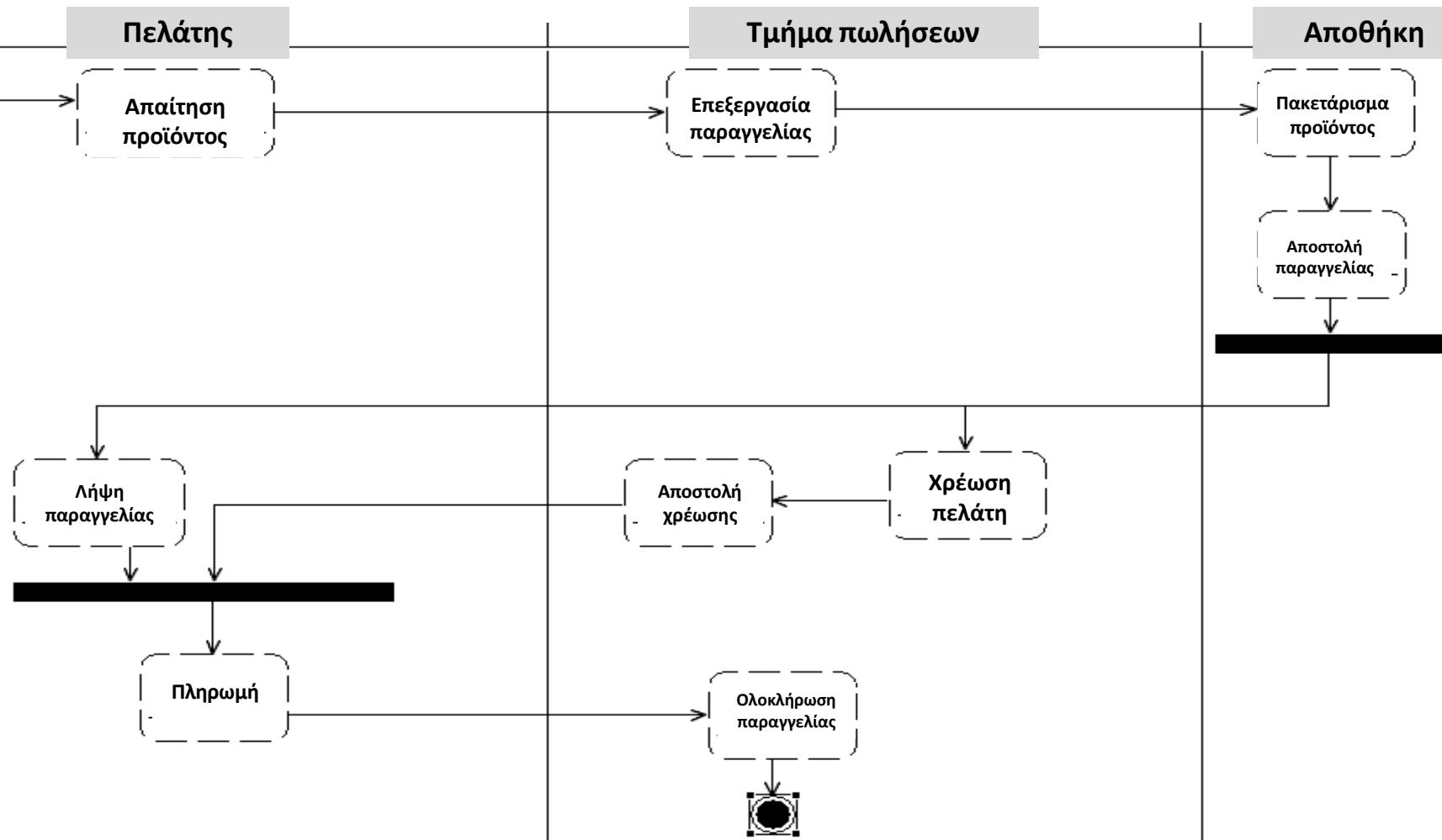
- Συγκεκριμένα, με τη λήψη μιας παραγγελίας οι δραστηριότητες διαχωρίζονται σε δυο παράλληλες ομάδες δραστηριοτήτων
- Η μία πλευρά συμπληρώνει και αποστέλλει την παραγγελία, ενώ η άλλη πλευρά διαχειρίζεται την πληρωμή (στέλνει το τιμολόγιο και λαμβάνει την πληρωμή)
- Μετά τη συμπλήρωση της παραγγελίας, η μέθοδος αποστολής αποφασίζεται ύστερα από την ικανοποίηση μιας συνθήκης
- Ανάλογα με τη συνθήκη πραγματοποιείται είτε η δραστηριότητα «Αποστολή Αυθημερόν» είτε η δραστηριότητα «Συνήθης Αποστολή»
- Τέλος, οι παράλληλες δραστηριότητες ενώνονται πριν την ολοκλήρωση της παραγγελίας



Παράδειγμα Εκτέλεσης Παραγγελίας



Παράδειγμα Εκτέλεσης Παραγγελίας



SwimLanes

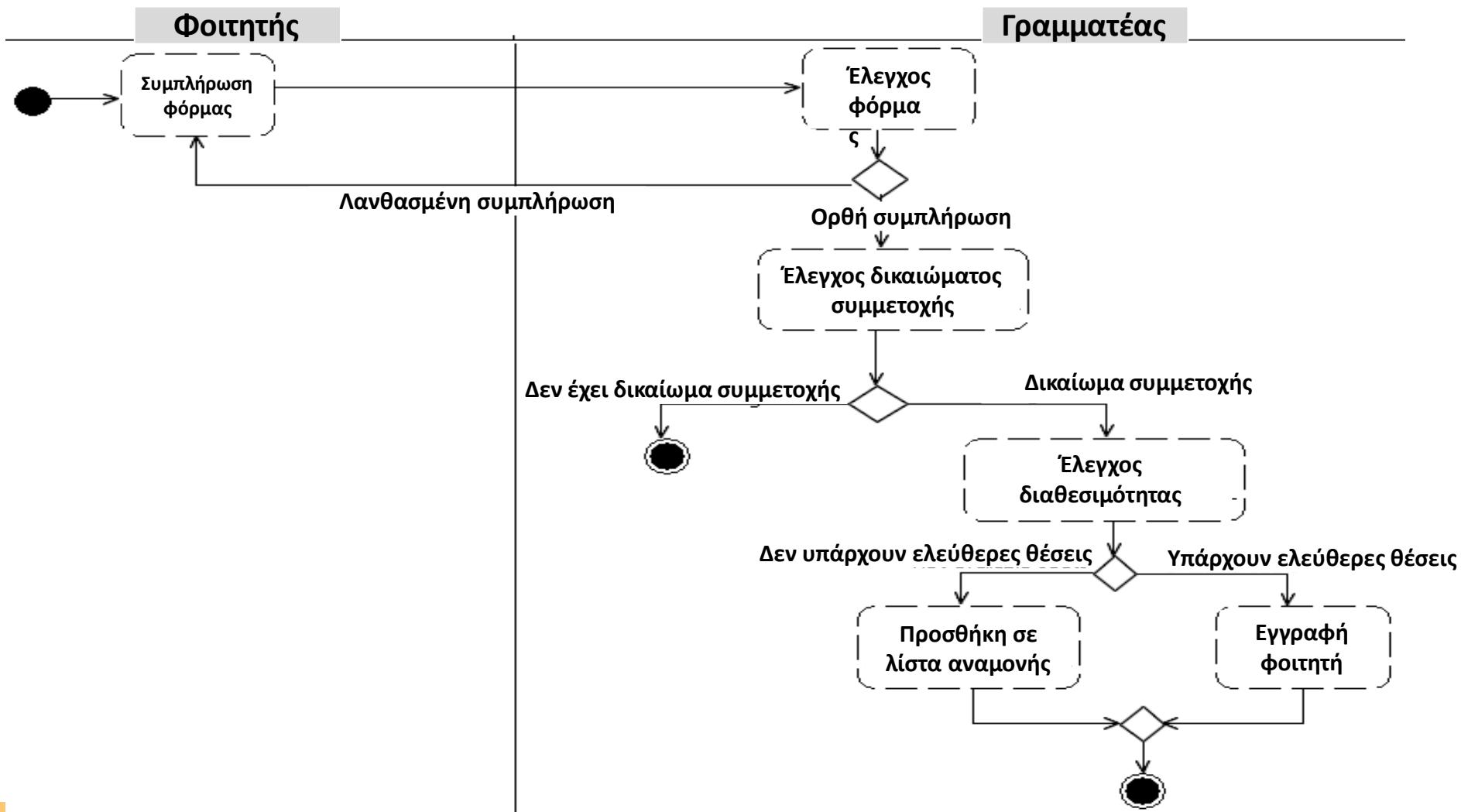
- Μερικές φορές είναι χρήσιμο όταν μοντελοποιούμε ροές εργασίας να διαχωρίσουμε τις δραστηριότητες των διαγραμμάτων δραστηριοτήτων σε ομάδες, όπου κάθε ομάδα αναπαριστά την επιχειρησιακή μονάδα του οργανισμού (υπάλληλος, τμήμα, κλπ.) που είναι υπεύθυνη για την εκτέλεση αυτών των δραστηριοτήτων
- •Στη UML αυτές οι ομάδες ονομάζονται **SwimLanes**
- •Στα διαγράμματα δραστηριοτήτων που διαχωρίζονται σε **SwimLanes**, κάθε δραστηριότητα ανήκει αποκλειστικά σε ένα **SwimLane**, αλλά οι μεταβάσεις από τη μία στην άλλη δραστηριότητα μπορεί να διαπερνούν τα **SwimLanes**

Παράδειγμα Σεμινάρια σε ΑΕΙ

- Διάγραμμα δραστηριοτήτων για την ακόλουθη διαδικασία:
 - Συμπλήρωση φόρμας συμμετοχής σε σεμινάριο από το φοιτητή
 - Έλεγχος της φόρμας από το γραμματέα
 - Αν υπάρχει λάθος στη φόρμα ο φοιτητής θα πρέπει να τη συμπληρώσει ξανά, διαφορετικά ο γραμματέας ελέγχει το αν ο φοιτητής έχει δικαίωμα συμμετοχής στο σεμινάριο
 - Σε περίπτωση που δεν έχει δικαίωμα η διαδικασία ολοκληρώνεται, διαφορετικά ο γραμματέας ελέγχει την ύπαρξη ελευθέρων θέσεων στο σεμινάριο
 - Σε περίπτωση που υπάρχουν ελεύθερες θέσεις ο γραμματέας γράφει το φοιτητή στο σεμινάριο, διαφορετικά τον βάζει σε λίστα αναμονής και η διαδικασία ολοκληρώνεται



Παράδειγμα Σεμινάρια σε ΑΕΙ



Ανάλυση Πληροφοριακών Συστημάτων

Εαρινό Εξάμηνο 2020-2021

Lec08 – 11/05/2021

Διδάσκων: Γεώργιος Χρ. Μακρής

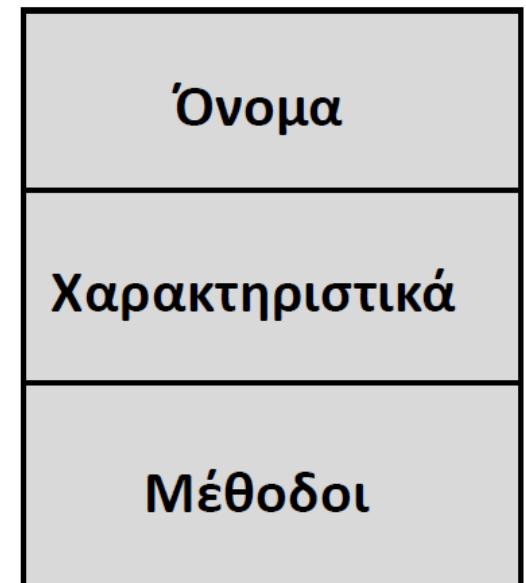
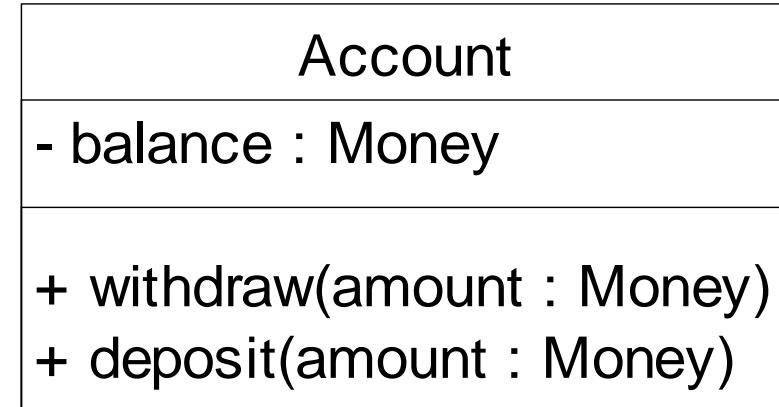


Διαγράμματα Κλάσεων

Τα βασικά στοιχεία

Κλάση

- Οι κλάσεις αποτελούν τη βάση της κατασκευής οποιουδήποτε αντικειμενοστρεφούς συστήματος. Ενσωματώνουν τα δεδομένα καθώς και τις λειτουργίες που επενεργούν στα δεδομένα αυτά.
- Συμβολίζονται με ένα παραλληλόγραμμο που έχει τρία διαμερίσματα. Στο πάνω διαμέρισμα αναγράφεται το όνομα της κλάσης, στο μεσαίο διαμέρισμα απεικονίζονται οι ιδιότητες, και στο κάτω διαμέρισμα οι λειτουργίες.



Διαγράμματα κλάσεων (Class Diagrams)

- Μια κλάση είναι ένα **πρότυπο** για ένα **σύνολο αντικειμένων** με παρόμοιες:
 - **ιδιότητες** (attributes),
 - **λειτουργίες** (operations),
 - **σχέσεις** (relationships) και
 - **σημασιολογία - όρους συμβολισμού** (semantics).
-
- Τα αντικείμενα μιας κλάσης έχουν **κατάσταση** (state) και **συμπεριφορά** (behavior).
 - Είναι στατικά διαγράμματα που δείχνουν κλάσεις, ιδιότητες, λειτουργίες και σχέσεις μεταξύ κλάσεων

Διαγράμματα κλάσεων (Class Diagrams)

- Χρησιμοποιούνται στην ανάλυση των απαιτήσεων και στη σχεδίαση του λογισμικού
- Περιγράφουν την δομή και συμπεριφορά των ΠΧ
- Δίνουν ένα εννοιολογικό μοντέλο του συστήματος, με τους όρους οντοτήτων και σχέσεων
- Τα αναλυτικά διαγράμματα κλάσεων χρησιμοποιούνται από τους προγραμματιστές

Κλάση

- Μια κλάση είναι μια περιγραφή ενός τύπου αντικειμένου
 - περιγράφει τα **χαρακτηριστικά** και τη **συμπεριφορά** του συγκεκριμένου τύπου αντικειμένου
- Όλα τα αντικείμενα είναι στιγμιότυπα μιας κλάσης
- Η σχέση ενός αντικειμένου με μια κλάση είναι παρόμοια με τη σχέση μιας μεταβλητής με τον τύπο της μεταβλητής σε μια γλώσσα προγραμματισμού
- •Οι κλάσεις χρησιμοποιούνται για τη **μοντελοποίηση** και **περιγραφή** συστημάτων
- •Η τεχνική της δημιουργίας ιεραρχίας υπάρχει και στην αντικειμενοστραφή μοντελοποίηση



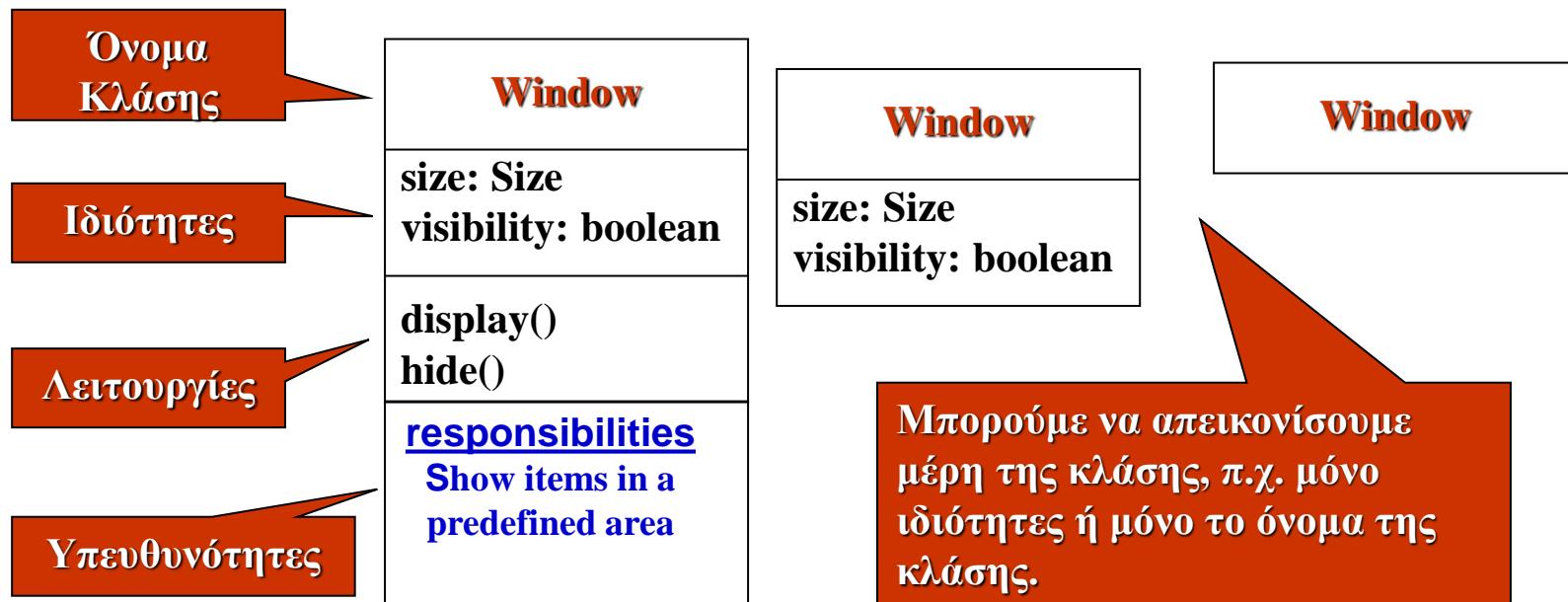
Κλάση

- Μια κλάση μπορεί να περιγράφει αντικείμενα οποιουδήποτε συστήματος
 - πληροφοριακό σύστημα
 - επιχειρησιακό σύστημα
 - τεχνικό σύστημα
 - κατανεμημένο σύστημα
 - λογισμικό, κλπ.
- Παραδείγματα κλάσεων σε πληροφοριακά ή επιχειρησιακά συστήματα είναι
 - πελάτης
 - παραγγελία
 - τιμολόγιο
 - δελτίο αποστολής, κλπ.



Απεικόνιση Κλάσεων (2)

- Στην UML οι κλάσεις απεικονίζονται με ένα τετράγωνο τριών τμημάτων που περιέχουν: το όνομα της κλάσης, τις ιδιότητες (attributes) της κλάσης, τις λειτουργίες (operations) της κλάσης και τις υπευθυνότητες (responsibilities) που είναι συμβόλαια ή υποχρεώσεις της κλάσης.



Διαγράμματα Κλάσεων - Ιδιότητες

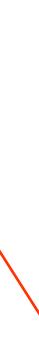
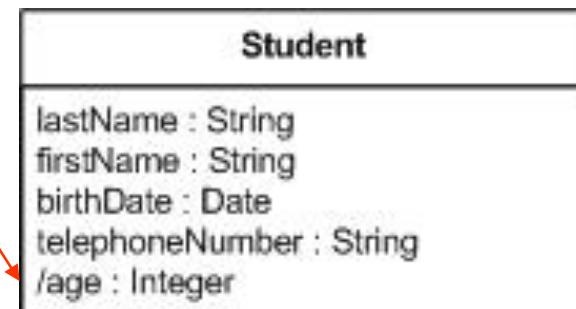
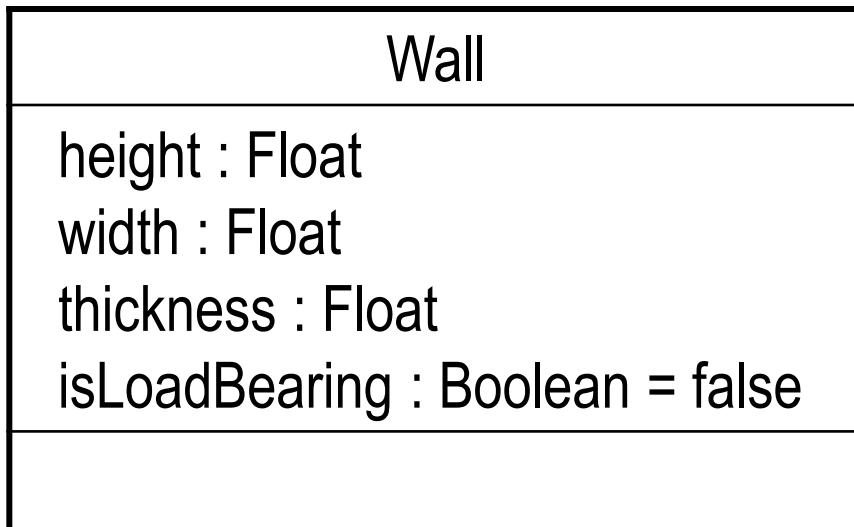
- Μια κλάση μπορεί να έχει **ιδιότητες** ή να μην έχει
- Οι ιδιότητες είναι **κοινές για όλα τα αντικείμενα της κλάσης**
- Σε οποιαδήποτε χρονική στιγμή ένα αντικείμενο της κλάσης θα έχει **συγκεκριμένες τιμές** για κάθε μια ιδιότητα της κλάσης
- Μπορεί να φαίνεται το **όνομα της ιδιότητας**, ο **τύπος των δεδομένων**, αλλά και **αρχική τιμή σε όλες ή κάποιες από αυτές**

Customer
name
address
phone
birthdate

Wall
height : Float
width : Float
thickness : Float
isLoadbearing : Boolean = false

Διαγράμματα Κλάσεων - Ιδιότητες

- Τυπική σύνταξη ιδιοτήτων
Όνομα : Τύπος = αρχική_τιμή
- Παραγόμενες (derived) ιδιότητες: η τιμή τους προκύπτει από άλλες ιδιότητες της κλάσης. Οι παραγόμενες ιδιότητες έχουν το σύμβολο / πριν από το όνομα.



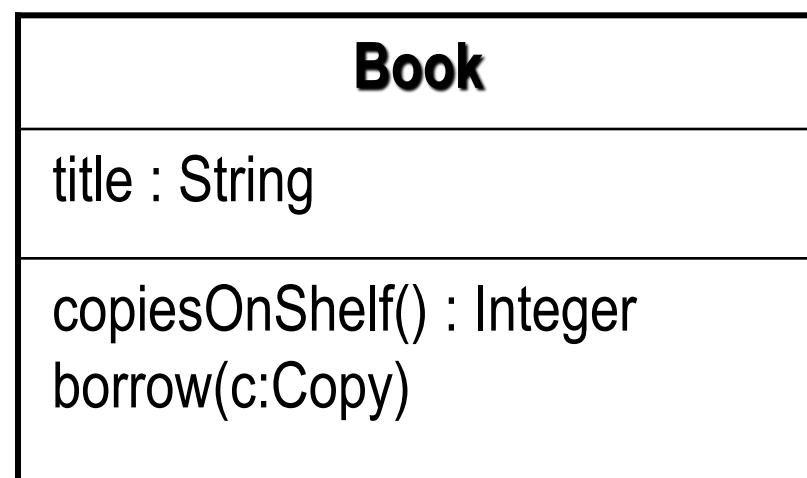
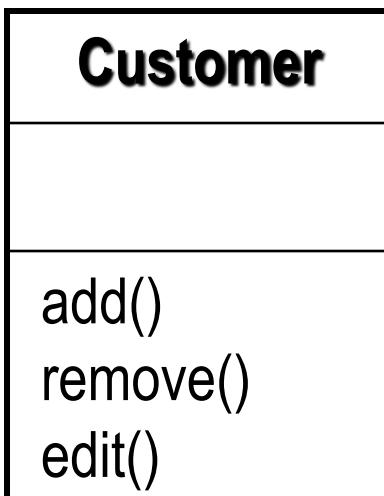
Διαγράμματα Κλάσεων - Λειτουργίες

- Λειτουργία είναι η **υλοποίηση μιας υπηρεσίας που απαιτείται από ένα αντικείμενο της κλάσης**
- Η λειτουργία επιδρά στην **συμπεριφορά του αντικειμένου** και συχνά οδηγεί στην **αλλαγή των δεδομένων ή της κατάστασης του**
- Μια κλάση μπορεί να έχει ή να μην έχει καθόλου λειτουργίες

Διαγράμματα Κλάσεων - Λειτουργίες

Στις λειτουργίες μπορεί να αναγράφεται:

- μόνο το όνομα της λειτουργίας
- το όνομα και η υπογραφή (signature), δηλαδή ο τύπος και οι αρχικές τιμές όλων των παραμέτρων και στις περιπτώσεις των συναρτήσεων ο τύπος του επιστρεφόμενου αποτελέσματος

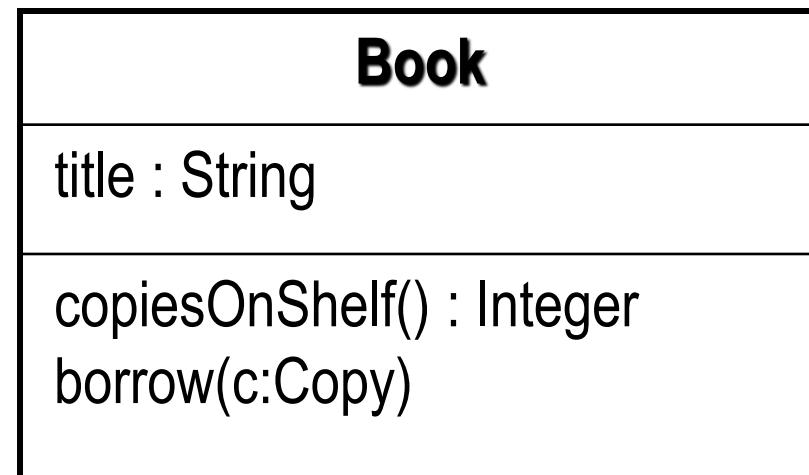
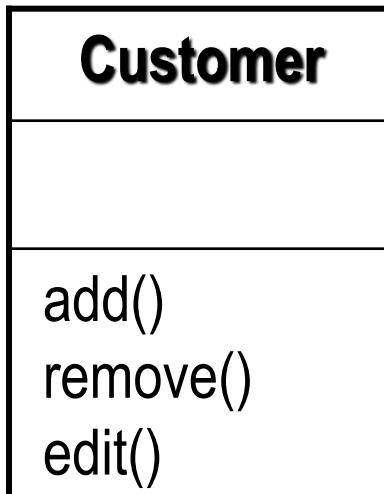


Διαγράμματα Κλάσεων - Λειτουργίες

Στις λειτουργίες μπορεί να αναγράφεται:

- μόνο το όνομα της λειτουργίας
- το όνομα και η υπογραφή (signature), δηλαδή ο τύπος και οι αρχικές τιμές όλων των παραμέτρων και ο τύπος του επιστρεφόμενου αποτελέσματος.

Όνομα(Παράμετρος1 : Τύπος1, Παράμετρος2 : Τύπος2, ...) : Τύπος_Επιστροφής



Προσπελασημότητα

Ορατότητα (Visibility)

Η ορατότητα ορίζει αν μια ιδιότητα ή λειτουργία είναι ορατή, δηλαδή μπορεί να προσπελαθεί από άλλες κλάσεις.

Η UML παρέχει τέσσερα σύμβολα για την ορατότητα :

- + **δημόσια ορατότητα (public).** Προσπέλαση και από άλλες κλάσεις.
- **ιδιωτική ορατότητα (private).** Προσπέλαση μόνο μέσα στην ίδια κλάση.
- # **προστατευμένη ορατότητα (protected).** Προσπέλαση έξω από το πακέτο αλλά από υποκλάσεις της τρέχουσας (κληρονομικότητα)
- ~ **ορατότητα πακέτου (package).** Προσπέλαση μόνο από το γκρουπ των κλάσεων του πακέτου

Μη χρήση συμβόλου πρόσβασης σημαίνει πως η πρόσβαση δεν φαίνεται, αλλά αυτό δε σημαίνει πως δεν υπάρχει πρόσβαση ή πως η πρόσβαση είναι δημόσια.

Δημόσια και Ιδιωτικά Χαρακτηριστικά

- Το συνηθισμένο σε μία κλάση είναι ότι οι ιδιότητες είναι **ιδιωτικές** (private) και οι λειτουργίες **δημόσιες** (public).
 - Ιδιωτικά είναι τα χαρακτηριστικά μιας κλάσης (ιδιότητες ή λειτουργίες) που δεν είναι προσπελάσιμα από άλλες κλάσεις
 - Δημόσια αυτά που είναι προσπελάσιμα από άλλες κλάσεις.
- Αρχή απόκρυψης δεδομένων (data hiding principle):
 - Εύκολη αλλαγή της εσωτερικής αναπαράστασης δεδομένων
 - Εφαρμογή πολιτικών ελέγχου από τις μεθόδους



Συντακτικό Ιδιοτήτων στη UML

- Το πλήρες συντακτικό για τη δήλωση ιδιοτήτων στη UML είναι:

**Προσδιοριστής-πρόσβασης όνομα-ιδιότητας: Τύπος
[πολλαπλότητα διάταξη] = Αρχική-τιμή
{συμβολοσειρά ιδιοτήτων}**

- Μερικά Παραδείγματα:
 - - **students : string[* ordered]**
 - - **interestRate : double = 0.035 {frozen}**



Συντακτικό Λειτουργιών στη UML

- Το πλήρες συντακτικό για τη δήλωση λειτουργιών στη UML είναι:

Προσδιοριστής-πρόσβασης όνομα (λίστα-παραμέτρων) : τύπος-επιστροφής {συμβολοσειρά ιδιοτήτων}

- Συντακτικό λίστας παραμέτρων:
 - **[*in* | *out* | *inout*] όνομα-παραμέτρου : τύπος = εξ ορισμού τιμή**
- Μερικά παραδείγματα:
 - + ***withdraw(in amount : Money) : boolean***
 - + ***getBalance():Money {query}***



Λειτουργίες με Εμβέλεια Κλάσης

- Κάποιες ιδιότητες ή λειτουργίες έχουν εμβέλεια κλάσης και όχι αντικειμένου.
- Έστω, για παράδειγμα, ότι θέλουμε να μπορούμε να δημιουργήσουμε αντικείμενα Account διαβάζοντας τα στοιχεία για τη δημιουργία κάθε τέτοιου αντικειμένου από μια βάση δεδομένων. Θα μπορούσαμε να δηλώσουμε μια λειτουργία με το εξής συντακτικό:
+createAccount(AccountID : String) : Account

Account
- balance : Money
+ <u>create(accountID : String) : Account</u>
+ deposit(amount : Money)
+ withdraw(amount : Money)



Παράδειγμα κλάσης και αντικειμένου

Αυτοκίνητο

αριθμ.πινακίδας
μοντέλο
ταχύτητα
κατεύθυνση

Ξεκίνα
Επιτάχυνε
Φρέναρε

Το αυτοκίνητο
του Γιώργου
Μακρή:
Αυτοκίνητο

NKK 1111
Nissan
220 Km
Δυτική

Όνομα

Χαρακτηριστικά

Μέθοδοι

Σχέσεις μεταξύ Κλάσεων (Relationships)

- Οι πιο σημαντικές σχέσεις στην αντικειμενοστρεφή τεχνολογία λογισμικού οι :
 - ✓ **Συσχετίσεις** (associations)
 - ✓ **Εξαρτήσεις** (dependencies)
 - ✓ **Γενικεύσεις** (generalizations)
 - ✓ **Συσσωματώσεις** (aggregations)
 - ✓ **Συνθέσεις** (Compositions)
- Διαφορετικές γραμμές και βέλη χαρακτηρίζουν τις σχέσεις

Διαγράμματα Κλάσεων Συσχετίσεις

- Η **συσχέτιση** (association) αναπαριστά κάποια σύνδεση των αντικειμένων δύο κλάσεων
- Απεικονίζει τη σχέση μεταξύ των κλάσεων
- Γραφικά απεικονίζονται με **συμπαγείς γραμμές**

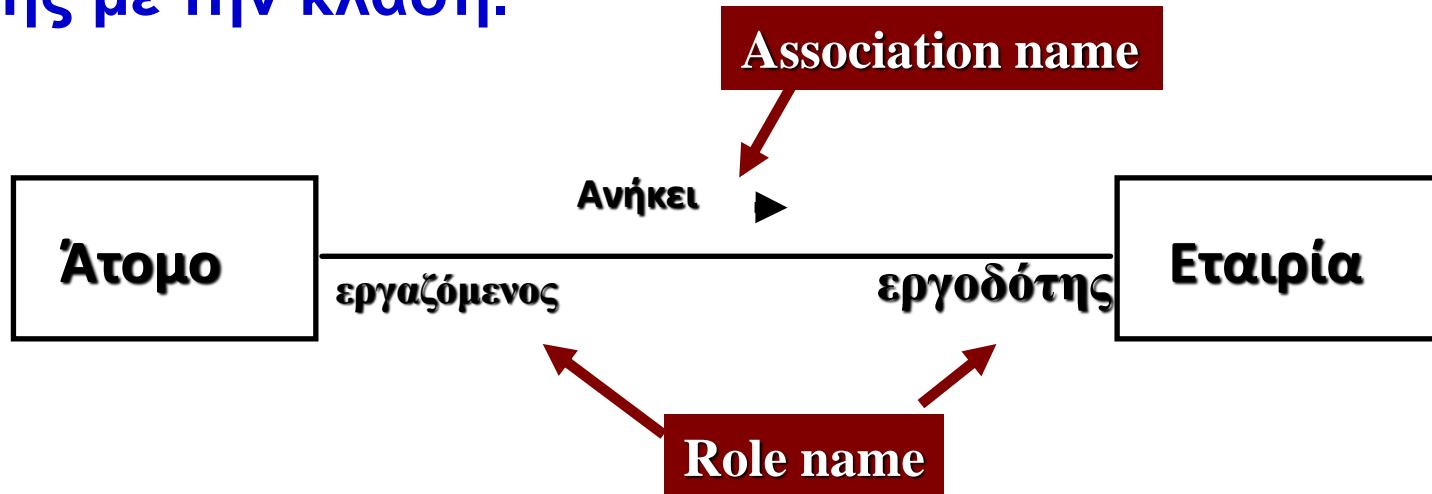
Παράδειγμα:

“Ένας υπάλληλος εργάζεται σε μία εταιρία”



Διαγράμματα Κλάσεων Συσχετίσεις

- **Όνομα σχέσης:** γράφεται σαν ετικέτα στο μέσο της γραμμής σύνδεσης (ρήμα). Το όνομα φανερώνει το είδος της συσχέτισης.
- Μία κλάση που μετέχει σε μία συσχέτιση έχει ένα **ρόλο** (role) σε αυτήν. Γράφουμε το όνομα του ρόλου (ουσιαστικό) στο τέλος της σχέσης, δηλαδή στο σημείο σύνδεσης με την κλάση.

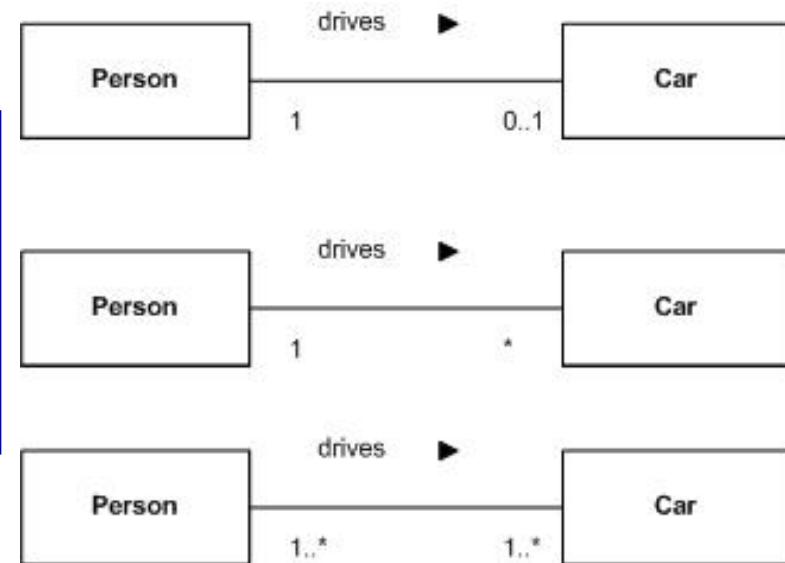


Διαγράμματα Κλάσεων Συσχετίσεις

Πολλαπλότητα (multiplicity)

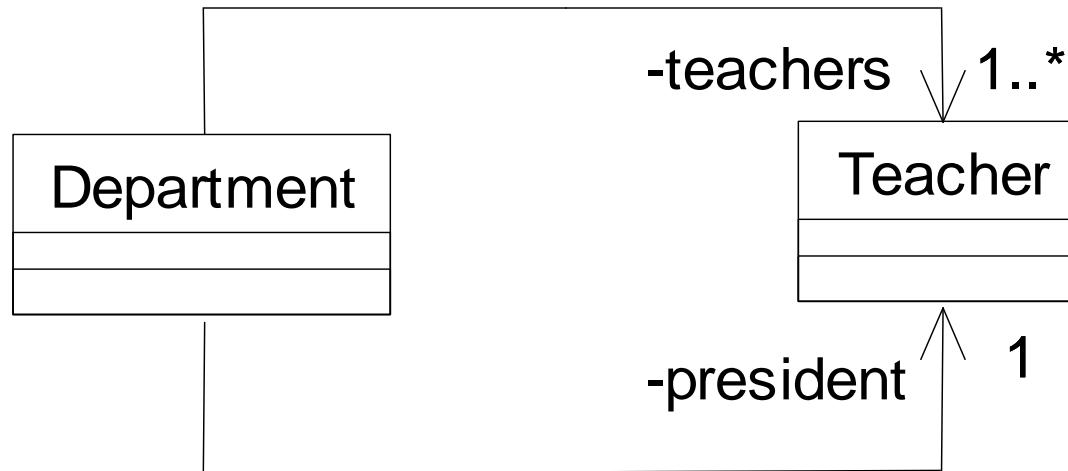
Καθορίζει πόσα αντικείμενα μετέχουν σε μια συσχέτιση
(από 0 έως οποιοδήποτε πλήθος).

- ✓ Ακριβώς ένα 1
- ✓ Μηδέν ή ένα 0..1
- ✓ Μηδέν ή περισσότερα 0..*
- ✓ Ένα ή περισσότερα 1..*
- ✓ Από..Έως m..n
- ✓ Πολλαπλά διαφορετικά όρια 2, 4..6, 8



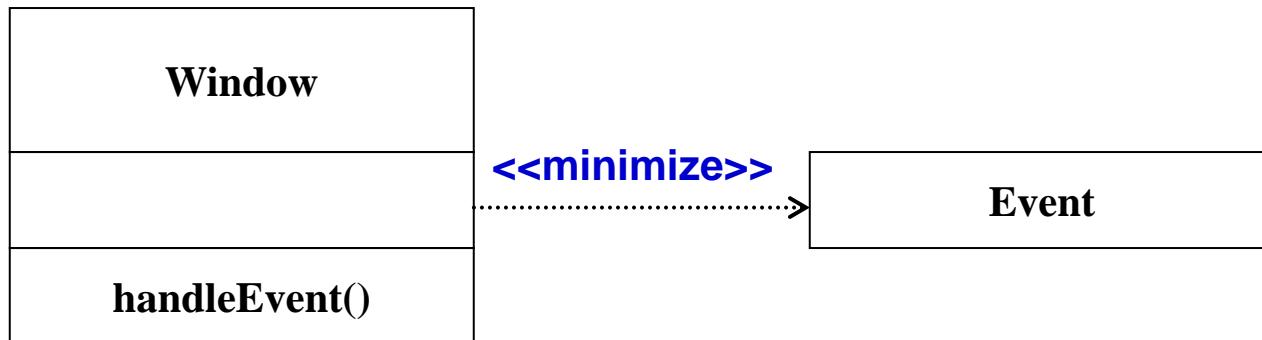
Όνόματα Άκρων Συσχέτισης

- Σε κάθε άκρο μίας συσχέτισης μπορούμε να βάλουμε ένα όνομα που υποδηλώνει το ρόλο αυτής της κλάσης στη συσχέτιση.
- Επειδή το όνομα του άκρου συσχέτισης προσδιορίζει το ρόλο μίας κλάσης στη συσχέτιση, μερικές φορές αναφέρεται και ως όνομα ρόλου (rolename)
- Απαραίτητο όταν δύο κλάσεις συνδέονται με περισσότερες από μία συσχετίσεις όπως στην Εικόνα



Εξαρτήσεις (dependencies)

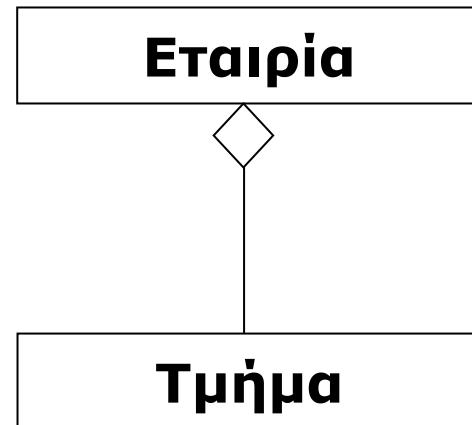
- Εξάρτηση είναι σχέση “χρήσης” δύο κλάσεων, δηλαδή μια αλλαγή στη χρήση της μιας από αυτές μπορεί να επιφέρει αλλαγές και στην άλλη
- Γραφικά απεικονίζεται με διακεκομένη γραμμή
- Ένα στερεότυπο χρησιμοποιείται για να δείξει το είδος της εξάρτησης
- Στο παράδειγμα : Η κλάση Window χρησιμοποιεί την Event



Διαγράμματα Κλάσεων

Συσσωμάτωση

- Η **συσσωμάτωση** (aggregation) είναι μία ειδική μορφή συσχέτισης τύπου “όλου / μερών...” ή “έχει...”, όπου μία μεγάλη κλάση αποτελείται από άλλες μικρότερες κλάσεις.
- Γραφικά απεικονίζεται με βέλος αιχμής σχήματος διαμαντιού

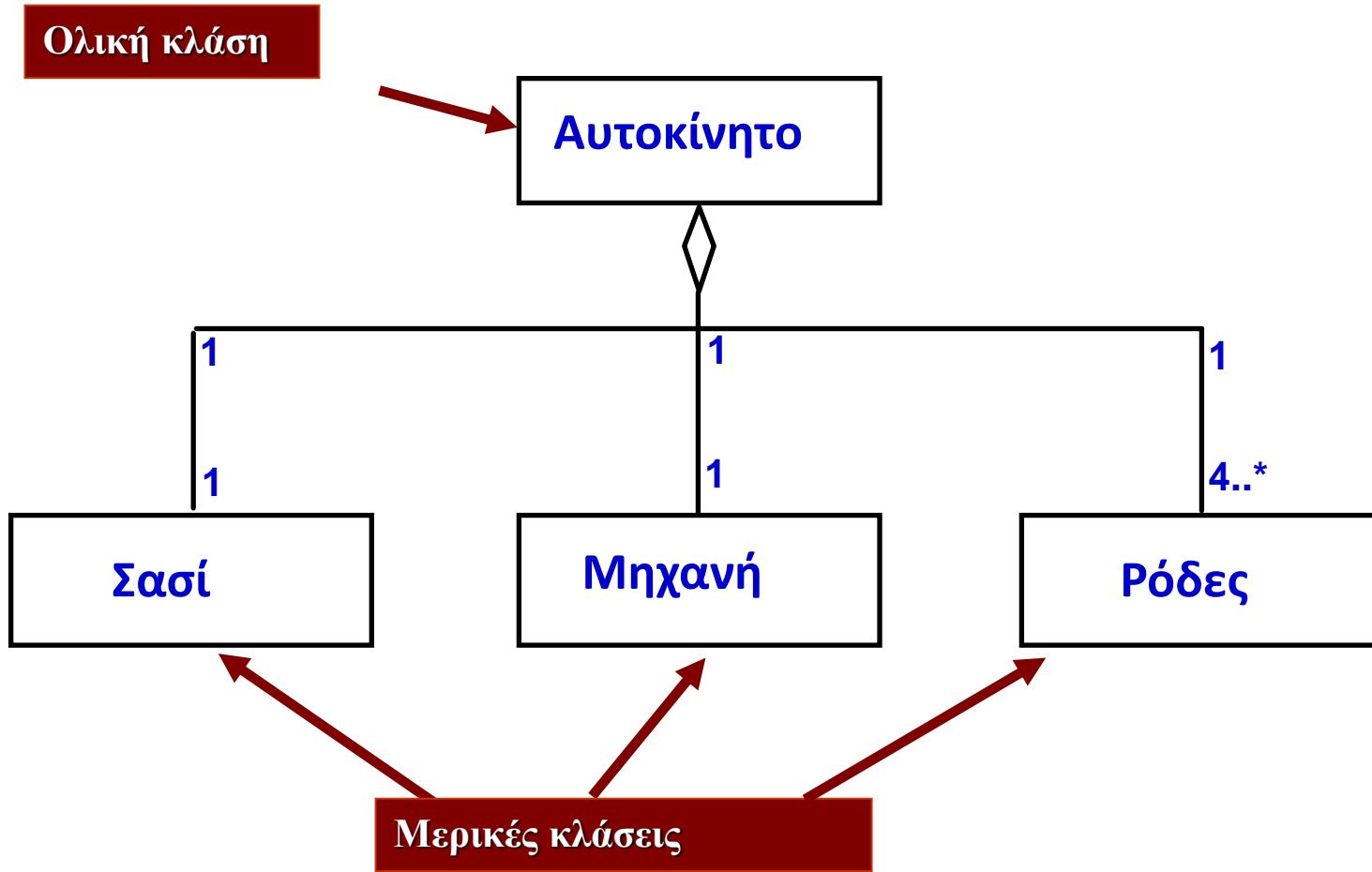


- Η κλάση Εταιρία αναπαριστά το «όλο» και η κλάση Τμήμα το «τμήμα»
- Το «όλο» δεν μπορεί να «λειτουργήσει» χωρίς το «τμήμα»



Διαγράμματα Κλάσεων

Συσσωμάτωση

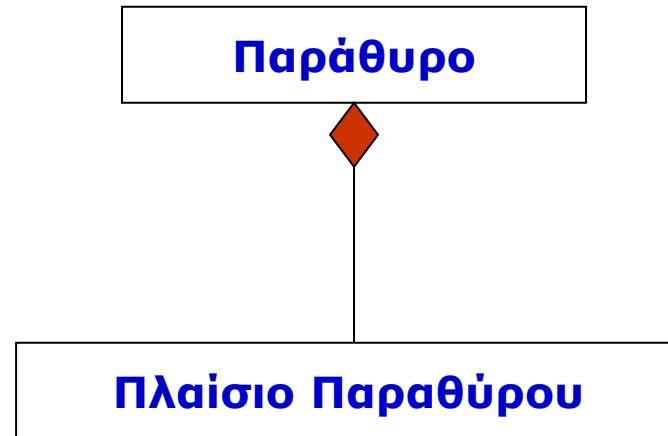


Διαγράμματα Κλάσεων - Σύνθεση (composition)

Είναι μια **ισχυρότερη σχέση** τύπου συσσωμάτωσης (όλου / μερών...) μόνο που τα μέρη θεωρούνται **αχώριστα** από το όλο

Το όλο **ελέγχει** πλήρως τα τμήματά (κύκλο ζωής τους). Για παράδειγμα η διαγραφή ενός αντικειμένου «όλου» διαγράφει και τα τμήματά του.

- Η πρόσβαση σε κάποιο «τμήμα» γίνεται αποκλειστικά μέσω του «όλου»
- Γραφικά απεικονίζεται με βέλος αιχμής σχήματος γεμάτου διαμαντιού



Διαγράμματα Κλάσεων - Σύνθεση (composition)

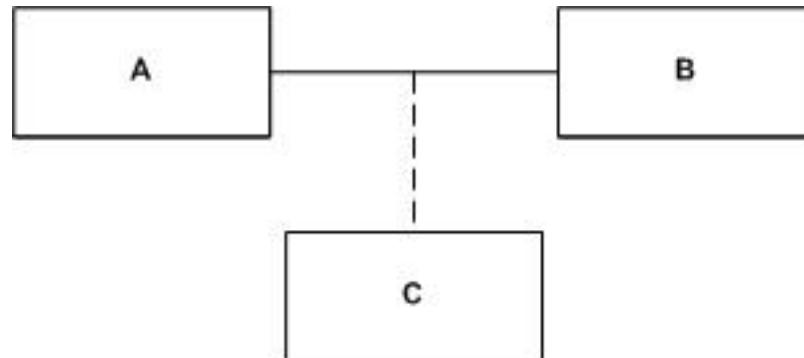
Παράδειγμα:



- Το αυτοκίνητο έχει ως αχώριστο τμήμα του τον κινητήρα.
- Ο κινητήρας δεν μπορεί να ανήκει ταυτόχρονα σε άλλο αυτοκίνητο.
- Η πρόσβαση στον κινητήρα μπορεί να γίνει μόνο μέσω του αυτοκινήτου.



Διαγράμματα Κλάσεων - Κλάση Συσχέτισης



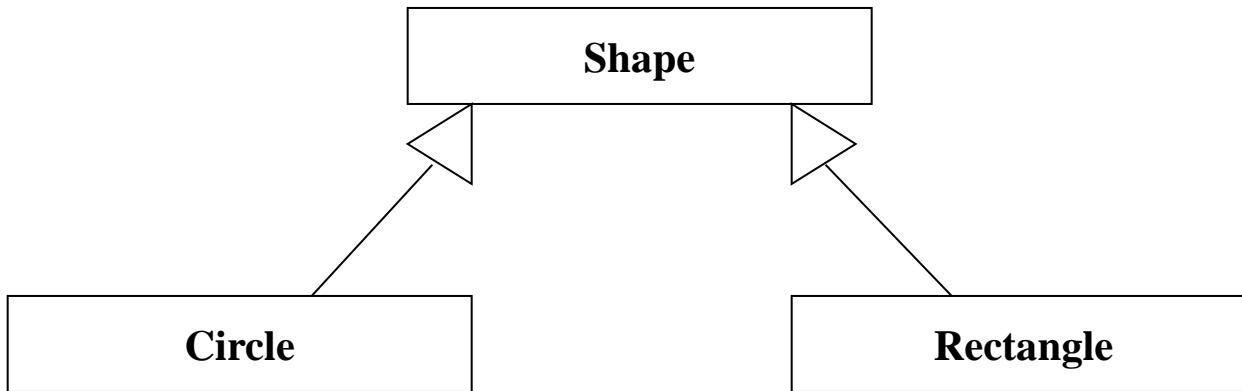
- Μία κλάση συσχέτισης (association class) αποδίδει ιδιότητες και λειτουργίες σε μία συσχέτιση
- Η κλάση C είναι η κλάση συσχέτισης των A και B
- Εκφράζει συσχετίσεις του τύπου «πολλά-προς-πολλά»



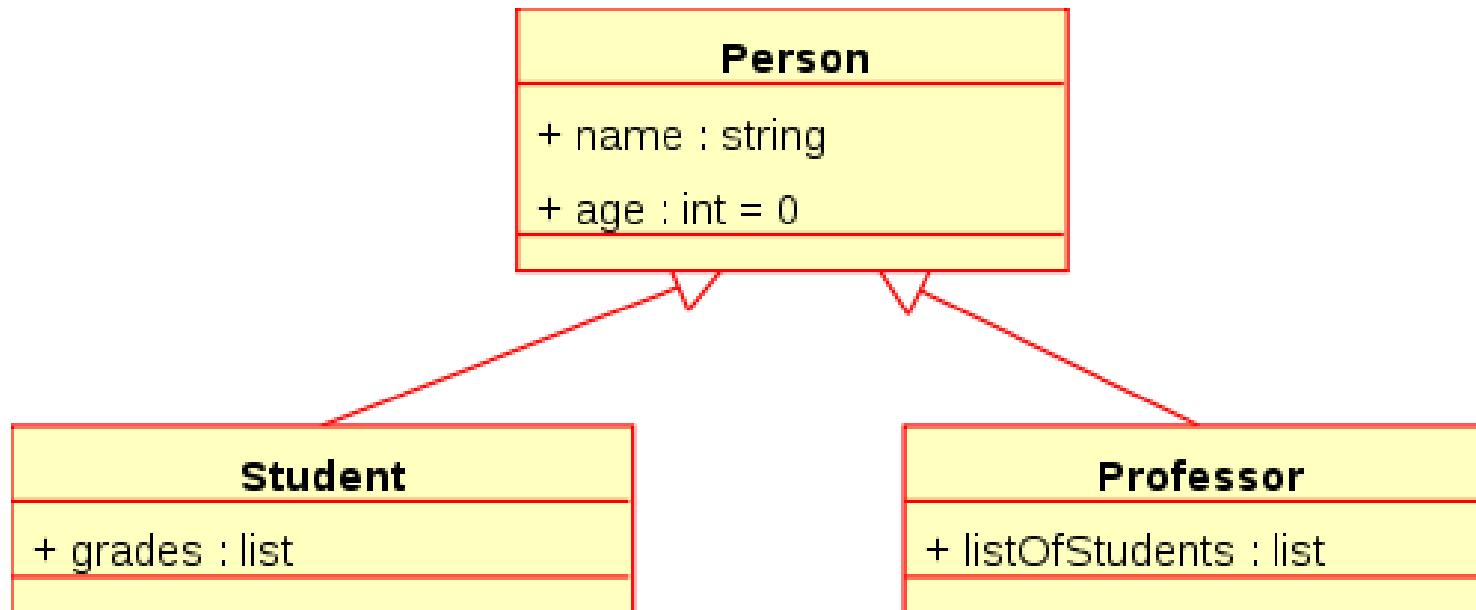
Διαγράμματα Κλάσεων

Γενίκευση - Κληρονομικότητα

- **Γενίκευση (generalization)** είναι η σχέση όπου μία κλάση (συνήθως υποκλάση ή “κλάση παιδί”) κληρονομεί τις λειτουργίες και τα δεδομένα από μία γενικότερη κλάση (υπερκλάση ή “κλάση γονέα”).
- Οι κλάσεις Circle και Rectangle είναι υποκλάσεις της Shape, και κληρονομούν ιδιότητες, συσχετίσεις και λειτουργίες της.
- Η υποκλάση μπορεί να **επαναορίσει (override)** λειτουργίες της υπερκλάσης της.



Διαγράμματα Κλάσεων Γενίκευση - Κληρονομικότητα



Παράδειγμα Συσχετίσεων στη Java

- Στο παράδειγμά μας έχουμε μία κλάση «**Teacher**», η οποία αφορά έναν εκπαιδευτικό. Κάθε εκπαιδευτικός έχει ένα όνομα που είναι αλφαριθμητικό (String). Επειδή ο εκπαιδευτικός δεν έχει την υποχρέωση να γνωρίζει το τμήμα στο οποίο ανήκει, η κλάση «Teacher» δεν αναφέρεται καθόλου στην κλάση «Department».



Κώδικας της Κλάσης Teacher

```
public class Teacher {  
    private String name;  
    public Teacher(String name) {  
        this.name = name;  
    }  
    public String getName() {  
        return name;  
    }  
    public boolean equals(Object o) {  
        Teacher t = (Teacher) o;  
        return t.name == this.name;  
    }  
    public int hashCode() {  
        return name.hashCode();  
    }  
}
```

Κώδικας της κλάσης Department (1)

...διάφορα import

```
public class Department {  
    private Teacher president;  
    private Set<Teacher> teachers;  
    private String name;  
    public Department(String name) {  
        this.name = name;  
        this.teachers = new HashSet<Teacher>();  
    }  
    public boolean addTeacher(Teacher t) {  
        return teachers.add(t);  
    }  
    public boolean removeTeacher(Teacher t) {  
        return teachers.remove(t);  
    }  
}
```



Κώδικας της κλάσης Department (2)

```
public boolean findTeacher(Teacher t) {  
    return teachers.contains(t);  
}  
public void setPresident(Teacher t) {  
    this.president = t;  
}  
public Teacher getPresident() {  
    return president;  
}  
public void printTeachers() {  
    Iterator<Teacher> iterator = teachers.iterator();  
    while (iterator.hasNext()) {  
        Teacher t = iterator.next();  
        System.out.println(t.getName());  
    }  
}
```



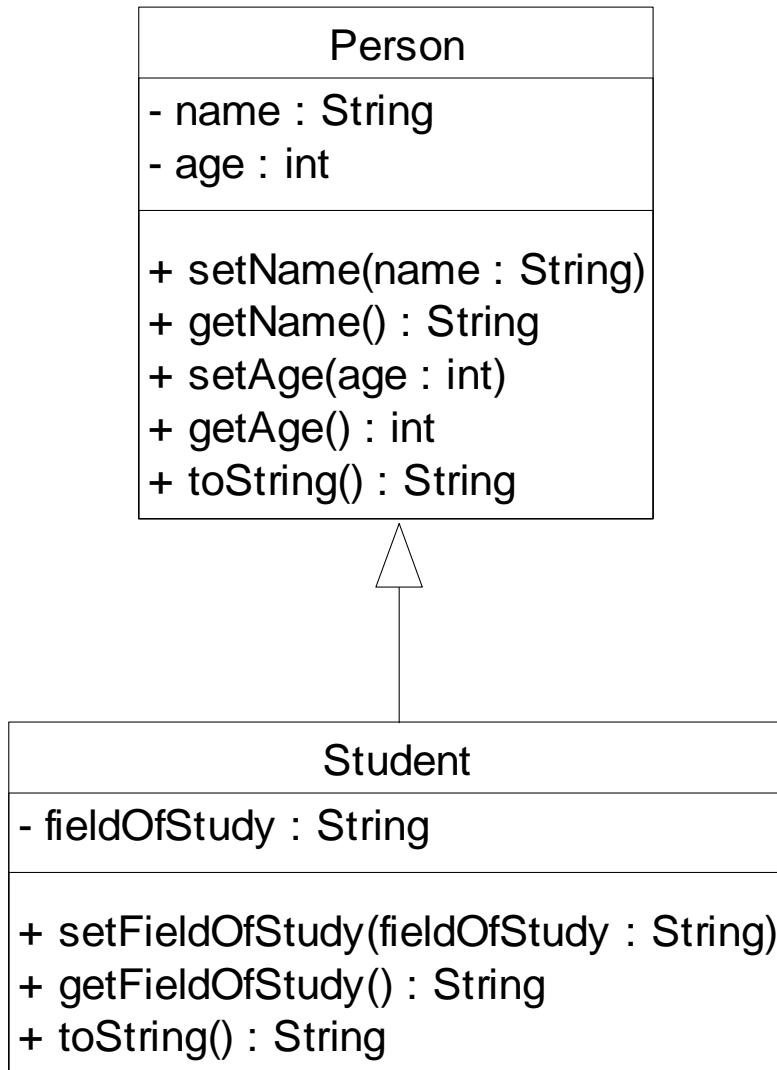
Μέθοδος Ελέγχου (main)

```
public static void main(String args[]) {  
    Teacher george = new Teacher("Γιώργος  
Κακαρόντζας");  
    Teacher george1 = new Teacher("Γιώργος  
Κακαρόντζας");  
    Department dept = new Department("Τμήμα XYZ");  
    dept.addTeacher(george);  
    //αυτήν η εισαγωγή θα αποτύχει!  
    boolean ok = dept.addTeacher(george1);  
    if (!ok) {  
        System.out.println("Η εισαγωγή απέτυχε");  
    }  
    System.out.println("Εκπαιδευτικοί:");  
    dept.printTeachers();  
    dept.setPresident(george);  
    System.out.println("Πρόεδρος:  
    "+dept.getPresident().getName());  
}
```



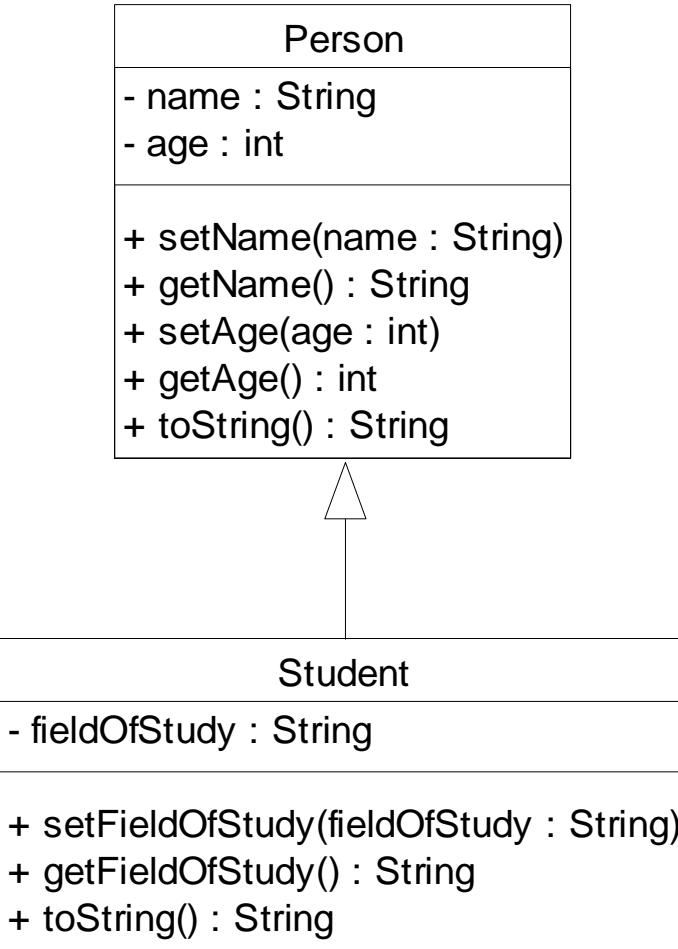
Συσχέτισης Γενίκευσης

- Η γενίκευση είναι μία ειδική μορφή συσχέτισης, κατά την οποία μία γενική κλάση αποτελεί τη βάση για τη δήλωση μίας ή περισσοτέρων ειδικότερων, υπό κάποια έννοια, κλάσεων.
- Η γενική κλάση ονομάζεται υπερκλάση και οι ειδικές κλάσεις ονομάζονται υποκλάσεις.
- **Το σύμβολο της γενίκευσης είναι ένα βέλος που δείχνει από την ειδική στη γενική κλάση.**



Παράδειγμα Γενίκευσης στη Java

- Οι κλάσεις «Person» και «Student» στην Java δίνονται στη Λίστα Κώδικα που ακολουθεί.
- Παρατηρήστε ότι η κλάση «Student» δηλώνεται ως υποκλάση της κλάσης «Person» με τη χρήση της φράσης *extends* στη δήλωσή της, επίσης, η κλάση «Student» υπερβαίνει τη μέθοδο *toString* της υπερκλάσης της δηλώνοντας τη μέθοδο *toString* εκ νέου.



Η Κλάση Person

```
public class Person {  
    protected String name;  
    private int age;  
    public Person() {}  
    public void setName(String name) {this.name = name;  
    }  
    public void setAge(int age) {this.age = age;}  
    public String getName() {return name;}  
    public int getAge() {return age;}  
    public String toString() {  
        return "Είμαι ο " + name + " και είμαι " + age + "  
ετών.>";  
    }  
}
```



Η Κλάση Student

```
public class Student extends Person {  
    private String fieldOfStudy;  
    public void setFieldOfStudy(String fieldOfStudy) {  
        this.fieldOfStudy = fieldOfStudy;  
    }  
    public String getField() {  
        return fieldOfStudy;  
    }  
    // Νέα δήλωση (υπέρβαση) της μεθόδου toString ειδικά για  
    // τους φοιτητές  
    public String toString() {  
        return "Είμαι ο " + name +  
            " και είμαι " + getAge() + " ετών. " + "Σπουδάζω " +  
            fieldOfStudy;  
    }  
}
```



Ανάλυση Πληροφοριακών Συστημάτων

Εαρινό Εξάμηνο 2020-2021

Lec07 – 11/05/2021

Διδάσκων: Γεώργιος Χρ. Μακρής

Διάγραμμα Περιπτώσεων Χρήσης (Use Case Diagram)

Τύποι Διαγραμμάτων της UML

- **Διάγραμμα Περιπτώσεων Χρήσης** (Use Case Diagram)
- **Διάγραμμα Κλάσεων** (Class Diagram)
- **Διαγράμματα Συμπεριφοράς** (Behavior Diagrams):
 - Διάγραμμα Καταστάσεων (Statechart Diagram)
 - Διάγραμμα Δραστηριοτήτων (Activity Diagram)
 - Διαγράμματα Αλληλεπίδρασης
 - Διάγραμμα Ακολουθίας (Sequence Diagram)
 - Διάγραμμα Συνεργασίας (Collaboration Diagram)
- **Διαγράμματα Υλοποίησης** (Implementation Diagrams):
 - Διάγραμμα Συστατικών (Component Diagram)
 - Διάγραμμα Διάταξης (Deployment Diagram)



■ Διαγράμματα Περιπτώσεων Χρήσης - Use Case Diagrams:

- Δείχνουν σχέσεις μεταξύ χρηστών και περιπτώσεων χρήσης και παρουσιάζουν την λειτουργικότητα του συστήματος. Είναι βασικός τύπος διαγράμματος της γλώσσας UML.

■ Διαγράμματα Δραστηριοτήτων – Activity Diagrams:

- Τα διαγράμματα δραστηριοτήτων χρησιμοποιούνται για την περιγραφή των επιχειρηματικών διαδικασιών, των ροών εργασιών, τον έλεγχο της ροής των αντικειμένων κ.λπ. Είναι βασικός τύπος διαγράμματος της γλώσσας UML.

■ Διαγράμματα Κλάσεων- Class Diagrams:

- Περιγράφουν τις κλάσεις καθώς και τις στατικές σχέσεις τους. Είναι βασικός τύπος διαγράμματος της γλώσσας UML.

■ Διαγράμματα Αντικειμένων – Object Diagrams:

- Περιγράφουν τα αντικείμενα του συστήματος και την αλληλεπίδρασή τους μια δεδομένη χρονική στιγμή.

■ Διαγράμματα Ακολουθίας - Sequence Diagrams:

- Περιγράφει την χρονική ακολουθία των αντικειμένων που συμμετέχουν στην συναλλαγή. Είναι βασικός τύπος διαγράμματος της γλώσσας UML.

■ Διαγράμματα Συνεργασίας – Collaboration Diagrams

- Παρουσιάζει τον τρόπο με τον οποίο συνεργάζονται τα αντικείμενα

■ Διαγράμματα Καταστάσεων – State Diagram

- Παρουσιάζει την εσωτερική κατάσταση ενός αντικειμένου καθώς και τις μεταβάσεις καταστάσεων.

■ Διαγράμματα «Συστατικών»- Component Diagrams

- Τα διαγράμματα αυτά παρουσιάζουν την δομή του κώδικα του συστήματος. Ένα component μπορεί να είναι ένα DLL, EXE, Java .class ή JAR αρχείο.

■ Διαγράμματα «Διάταξης» - Deployment Diagrams

- Τα διαγράμματα αυτά παρουσιάζουν τον τρόπο με το οποίο κατανέμονται οι διαδικασίες επεξεργαστικά στοιχεία.

Ποια διαγράμματα να χρησιμοποιήσω...

- Η UML είναι γλώσσα και μπορείς να περιγράψεις ένα σύστημα με πολλούς τρόπους
- Η UML δεν ορίζει πια διαδικασία να ακολουθήσουμε
- Μερικά από τα διαγράμματα είναι ισοδύναμα
- Τα διαγράμματα περιγράφουν το σύστημα από διαφορετικές οπτικές γωνίες (views)
- Η γλώσσα είναι δυναμική και σου επιτρέπει να χρησιμοποιήσεις – φτιάξεις την δική σου διάλεκτο



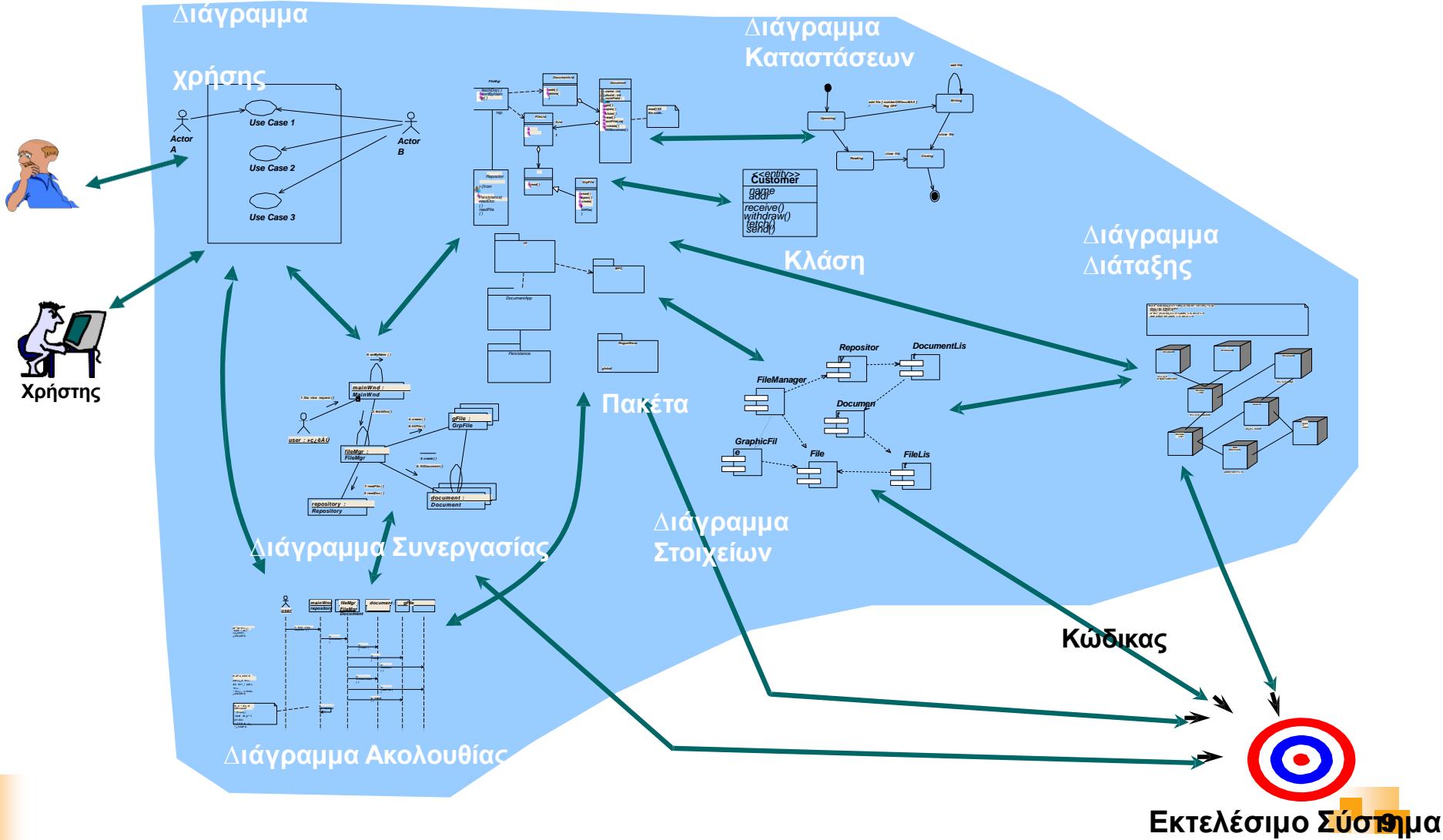
Τα διαγράμματα είναι όψεις ενός μοντέλου



ένα μοντέλο είναι μια πλήρης περιγραφή του συστήματος από μια οπτική γωνία



Μοντελοποίηση με UML διαγράμματα



Η λεκτική περιγραφή μιας περίπτωσης χρήσης

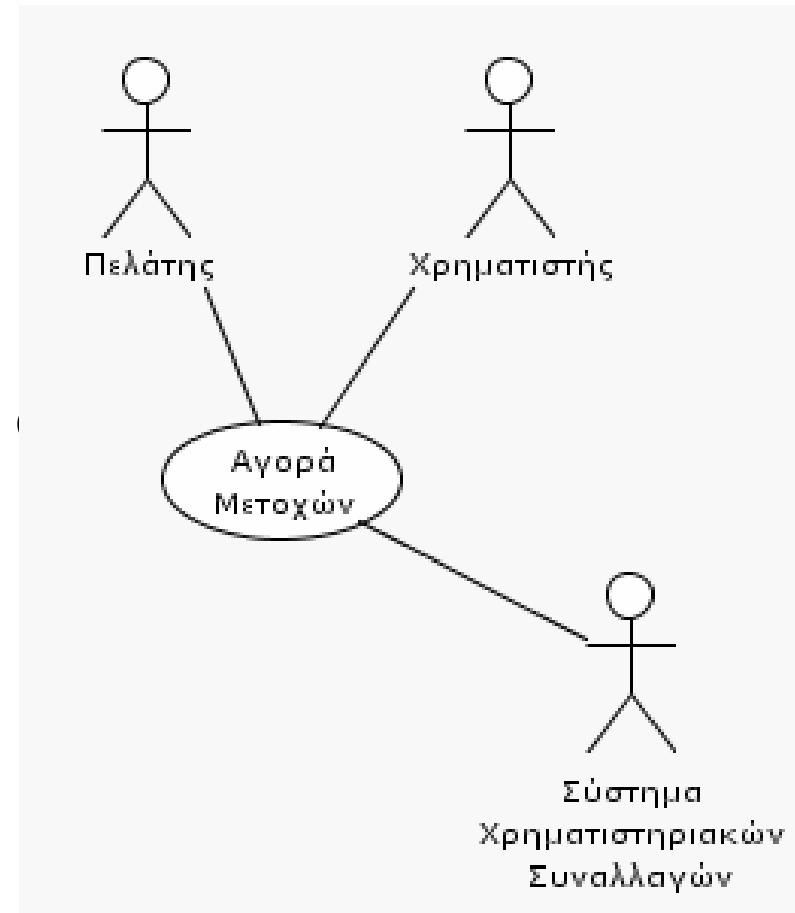
- Η λεκτική περιγραφή της περίπτωσης χρήσης είναι εκεί που περιγράφονται οι λεπτομέρειες που απαιτούνται
- Η γραφική αναπαράσταση δίνει **μόνο την γενική εικόνα** του συστήματος
- Υπάρχουν διάφορα πρότυπα λεκτικής περιγραφής περιπτώσεων χρήσης
 - Απλή περιγραφή
 - Σε μορφή πίνακα
 - Αναλυτική περιγραφή
 - **To δικό σας πρότυπο**

Η λεκτική περιγραφή μιας περίπτωσης χρήσης - παράδειγμα

Τίτλος: Αγορά Μετοχών

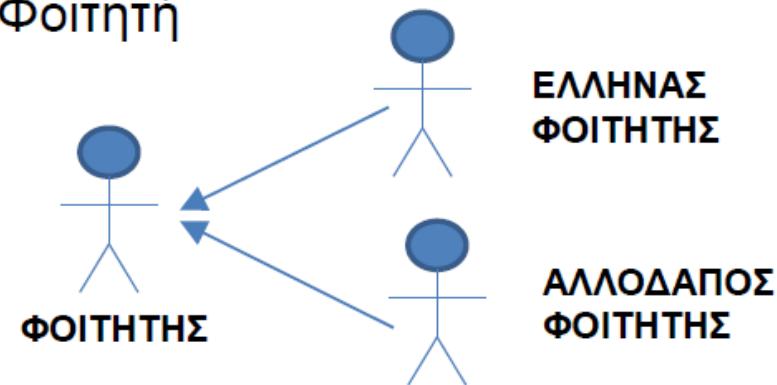
Βασική ροή:

- 1. Εισαγωγή όνομα πελάτη και λογαριασμού τράπεζας**
- 2. Εισαγωγή αριθμού μετοχών και ID μετοχής**
- 3. Καθορισμός τιμής**
- 4. Έλεγχος υπολοίπου λογαριασμού**
- 5. Προετοιμασία εντολής αγοράς**
- 6. Λήψη επιβεβαίωσης και αποθήκευση αριθμού συναλλαγής**



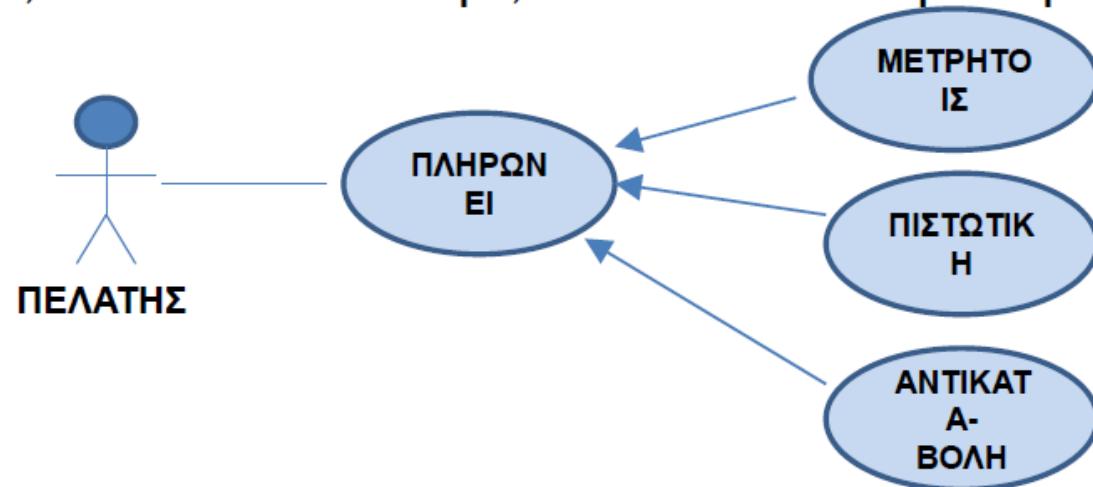
Γενίκευση μεταξύ Χρηστών

- Δύο χρήστες μπορεί να συνδέονται με συσχέτιση γενίκευσης/ εξειδίκευσης
 - Παράδειγμα: φοιτητής: έλληνας φοιτητής – αλλοδαπός φοιτητής
- Αυτό γίνεται: α) για λόγους απλοποίησης και β) γιατί διαφορετικοί χρήστες συμμετέχουν σε διαφορετικές περιπτώσεις χρήσης
 - Παράδειγμα: οι αλλοδαποί φοιτητές δεν συμμετέχουν στην περίπτωση χρήσης «Λήψη επιδόματος»
 - Για τις περιπτώσεις χρήσης που συμμετέχουν όλοι οι Φοιτητές, η σύνδεση γίνεται με τον Φοιτητή



Γενίκευση μεταξύ Περιπτώσεων Χρήσης

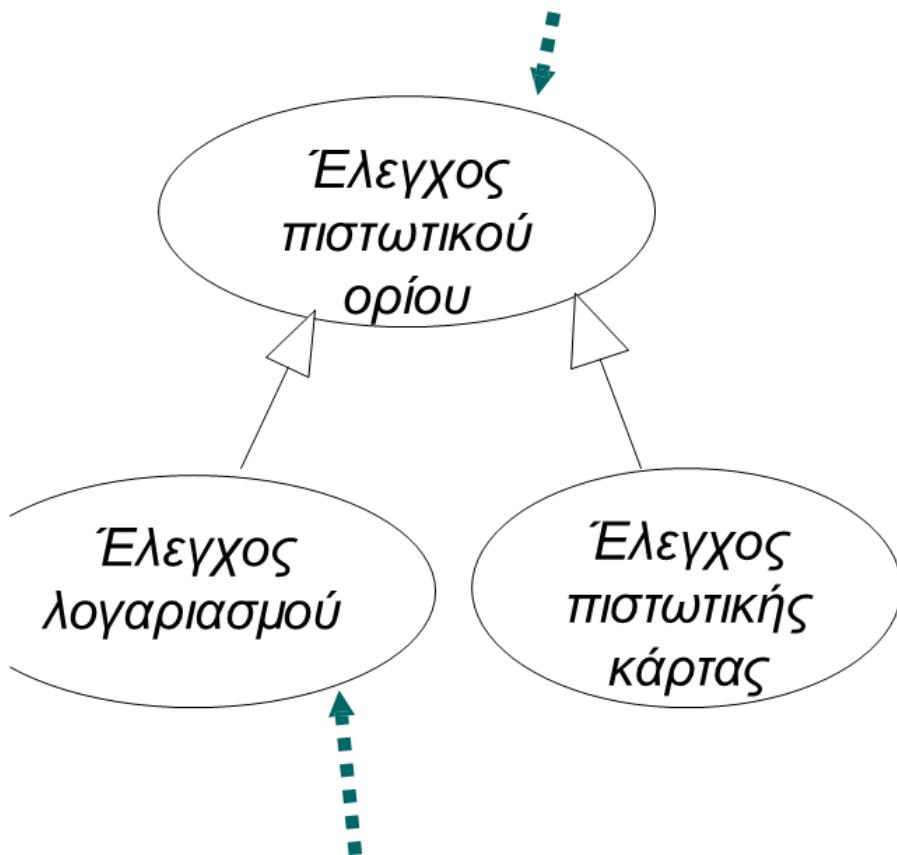
- Δύο Περιπτώσεις χρήσης μπορεί να συνδέονται με σχέση γενίκευσης/ εξειδίκευσης
- Η περίπτωση χρήσης «γονέας» κληρονομεί τη συμπεριφορά και τις σχέσεις της περίπτωσης χρήσης «παιδί».
 - Παράδειγμα: η Πληρωμή μπορεί να είναι «Μετρητοίς», «Με Πιστωτική», «Με Αντικαταβολή»



Πως οργανώνονται οι περιπτώσεις χρήσης

Συσχέτιση γενίκευσης (generalisation)

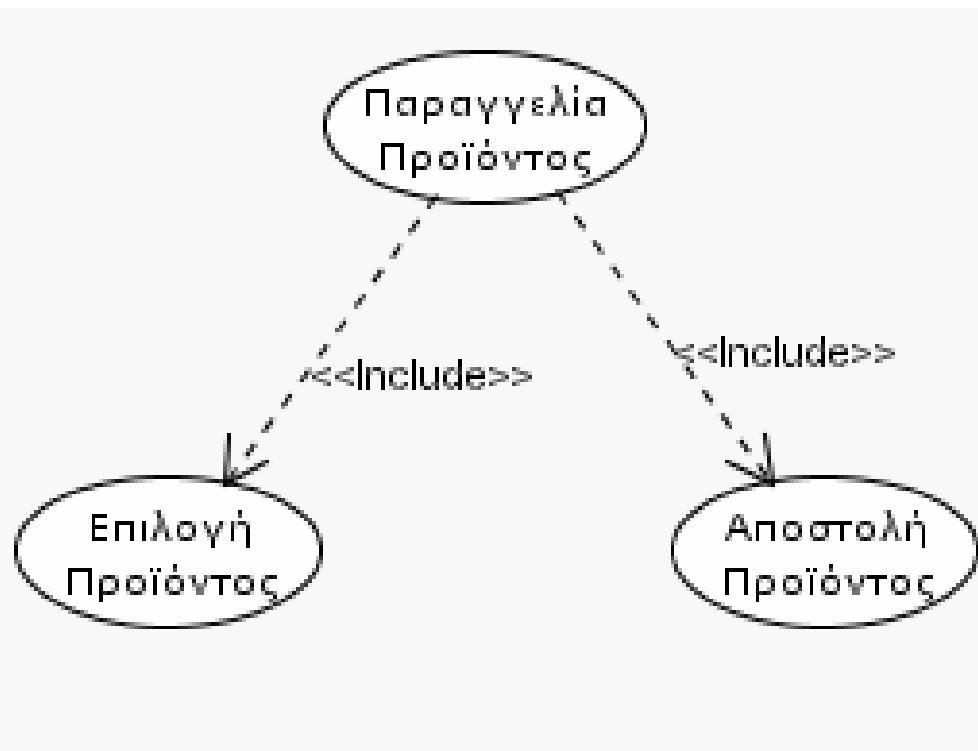
περίπτωση χρήσης - πατέρας



περίπτωση χρήσης - παιδί

- **Η περίπτωση χρήσης παιδί κληρονομεί την συμπεριφορά και το νόημα από την περίπτωση χρήσης πατέρας**
- **Η περίπτωση χρήσης παιδί προσθέτει στην συμπεριφορά του πατέρα**
- **Η περίπτωση χρήσης παιδί μπορεί να αντικαθιστά την περίπτωση χρήσης πατέρας**

Πως οργανώνονται οι περιπτώσεις χρήσης Συσχέτιση <<include>> (περιλαμβάνει)



- **Χρησιμοποιείται για την αποφυγή επανάληψης στην περιγραφή κοινών σεναρίων.**
- **Αποφεύγουμε copy & paste τμημάτων περιπτώσεων χρήσης**

Παράδειγμα “Include”

Τίτλος: Παραγγελία προϊόντος

**Κατάσταση Εισόδου: Ο υπάλληλος έχει κάνει
login**

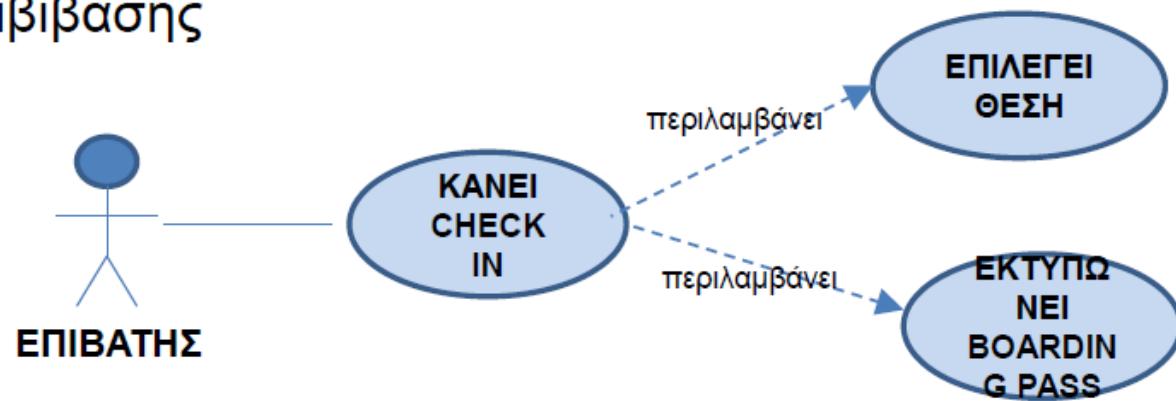
Βασική ροή:

- 1. *Include (Επιλογή προϊόντος)***
- 2. *Πληρωμή προϊόντος***
- 3. *Include (Αποστολή προϊόντος)***
- 4. *Ενημέρωση πελάτη***

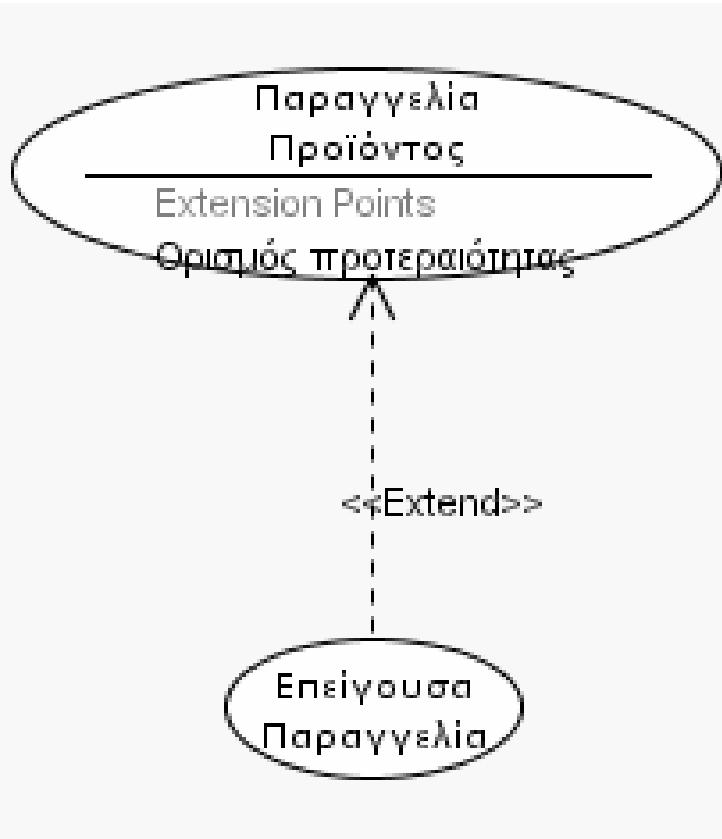
Σχέση «Περιλαμβάνει» μεταξύ περιπτώσεων χρήσης

Α «περιλαμβάνει» Β σημαίνει ότι

- η Β είναι μια υπολειτουργία της Α
- η Β είναι απαραίτητο να γίνει για τη λειτουργικότητα της Α
- η Β συνήθως δεν μπορεί να σταθεί μόνη της ως λειτουργία, αλλά είναι μέρος της Α
- Παράδειγμα:
 - Η ανάληψη μετρητών προϋποθέτει τον έλεγχος υπολοίπου
 - το check in περιλαμβάνει την επιλογή θέσης και την εκτύπωση της κάρτας επιβίβασης



Πως οργανώνονται οι περιπτώσεις χρήσης Συσχέτιση <<extend>> (επεκτείνει)



- Επιτρέπει να περιγράψουμε το σημείο της περίπτωσης χρήσης που ένας χρήστης βλέπει ως προαιρετικό
- Επιτρέπει να περιγράψουμε εναλλακτικές ροές που εξαρτώνται από συνθήκες
- Επιτρέπει να εισάγουμε εναλλακτικές ροές στα σημεία που εξαρτώνται από την αλληλεπίδραση του χρήστη.
- Χρειάζεται να ορίσουμε τα σημεία επέκτασης (extension points)

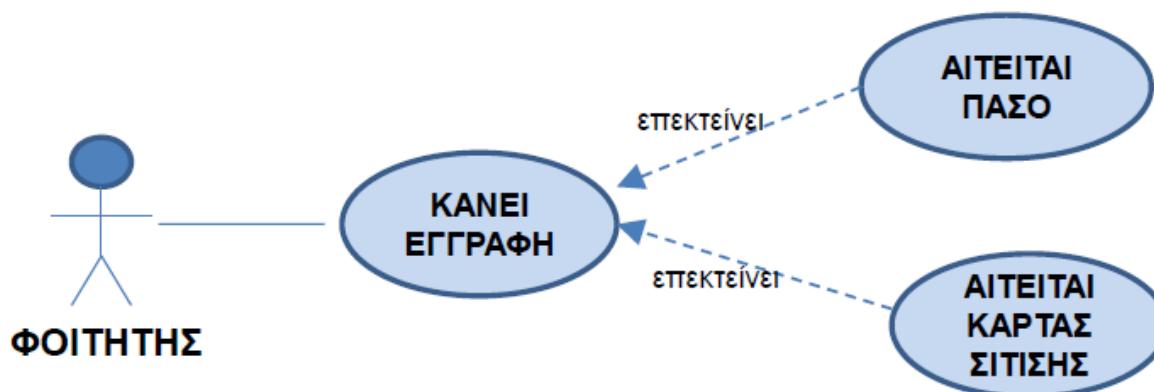
Χρησιμοποιώντας την συσχέτιση “Extends”

- Επιτρέπει να καταγράψουμε την βασική περίπτωση χρήσης
- Εαν επιθυμούμε μπορούμε να τροποποιήσουμε ή να διαγράψουμε την επέκταση χωρίς να χρειαστεί να τροποποιήσουμε τη βασική ροή.
- Σε κάθε βήμα ρωτάμε
 - τι μπορεί να πάει λάθος;
 - Πως μπορεί να γίνει διαφορετικά;

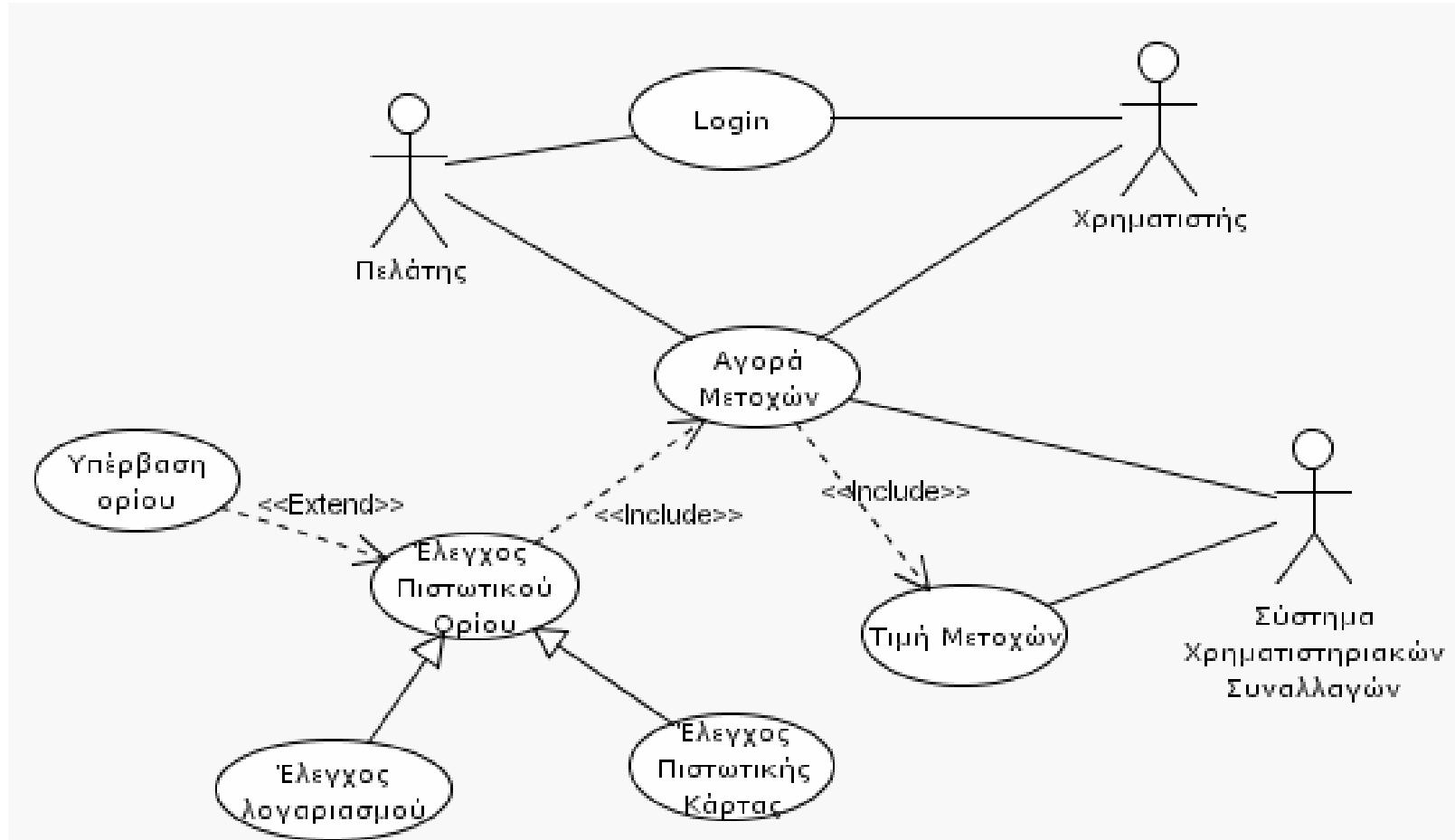
Σχέση «Επεκτείνει» μεταξύ περιπτώσεων χρήσης

B «επεκτείνει» A σημαίνει ότι η

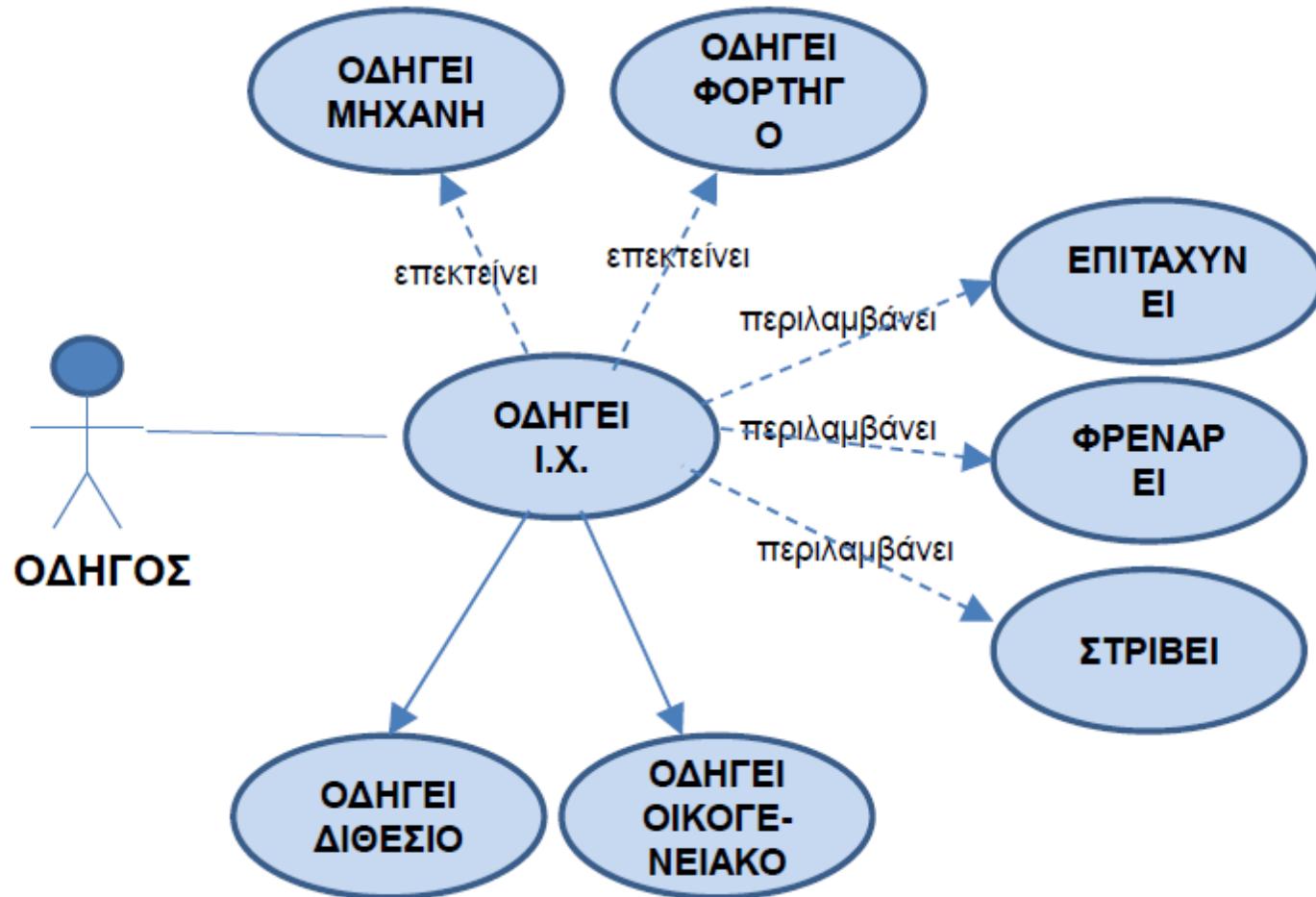
- Η B είναι μία εξειδικευμένη περίπτωση της A που περιλαμβάνει τη γενική λειτουργικότητα της A και προσθέτει επιπλέον λειτουργικότητα
- Η B προσθέτει/ επεκτείνει τη λειτουργικότητα της A με επιπλέον χαρακτηριστικά
- Η B μπορεί να είναι προαιρετική σε σχέση με την A



Παράδειγμα (01) διαγράμματος περιπτώσεων χρήσης



Παράδειγμα (02) διαγράμματος περιπτώσεων χρήσης



Kανόνες για το σωστό ορισμό των περιπτώσεων χρήσης [1/3]

- Οι περιπτώσεις χρήσης του συστήματος πρέπει να είναι ευκολοδιάβαστες. Πρέπει να είμαστε σύντομοι, ειδικά όταν έχουμε μεγάλο αριθμό περιπτώσεων χρήσης.
 - Ο τίτλος των περιπτώσεων χρήσης πρέπει να είναι σαφής, σύντομος και ξεκάθαρος.
 - Χρησιμοποιείστε ρήματα ενεργητικής φωνής όσο είναι δυνατόν.
 - Οι χειριστές πρέπει να εμφανίζονται σε κάθε βήμα (ποιος κάνει τι).
 - Ξεκινήστε με το γεγονός που σηματοδοτεί την έναρξη της περίπτωση χρήσης και καταλήξτε στην επίτευξη του σκοπού (επιτυχία).
 - Περιγράψτε τις συνθήκες (if) σαν εναλλακτικά σενάρια.
 - Περιγράψτε με διακριτό τρόπο σαν εναλλακτικά σενάρια, τις περιπτώσεις αποτυχίας της περίπτωσης χρήσης.

Kανόνες για το σωστό ορισμό των περιπτώσεων χρήσης [2/3]

- Η συγγραφή των περιπτώσεων χρήσης είναι ισοδύναμη με τη **συγγραφή μιας έκθεσης**.
- Ιδιαίτερη σημασία λοιπόν πρέπει να δοθεί στη **λεκτική περιγραφή** των περιπτώσεων χρήσης και λιγότερη στη **γραφική τους αναπαράσταση** με τη χρήση διαγραμμάτων.
- Κάθε βήμα του σεναρίου να είναι μια απλή σύντομη πρόταση σε ενεργητική φωνή.
- Προτιμήστε πολλές μικρές προτάσεις αντί για μια μεγάλη.
- Χρησιμοποιείστε υπο-περιπτώσεις χρήσης όπου μπορείτε για να απλοποιήσετε την περιγραφή
- Ο αριθμός των βημάτων της βασικής ροής μιας περίπτωσης χρήσης δε μπορεί να είναι μεγάλος. Ο ιδανικός αριθμός είναι μικρότερος του 10.
- Μια περίπτωση χρήσης περιγράφει το τι μπορεί να κάνει ένας χειριστής μέσα σε λίγα λεπτά (2-20 λεπτά).

Kανόνες για το σωστό ορισμό των περιπτώσεων χρήσης [3/3]

- Οι περιπτώσεις χρήσης περιγράφουν την πρόθεση του χειριστή να επιτύχει το σκοπό του και όχι τις κινήσεις που πρέπει να κάνει αυτός στο user interface.
- Οι περιπτώσεις χρήσης δεν περιγράφουν το user interface αλλά την πρόθεση του χειριστή. Αυτό ισχύει γιατί αν περιγράψουμε το user interface.
 - Το έγγραφο των περιπτώσεων χρήσης γίνεται πάρα πολύ μεγάλο.
 - Οι αλλαγές στο user interface είναι συχνές, το οποίο σημαίνει ότι θα πρέπει να αλλάζουμε/συντηρούμε το έγγραφο των περιπτώσεων χρήσης συχνά
 - Ο σχεδιασμός του user interface ακολουθεί γενικότερους κανόνες και είναι πρόβλημα σχεδίασης και όχι καταγραφής απαιτήσεων
- Κάθε περίπτωση χρήσης μπορεί να τελειώσει τουλάχιστον με δύο τρόπους: την επιτυχία και την αποτυχία.



Δημιουργία Διαγραμμάτων Χρήσης

- Συλλογή πληροφοριών
- Εντοπισμός εξωτερικών χρηστών: ποιοι πελάτες να χρησιμοποιούν τα προϊόντα και τις υπηρεσίες του συστήματος;
- Εντοπισμός περιπτώσεων χρήσης προς τους εξωτερικούς πελάτες: ποιες χρήσεις του συστήματος κάνουν οι πελάτες;
- Εντοπισμός εσωτερικών χρηστών: ποιοι χειριστές λειτουργούν το σύστημα;
- Εντοπισμός περιπτώσεων χρήσης προς τους εσωτερικούς πελάτες: σε ποιες χρήσεις του συστήματος συμμετέχουν οι χειριστές του συστήματος;
- Συσχέτιση χρηστών και περιπτώσεων χρήσης
- Αναζήτηση επιπλέον χρηστών και περιπτώσεων χρήσης
- Επαλήθευση του μοντέλου

Συλλογή Πληροφοριών [1/2]

Συλλέγουμε πληροφορίες από:

- Διαχειριστές του συστήματος
- Τους χειριστές που εμπλέκονται στην λειτουργία του συστήματος
- Τους χρήστες ομοειδών ή συναφών συστημάτων
- Πελάτες
- Συνεργάτες
- Ειδικούς του χώρου

Συλλογή Πληροφοριών [2/2]

Χρήσιμες πληροφορίες μπορεί να προέρχονται ακόμη:

- Παρατηρώντας τους εργαζόμενους κατά την εργασία τους
- Συμμετέχοντας στις επιχειρηματικές διαδικασίες που ερευνώνται
- Κάνοντας τον πελάτη
- Κάνοντας έρευνα
- Με συνεντεύξεις
- Συζητώντας με ειδικούς του χώρου
- Εξετάζοντας έγγραφα, προδιαγραφές και εγχειρίδια εργασίας, περιγραφές καθηκόντων
- Εξετάζοντας οργανογράμματα

Αναζήτηση Χρηστών [1/2]

- Πιθανοί χρήστες μπορούν να βρεθούν απαντώντας στις ακόλουθες ερωτήσεις:
- Ποιοι είναι οι πελάτες του συστήματος;
- Ποιοι είναι οι χειριστές και διαχειριστές του συστήματος;
- Ποιοι άλλοι χρησιμοποιούν τα αποτελέσματα του συστήματος;
- Ποιοι είναι οι συνεργάτες του συστήματος;
- Ποια άλλα συστήματα και πόροι χρησιμοποιούνται κατά τη λειτουργία του συστήματος;

Αναζήτηση Χρηστών [2/2]

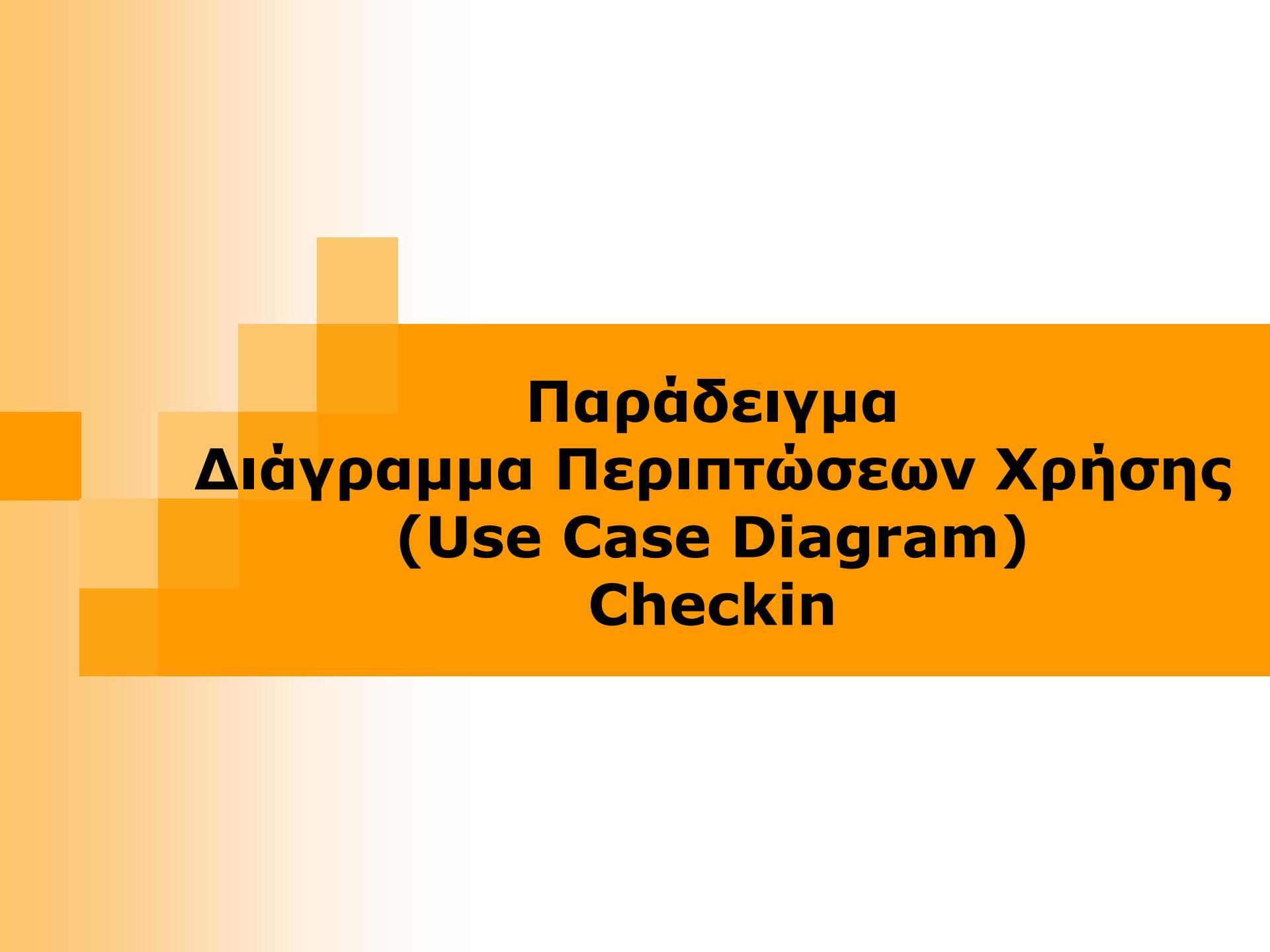
- Φοιτητές
- Καθηγητές
- Διοικητικό προσωπικό
- Προσωπικό βιβλιοθήκης
- ...
- Αρκεί να έχει σχέση και σημασία για το «σύστημα» που αναλύουμε

Αναζήτηση Περιπτώσεων Χρήσης [1/2]

- Ποια προϊόντα ή υπηρεσίες χρησιμοποιούνται από τον πελάτη;
- Ποια προϊόντα ή υπηρεσίες παρέχονται από το σύστημα;
- Ποιες λειτουργίες και δραστηριότητες εκτελούν οι εργαζόμενοι;
- Ποια γεγονότα ενεργοποιούν τις λειτουργίες και δραστηριότητες;

Αναζήτηση Περιπτώσεων Χρήσης [2/2]

- Παρακολουθούν διαλέξεις
- Παρακολουθούν εργαστήρια
- Κάνουν εργασίες
- Σιτίζονται
- Διασκεδάζουν
- Αιτούνται έγγραφα
- Μελετούν



Παράδειγμα Διάγραμμα Περιπτώσεων Χρήσης (Use Case Diagram) Checkin

Παράδειγμα: Check in

Κάντε web check-in από τον υπολογιστή σας ή μέσω του ταξιδιωτικού σας γραφείου, από 48 ώρες έως και 30 λεπτά πριν την ώρα αναχώρησης της πτήσης σας.

Επιλέξτε τη θέση σας στο αεροσκάφος, τυπώστε την κάρτα επιβίβασής σας και είστε έτοιμοι προς επιβίβαση!

Η διαδικασία είναι απλή και γρήγορη

- Εισάγετε, είτε τον αριθμό του ηλεκτρονικού εισιτηρίου, είτε τον κωδικό της κράτησης σας και το επώνυμό σας, όπως αναφέρονται στο σύστημα της Aegean Airlines.
- Επιλέξτε την πτήση και τη θέση της προτίμησής σας μέσα από τη γραφική αναπαράσταση του αεροσκάφους.
- Λάβετε την κάρτα επιβίβασής σας με 4 τρόπους:
 - Εκτύπωση της κάρτας επιβίβασής σας
 - Λήψη της κάρτας επιβίβασής σας μέσω e-mail
 - Λήψη της κάρτας επιβίβασής σας στο κινητό σας τηλέφωνο
 - Παραλαβή της κάρτας επιβίβασής σας από ένα Self check-in Kiosk ή check-in counter στο αεροδρόμιο
- Παραδώστε τις αποσκευές σας στο αεροδρόμιο στα ειδικά Check-in counters.
- Εάν δεν μεταφέρετε αποσκευές κατευθυνθείτε στη έξοδο επιβίβασης της πτήσης σας.

Παράδειγμα: Check in

Κάντε web check-in από τον υπολογιστή σας ή μέσω του ταξιδιωτικού σας γραφείου, από 48 ώρες έως και 30 λεπτά πριν την ώρα αναχώρησης της πτήσης σας.

Επιλέξτε τη θέση σας στο αεροσκάφος, τυπώστε την κάρτα επιβίβασής σας και είστε έτοιμοι προς επιβίβαση!

Η διαδικασία είναι απλή και γρήγορη

- **Εισάγετε**, είτε τον αριθμό του ηλεκτρονικού εισιτηρίου, είτε τον κωδικό της κράτησης σας και το επώνυμο σας, όπως αναφέρονται στο σύστημα της Aegean Airlines.
- **Επιλέξτε την πτήση και τη θέση** της προτίμησής σας μέσα από τη γραφική αναπαράσταση του αεροσκάφους.
- **Λάβετε την κάρτα επιβίβασής σας** με 4 τρόπους:
 - **Εκτύπωση** της κάρτας επιβίβασής σας
 - **Λήψη** της κάρτας επιβίβασής σας μέσω e-mail
 - **Λήψη** της κάρτας επιβίβασής σας στο **κινητό σας τηλέφωνο**
 - **Παραλαβή** της κάρτας επιβίβασής σας από ένα **Self check-in Kiosk** ή **check-in counter** στο αεροδρόμιο
- **Παραδώστε τις αποσκευές** σας στο αεροδρόμιο στα ειδικά Check-in counters.
- Εάν δεν μεταφέρετε αποσκευές κατευθυνθείτε στη έξοδο επιβίβασης της πτήσης σας.

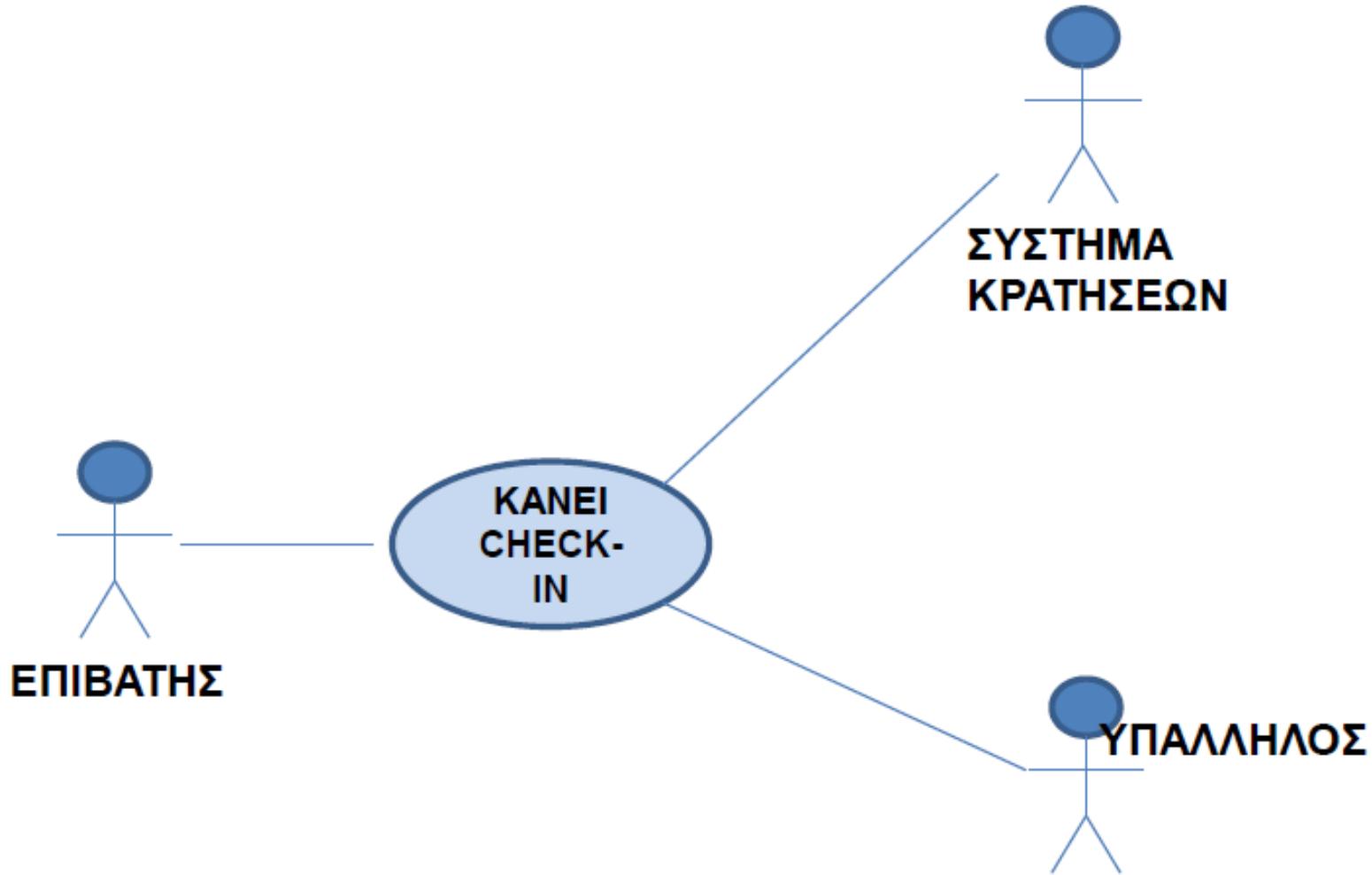
Παράδειγμα: Check in

- Παράδοση αποσκευών
 - Οι επιβάτες που κάνουν web ή mobile check-in και αναχωρούν από το Διεθνή Αερολιμένα Αθηνών, εξοικονομούν ακόμα περισσότερο χρόνο, καθώς παραδίδουν τις αποσκευές τους σε ειδικό counter παράδοσης αποσκευών (web & mobile baggage drop off).
 - Για τους επιβάτες που ταξιδεύουν με αποσκευές και έχουν πραγματοποιήσει τον έλεγχο του εισιτηρίου τους με έναν από τους παραπάνω τρόπους, το προσωπικό μας στο αεροδρόμιο είναι διαθέσιμο για την άμεση εξυπηρέτηση τους στα ειδικά Check-in counters (drop off baggage) προκειμένου να τις παραδώσουν.
 - Για τους επιβάτες που ταξιδεύουν Business Class ή είναι Gold Members του προγράμματος Miles+Bonus, θα συνεχίσουν να λειτουργούν ξεχωριστά check-in counters.
 - Ειδική μέριμνα θα υπάρχει για την εξυπηρέτηση των επιβατών μας με ειδικές ανάγκες, για όσους ταξιδεύουν με κατοικίδια και για τα ασυνόδευτα παιδιά.
- Στο αεροδρόμιο
 - Κατευθυνθείτε στην έξοδο επιβίβασης, το λιγότερο 30 λεπτά πριν την ώρα αναχώρησης της πτήσης σας, χωρίς να περάσετε από το check in counter. Στην έξοδο επιβίβασης θα πρέπει να προσκομίσετε την ταυτότητα ή το διαβατήριό σας, καθώς και την κάρτα επιβίβασης που θα έχετε εκτυπώσει μετά την ολοκλήρωση της διαδικασίας του ηλεκτρονικού check in.
 - Οι επιβάτες της Διακεκριμένης Θέσης πρέπει να έχουν εισέλθει στον Έλεγχο Ασφαλείας τουλάχιστον 20 λεπτά πριν την προγραμματισμένη ώρα αναχώρησης
 - Οι επιβάτες της Οικονομικής Θέσης πρέπει να έχουν εισέλθει στον Έλεγχο Ασφαλείας τουλάχιστον 30 λεπτά πριν την προγραμματισμένη ώρα αναχώρησης
- Το προσωπικό μας σε όλα τα αεροδρόμια θα είναι εκεί για να σας καλωσορίσει και διαθέσιμο για να φροντίσει όλες τις επιπλέον ανάγκες σας.

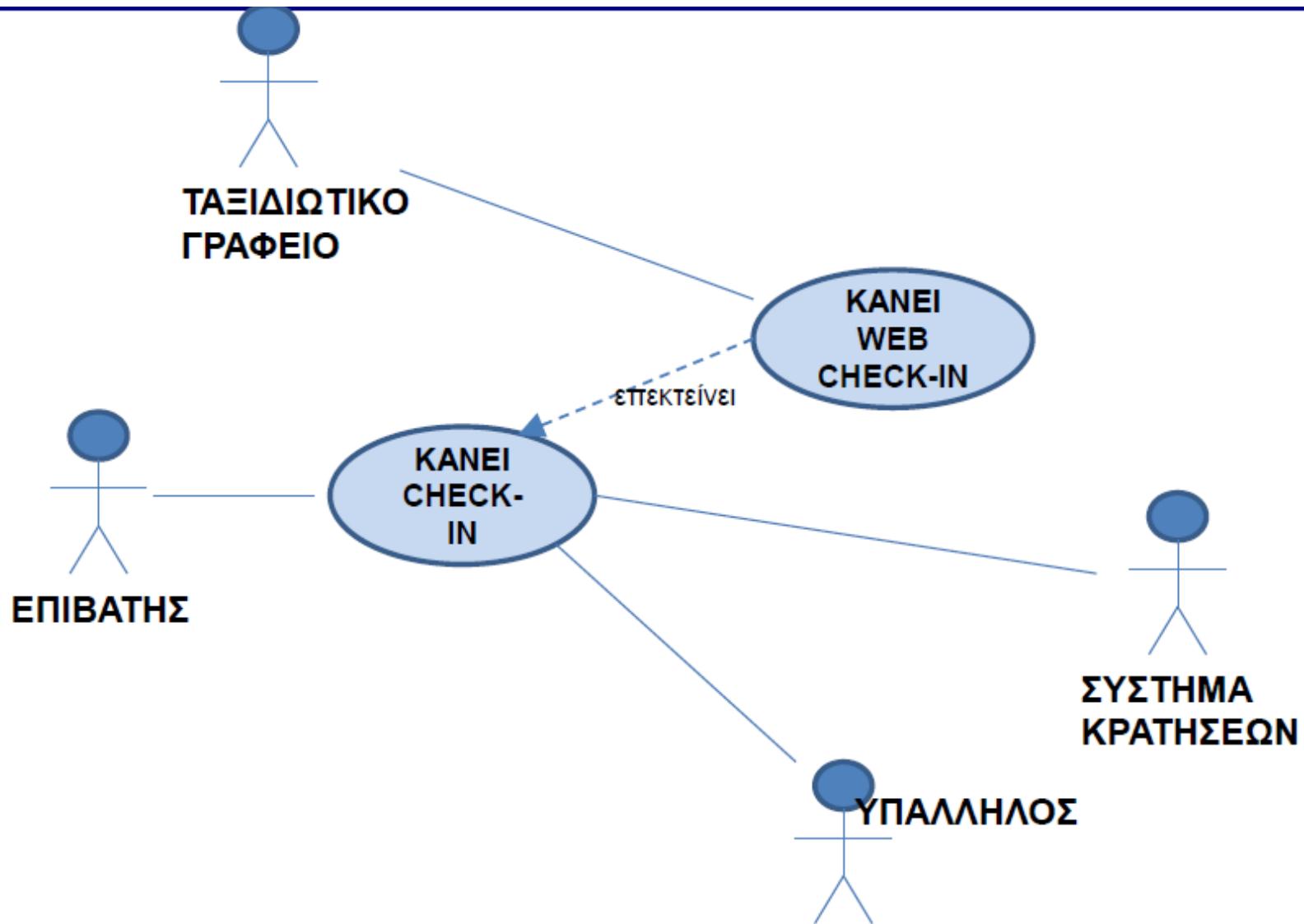
Παράδειγμα: Check in

- Παράδοση αποσκευών
 - Οι επιβάτες που κάνουν web ή mobile check-in και αναχωρούν από το Διεθνή Αερολιμένα Αθηνών, εξοικονομούν ακόμα περισσότερο χρόνο, καθώς παραδίδουν τις αποσκευές τους σε ειδικό counter παράδοσης αποσκευών (web & mobile baggage drop off).
 - Για τους επιβάτες που ταξιδεύουν με αποσκευές και έχουν πραγματοποιήσει τον έλεγχο του εισιτηρίου τους με έναν από τους παραπάνω τρόπους, το προσωπικό μας στο αεροδρόμιο είναι διαθέσιμο για την άμεση εξυπηρέτηση τους στα ειδικά Check-in counters (drop off baggage) προκειμένου να τις παραδώσουν.
 - Για τους επιβάτες που ταξιδεύουν Business Class ή είναι Gold Members του προγράμματος Miles+Bonus, θα συνεχίσουν να λειτουργούν ξεχωριστά check-in counters.
 - Ειδική μέριμνα θα υπάρχει για την εξυπηρέτηση των επιβατών μας με ειδικές ανάγκες, για όσους ταξιδεύουν με κατοικίδια και για τα ασυνόδευτα παιδιά.
- Στο αεροδρόμιο
 - Κατευθυνθείτε στην έξοδο επιβίβασης, το λιγότερο 30 λεπτά πριν την ώρα αναχώρησης της πτήσης σας, χωρίς να περάσετε από το check in counter. Στην έξοδο επιβίβασης θα πρέπει να προσκομίσετε την ταυτότητα ή το διαβατήριό σας, καθώς και την κάρτα επιβίβασης που θα έχετε εκτυπώσει μετά την ολοκλήρωση της διαδικασίας του ηλεκτρονικού check in.
 - Οι επιβάτες της Διακεκριμένης Θέσης πρέπει να έχουν εισέλθει στον Έλεγχο Ασφαλείας τουλάχιστον 20 λεπτά πριν την προγραμματισμένη ώρα αναχώρησης
 - Οι επιβάτες της Οικονομικής Θέσης πρέπει να έχουν εισέλθει στον Έλεγχο Ασφαλείας τουλάχιστον 30 λεπτά πριν την προγραμματισμένη ώρα αναχώρησης
- Το προσωπικό μας σε όλα τα αεροδρόμια θα είναι εκεί για να σας καλωσορίσει και διαθέσιμο για να φροντίσει όλες τις επιπλέον ανάγκες σας.

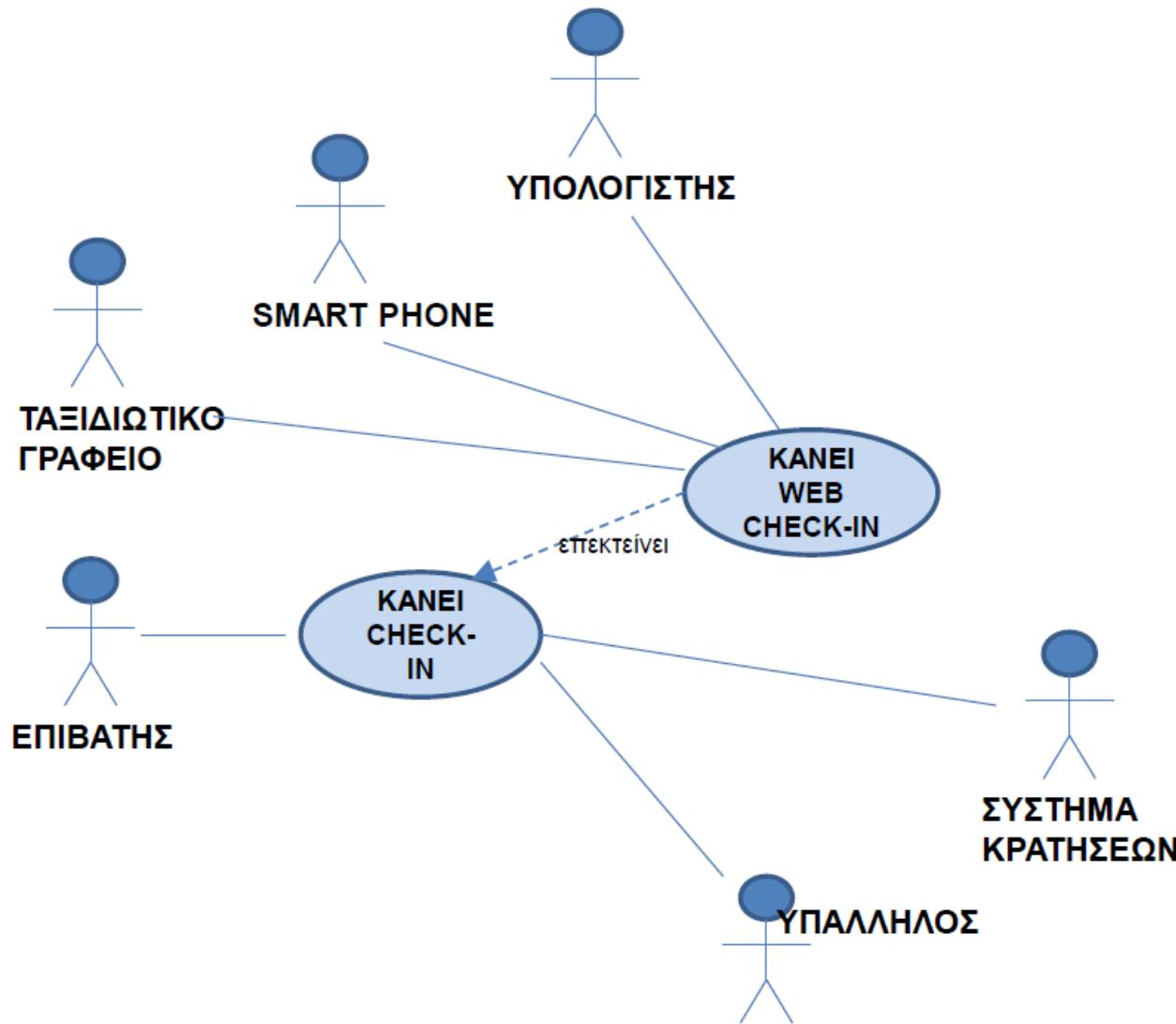
Παράδειγμα: Check in



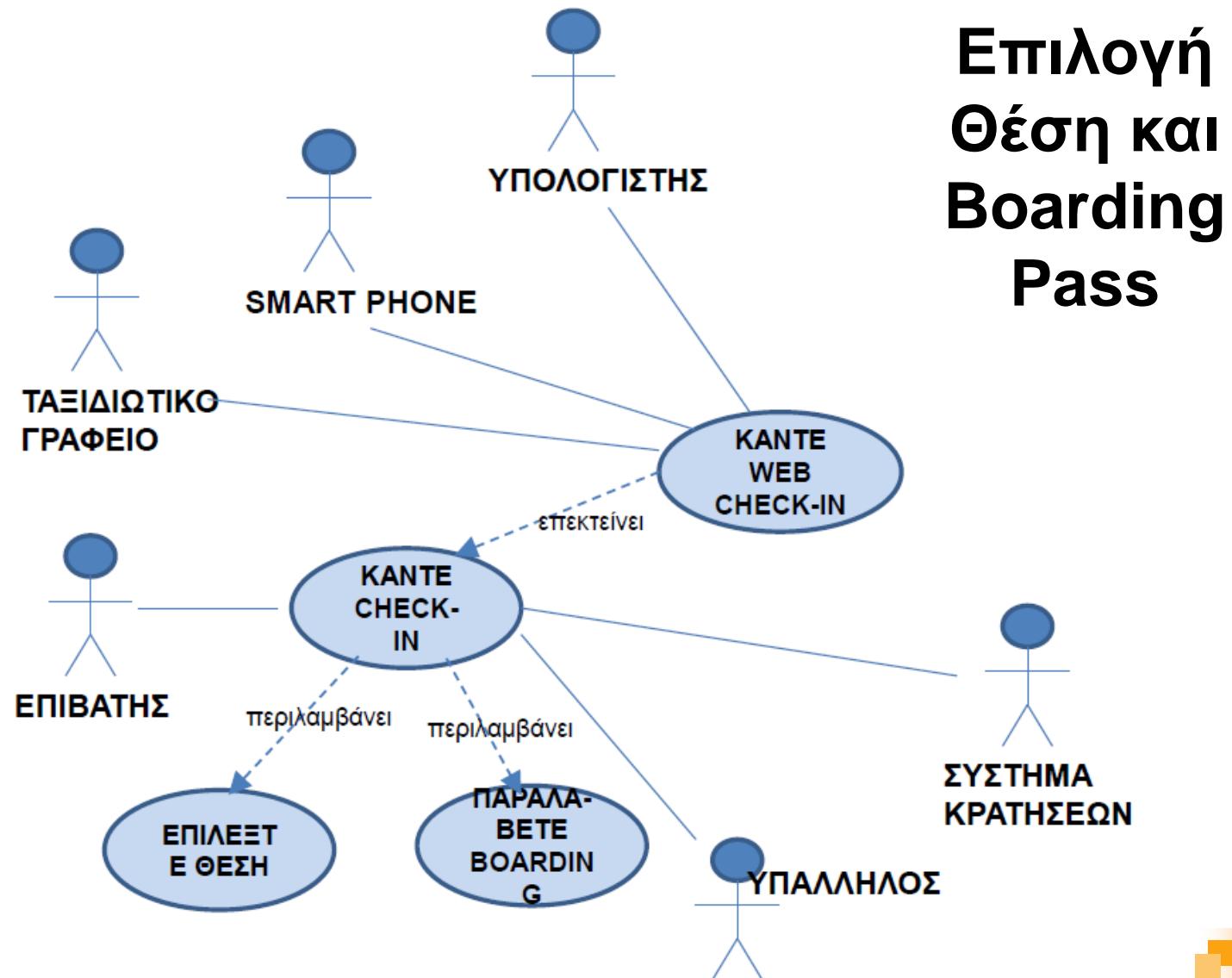
Παράδειγμα: Web Check in



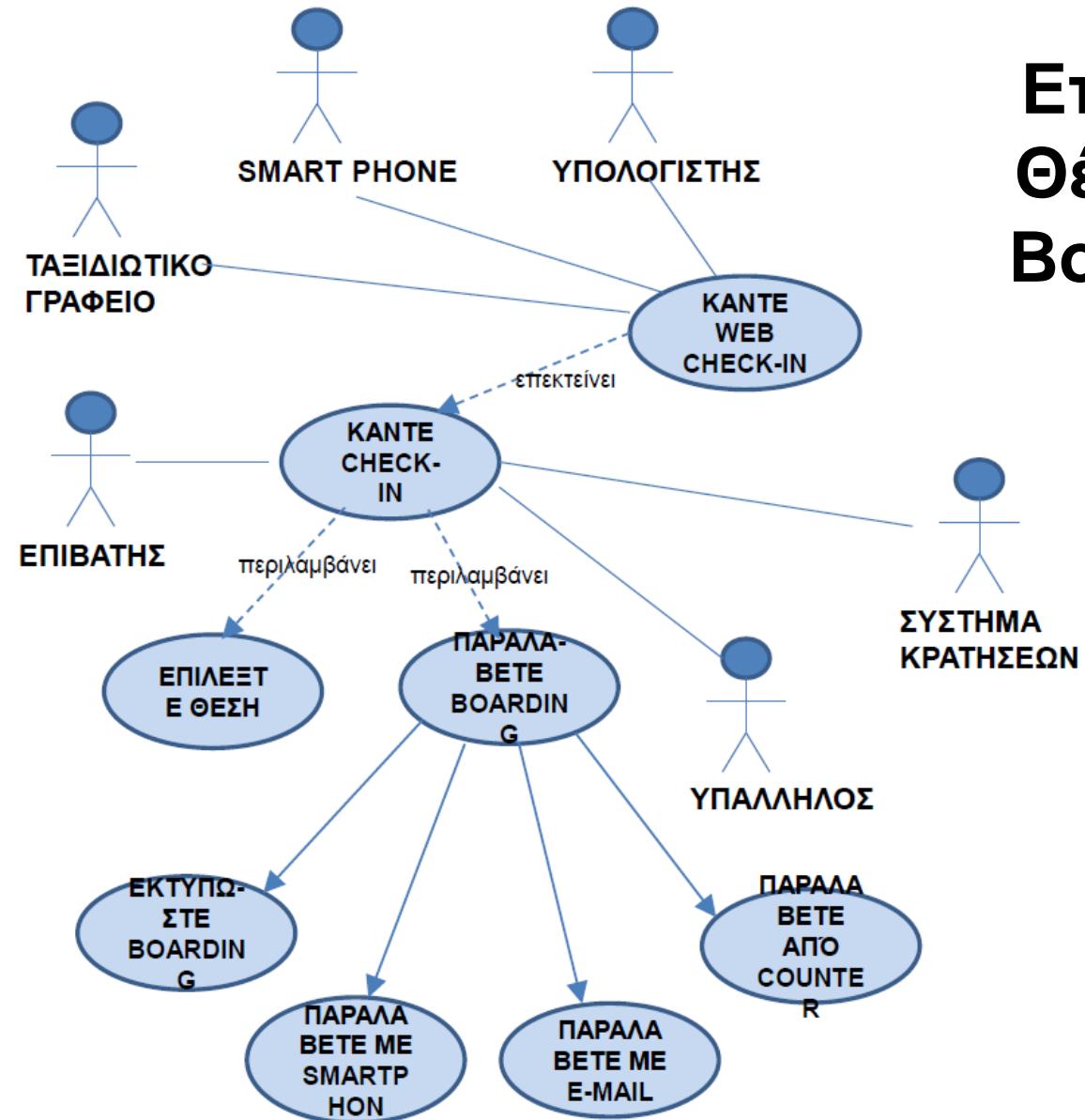
Παράδειγμα: Web Check in



Παράδειγμα: Web Check in

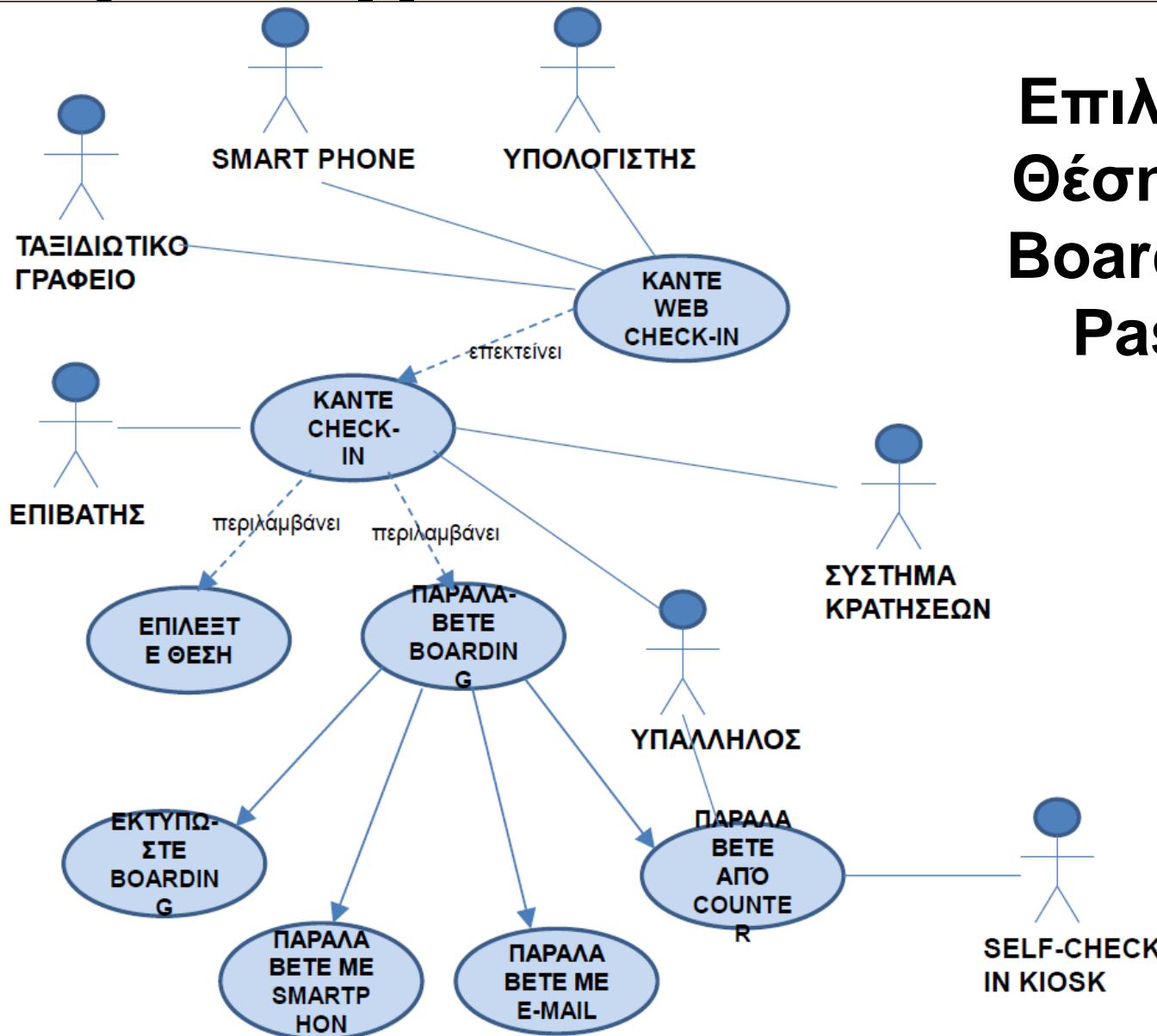


Παράδειγμα: Web Check in



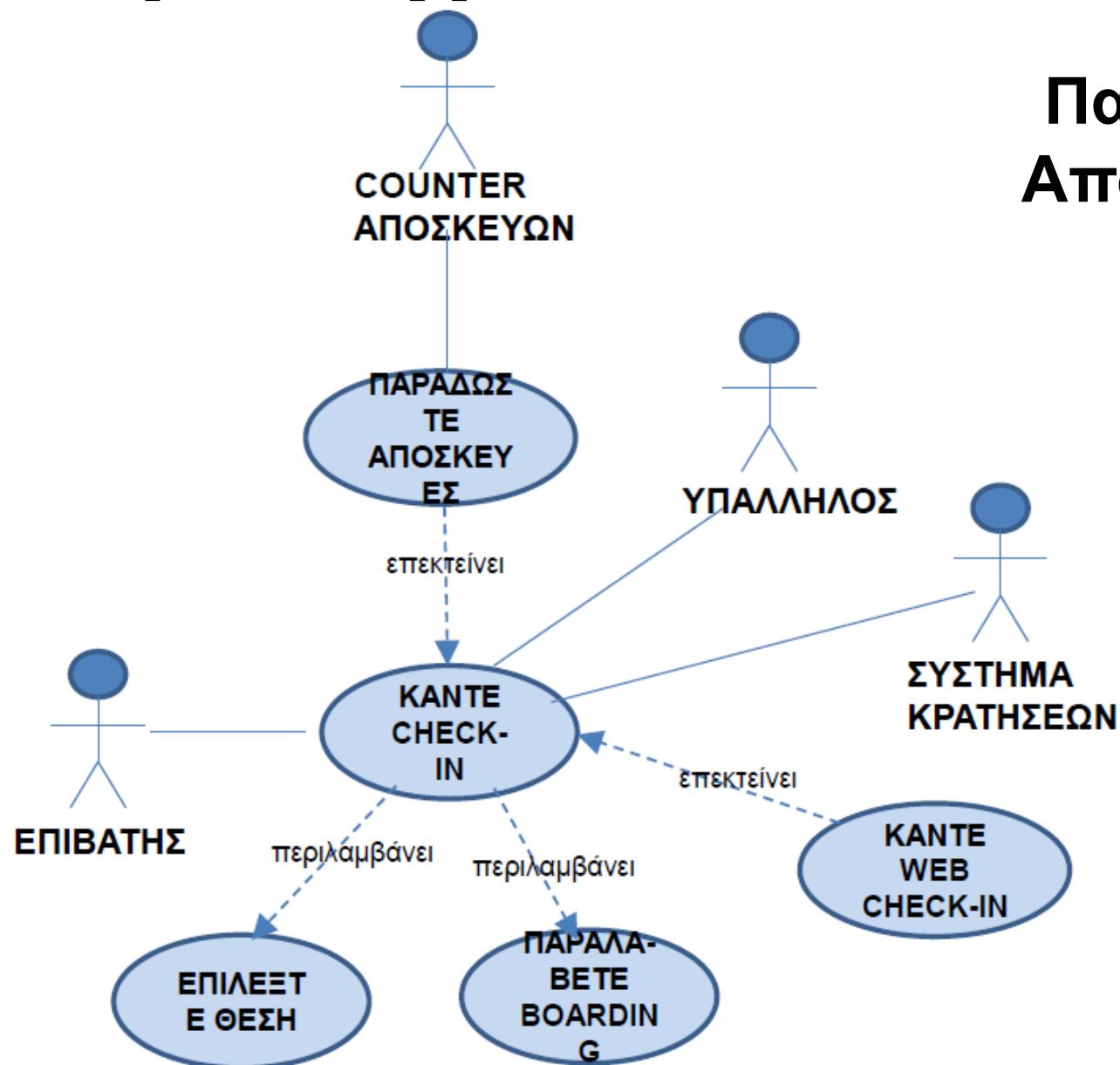
Επιλογή
Θέση και
Boarding
Pass

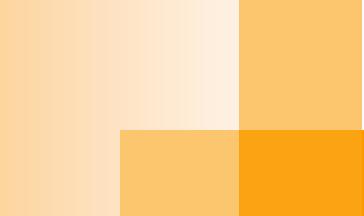
Παράδειγμα: Web Check in



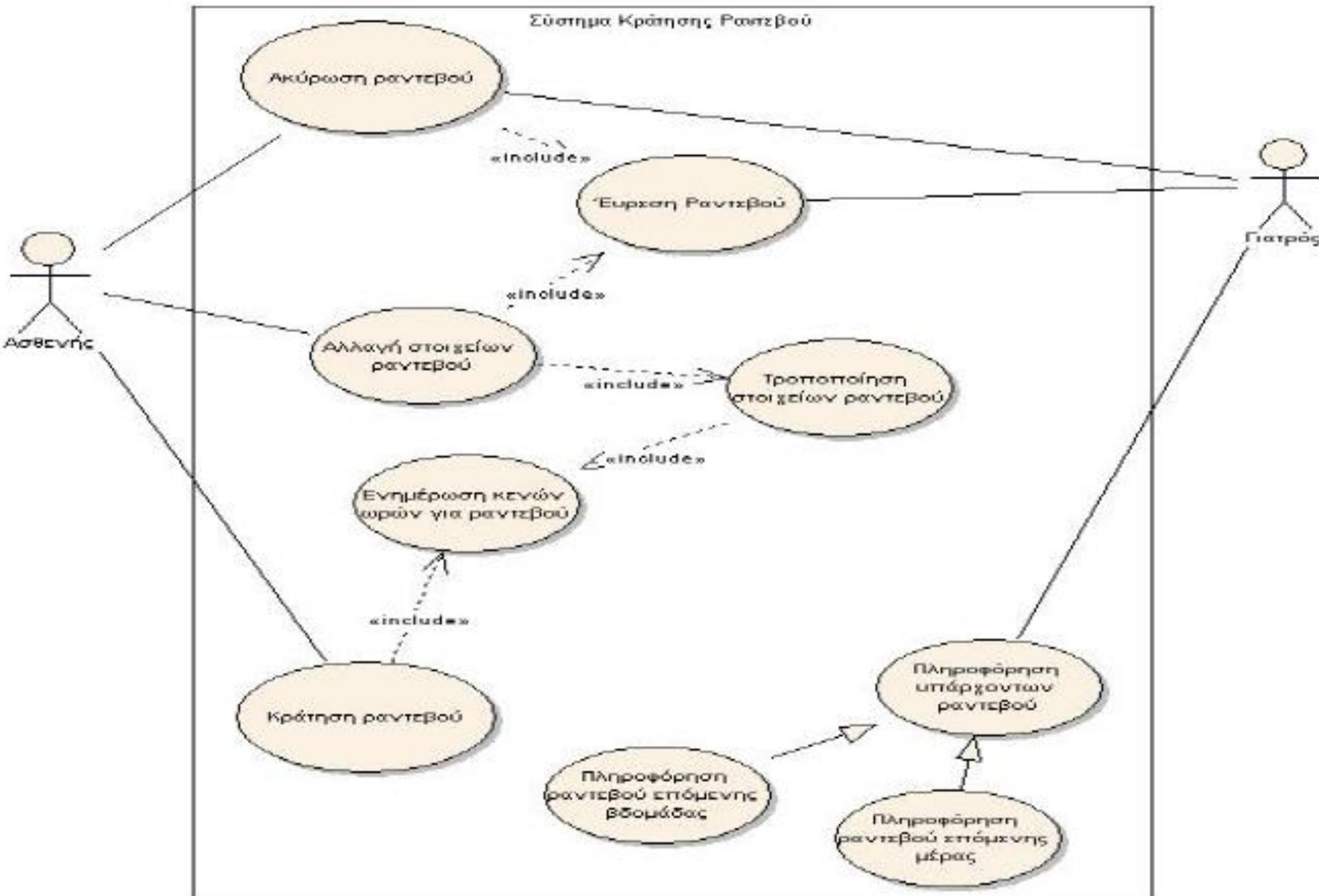
Επιλογή
Θέση και
Boarding
Pass

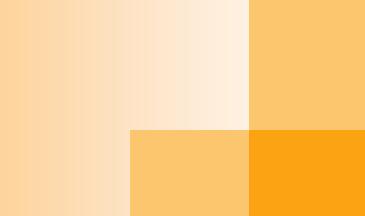
Παράδειγμα: Web Check in





Παράδειγμα Διάγραμμα Περιπτώσεων Χρήσης (Use Case Diagram) κράτη ραντεβού σε ένα ιατρείο



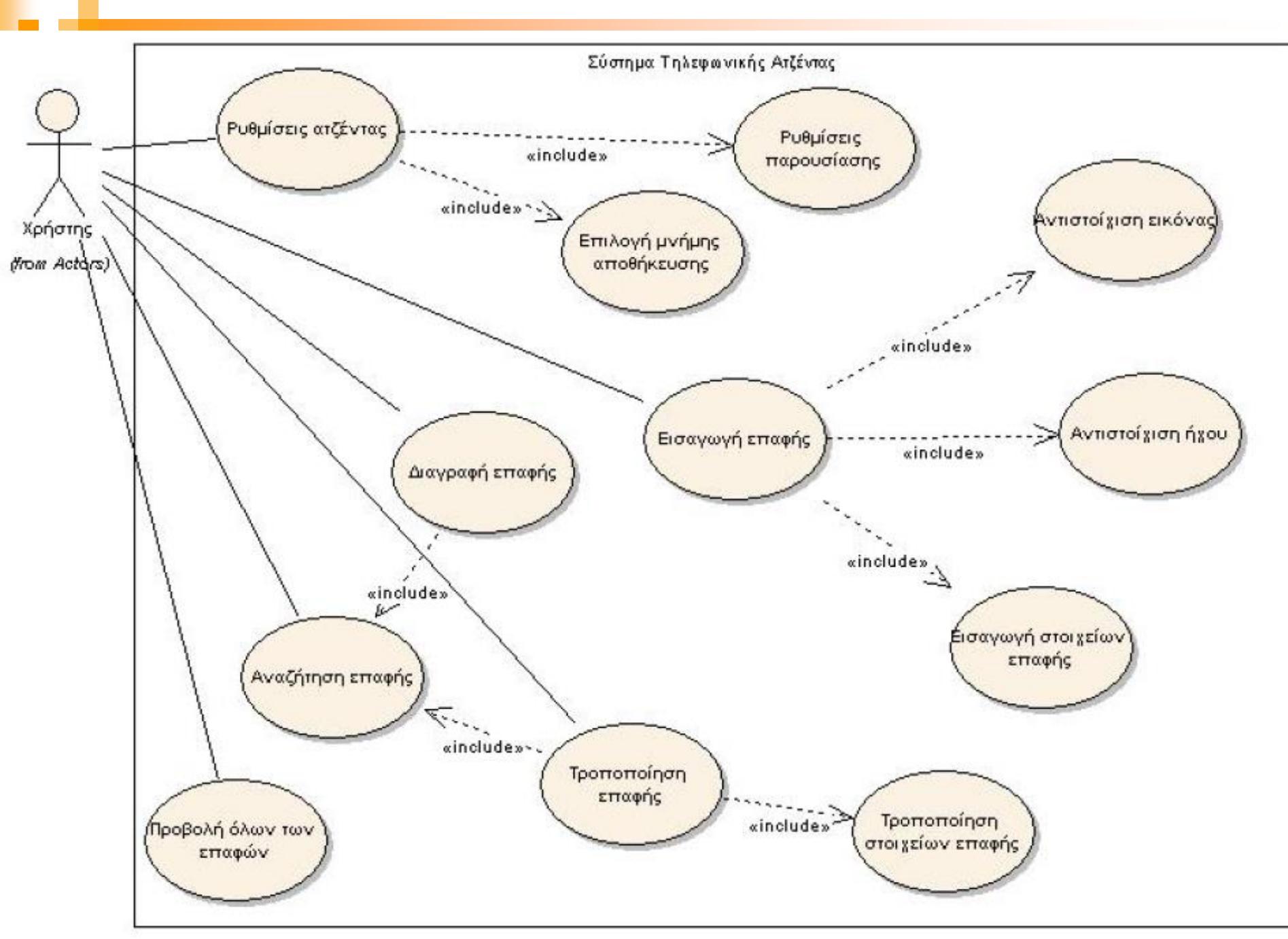


Παράδειγμα

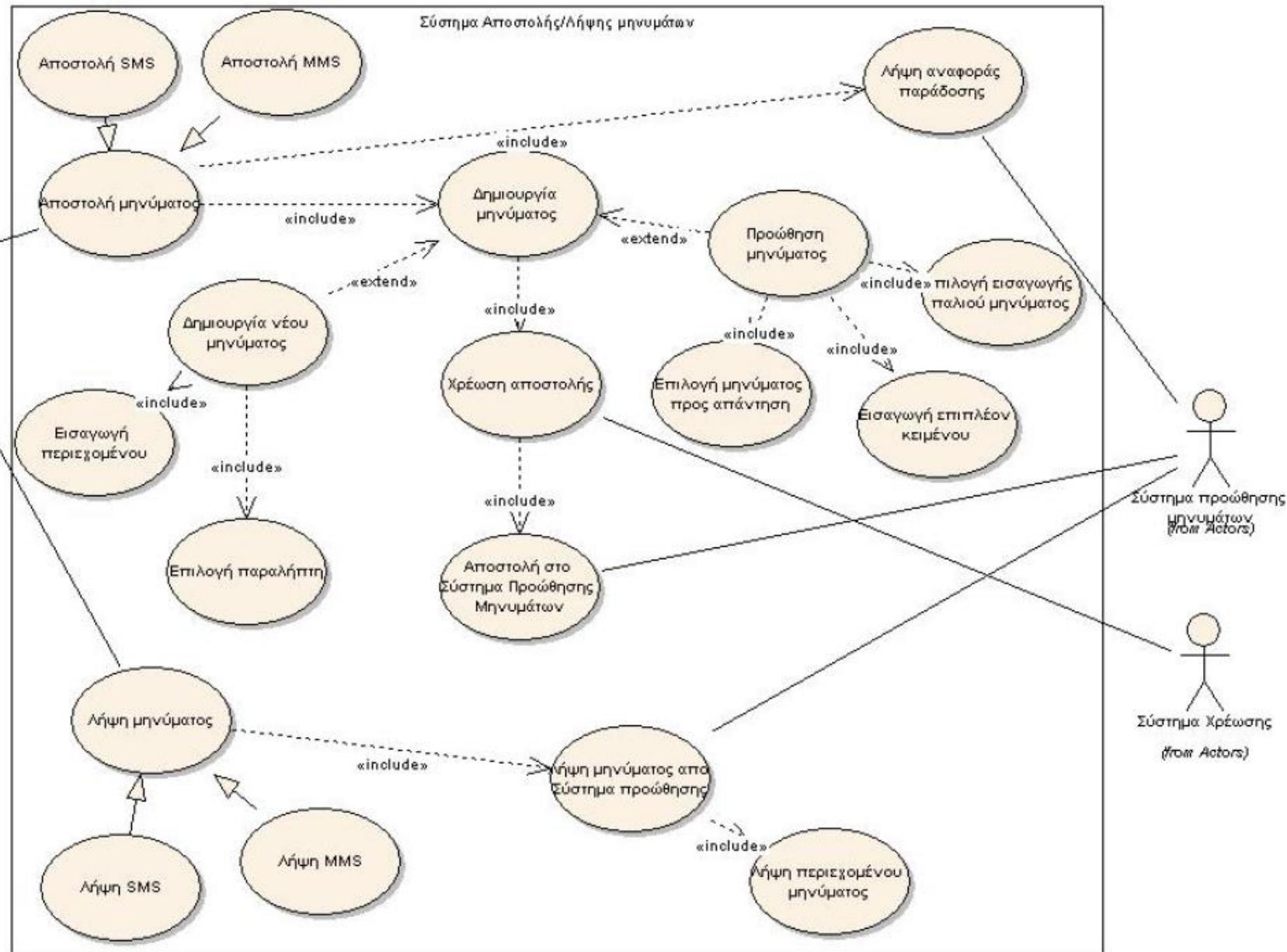
Διάγραμμα Περιπτώσεων Χρήσης

(Use Case Diagram)

τηλεφωνική ατζέντα σε κινητό



Παράδειγμα Διάγραμμα Περιπτώσεων Χρήσης (Use Case Diagram) αποστολή και λήψη μηνυμάτων σε κινητό

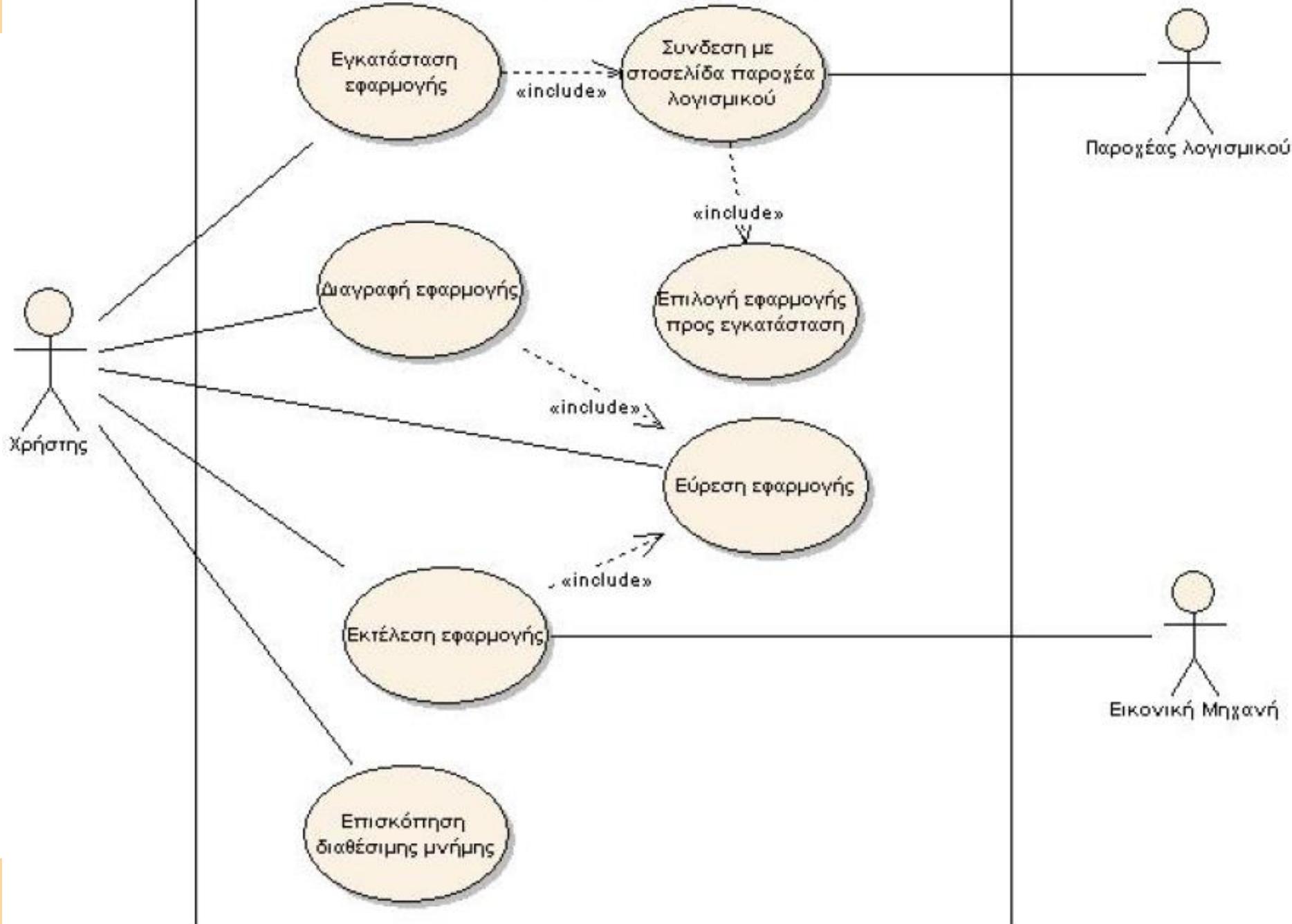


Παράδειγμα

Διάγραμμα Περιπτώσεων Χρήσης (Use Case Diagram)

Παιχνιδιών και άλλων εφαρμογών σε κινητό

Σύστημα Διαχείρισης Εφαρμογών Κινητού Τηλεφάνου

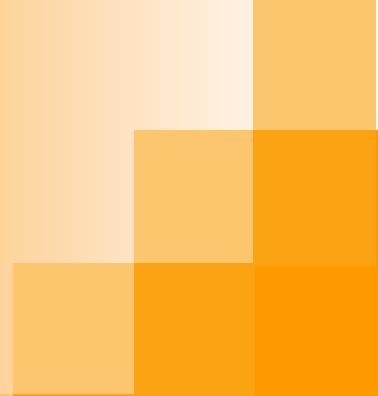


Ανάλυση Πληροφοριακών Συστημάτων

Εαρινό Εξάμηνο 2020-2021

Lec06 – 13/04/2021

Διδάσκων: Γεώργιος Χρ. Μακρής



UML

Εισαγωγικά

Διάγραμμα Περιπτώσεων Χρήσης

(Use Case Diagram)

Τι είναι η UML

- Η UML αποτελεί μία γλώσσα απεικόνισης ή μοντελοποίησης ενός πληροφοριακού συστήματος βασισμένου σε αντικείμενα (αντικειμενοστραφούς συστήματος).
- Η UML αποτελεί την πρότυπη (standard) γλώσσα μοντελοποίησης αντικειμενοστραφών συστημάτων.
- Η UML αποτελεί πρότυπο του OMG (Object Management Group – www.omg.org).



Γιατί μοντελοποιούμε συστήματα πληροφορικής;

- Η μοντελοποίηση ενός συστήματος παρέχει
 - την δυνατότητα της αφαίρεσης των ασήμαντων με αυτό λεπτομερειών και της **εστίασης στις σημαντικές λεπτομέρειες του συστήματος**
 - την δυνατότητα του πειραματισμού με **διαφορετικές λύσεις ή προσεγγίσεις** για το ίδιο πρόβλημα
 - την δυνατότητα **ανάλυσης, σχεδιασμού, καταγραφής και παρακολούθησης της προόδου** ενός έργου πληροφορικής
 - **μία κοινή γλώσσα για την επικοινωνία** όσων εμπλέκονται στην κατασκευή του συστήματος
- Χωρίς ένα μοντέλο δεν είναι δυνατόν να προσεγγίσει κανείς την πολυπλοκότητα των σύγχρονων πληροφοριακών συστημάτων



Γιατί Αντικειμενοστραφές Λογισμικό;

- Το αντικειμενοστραφές λογισμικό είναι ευκολότερο στην αρχική του σύλληψη μια και **τα αντικείμενα είναι – εν μέρει – οντότητες του υπαρκτού κόσμου**
- Το αντικειμενοστραφές λογισμικό είναι **ευκολότερο στην εξέλιξή του.**
- Η αντικειμενοστραφής προσέγγιση επιτρέπει την **δημιουργία λογισμικού με βάση τα συστατικά (components)**
- Σύγχρονες τεχνολογίες κατασκευής κατανεμημένων συστημάτων προσανατολισμένων στην σύνδεση επιχειρήσεων (Business-to-business – B2B) και στην σύνδεση επιχειρηματικών εφαρμογών (Enterprise Application Integration – EAI), όπως οι Υπηρεσίες του παγκόσμιου ιστού (Web Services), έχουν σαν **τεχνολογικό υπόβαθρο αντικειμενοστραφείς γλώσσες προγραμματισμού (π.χ. Java, C#).**



Μεθοδολογίες Ανάπτυξης Λογισμικού

- Μία μεθοδολογία ανάπτυξης λογισμικού παρέχει μία συστηματική προσέγγιση στην διαδικασία **ανάλυσης, σχεδίασης, κατασκευής και εξέλιξης** ενός έργου πληροφορικής.
- Είναι μία **σειρά σταδίων τα οποία περιγράφουν συγκεκριμένες εργασίες.**
- **Η UML είναι ουδέτερη σε σχέση με τις μεθοδολογίες χωρίς να επιβάλλει κάποια συγκεκριμένη μεθοδολογία**
- Υπάρχουν πολλές μεθοδολογίες ανάπτυξης λογισμικού (όπως η Unified Process, eXtreme Programming, Catalysis, Syntropy κοκ.). Κάθε μία από αυτές θεωρείται πως είναι πιο κατάλληλη για κάποιον συγκεκριμένο τύπο λογισμικού, για παράδειγμα η Catalysis θεωρείται πως είναι πιο κατάλληλη για συστήματα βασισμένα σε συστατικά (component based systems)



Η Ενοποιημένη Προσέγγιση

- Ίσως η πιο διαδεδομένη μεθοδολογία είναι η Ενοποιημένη Προσέγγιση (Unified Process ή εν συντομίᾳ UP). Η UP αναπτύχθηκε αρχικά από την εταιρία Rational.
- Τα βασικά στάδια ή φάσεις της UP είναι τα εξής:
 1. Σύλληψη (Inception)
 2. Λεπτομερής Επεξεργασία (Elaboration)
 3. Κατασκευή (Construction)
 4. Μετάβαση (Transition)



Η Φάση της Σύλληψης

- Σε αυτό το σύντομο αρχικό στάδιο **διερευνάται η σκοπιμότητα του έργου**: είναι αναγκαίο, κερδοφόρο κλπ. Απλά διαπιστώνεται αν θα συνεχίσει η ομάδα με την εκτέλεση του έργου ή όχι και αν χρειάζεται εξασφαλίζεται η δέσμευση για την χρηματοδότηση του έργου.



Η Φάση της Λεπτομερούς Επεξεργασίας

- Αυτό το στάδιο διαρκεί από μερικές εβδομάδες μέχρι και λίγους μήνες, ανάλογα με το μέγεθος του έργου.
- **Αποτελείται από επαναλήψεις** (iterations) καθορισμένης χρονικής διάρκειας (time-boxed) **στις οποίες βασικός στόχος είναι η καταγραφή των προδιαγραφών του συστήματος.**
- Εντοπίζονται τα τυχόν «**επικίνδυνα σημεία**» του έργου
- **Αναπτύσσεται και το χρονοδιάγραμμα** του έργου στο οποίο καταγράφεται αναλυτικά πότε θα έχει παραδοθεί η κάθε προδιαγραφή του συστήματος.
- Αρχίζει να αναπτύσσεται το έργο, να δημιουργείται δηλ. κώδικας επιπέδου παραγωγής. Τυπικά το 15% περίπου του έργου θα έχει προγραμματιστεί πριν να περάσουμε στην επόμενη φάση.



Η Φάση της Κατασκευής

- Σε αυτή τη φάση γίνεται η κατασκευή του έργου σύμφωνα με το χρονοδιάγραμμα που δημιουργήθηκε στην φάση της επεξεργασίας.
- Είναι επαναληπτική με επαναλήψεις σταθερής χρονικής διάρκειας. Σε κάθε επανάληψη αναπτύσσεται ένα σύνολο προδιαγραφών του συστήματος
- Η πρόοδος του έργου είναι απολύτως ελέγχιμη
 - Οποιαδήποτε καθυστέρηση θα γίνει πολύ γρήγορα αντιληπτή



Η Φάση της Μετάβασης

- Σε αυτή τη φάση γίνονται εργασίες όπως η **τεκμηρίωση** του συστήματος (documentation), **έλεγχος ορθότητας** (beta testing), **ρύθμιση της απόδοσης του συστήματος** (performance tuning), κλπ.



Τύποι Διαγραμμάτων της UML

- **Διάγραμμα Περιπτώσεων Χρήσης** (Use Case Diagram)
- **Διάγραμμα Κλάσεων** (Class Diagram)
- **Διαγράμματα Συμπεριφοράς** (Behavior Diagrams):
 - Διάγραμμα Καταστάσεων (Statechart Diagram)
 - Διάγραμμα Δραστηριοτήτων (Activity Diagram)
 - Διαγράμματα Αλληλεπίδρασης
 - Διάγραμμα Ακολουθίας (Sequence Diagram)
 - Διάγραμμα Συνεργασίας (Collaboration Diagram)
- **Διαγράμματα Υλοποίησης** (Implementation Diagrams):
 - Διάγραμμα Συστατικών (Component Diagram)
 - Διάγραμμα Διάταξης (Deployment Diagram)



Γιατί τόσοι διαφορετικοί τύποι διαγραμμάτων;

- Το σύστημα έχει στατικά στοιχεία, δυναμικά στοιχεία, στοιχεία υλοποίησης κοκ.
- Δεν είναι δυνατόν ένας και μόνο τύπος διαγράμματος να περιγράψει όλες αυτές τις διαφορετικές οπτικές γωνίες ενός συστήματος.
- Έτσι τυπικά χρησιμοποιούμε διαγράμματα κλάσεων για την περιγραφή των στατικών σχέσεων μεταξύ των κλάσεων, διαγράμματα συμπεριφοράς (κατάστασης, ακολουθίας κοκ) για την περιγραφή της δυναμικής συμπεριφοράς του συστήματος και διαγράμματα υλοποίησης (συστατικών, διάταξης) για την καταγραφή των λεπτομερειών υλοποίησης του συστήματος. Τα διαγράμματα περιπτώσεων χρήσης είναι μία ειδική περίπτωση υπό την έννοια πως δεν έχουν κάποια σχέση με αντικείμενα ή αντικειμενοστραφή συστήματα.



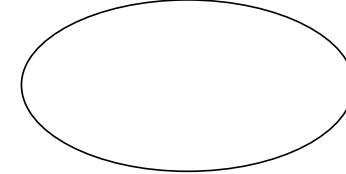
Ανάλυση Προδιαγραφών

- Το βασικότερο θέμα στην κατασκευή ενός πληροφοριακού συστήματος είναι η κατασκευή του ορθού συστήματος: του συστήματος
- Το μοντέλο **περιπτώσεων χρήσης** δίνει έμφαση στην λειτουργικότητα ενός συστήματος, όπως αυτή είναι ορατή από τους εξωτερικούς χρήστες του.
- Μια περίπτωση χρήσης διαμερίζει την λειτουργικότητα ενός συστήματος σε συναλλαγές («περιπτώσεις χρήσης») που έχουν νόημα για τους χρήστες του συστήματος («χειριστές»).



Περίπτωση Χρήσης

- Αναπαριστά έναν στόχο (user goal) για έναν εξωτερικό χειριστή (actor) του συστήματος.
- Οι χειριστές ενός συστήματος μπορεί να είναι άνθρωποι (π.χ. γραμματέας, ταμίας) αλλά ενδέχεται να είναι και εξωτερικά συστήματα (π.χ. Διατραπεζικό Σύστημα Συναλλαγών)
- Το σύμβολο για μία περίπτωση χρήσης είναι η έλλειψη στην οποία αναγράφεται ένα όνομα για την περίπτωση χρήσης.



Ανάληψη Μετρητών



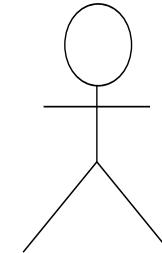
Σενάρια Περίπτωσης Χρήσης

- Μία περίπτωση χρήσης συνήθως περιλαμβάνει πολλά εναλλακτικά σενάρια τα οποία ονομάζονται επεκτάσεις (extensions)
 - Ένα σενάριο όπου όλα πάνε καλά (happy path)
 - Αρκετά σενάρια όπου κάτι δεν πάει καλά ή προκύπτει κάποια εξαίρεση σε σχέση με το φυσιολογικό
- Οι εξαίρεσις διαφοροποιούν το βασικό σενάριο σε συγκεκριμένα σημεία



Χειριστές

- Ο χειριστής (actor) ενός συστήματος μπορεί να είναι άνθρωπος ή υποσύστημα
- Το σύμβολο των διαγραμμάτων περιπτώσεων χρήσης για τους χειριστές είναι μία φιγούρα (stickman)
- Αν πρόκειται για υποσύστημα μπορούμε να χρησιμοποιήσουμε το σύμβολο της κλάσης με το στερεότυπο <<Actor>>



Πελάτης

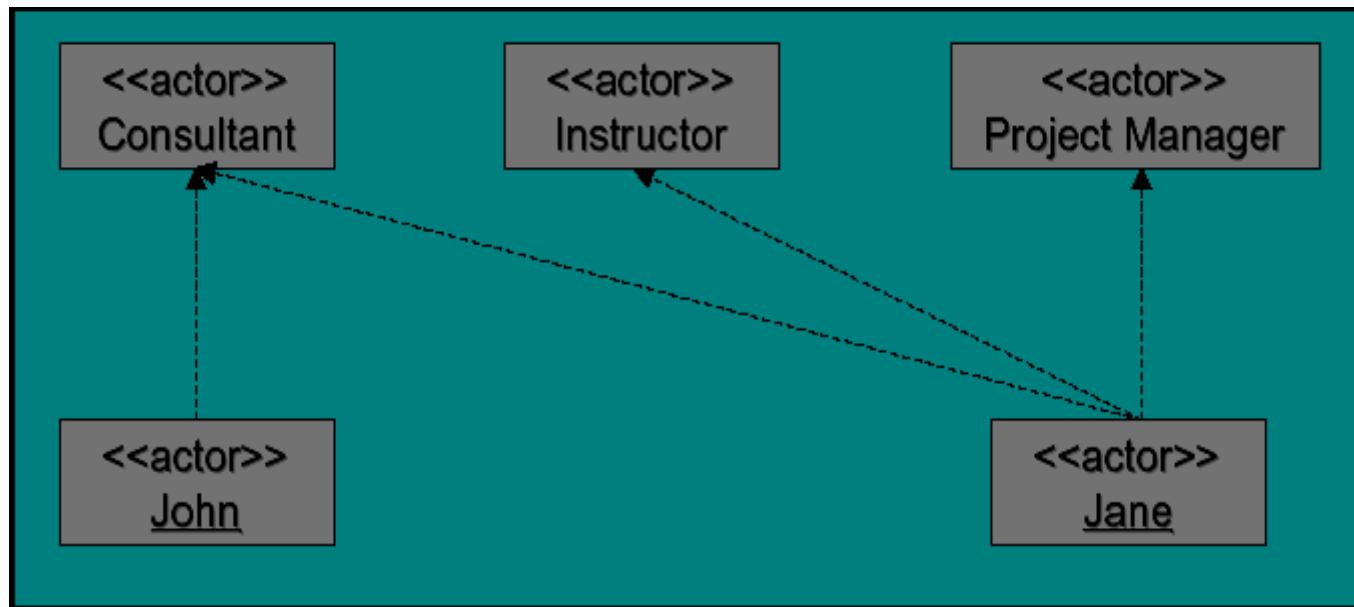
<<Actor>>
Διατραπεζικό Σύστημα Συναλλαγών



Ο χειριστής είναι ρόλος

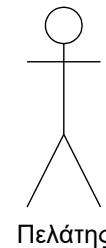
Ένας χειριστής ορίζει ένα ρόλο που παίζουν οι χρήστες στην αλληλεπίδρασή τους με το σύστημα

- ➔ Πολλοί χρήστες μπορούν να παίζουν ένα ρόλο
- ➔ Ένας χρήστης μπορεί να παίξει πολλούς ρόλους

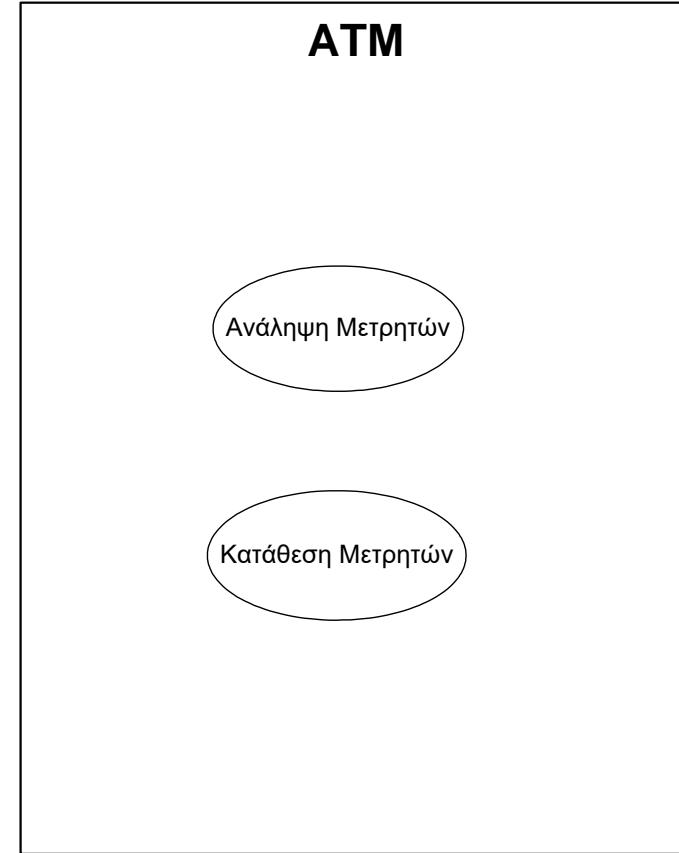


Πλαισιο Συστήματος

- Για να διακρίνουμε τις προδιαγραφές που βρίσκονται μέσα στα πλαισια του υπό ανάπτυξη συστήματος από τα πιθανά εξωτερικά συστήματα και τους χρήστες, περιλαμβάνουμε τις περιπτώσεις χρήσης σε ένα πλαισιο με τίτλο το όνομα του συστήματος
- Μέσα στο πλαισιο του συστήματος τοποθετούμε τις περιπτώσεις χρήσης και έξω από αυτό τους χειριστές του συστήματος.

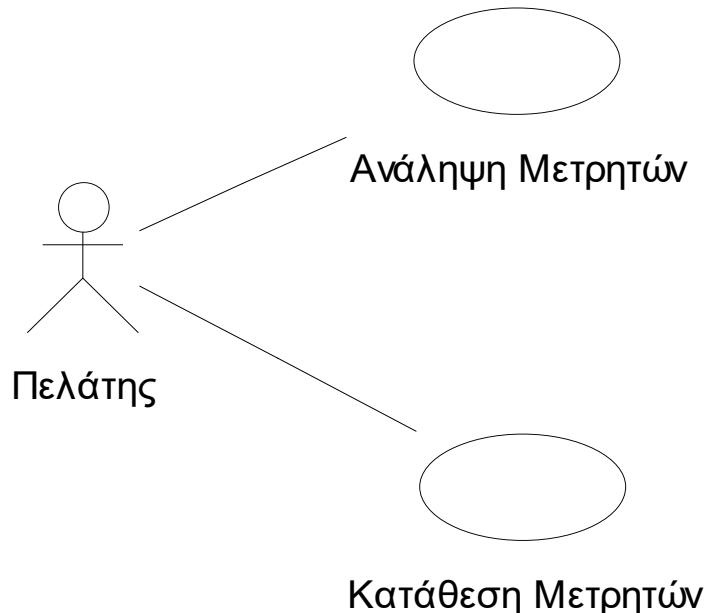


Πελάτης



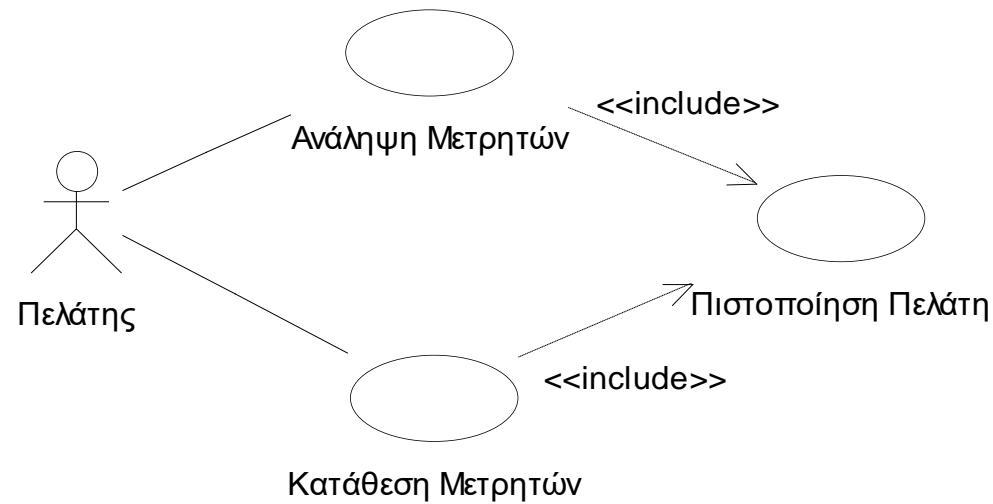
Σχέση

- Υποδηλώνει την σχέση ενός χειριστή με μία περίπτωση χρήσης.
- Ενδέχεται ένας χειριστής να είναι ο βασικός (primary) για μία περίπτωση χρήσης αλλά και άλλοι χειριστές ή εξωτερικά συστήματα να σχετίζονται επίσης με αυτήν.



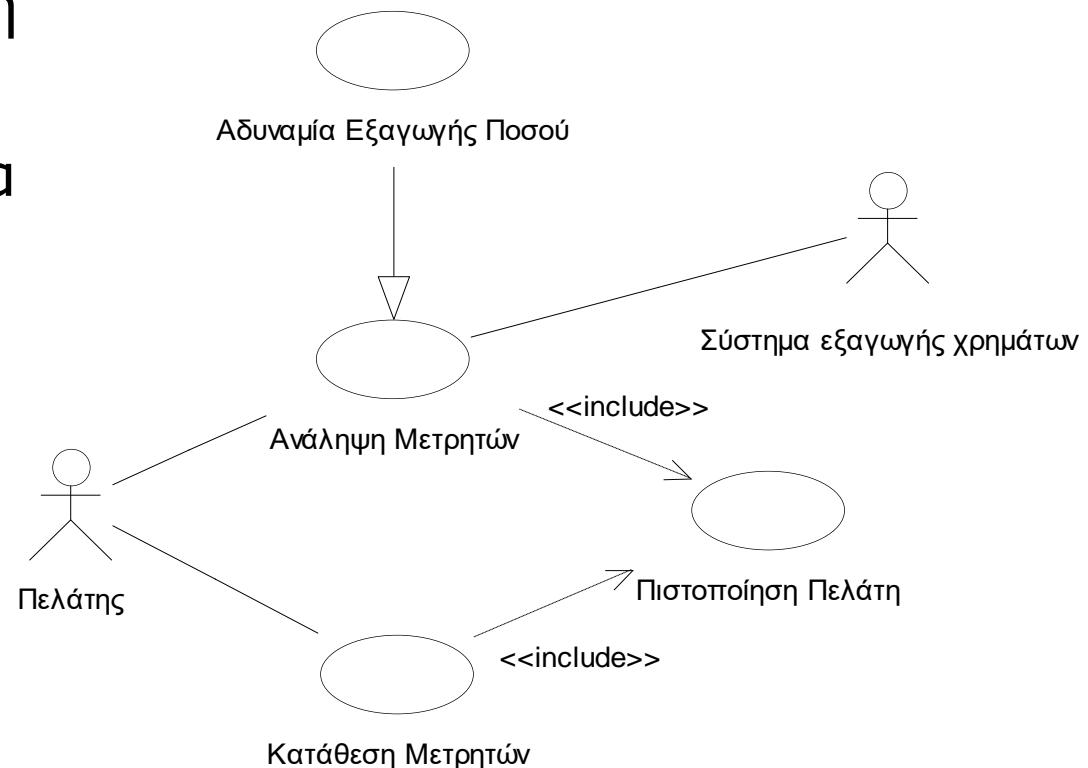
Συμπερίληψη

- Η συμπερίληψη είναι μία ειδική περίπτωση σχέσης στην οποία σχετίζουμε δύο περιπτώσεις χρήσης. Η μία περίπτωση χρήσης συμπεριλαμβάνει την άλλη.
- Η έννοια της συμπερίληψης είναι υποχρεωτική, δηλαδή **πάντα** η μία περίπτωση χρήσης θα συμπεριλαμβάνει την άλλη.
- Η φορά του βέλους στην συμπερίληψη είναι από την περίπτωση χρήσης που συμπεριλαμβάνει προς αυτήν που συμπεριλαμβάνεται και επισημαίνεται με το στερεότυπο <<include>>



Επέκταση

- Η επέκταση όπως και η συμπερίληψη καθορίζει μία σχέση μεταξύ δύο περιπτώσεων χρήσης, στην οποία μία περίπτωση χρήσης επεκτείνεται προαιρετικά από μία άλλη ανάλογα με τις επιλογές ή την κατάσταση κάποιου χειριστή (βασικού ή δευτερεύοντος)



Καταγραφή προδιαγραφών με κείμενο

- Η εύρεση των προδιαγραφών του συστήματος δεν είναι μία εργασία που έχει άμεση σχέση με την καταγραφή αυτών των προδιαγραφών σε ένα διάγραμμα περιπτώσεων χρήσης. Το διάγραμμα περιπτώσεων χρήσης κατά τη διαδικασία εύρεσης των προδιαγραφών του συστήματος πρέπει να χρησιμοποιείται επικουρικά.
- Εναλλακτικά ή και συμπληρωματικά με τα διαγράμματα περιπτώσεων χρήσης θα μπορούσε η καταγραφή των προδιαγραφών να γίνει με κείμενο σε έναν πίνακα



Πρότυπο Προδιαγραφής Περίπτωσης Χρήσης

1. Τίτλος περίπτωσης χρήσης
2. Σύντομη περιγραφή
3. Ροή γεγονότων
 - 3.1 Βασική ροή
(ακολουθία των ενεργειών)
 - 3.2 Εναλλακτικές ροές
(π.χ. σφάλματος)
4. Μη λειτουργικές απαιτήσεις
(απαιτήσεις επίδοσης ή περιβάλλοντος)
5. Κατάσταση εισόδου
(pre-conditions)
6. Κατάσταση εξόδου
(post-conditions)



Το παράδειγμα Π.Χ. «Επίκουρος»

■ Από το βιβλίο Τεχνολογία Λογισμικού II του ΕΑΠ – του Β. Βεσκουκη

- «ΕΠΙΚΟΥΡΟΣ»: Μια εφαρμογή λογισμικού για την υποστήριξη της γραμματείας εκπαιδευτικού φορέα.
- Η εφαρμογή θα πρέπει να τηρεί αρχεία σπουδαστών, καθηγητών, μαθημάτων, εγγραφής σε μαθήματα, καθώς και αποτελέσματα βαθμολογίας.
- Η εφαρμογή θα πρέπει να εκτυπώνει καταστάσεις σπουδαστών, καθηγητών, μαθημάτων και βαθμολογίας με κριτήρια που θα δίνει ο χρηστης.
- Η εφαρμογή δεν θα πρέπει να επιτρέπει τη διαγραφή ενός σπουδαστή ή καθηγητή από το αρχείο αν αυτός έχει εγγραφεί ή διδάξει μάθημα, αντίστοιχα. Το περιβάλλον λειτουργίας θα είναι ένας αυτόνομος ηλεκτρονικός υπολογιστής με Windows.



Απαιτήσεις από το λογισμικό (1)

- 1. Ο «ΕΠΙΚΟΥΡΟΣ» θα τρέχει σε αυτόνομο υπολογιστή κάτω από το λειτουργικό σύστημα Windows. Δεν απαιτείται σύνδεση σε δίκτυο.
- 2. Ζητείται η τήρηση αρχείων μαθητών, καθηγητών και μαθημάτων.
- 3. Κάθε μάθημα διδάσκεται από έναν καθηγητή σε κάθε ακαδημαϊκό έτος.
- 4. Κάθε σπουδαστής μπορεί να εγγράφεται σε κάθε μάθημα όσες φορές θέλει.
- 5. Κάθε σπουδαστής αξιολογείται σε μαθήματα στα οποία έχει εγγραφεί. Η αξιολόγηση αυτή μπορεί να γίνεται περισσότερες από μία φορές τόσο κατά τη διάρκεια του ακαδημαϊκού έτους (ενδιάμεση εξέταση), όσο και με τελικό γραπτό.



Απαιτήσεις από το λογισμικό (2)

- 6. Δεν πρέπει να επιτρέπεται η καταχώρηση βαθμολογίας σε μάθημα στο οποίο δεν έχει γίνει εγγραφή.
- 7. Επιτρέπεται η διαγραφή σπουδαστή μόνο αν δεν έχει εγγραφεί σε κανένα μάθημα.
- 8. Επιτρέπεται η διαγραφή καθηγητή μόνο αν δεν έχει διδάξει κανένα μάθημα.
- 9. Επιτρέπεται η διαγραφή μαθήματος μόνο αν δεν έχουν υπάρξει εγγραφές ή εξετάσεις που να το αφορούν.
- 10. Ζητείται αλφαριθμητική εκτύπωση ολόκληρου του αρχείου των σπουδαστών.
- 11. Ζητείται αλφαριθμητική εκτύπωση των εγγεγραμμένων σε κάθε μάθημα σπουδαστών.
- 12. Ζητείται αλφαριθμητική εκτύπωση ολόκληρου του αρχείου καθηγητών.
- 13. Ζητείται αλφαριθμητική εκτύπωση των φοιτητών με τη βαθμολογία σε κάθε μάθημα.

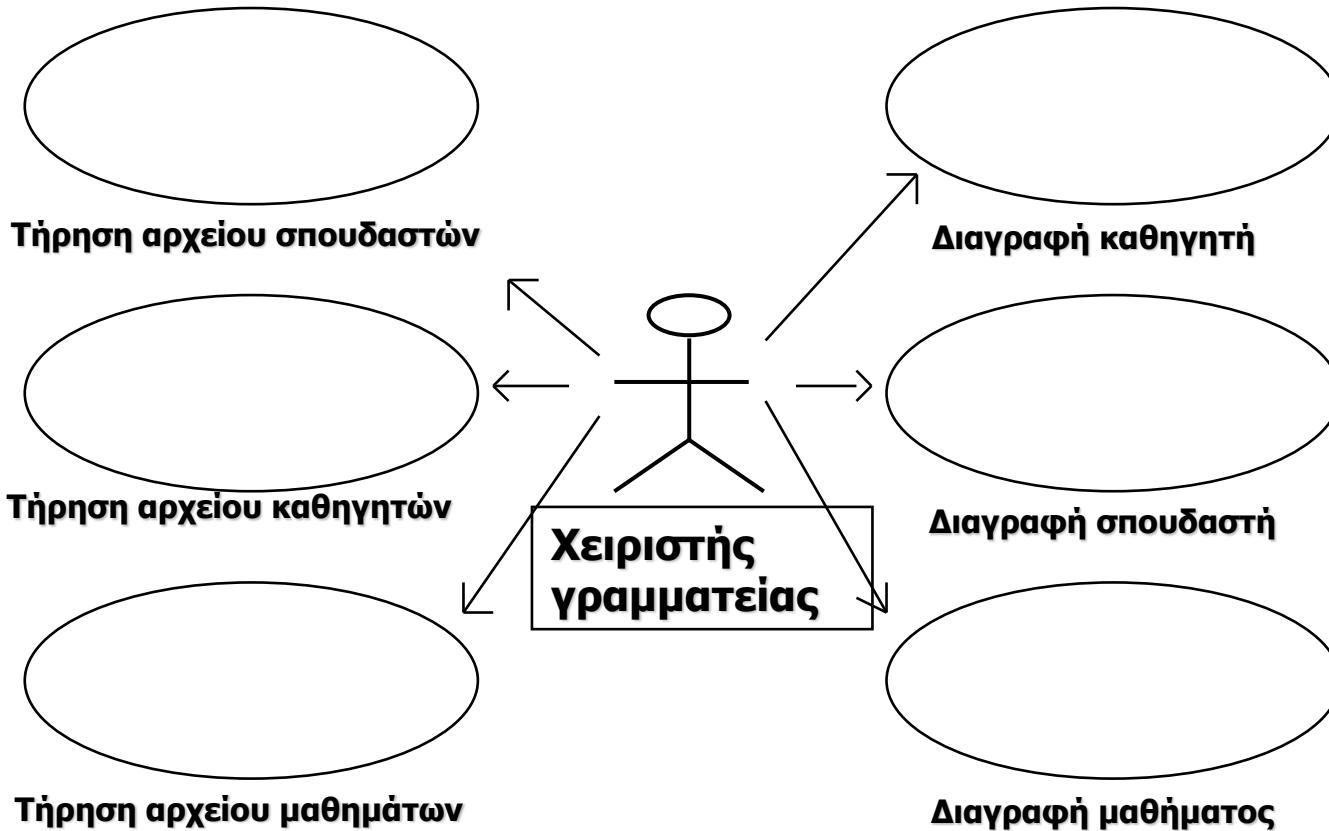


Άσκηση 1

- Να κατασκευάσουμε ένα διάγραμμα περιπτώσεων χρήσης που αντιστοιχεί στις απαιτήσεις 2, 7, 8 και 9. Να τις θυμηθούμε:
 - 2. Ζητείται η τήρηση αρχείων μαθητών, καθηγητών και μαθημάτων.
 - 7. Επιτρέπεται η διαγραφή σπουδαστή μόνο αν δεν έχει εγγραφεί σε κανένα μάθημα.
 - 8. Επιτρέπεται η διαγραφή καθηγητή μόνο αν δεν έχει διδάξει κανένα μάθημα.
 - 9. Επιτρέπεται η διαγραφή μαθήματος μόνο αν δεν έχουν υπάρξει εγγραφές ή εξετάσεις που να το αφορούν.



Απάντηση



Άσκηση 2

Να προσθέσουμε και να υλοποιήσουμε δύο ακόμη απαιτήσεις:

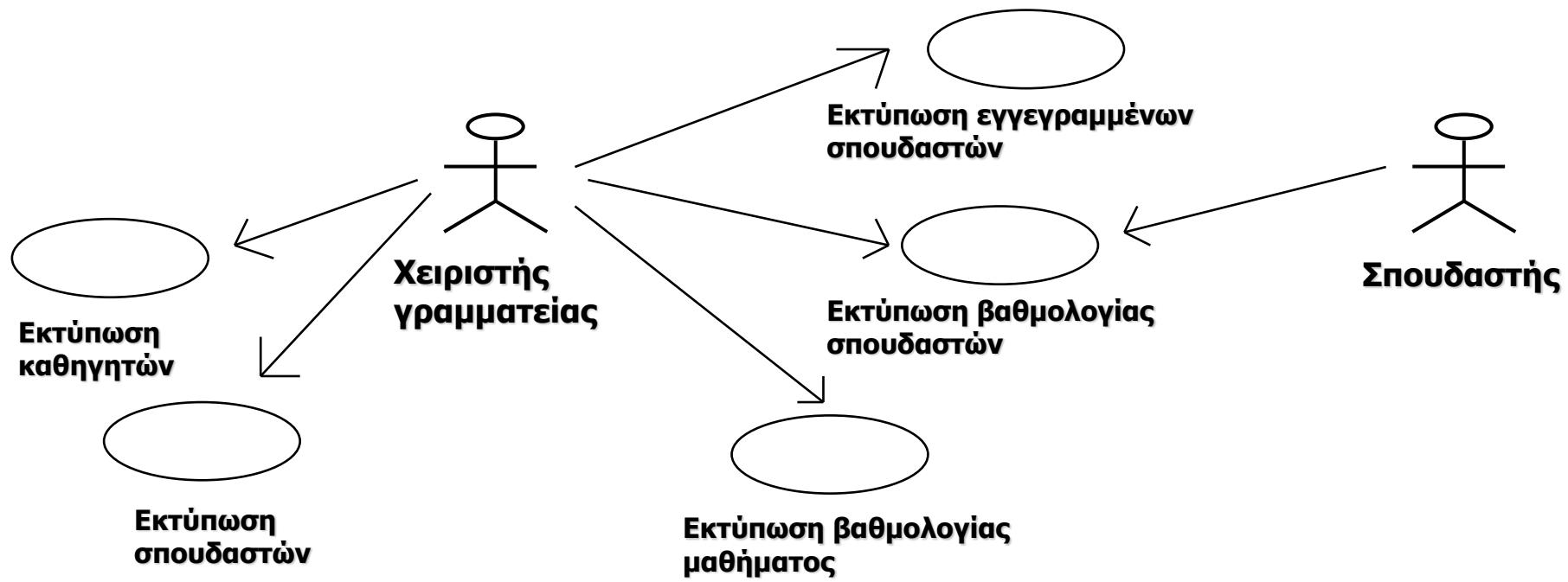
14. Ο σπουδαστής μπορεί να χρησιμοποιεί ένα τερματικό του συστήματος και να βλέπει ο ίδιος την βαθμολογία του σε όλα τα μαθήματα.
15. Ο διδάσκων ενός μαθήματος είναι δυνατόν να καταχωρεί ο ίδιος τη βαθμολογία των σπουδαστών σε αυτό.

Προσοχή!!!

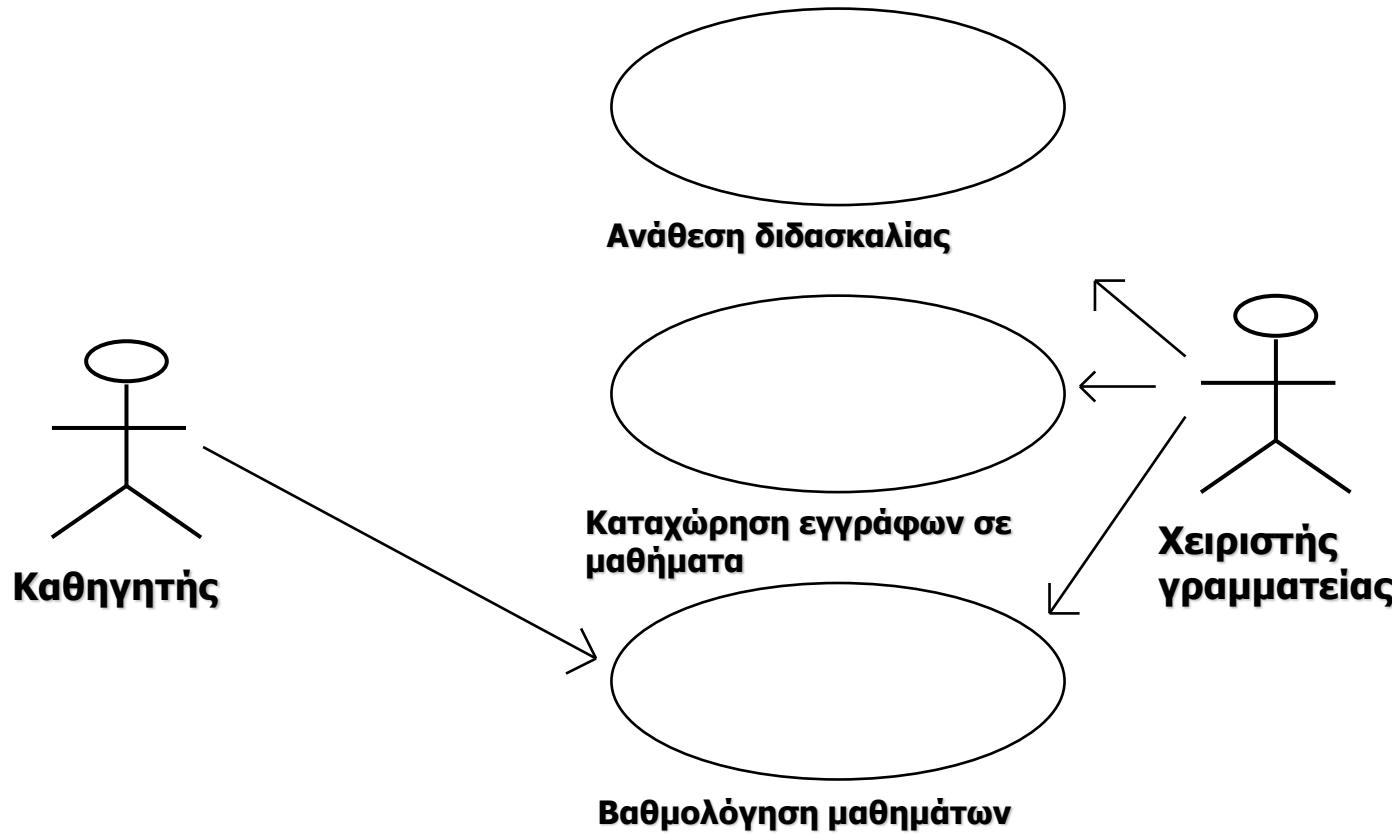
Στις δύο αυτές περιπτώσεις εμπλέκονται περισσότεροι του ενός χειριστές με περισσότερες εργασίες...



Επίκουρος - Απαίτηση 14η



Επίκουρος - Απαίτηση 15η



Βασικές Αρχές

- Το όνομα της Π.Χ. δείχνει ενέργεια
- Η Π.Χ. πρέπει να περιγράφει μία πλήρη συμπεριφορά
- Η Π.Χ. θα πρέπει να μπορεί να ολοκληρωθεί
- Μία συμπεριφορά για κάθε Π.Χ.
- Χρήση της ορολογίας του πεδίου του προβλήματος



Συχνά λάθη στα διαγράμματα περιπτώσεων χρήσης

- Συνωστισμένα Διαγράμματα (>20 ΠΧ)
- Ασαφείς Π. Χ. ή πολλές περιπτώσεις χρήσης για παρόμοιο σκοπό.
- Λανθασμένη/μπερδεμένη χρήση include και extend
 - Λάθος κατεύθυνση βέλους
 - Εκεί που ταιριάζει include χρησιμοποιείται το extend και αντίστροφα
- Λανθασμένη ονοματολογία. Τα ονόματα των περιπτώσεων χρήσης δεν δείχνουν ενέργεια

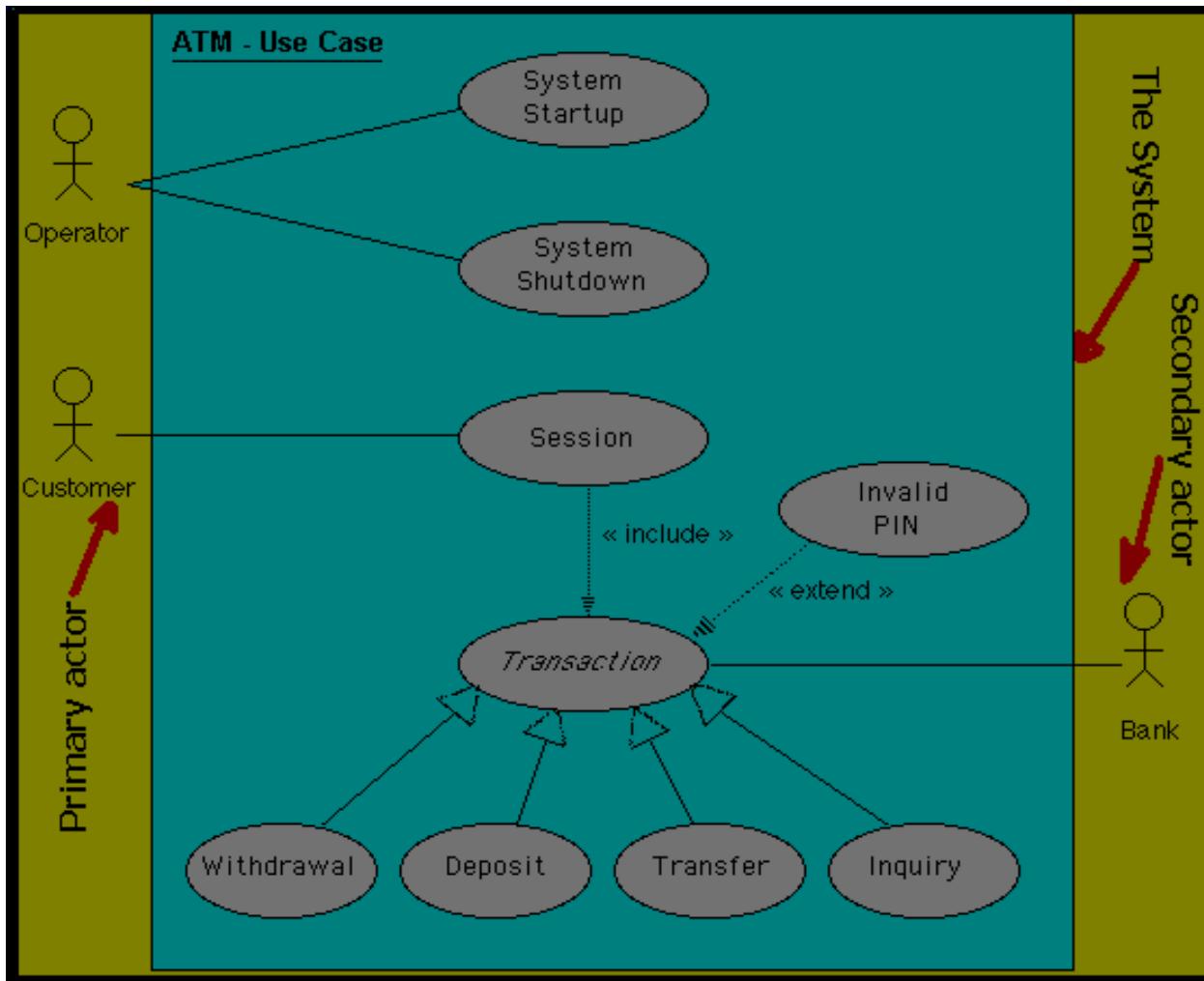


Προδιαγραφή περιπτώσεων χρήσης - Λάθη

- Βασικό πρόβλημα είναι η μη αντιστοιχία των περιγραφών με το διάγραμμα περιπτώσεων χρήσης.
 - Μη περιγραφή όλων των περιπτώσεων χρήσης του διαγράμματος
 - Χρήση διαφορετικών τίτλων για τις περιπτώσεις χρήσης
 - Εμφάνιση νέων περιπτώσεων χρήσης
- Δεν αναφέρεται με σαφήνεια ο χειριστής της περίπτωσης χρήσης.
- Συσσώρευση διαφορετικών λειτουργιών στη βασική ροή μιας ΠΧ (π.χ. Εισαγωγή/Τροποποίηση/Διαγραφή/Ανάκληση εγγραφών Πελάτη). Το πιο συνηθισμένο παράδειγμα είναι η περιγραφή της περίπτωσης χρήσης login μέσα στις άλλες
- Περιγραφή μόνο των ενεργειών του χρήστη και όχι των αποκρίσεων του συστήματος
- Συνοπτικές περιγραφές (π.χ. βασική ροή που κάθε ενέργειά της θα μπορούσε να περιγραφεί σαν μια ξεχωριστή ΠΧ)
- Σύνθεση διαφορετικών ΠΧ του διαγράμματος σε μια περιγραφή ΠΧ με συνέπεια πολύπλοκες περιγραφές
- Οι σχέσεις include και extend δεν λαμβάνονται υπόψη στις προδιαγραφές



Παράδειγμα ΑΤΜ – Περίπτωση χρήσης - 1



Παράδειγμα ATM – Περίπτωση χρήσης - 2

System Startup – Use Case

- ➔ **ο χειριστής πατά τον διακόπτη - ON**
- ➔ **τοποθετεί χρήματα στον διανομέα**
- ➔ **το ATM συνδέεται με την τράπεζα και ο πελάτης μπορεί να ξεκινήσει τις κινήσεις**

System Shutdown - Use Case

- ➔ **ο χειριστής πατά τον διακόπτη - OFF**
- ➔ **αποσυνδέεται το ATM από την τράπεζα**
- ➔ **ο χειριστής παίρνει τους φακέλους κατάθεσης, και κάνει service το ATM.**



Παράδειγμα ATM – Περίπτωση χρήσης - 3

Session – Use Case

- ➔ Ξεκινά με την είσοδο της κάρτας. Διαβάζει την κάρτα και αν υπάρχει πρόβλημα την απορρίπτει
- ➔ Διαφορετικά επιτρέπει την είσοδο του PIN και την εκτέλεση κίνησης
- ➔ Εκτελούνται μια ή περισσότερες κινήσεις και στο τέλος επιστρέφει την κάρτα
- ➔ Η κάρτα δεσμεύεται αν η είσοδος του PIN είναι λανθασμένη (> = 3 φορές)

Transaction – Use Case

- ➔ αφαιρεση γενίκευσης κίνησης. Ξεκινά μέσα σε ένα session όταν ο χρήστης επιλέγει από το μενού μία κίνηση
- ➔ **Μηνύματα** - από και προς την τράπεζα καταχωρούνται στο log – αρχείο του ATM



Παράδειγμα ATM – Περίπτωση χρήσης - 4

Withdrawal - Use Case

- ◆ ο πελάτης επιλέγει τύπο λογαριασμού ανάληψης από το μενού και πληκτρολογεί το ποσό
- ◆ το σύστημα επαληθεύει την ύπαρξη του ποσού πριν επιτρέψει τον πελάτη να κάνει την ανάληψη
- ◆ αν επιτραπεί η κίνηση το ποσό ανάληψης χορηγείται από τον διανομέα και εκδίδεται η απόδειξη
- ◆ η κίνηση καταχωρείται στο log – του ATM. Ο πελάτης μπορεί να ακυρώσει με το CANCEL την ανάληψη

Deposit - Use Case

- ◆ ο πελάτης επιλέγει τύπο λογαριασμού κατάθεσης από το μενού και πληκτρολογεί το ποσό
- ◆ η κίνηση στέλνεται στην τράπεζα για επαλήθευση ότι το ATM μπορεί να δεχτεί την κατάθεση



Παράδειγμα ATM – Περίπτωση χρήσης - 6

Inquire - Use Case

- **ο πελάτης επιλέγει τον λογαριασμό για τον οποίο θέλει ενημέρωση.**
- **η κίνηση εγκρίνεται από την τράπεζα και εκδίδεται απόδειξη**

Περιγράψτε εσείς την επέκταση ελέγχου του PIN...



Παράδειγμα ATM – Περίπτωση χρήσης - 3

Session – Use Case

- ➔ Ξεκινά με την είσοδο της κάρτας. Διαβάζει την κάρτα και αν υπάρχει πρόβλημα την απορρίπτει
- ➔ Διαφορετικά επιτρέπει την είσοδο του PIN και την εκτέλεση κίνησης
- ➔ Εκτελούνται μια ή περισσότερες κινήσεις και στο τέλος επιστρέφει την κάρτα
- ➔ Η κάρτα δεσμεύεται αν η είσοδος του PIN είναι λανθασμένη (> = 3 φορές)

Transaction – Use Case

- ➔ αφαιρεση γενίκευσης κίνησης. Ξεκινά μέσα σε ένα session όταν ο χρήστης επιλέγει από το μενού μία κίνηση
- ➔ **Μηνύματα** - από και προς την τράπεζα καταχωρούνται στο log – αρχείο του ATM



Παράδειγμα ATM – Περίπτωση χρήσης - 4

Withdrawal - Use Case

- ◆ ο πελάτης επιλέγει τύπο λογαριασμού ανάληψης από το μενού και πληκτρολογεί το ποσό
- ◆ το σύστημα επαληθεύει την ύπαρξη του ποσού πριν επιτρέψει τον πελάτη να κάνει την ανάληψη
- ◆ αν επιτραπεί η κίνηση το ποσό ανάληψης χορηγείται από τον διανομέα και εκδίδεται η απόδειξη
- ◆ η κίνηση καταχωρείται στο log – του ATM. Ο πελάτης μπορεί να ακυρώσει με το CANCEL την ανάληψη

Deposit - Use Case

- ◆ ο πελάτης επιλέγει τύπο λογαριασμού κατάθεσης από το μενού και πληκτρολογεί το ποσό
- ◆ η κίνηση στέλνεται στην τράπεζα για επαλήθευση ότι το ATM μπορεί να δεχτεί την κατάθεση



Παράδειγμα ATM – Περίπτωση χρήσης - 5

- ➔ αν η κίνηση γίνει αποδεκτή ο πελάτης εισάγει τον φάκελο που γίνεται αποδεκτός και εκδίδεται η απόδειξη
- ➔ μόλις ληφθεί ο φάκελος στέλνεται νέο μήνυμα στην τράπεζα για την κατάθεση ώστε να πιστωθεί ο λογαριασμός αργότερα
- ➔ η λήψη του φακέλου καταχωρείται στο log – του ATM. Ο πελάτης μπορεί να ακυρώσει με το CANCEL την κατάθεση πριν βάλει το φάκελο

Transfer - Use Case

- ➔ Ο πελάτης επιλέγει τους λογαριασμούς από και προς μεταφορά
- ➔ πληκτρολογεί το ποσό, αφού εγκρίνει η τράπεζα και εκδίδεται η απόδειξη



Παράδειγμα ATM – Περίπτωση χρήσης - 6

Inquire - Use Case

- **ο πελάτης επιλέγει τον λογαριασμό για τον οποίο θέλει ενημέρωση.**
- **η κίνηση εγκρίνεται από την τράπεζα και εκδίδεται απόδειξη**

Περιγράψτε εσείς την επέκταση ελέγχου του PIN...



Ανάλυση Πληροφοριακών Συστημάτων

Εαρινό Εξάμηνο 2020-2021

Lec05: Προγραμματισμός Έργων

30/03/2021

Διδάσκων: Δρ. Γεώργιος Χρ. Μακρής

Ανάλυση έργου και διαμόρφωση δικτυωτού διαγράμματος

Βασικές έννοιες

- **Έργο** ονομάζεται ένα σύνολο αλληλένδετων ενεργειών ή δραστηριοτήτων, οι οποίες εκτελούνται με συγκεκριμένη σειρά και αποβλέπουν στην επίτευξη ενός συγκεκριμένου σκοπού.
- **Δραστηριότητα** ή **εργασία**, είναι το δομικό υλικό του έργου.
- **Υποέργο**, είναι μέρος του έργου που εξυπηρετεί γεωγραφικές, οργανωτικές ή λειτουργικές ανάγκες.
- Πρόγραμμα έργων, είναι ένα σύνολο αλληλοσχετιζόμενων έργων που αποσκοπεί στην εκπλήρωση πολλών στόχων.

Η έννοια του έργου (συνέχεια)

- Χαρακτηριστικό ενός έργου είναι ότι πρόκειται για ένα σύνολο από αλληλοεξαρτώμενες μεταξύ τους εργασίες, που όλες μαζί έχουν σκοπό την επίτευξη συγκεκριμένων στόχων.
- Οι δραστηριότητες του έργου πρέπει να ολοκληρωθούν μέσα σε προκαθορισμένα όρια κόστους και χρόνου υλοποίησης, ικανοποιώντας ταυτόχρονα τις προδιαγραφές ποιότητας.

Βασικά Χαρακτηριστικά του Έργου

- Είναι μοναδικό
- Έχει αρχή και τέλος
- Απαιτεί την ανάλωση χρόνου, χρήματος και πόρων
- Πρέπει να ικανοποιεί συγκεκριμένες ποιοτικές προδιαγραφές
- Περιέχει στοιχεία επιχειρηματικού κινδύνου

Προβλήματα στην υλοποίηση ενός έργου

- σχετικά με τον χρόνο και
- σχετικά με το κόστος και τα μέσα παραγωγής.

Προβλήματα στην υλοποίηση ενός έργου σχετικά με το χρόνο

- Ποιος είναι ο χρόνος που θα χρειαστούμε για την ολοκλήρωση του έργου;
- Ποιος είναι ο χρόνος έναρξης και λήξης κάθε δραστηριότητας;
- Πως θα παρακολουθήσουμε τη χρονική εξέλιξη του έργου και πως θα αντιμετωπίσουμε απρόβλεπτες καθυστερήσεις;
- Ποια είναι η πιθανότητα να ολοκληρωθεί το έργο μέσα σε καθορισμένο χρονικό διάστημα;

Προβλήματα στην υλοποίηση ενός έργου σχετικά με το κόστος

- Ποια είναι τα μέσα παραγωγής;
- Πόσο θα μας κοστίσει το έργο συνολικά;
- Πώς μπορούμε να αντιμετωπίσουμε προβλήματα διαθεσιμότητας των μέσων παραγωγής;
- Ποιος είναι ο χρόνος έναρξης και λήξης καθεμιάς κάθε δραστηριότητας;

Μέθοδοι δικτυωτής ανάλυσης

- Μέθοδοι Δικτυωτής Ανάλυσης, που χρησιμοποιούν κατάλληλα δικτυωτά διαγράμματα (δίκτυα) για να απεικονίσουν τα διάφορα δομικά στοιχεία ενός έργου.
- Μέθοδος PERT και μέθοδος CPM.
- Οι δύο μέθοδοι έχουν κοινά χαρακτηριστικά και ομοιότητες, και χρησιμοποιούνται αδιακρίτως για τη διοίκηση έργων με τον όρο μέθοδος PERT/CPM.

Δομική ανάλυση του έργου

- Στο πρώτο στάδιο, αναλύουμε το σύνολο του έργου σε διακριτές κύριες φάσεις, σε αυτοτελή δηλαδή τμήματά του.
- Και στο δεύτερο στάδιο, αναλύουμε κάθε διακριτή φάση του έργου σε πακέτα εργασίας και αυτά σε δραστηριότητες.
- Τρία ή τέσσερα επίπεδα ανάλυσης του έργου αρκούν για να επιτευχθεί το επιθυμητό επίπεδο προγραμματισμού και ελέγχου

Και άλλες έννοιες

- Πρόγραμμα έργων
- Φάση / Υποέργο.
- Δραστηριότητα / Πακέτο εργασίας

Δομική ανάλυση του έργου (συνέχεια)

- Αναγνώριση και καταγραφή όλων των δραστηριοτήτων
- Αναγνώριση των τεχνολογικών περιορισμών που υπάρχουν στη σειρά εκτέλεσής τους.
- Σχέσεις εξάρτησης, ενώ υπόκεινται και σε διάφορους περιορισμούς
- Ποιες δραστηριότητες είναι άμεσα προηγούμενες της δραστηριότητας που εξετάζουμε;
- Ποιες δραστηριότητες είναι άμεσα επόμενες της δραστηριότητας που εξετάζουμε;
- Ποιες δραστηριότητες είναι ανεξάρτητες από αυτή που εξετάζουμε και μπορούν να εκτελούνται ταυτόχρονα;

Διαμόρφωση του δικτύου

- Τοξωτά δίκτυα ή δίκτυα με δραστηριότητες στα βέλη (Activity On Arrow).
- Κομβικά δίκτυα ή δίκτυα με δραστηριότητες στους κόμβους (Activity On Node).

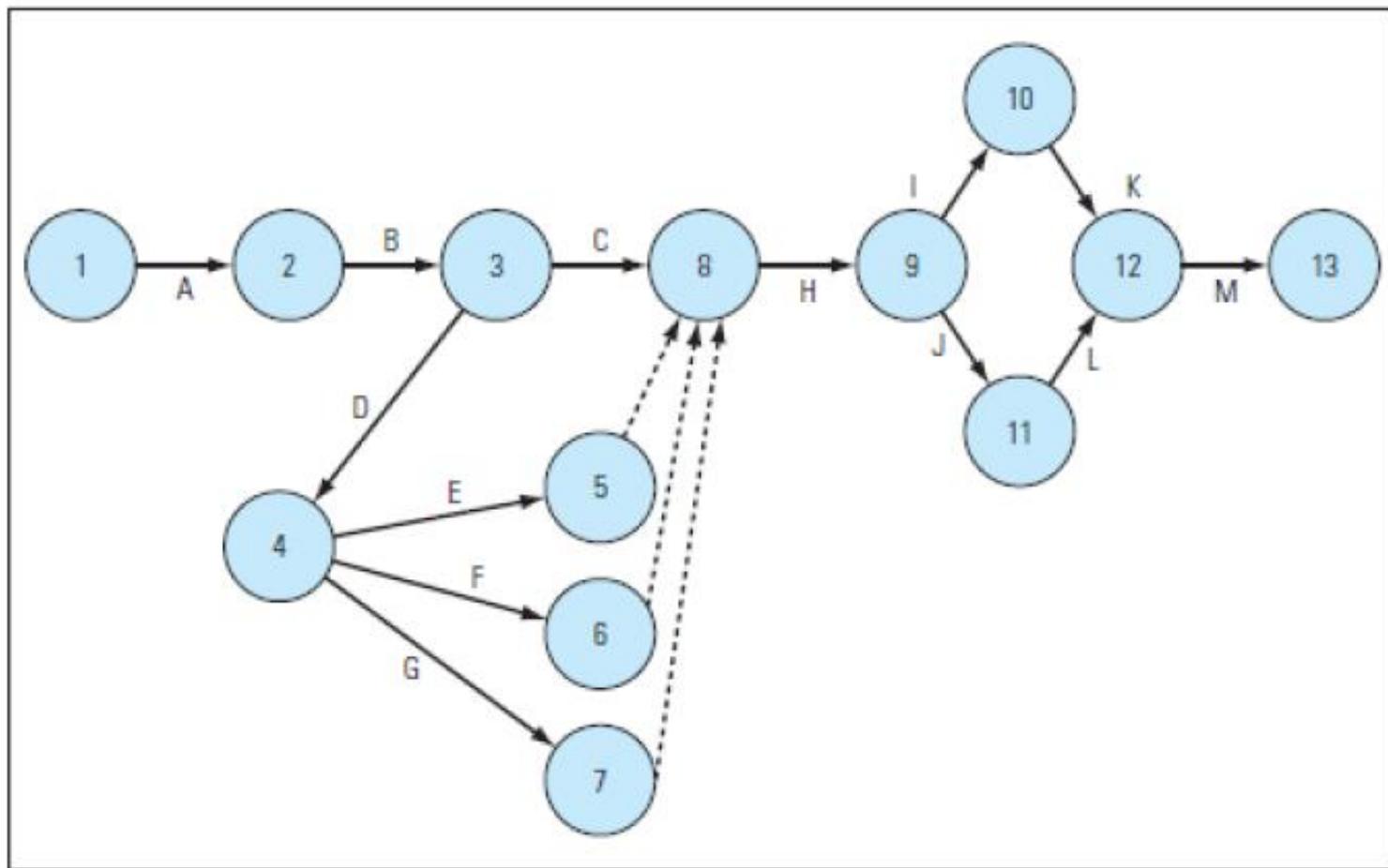
Διαμόρφωση του δικτύου (συνέχεια)

Ένα **δίκτυο** αποτελείται από **κόμβους** και **τόξα**. Ανάλογα με τον τρόπο ερμηνείας των κόμβων και τόξων, τα δικτυωτά γραφήματα περιλαμβάνουν τις κάτωθι δύο κατηγορίες:

1. **Toξωτά δίκτυα** με «**δραστηριότητες στα τόξα**» (activity-on-arrow ή πιο σύντομα AOA), στα οποία τα τόξα αντιπροσωπεύουν τις δραστηριότητες και οι κόμβοι συμβολίζουν την έναρξη και το πέρας των δραστηριοτήτων. Οι κόμβοι αρχής και λήξης κάθε δραστηριότητας ονομάζονται «**γεγονότα**» ή «**συμβάντα**».
2. **Κομβικά δίκτυα** με «**δραστηριότητες στους κόμβους**» (activity-on-node ή συντομογραφικά AON), στα οποία οι κόμβοι αντιπροσωπεύουν τις δραστηριότητες ενώ τα τόξα υποδεικνύουν τις σχέσεις αλληλεξάρτησης ανάμεσα στις δραστηριότητες (κόμβους) του έργου.

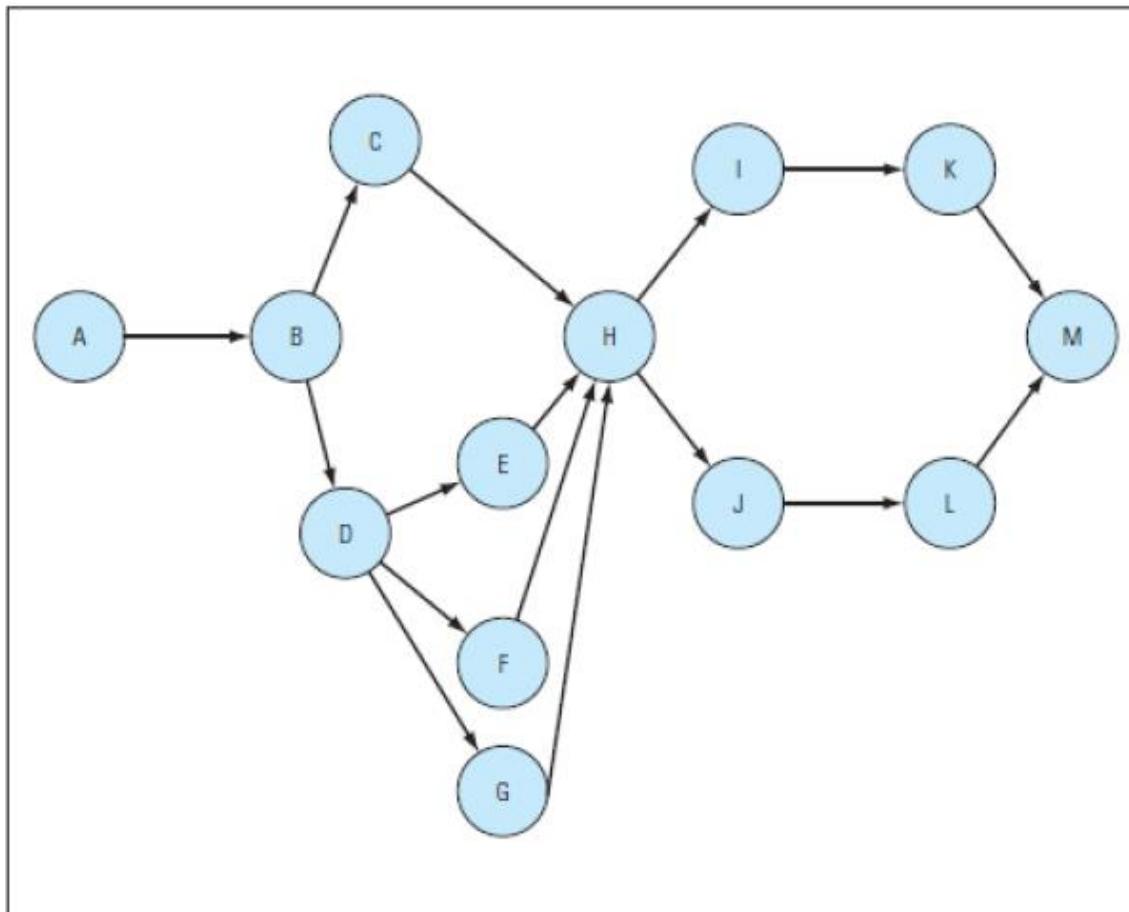
Διαμόρφωση του δικτύου (συνέχεια)

Τοξωτά δίκτυα με «δραστηριότητες στα τόξα» (activity-on-arrow ή πιο σύντομα AOA), στα οποία τα τόξα αντιπροσωπεύουν τις δραστηριότητες και οι κόμβοι συμβολίζουν την έναρξη και το πέρας των δραστηριοτήτων. Οι κόμβοι αρχής και λήξης κάθε δραστηριότητας ονομάζονται «γεγονότα» ή «συμβάντα».



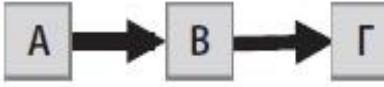
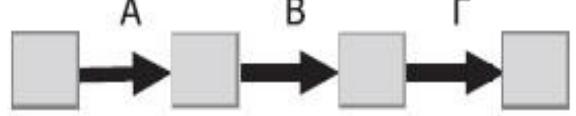
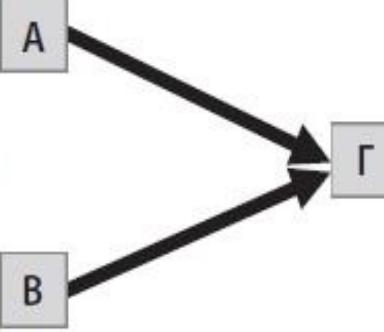
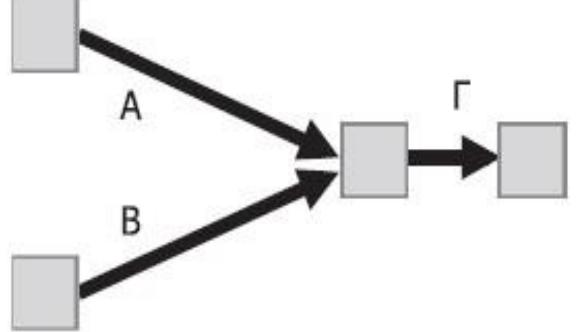
Διαμόρφωση του δικτύου (συνέχεια)

Κομβικά δίκτυα με «δραστηριότητες στους κόμβους» (activity-on-node ή συντομογραφικά AON), στα οποία οι κόμβοι αντιπροσωπεύουν τις δραστηριότητες ενώ τα τόξα υποδεικνύουν τις σχέσεις αλληλεξάρτησης ανάμεσα στις δραστηριότητες (κόμβους) του έργου.



Διαμόρφωση του δικτύου (συνέχεια)

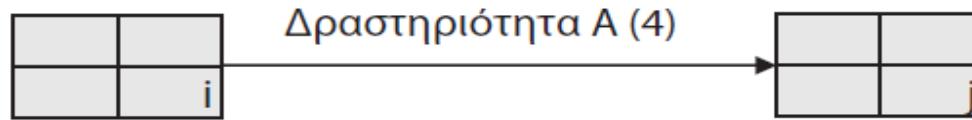
Σχήμα 8.1: Διαφορές στη γραφική παράσταση σχέσεων αλληλεξάρτησης έργου με τη βοήθεια τοξωτού ή κομβικού δικτύου

Κομβικά Δίκτυα (AoN)	Σχέσεις εξάρτησης	Τοξωτά Δίκτυα (AoA)
(α) 	Η δραστηριότητα Α προηγείται άμεσα της Β, η οποία προηγείται άμεσα της Γ.	
(β) 	Οι δραστηριότητες Α και Β είναι άμεσα προηγούμενες της δραστηριότητας Γ.	

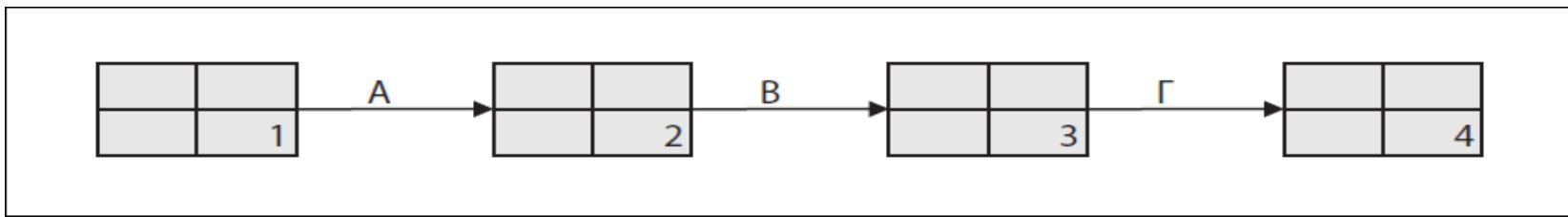
Διαμόρφωση του δικτύου (συνέχεια)

- Τα κομβικά δίκτυα είναι πράγματι πολύ πιο απλά και κατανοητά.
- Τα διάφορα μεγέθη του χρονικού και οικονομικού προγραμματισμού είναι πιο κατανοητά με τη χρήση τοξωτών δικτύων.
- Θα χρησιμοποιήσουμε αποκλειστικά και μόνο τοξωτά δίκτυα στη συνέχεια.

Διαμόρφωση του δικτύου (συνέχεια)

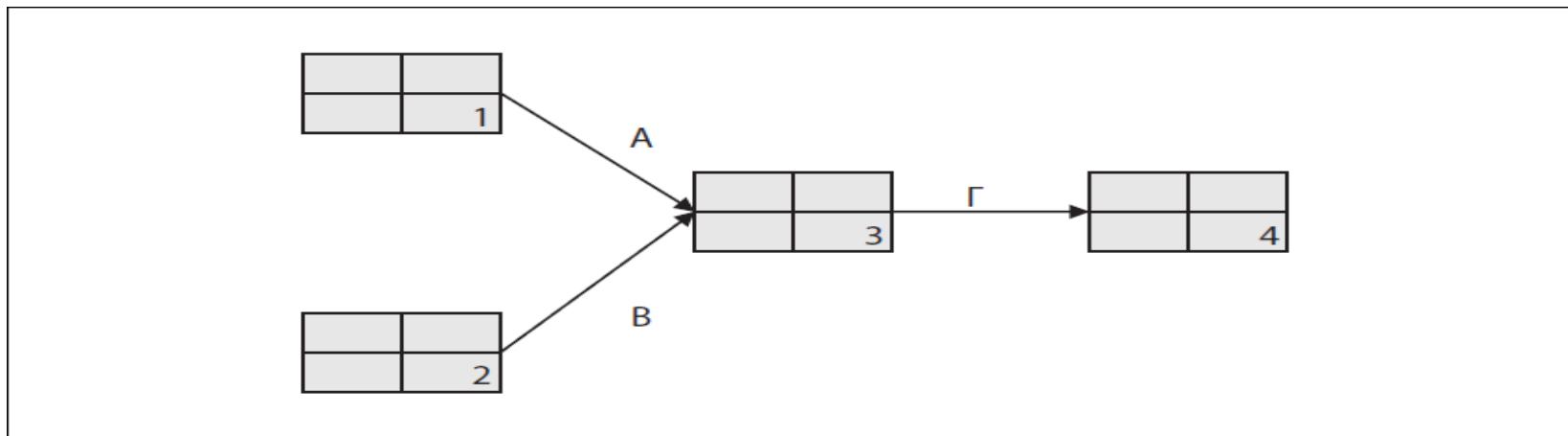


Διαμόρφωση του δικτύου (συνέχεια)



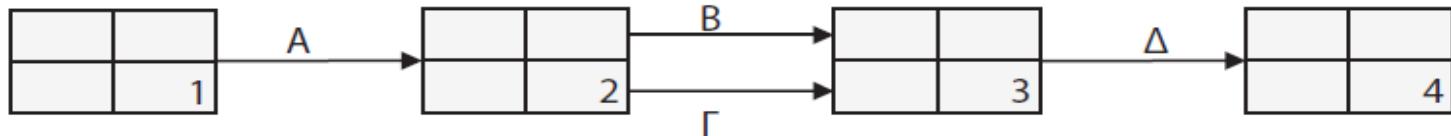
Διαμόρφωση του δικτύου (συνέχεια)

- Οι δύο δραστηριότητες A και B είναι άμεσα προηγούμενες της δραστηριότητας Γ

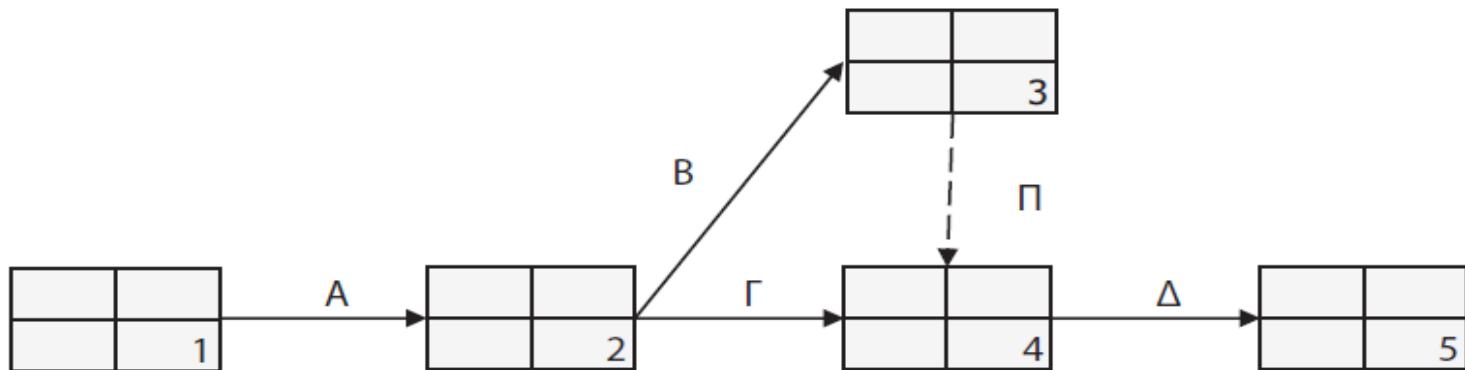


Διαμόρφωση του δικτύου (συνέχεια)

Σχήμα 8.5: Χρήση πλασματικής δραστηριότητας σε τοξωτά δίκτυα



(α): Λανθασμένος τρόπος παράστασης των σχέσεων εξάρτησης



Π: Πλασματική δραστηριότητα με μηδενική διάρκεια

(β): Σωστός τρόπος παράστασης των σχέσεων εξάρτησης

Γιατί η δικτυωτή ανάλυση;

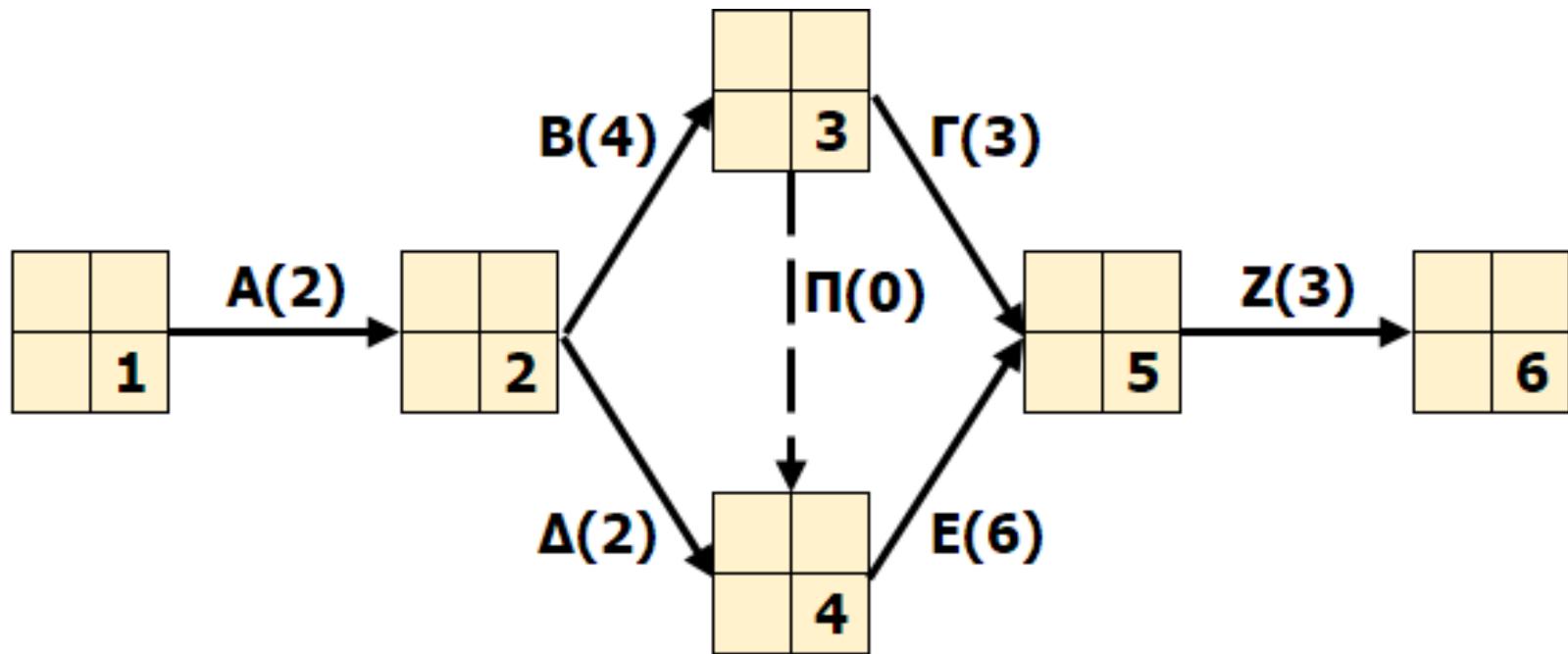
1. Ποια είναι η αναμενόμενη ημερομηνία ολοκλήρωσης (διάρκεια) του έργου;
2. Ποια είναι η πιθανότητα «μεταβλητότητας» αυτής της ημερομηνίας (διάρκειας);
3. Ποιες είναι οι προγραμματισμένες ημερομηνίες έναρξης και λήξης για κάθε δραστηριότητα;
4. Ποιες δραστηριότητες είναι κρίσιμες υπό την έννοια ότι πρέπει να ολοκληρωθούν ακριβώς όπως έχουν προγραμματισθεί προκειμένου να επιτευχθεί ο στόχος της ολοκλήρωσης του έργου;
5. Πόσο χρονικό διάστημα μπορούν να καθυστερήσουν οι **μη κρίσιμες** δραστηριότητες χωρίς να προκληθεί καθυστέρηση στη συνολική διάρκεια του έργου;
6. Πώς θα μπορούσαν οι απαιτούμενοι πόροι του έργου να κατανεμηθούν πιο αποτελεσματικά στις δραστηριότητες, προκειμένου να επιταχυνθεί η ολοκλήρωση του έργου;
7. Ποιοί έλεγχοι πρέπει να ασκηθούν στις ταμειακές εκροές (δαπάνες) υλοποίησης των διάφορων δραστηριοτήτων κατά τη διάρκεια του έργου, ώστε να μην προκύψει υπέρβαση του συνολικού προϋπολογισμού του έργου;

Άσκηση

Δραστηριότητα	Άμεσα προηγούμενη δραστηριότητα	Διάρκεια [σε μήνες]
A	--	2,0
B	A	4,0
Γ	B	3,0
Δ	A	2,0
E	B, Δ	6,0
Z	Γ, E	3,0

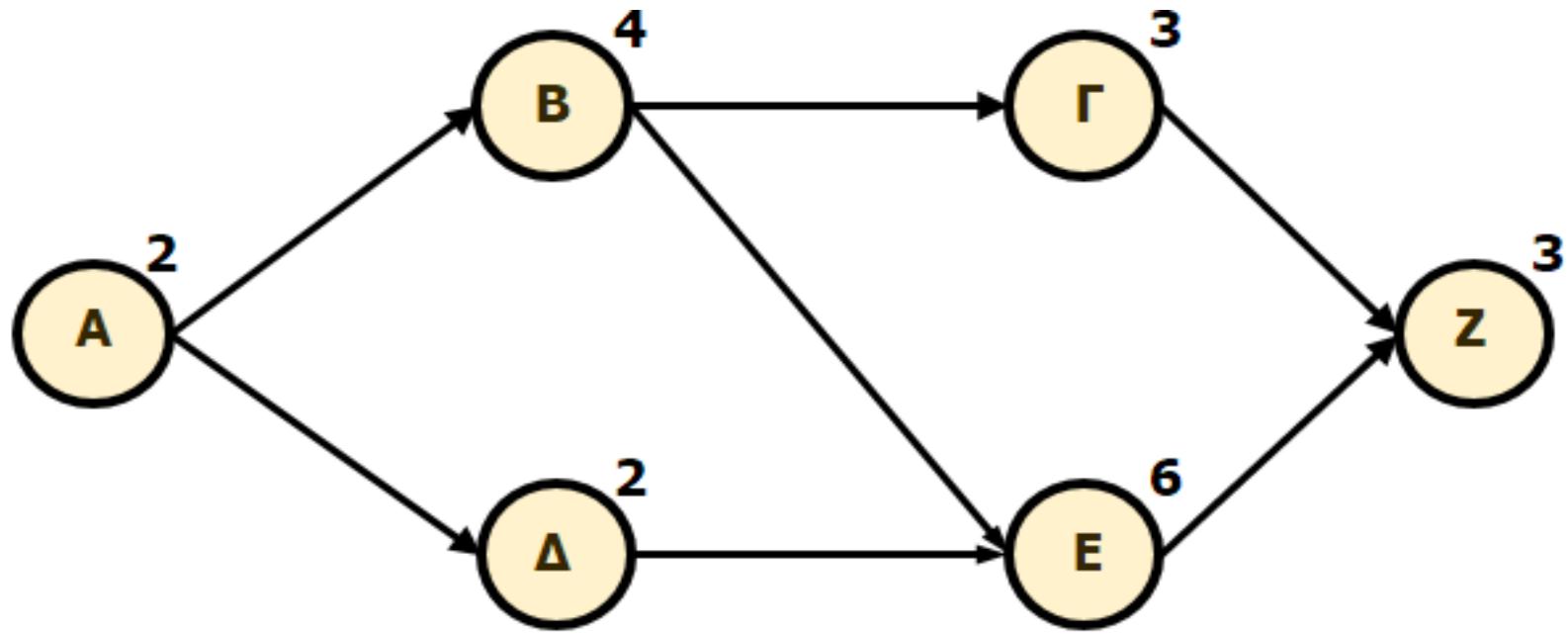
Με τα στοιχεία αυτά να διαμορφώσετε ένα κατάλληλο δίκτυο, που να απεικονίζει γραφικά το έργο.

Ενδεικτική απάντηση



(α) Τοξωτό δίκτυο γραφικής απεικόνισης του έργου

Ενδεικτική απάντηση (συνέχεια)



(β) Κομβικό δίκτυο γραφικής απεικόνισης του έργου

Παράδειγμα

Παράδειγμα 8.1: Για την ολοκλήρωση ενός έργου απαιτείται η εκτέλεση ενός αριθμού δραστηριοτήτων. Οι δραστηριότητες αυτές, οι διάρκειες τους και οι περιορισμοί που υπάρχουν στην εκτέλεσή τους, σημειώνονται στον πίνακα που ακολουθεί:

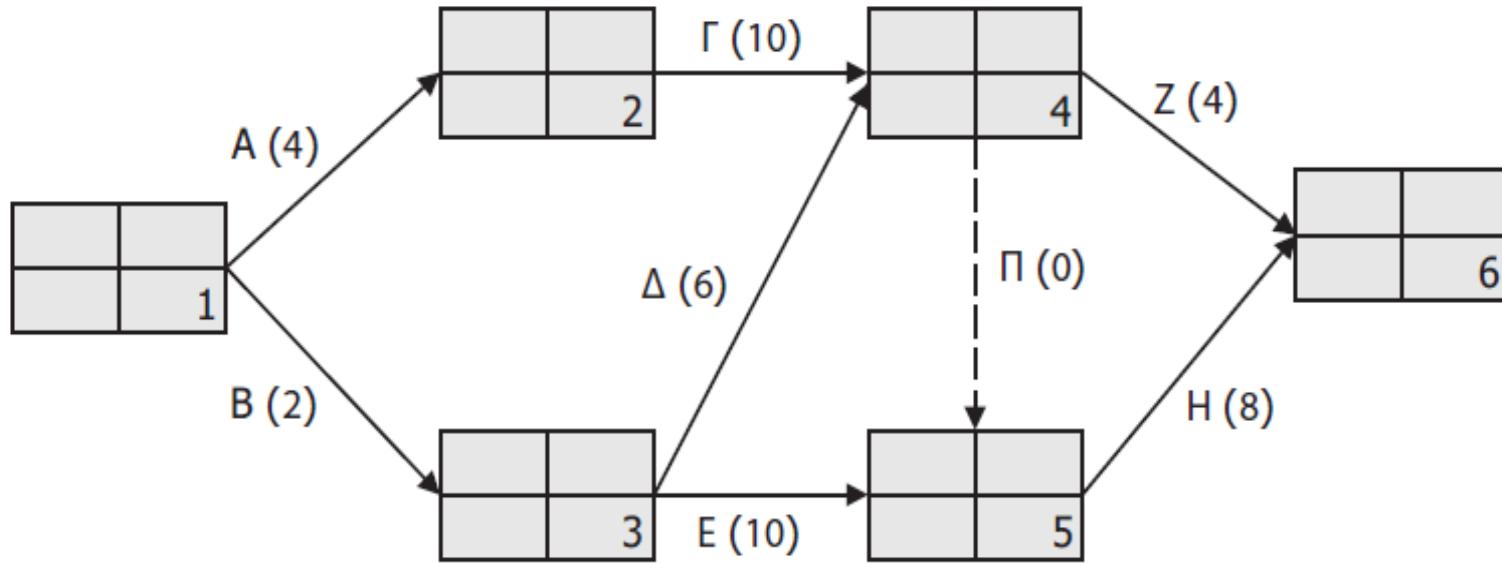
Πίνακας 8.1: Αριθμητικά δεδομένα Παραδείγματος 8.1

Δραστηριότητα	Άμεσα προηγούμενη δραστηριότητα	Διάρκεια (σε ημέρες)
A	—	4,0
B	—	2,0
Γ	A	10,0
Δ	B	6,0
Ε	B	10,0
Ζ	Γ, Δ	4,0
Η	Γ, Δ, Ε	8,0

Ζητείται να διαμορφωθεί το κατάλληλο τοξωτό δίκτυο που θα απεικονίζει γραφικά το έργο που μας απασχολεί.

Παράδειγμα

Σχήμα 8.6: Τοξωτό δίκτυο του Παραδείγματος 8.1



Π: Πλασματική δραστηριότητα με μηδενική διάρκεια

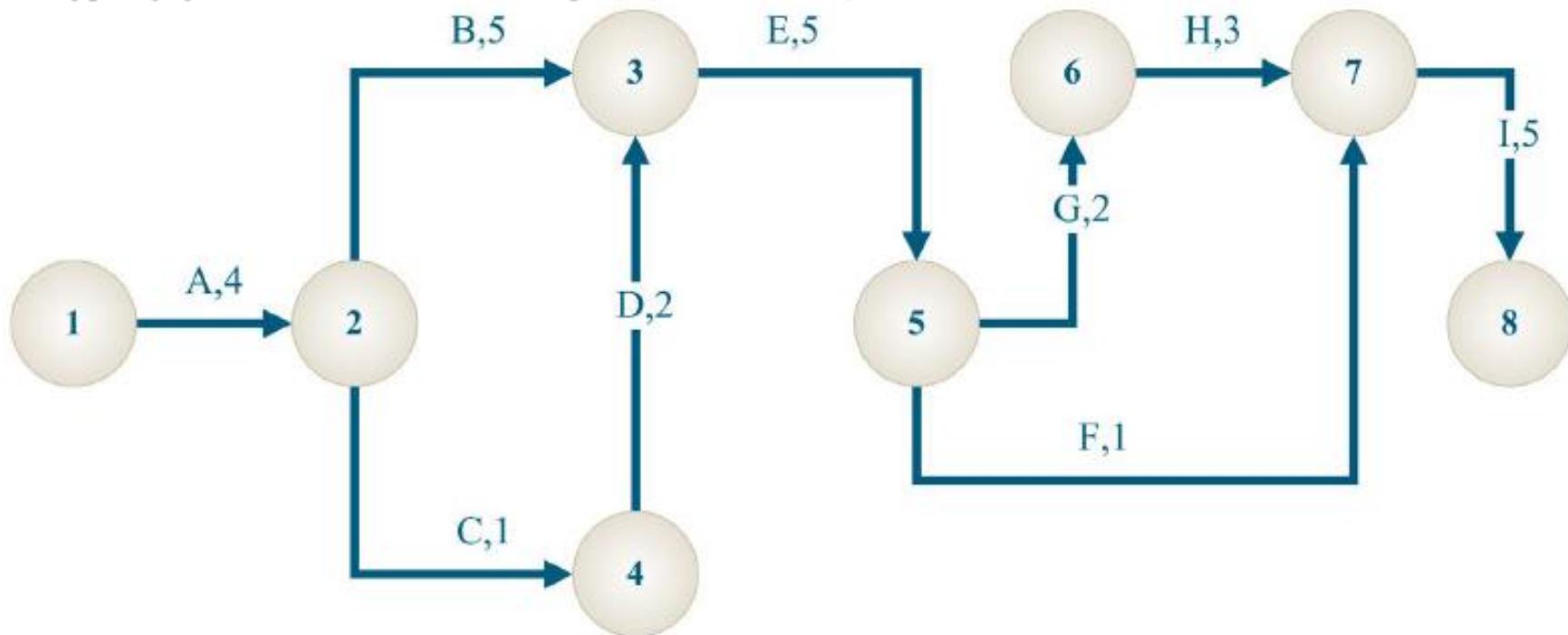
Παράδειγμα

Δραστηριότητα	Προηγούμενη	Διάρκεια
A		4
B	A	5
C	A	1
D	C	2
E	B, D	5
F	E	1
G	E	2
H	G	3
I	F, H	5

Παράδειγμα

Δραστηριότητα	Προηγούμενη	Διάρκεια
A		4
B	A	5
C	A	1
D	C	2
E	B, D	5
F	E	1
G	E	2
H	G	3
I	F, H	5

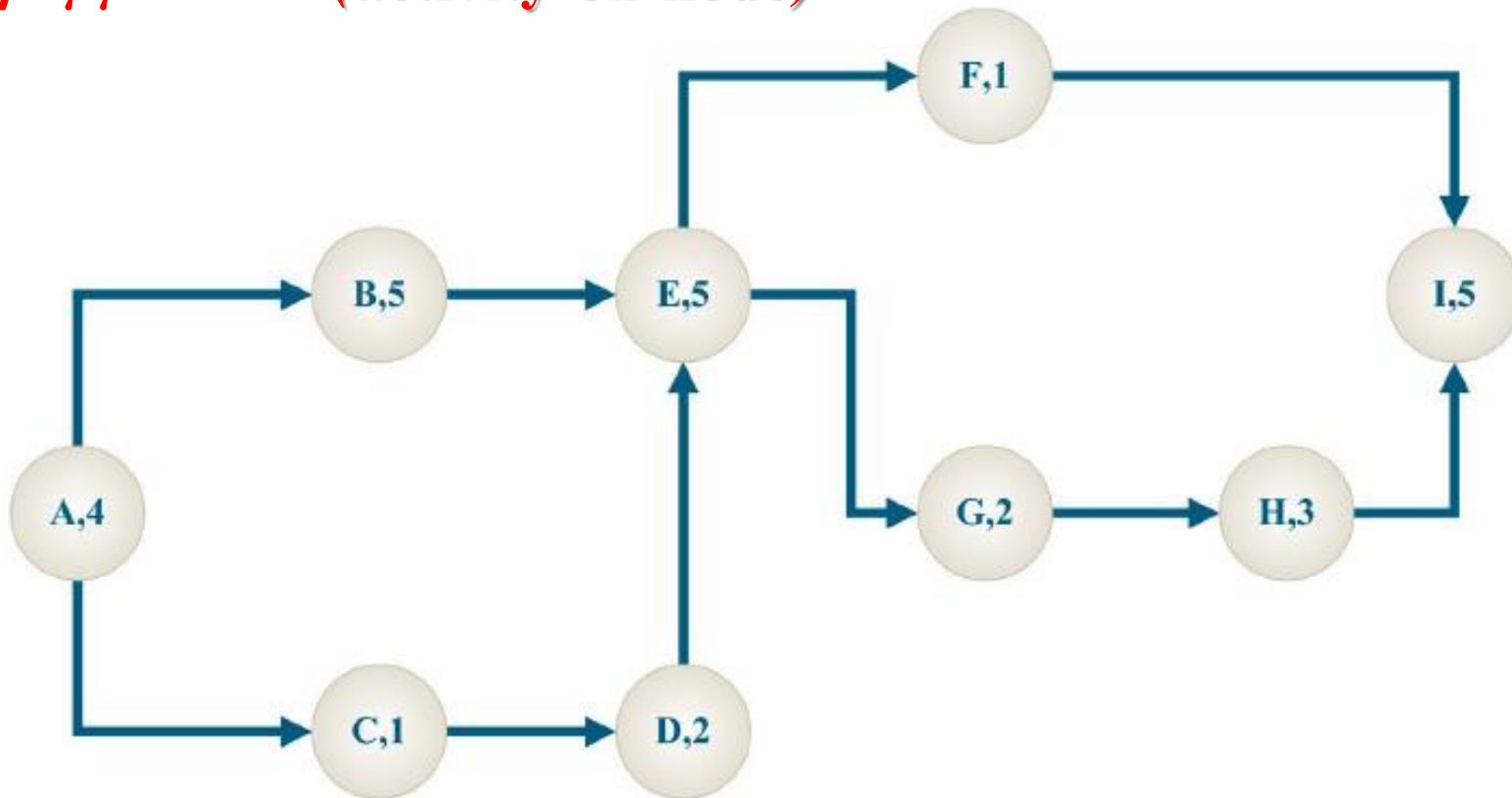
Διάγραμμα AOA (activity-on-arrow)



Παράδειγμα

Δραστηριότητα	Προηγούμενη	Διάρκεια
A		4
B	A	5
C	A	1
D	C	2
E	B, D	5
F	E	1
G	E	2
H	G	3
I	F, H	5

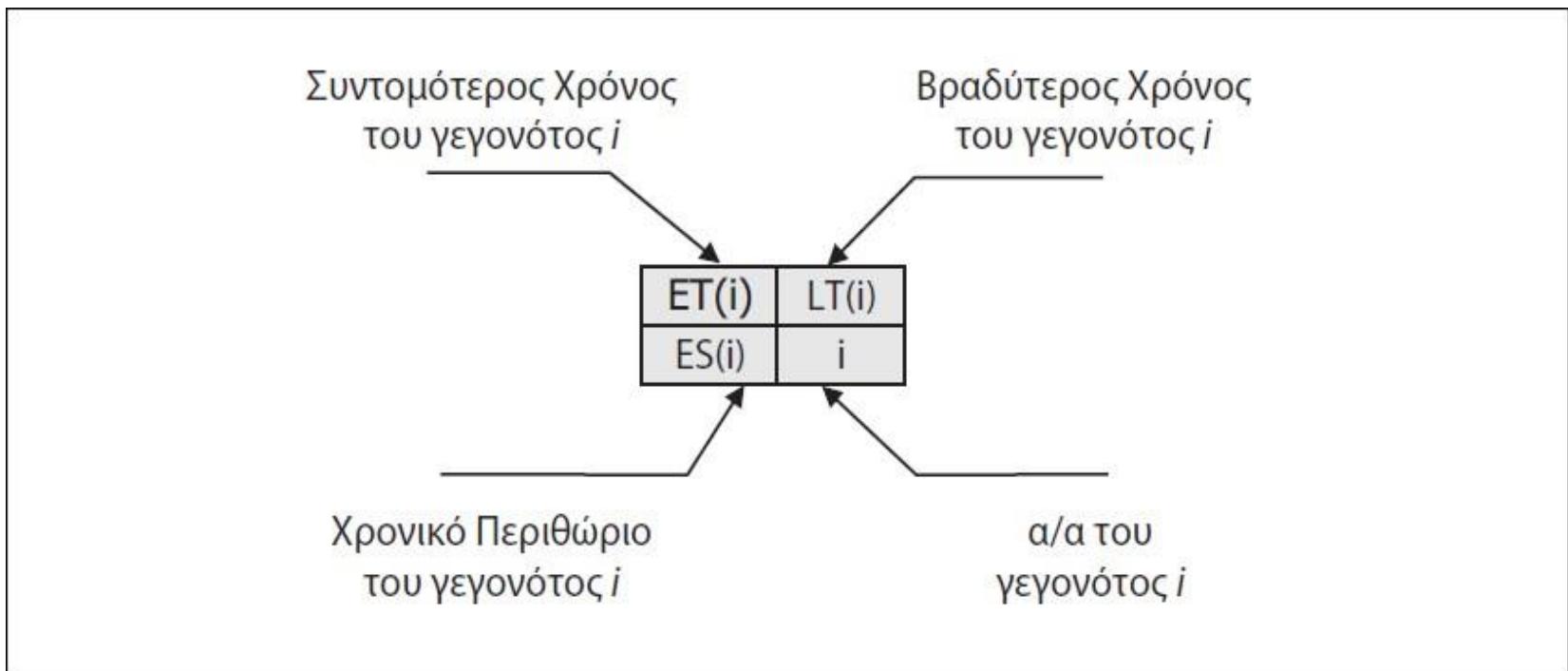
Διάγραμμα AON (activity-on-node)



Προγραμματισμός έργου με τη μέθοδο PERT/CPM

Προγραμματισμός έργου με τη μέθοδο PERT/CPM

- Το πρώτο στάδιο της μεθόδου PERT/CPM είναι η επίλυση του τοξωτού δικτύου, που περιλαμβάνει τον υπολογισμό τριών αριθμητικών μεγεθών για καθένα από τα γεγονότα του δικτύου



Προγραμματισμός έργου με τη μέθοδο PERT/CPM

$$ET(i) = \max \{ET(k) + t(ki)\}, \forall k \in P, i = 1, 2, \dots, n \text{ και } ET(1) = 0 \quad (8.1)$$

όπου: P το σύνολο των γεγονότων που προηγούνται του γεγονότος i και συνδέονται άμεσα με αυτό, και
 $t(ki)$ η διάρκεια της δραστηριότητας με γεγονός έναρξης το γεγονός k και γεγονός λήξης το γεγονός i .

$$LT(i) = \min \{LT(k) - t(ik)\}, \forall k \in S, i = n, n-1, \dots, 1 \text{ και } LT(n) = ET(n) \quad (8.2)$$

όπου: S το σύνολο των γεγονότων που έπονται του γεγονότος i και συνδέονται άμεσα με αυτό, και
 $t(ik)$ η διάρκεια της δραστηριότητας με γεγονός έναρξης το γεγονός k και γεγονός λήξης το γεγονός i .

$$ES(i) = LT(i) - ET(i), i = 1, 2, \dots, n \quad (8.3)$$

Προγραμματισμός έργου με τη μέθοδο PERT/CPM

$$ES(ij) = ET(i) \quad (8.4)$$

$$EF(ij) = ES(ij) + t(ij) = ET(i) + t(ij) \quad (8.5)$$

$$LF(ij) = LT(j) \quad (8.6)$$

$$LS(ij) = LF(ij) - t(ij) = LT(j) - t(ij) \quad (8.7)$$

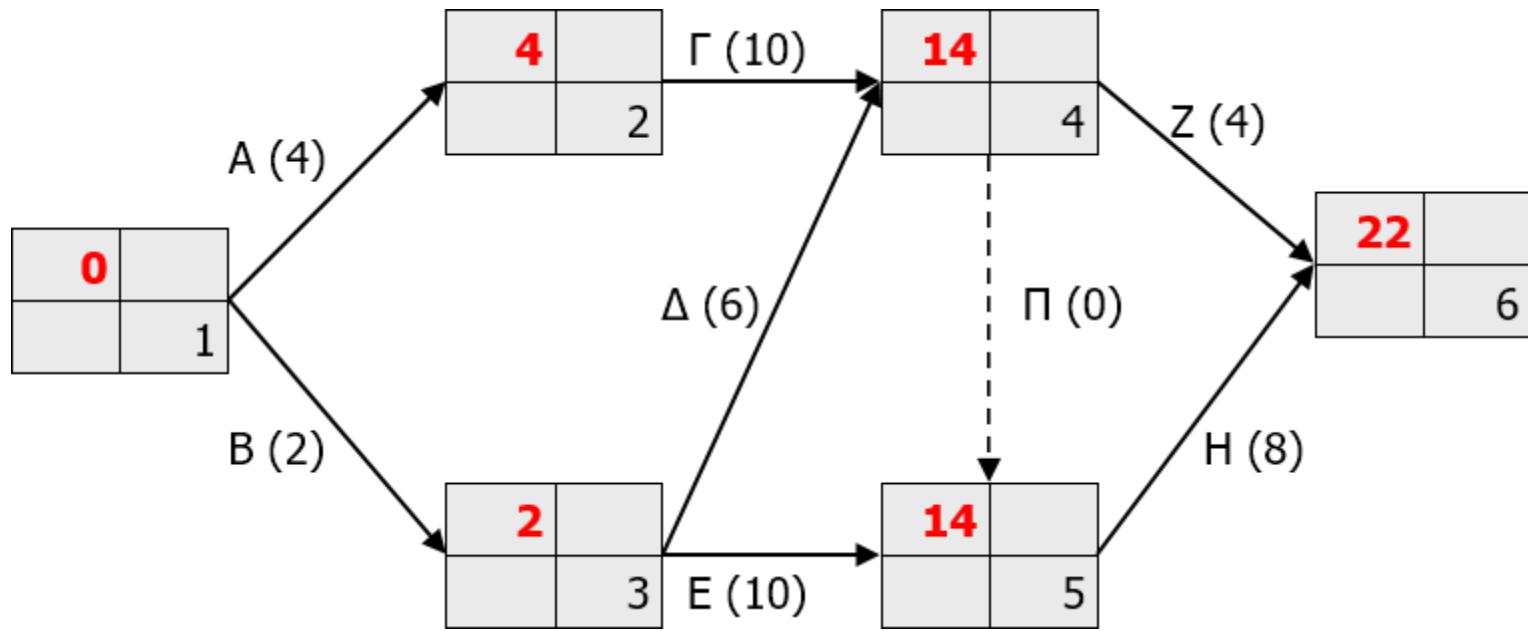
$$ST(ij) = LS(ij) - ES(ij) = LF(ij) - EF(ij) \quad (8.8)$$

Παράδειγμα 8.2

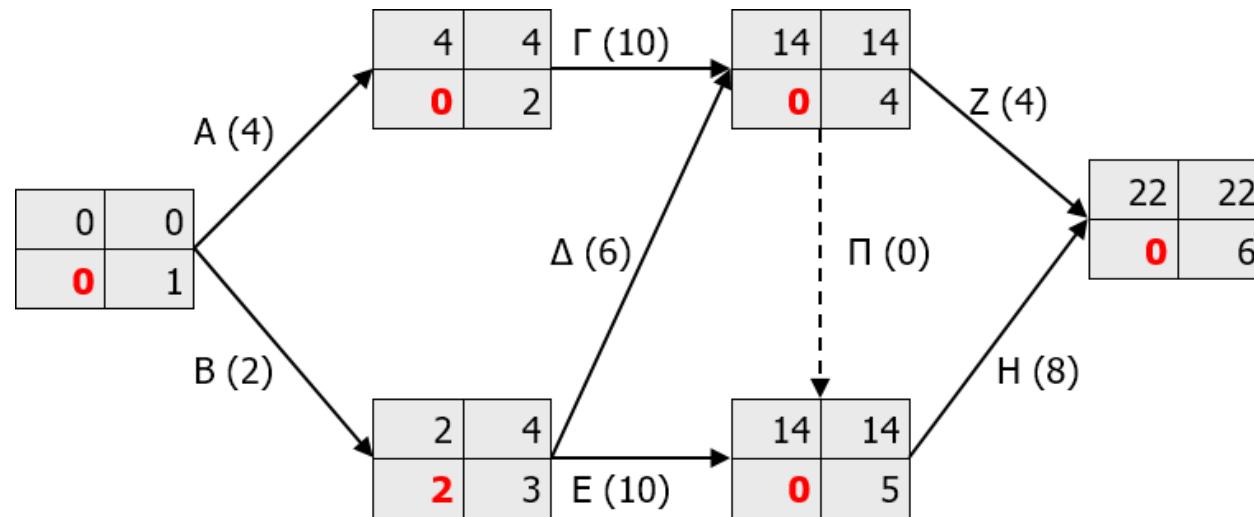
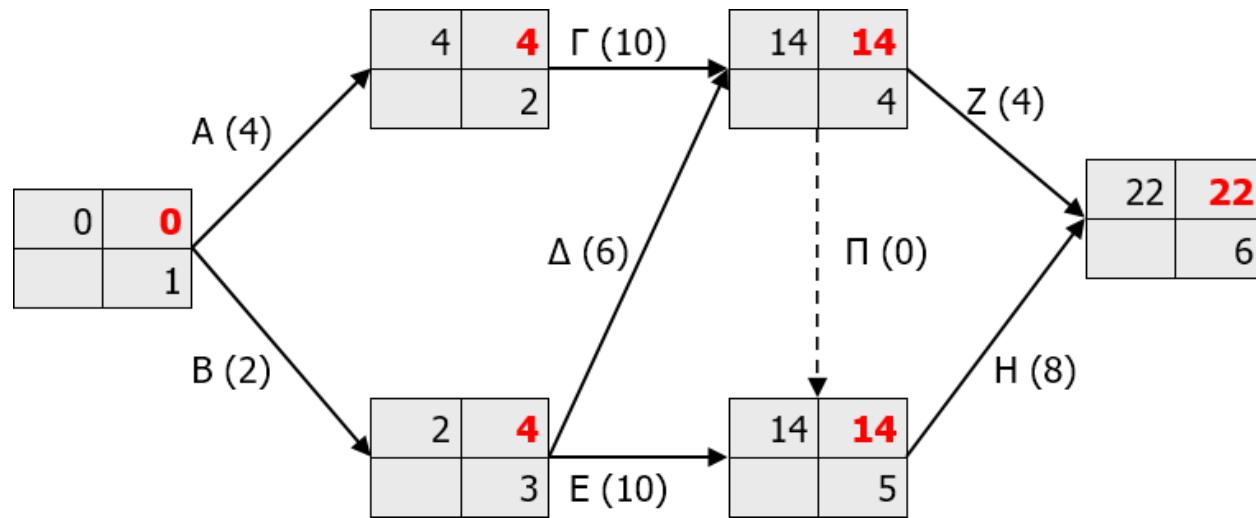
Παράδειγμα 8.2: Για τα αριθμητικά δεδομένα του έργου του Παραδείγματος 8.1, που μας απασχόλησε προηγουμένως, ζητούνται τα εξής:

- (α) Ποιος είναι ο μικρότερος χρόνος ολοκλήρωσης του έργου; Ποια θα είναι η ημερομηνία ολοκλήρωσης του έργου αν υποθέσουμε ότι η έναρξη των εργασιών του θα ξεκινήσει τη Δευτέρα 23 Μαρτίου 2020;
- (β) Πότε πρέπει να προγραμματιστεί η έναρξη και η λήξη των επιμέρους εργασιών, ώστε να ολοκληρωθεί το έργο στον μικρότερο χρόνο του;
- (γ) Ποιες είναι οι δραστηριότητες που πρέπει να ολοκληρωθούν χωρίς καθυστέρηση σε προκαθορισμένα στενά χρονικά περιθώρια και
- (δ) Ποιες είναι οι δραστηριότητες, που αν για διάφορους λόγους καθυστερήσει η ολοκλήρωσή τους αυτό δεν θα επηρεάσει το χρόνο ολοκλήρωσης του έργου συνολικά;

Παράδειγμα 8.2



Παράδειγμα 8.2

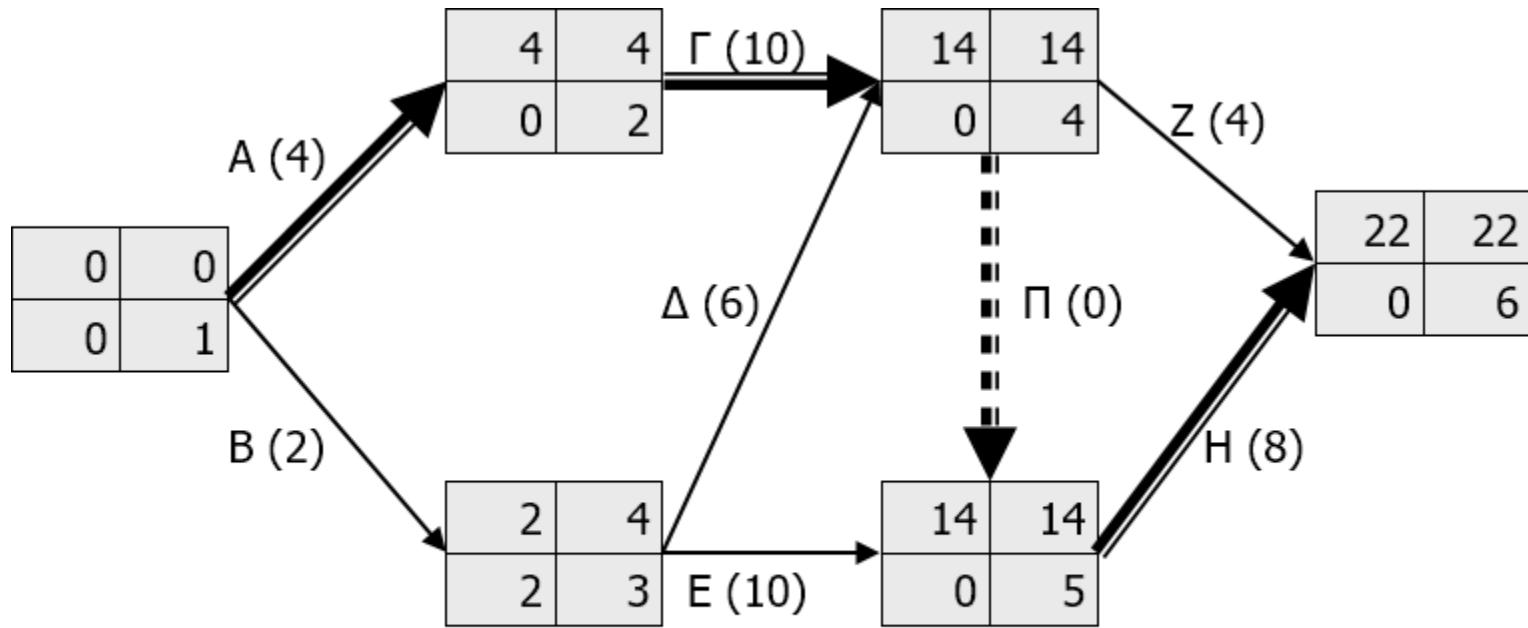


Παράδειγμα 8.2

Πίνακας 8.2: Προγραμματισμός δραστηριοτήτων του έργου του Παραδείγματος 8.2

Δραστηριότητα	Διάρκεια	ES	EF	LS	LF	ST
A	4	0	4	0	4	0
B	2	0	2	2	4	2
Γ	10	4	14	4	14	0
Δ	6	2	8	8	14	6
Ε	10	2	12	4	14	2
Π	0	14	14	14	14	0
Ζ	4	14	18	18	22	4
Η	8	14	22	14	22	0

Παράδειγμα 8.2



A - Γ - Π - H : κρίσιμη διαδρομή

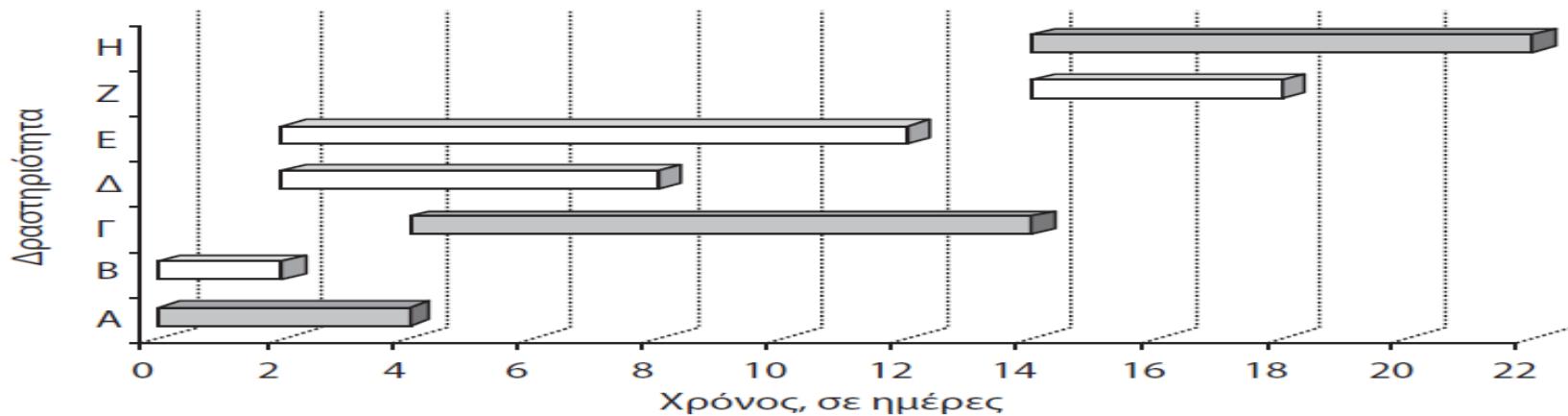
Παράδειγμα 8.2

- Η έναρξη κάθε δραστηριότητας του έργου μπορεί να προγραμματιστεί σε οποιαδήποτε χρονική στιγμή ανάμεσα στον συντομότερο και στον βραδύτερο χρόνο έναρξής της, χρόνοι που σημειώνονται στις στήλες ES και LS αντίστοιχα στον Πίνακα 8.2.
- Το διάγραμμα Gantt είναι στην πραγματικότητα ένα απλό γραμμικό ημερολόγιο, πάνω στο οποίο σημειώνουμε τους χρόνους έναρξης και λήξης των δραστηριοτήτων του έργου που μας απασχολεί.

Παράδειγμα 8.2

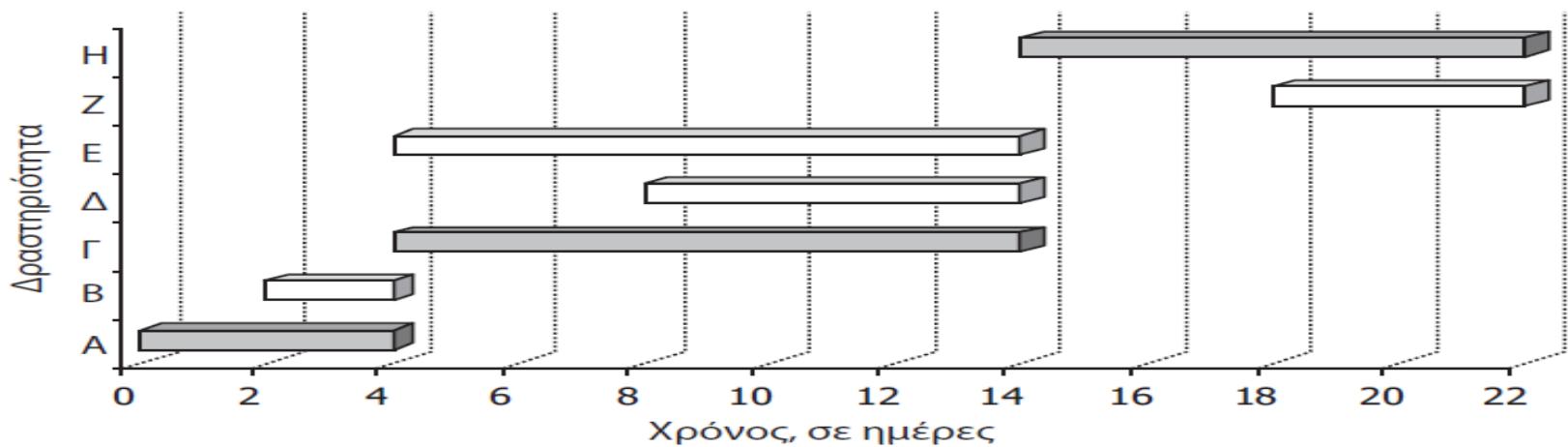
Σχήμα 8.9: Προγραμματισμός με βάση τους συντομότερους χρόνους έναρξης

Διάγραμμα Gantt: Προγραμματισμός στον συντομότερο χρόνο έναρξης



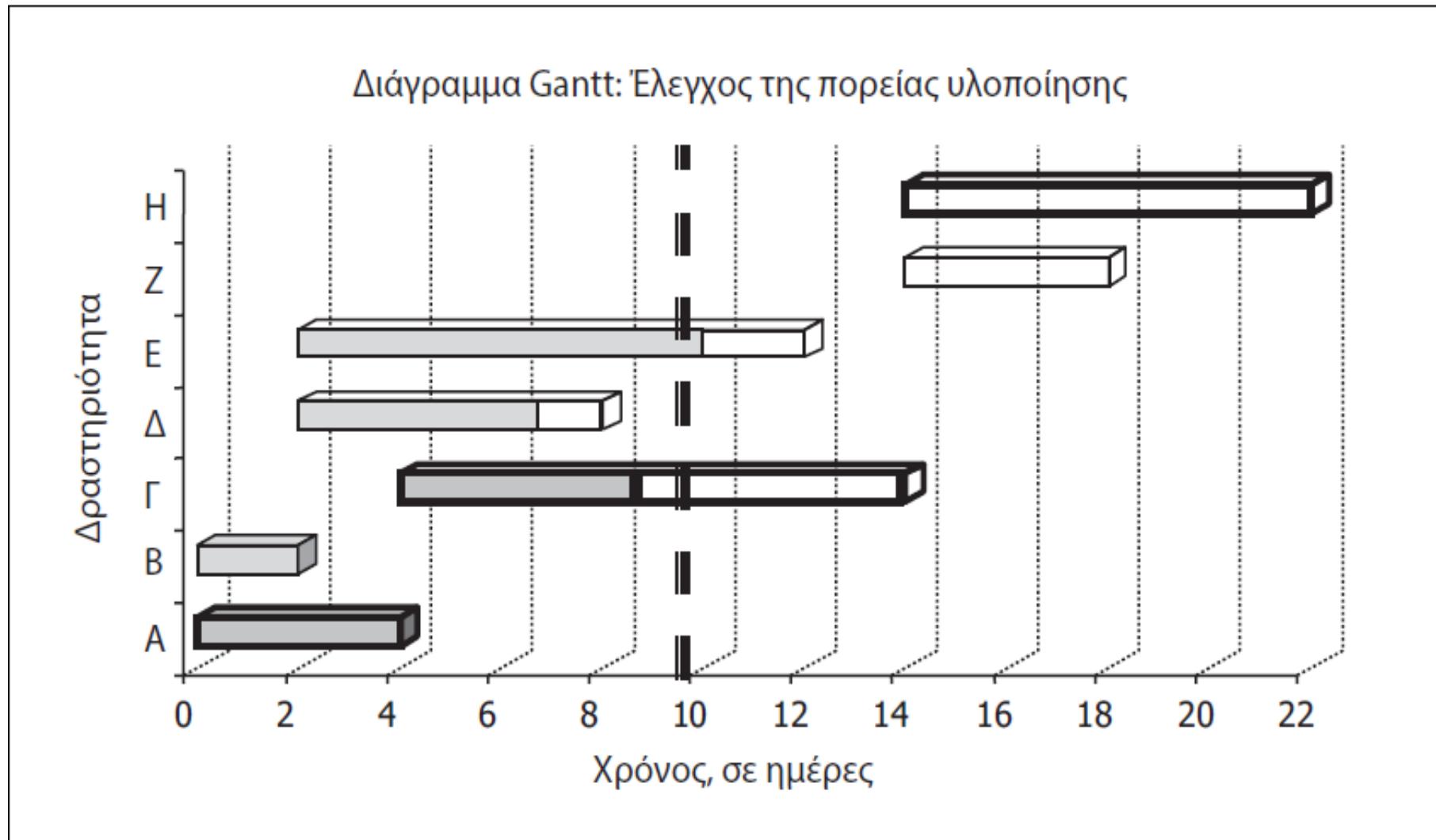
Σχήμα 8.11: Προγραμματισμός με βάση τους βραδύτερους χρόνους έναρξης

Διάγραμμα Gantt: Προγραμματισμός στον βραδύτερο χρόνο έναρξης



Παράδειγμα 8.2

Σχήμα 8.10: Διάγραμμα Gantt για τον έλεγχο της πορείας υλοποίησης του έργου



Παράδειγμα 8.3

Παράδειγμα 8.3: Για την ολοκλήρωση ενός έργου απαιτείται η εκτέλεση ενός αριθμού δραστηριοτήτων για καθεμία από τις οποίες θα απασχοληθεί ένας ειδικευμένος εργαζόμενος. Οι δραστηριότητες αυτές, οι διάρκειες τους και οι περιορισμοί, που υπάρχουν στην εκτέλεσή τους, σημειώνονται στον Πίνακα 8.3, που ακολουθεί:

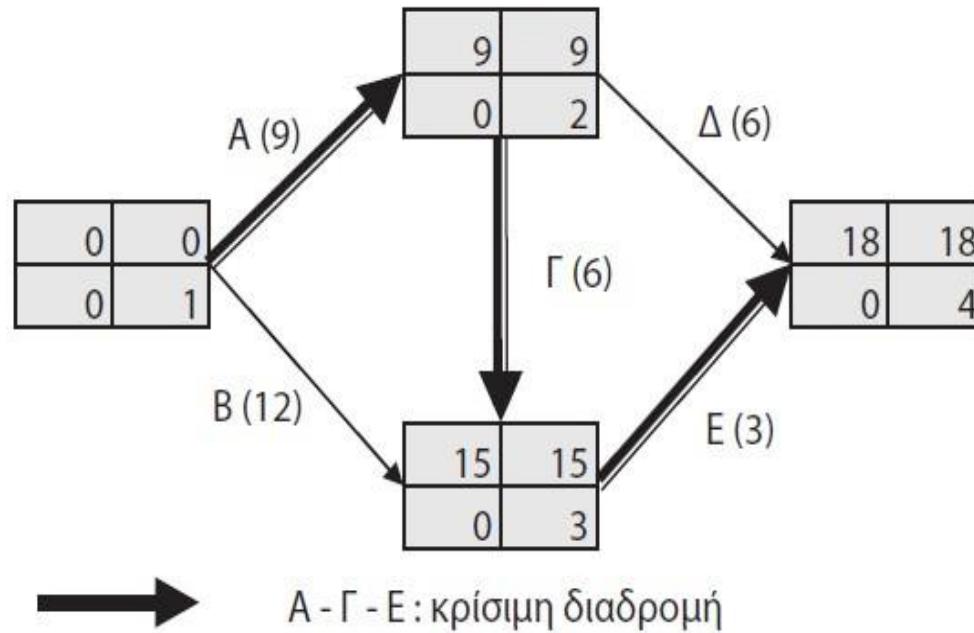
Πίνακας 8.3: Αριθμητικά δεδομένα Παραδείγματος 8.3

Δραστηριότητα	Άμεσα προηγούμενη δραστηριότητα	Διάρκεια (σε ημέρες)
A	–	9,0
B	–	12,0
Γ	A	6,0
Δ	A	6,0
Ε	B, Γ	3,0

Ζητείται να διαμορφωθεί ο κατάλληλος προγραμματισμός υλοποίησης του έργου, έτσι ώστε το έργο να ολοκληρωθεί στη συντομότερη διάρκειά του και η ζήτηση σε ειδικευμένο ανθρώπινο δυναμικό στη διάρκεια του χρόνου υλοποίησης να είναι η μικρότερη δυνατή.

Παράδειγμα 8.3

Σχήμα 8.12: Επίλυση δικτύου του Παραδείγματος 8.3



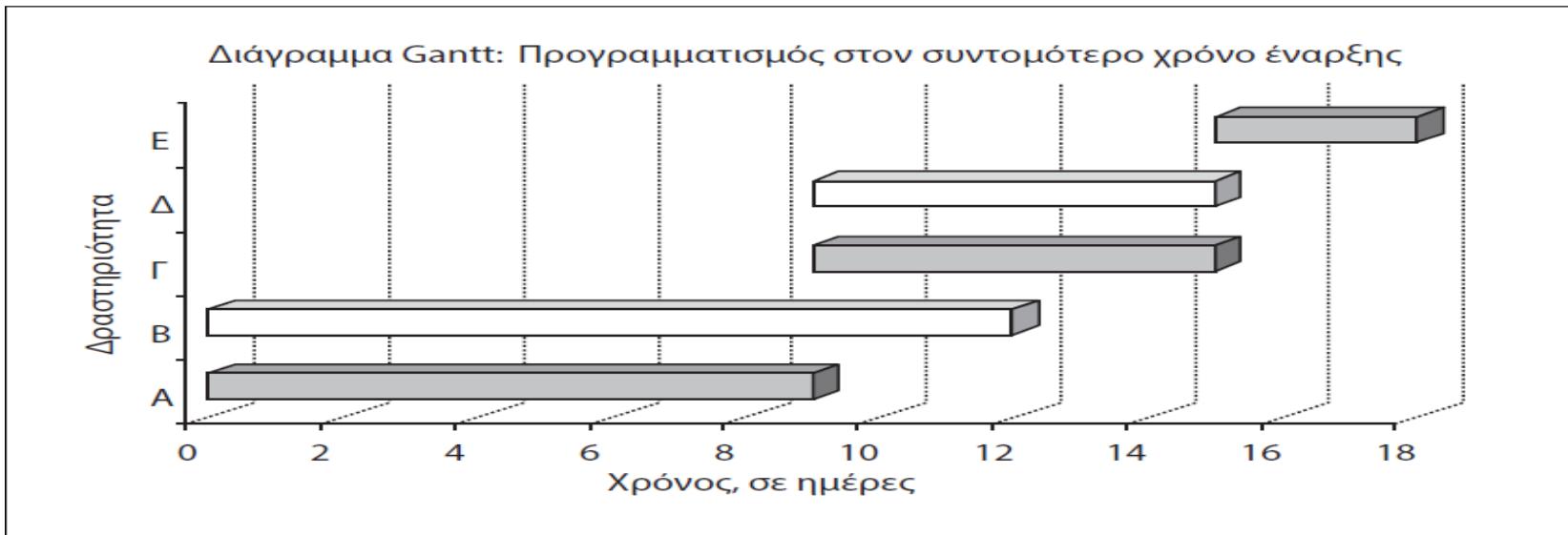
Παράδειγμα 8.3

Πίνακας 8.4: Προγραμματισμός δραστηριοτήτων του έργου του Παραδείγματος 8.3

Δραστηριότητα	Διάρκεια	ES	EF	LS	LF	ST
A	9	0	9	0	9	0
B	12	0	12	3	15	3
Γ	6	9	15	9	15	0
Δ	6	9	15	12	18	3
Ε	3	15	18	15	18	0

Παράδειγμα 8.3

Σχήμα 8.13: Προγραμματισμός με βάση τους συντομότερους χρόνους έναρξης



Σχήμα 8.14: Ζήτηση σε δυναμικό με βάση τους συντομότερους χρόνους έναρξης



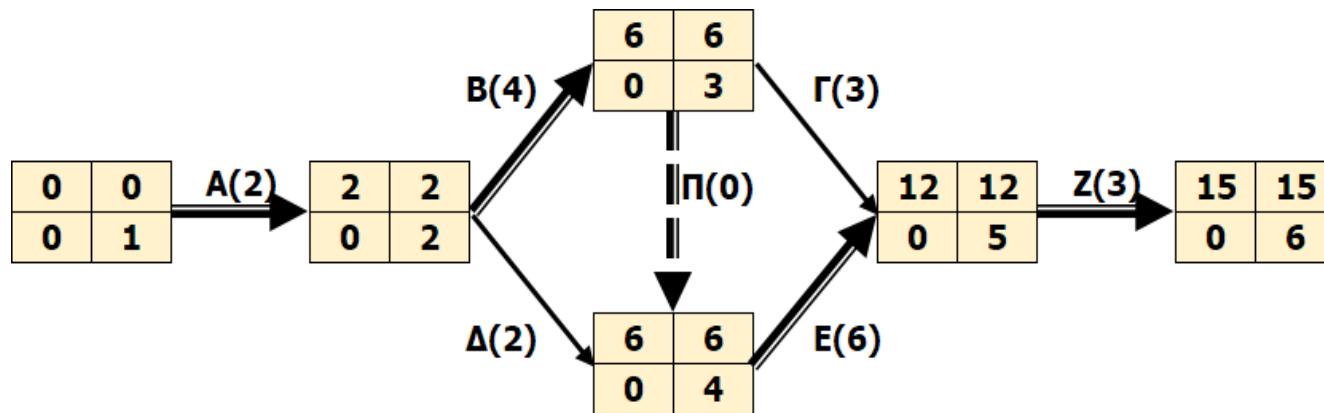
Άσκηση / Δραστηριότητα :

- Για τα δεδομένα του έργου, που σημειώνονται στον πίνακα που ακολουθεί, να βρεθούν ο συντομότερος χρόνος ολοκλήρωσης του έργου συνολικά, οι συντομότεροι και οι βραδύτεροι χρόνοι έναρξης και λήξης των επί μέρους δραστηριοτήτων του, τα συνολικά χρονικά τους περιθώρια και η κρίσιμη διαδρομή.

<i>Δραστηριότητα</i>	<i>Άμεσα προηγούμενη δραστηριότητα</i>	<i>Διάρκεια [σε μήνες]</i>
<i>A</i>	--	<i>2,0</i>
<i>B</i>	<i>A</i>	<i>4,0</i>
<i>Γ</i>	<i>B</i>	<i>3,0</i>
<i>Δ</i>	<i>A</i>	<i>2,0</i>
<i>E</i>	<i>B, Δ</i>	<i>6,0</i>
<i>Z</i>	<i>Γ, E</i>	<i>3,0</i>

Αν όλες οι άλλες δραστηριότητες ολοκληρωθούν στην προκαθορισμένη διάρκεια τους, ποιος είναι ο μέγιστος χρόνος, που η δραστηριότητα Δ μπορεί να καθυστερήσει, χωρίς αυτό να επηρεάσει τον συντομότερο χρόνο ολοκλήρωσης του έργου συνολικά;

Ενδεικτική απάντηση:



Δραστηριότητα	Διάρκεια	ES	EF	LS	LF	ST
A	2	0	2	0	2	0
B	4	2	6	2	6	0
Γ	3	6	9	9	12	3
Δ	2	2	4	4	6	2
Π	0	6	6	6	6	0
E	6	6	12	6	12	0
Z	3	12	15	12	15	0

Ασκήσεις για λύση

Ζητούνται:

- 1 Να σχεδιαστεί το δίκτυο και να υπολογιστεί το κρίσιμο μονοπάτι και ο αναμενόμενος χρόνος ολοκλήρωσης του έργου.**
- 2 να υπολογισθούν, ο νωρίτερος χρόνος έναρξης, ο βραδύτερος χρόνος έναρξης, ο νωρίτερος χρόνος ολοκλήρωσης και ο βραδύτερος χρόνος ολοκλήρωσης για κάθε δραστηριότητα του έργου.**
- 3 Να σχεδιαστεί το διάγραμμα Gantt απεικονίζοντας το συνολικό περιθώριο**

Άσκηση για λύση 1^η

Δραστηριότητα	Χρονική διάρκεια	Άμεσα προηγούμενες δραστηριότητες
A	5	-
B	3	A
C	4	B
D	5	B
E	6	C
F	3	C
G	4	E, F
H	8	G
I	7	D, G

Άσκηση για λύση 2^η

Δραστηριότητα	Χρονική διάρκεια	Άμεσα προηγούμενες δραστηριότητες
A	1	-
B	2	-
C	15	A, B
D	20	C
E	3	D
F	7	D
G	3	F
H	10	D
I	1	F
J	5	H
K	2	H, I
L	1	J, K

Άσκηση για λύση 3^η

Δραστηριότητα	Χρονική διάρκεια	Άμεσα προηγούμενες δραστηριότητες
A	6	-
B	4,5	A
C	6,17	B
D	8	C
E	4,83	D
F	12,33	C
G	5,33	F
H	3	E, G
I	4	E, G
J	5	H, I
K	6,33	J
L	7,33	J

Άσκηση για λύση 4^η

Δραστηριότητα	Χρονική διάρκεια	Άμεσα προηγούμενες δραστηριότητες
A	4	-
B	5	A
C	4	A
D	3	B
E	3	C
F	4	B, E

Άσκηση για λύση 5^η

Δραστηριότητα	Χρονική διάρκεια	Άμεσα προηγούμενες δραστηριότητες
A	5	-
B	7	A
C	9	A
D	11	B
E	3	C
F	6	C
G	6	D, F
H	8	E

Άσκηση για λύση 6^η

Δραστηριότητα	Χρονική διάρκεια	Άμεσα προηγούμενες δραστηριότητες
A	5	-
B	7	A
C	9	A
D	11	B
E	3	C
F	6	C
G	6	D, F
H	8	E

Άσκηση για λύση 7^η

Δραστηριότητα	Χρονική διάρκεια	Άμεσα προηγούμενες δραστηριότητες
A	16,5	-
B	17,5	A
C	18,5	A
D	16	B
E	20,3	B, C
F	23,3	B
G	8,5	D, E
H	23	F, G
I	25	B, C
J	13	I, H

Άσκηση για λύση 8^η

	ΔΡΑΣΤΗΡΙΟΤΗΤΑ	ΠΡΟΗΓΟΥΜΕΝΗ/ΕΣ	ΔΙΑΡΚΕΙΑ
A	Αξιολόγηση και επιλογή Η/W και S/W	-	5
B	Παραγγελία Η/W	A	8
C	Εύρεση χώρου	-	6
D	Παραγγελία S/W	A	8
E	Καλωδίωση	C	5
F	Παραγγελία επίπλων	C	3
G	Τοποθέτηση επίπλων	E,F	3
H	Πρόσληψη υπεύθυνου PC Lab	-	10
I	Μεταφορά/Τοποθέτηση υπολογιστών	B, G	2
J	Εγκατάσταση λογισμικού	D, H, I	2

Άσκηση για λύση 9^η

Δραστηριότητα	Προηγούμενες Δραστηριότητες	Χρονική Διάρκεια
A	-	6
B	A	9
C	A	8
D	-	7
E	B,C	10
F	C,D	10
G	E,F	4
H	D,E,F	3
I	D	12

Άσκηση για λύση 10ⁿ

Δραστηριότητα	Προηγούμενη	Διάρκεια
A	-	10
B	-	10
C	-	5
D	A	10
E	B	5
F	B	15
G	C	10
H	D, E	5
I	G	10
J	F, H, I	10

Άσκηση για λύση 11^η

	ΔΡΑΣΤΗΡΙΟΤΗΤΑ	ΠΡΟΗΓΟΥΜΕΝΕΣ	ΔΙΑΡΚΕΙΑ
A	Συμφωνία για τη στρατηγική του προϊόντος	-	2
B	Σχεδιασμός καλλιτεχνικού υλικού	A	8
Γ	Ονομασία προϊόντος	A	2
Δ	Διεξαγωγή έρευνας καταναλωτή	A	4
Ε	Αξιολόγηση συστατικών	A	4
Z	Οριστικοποίηση απαιτήσεων παραγωγής	A	3
Η	Προετοιμασία διαφημιστικής εκστρατείας στα μέσα	B	12
Θ	Σχεδιασμός συσκευασίας	Γ	6
I	Παραγγελία, εγκατάσταση και λειτουργία νέου εργοστασίου	Z	24
K	Διανομή διαφήμισης	H	2
Λ	Εκτύπωση και παράδοση συσκευασίας	Θ	14
M	Δοκιμή προϊόντος και συσκευασίας στην αγορά	Δ,Θ	4
N	Προμήθεια πρώτων υλών για παραγωγή	Δ,Ε,Θ	12
Ξ	Αρχικές δοκιμές παραγωγής	I,N	3
Ο	Τελικές δοκιμές παραγωγής και διανομής	Λ,Ξ	3
Π	Συγκέντρωση αποθεμάτων για κυκλοφορία στην αγορά	M,O	6
P	Προβολή προϊόντος	K	Ενεργοποιήστε τα Windows Μετάβαση στις ουθμίσεις πανεύοντας

Ανάλυση Πληροφοριακών Συστημάτων

Εαρινό Εξάμηνο 2020-2021

Lec04: Enterprise Resource Planning

23/03/2021

Διδάσκων: Δρ. Γεώργιος Χρ. Μακρής

Προγραμματισμός Επιχειρησιακών Πόρων – Enterprise Resource Planning

Βιβλίο:

Σύγχρονα Πληροφοριακά Συστήματα Επιχειρήσεων
***ERP-CRM-BPR* (Πάνος Φιτσιλής)**

Παραδοσιακές Εφαρμογές (Legacy Applications) [1/2]

- «Μονολιθικά» σχεδιασμένες εφαρμογές, που αποτελούνται από **μια σειρά διεργασιών, χωρίς συγκεκριμένη λογική διασύνδεσης** και οι οποίες **δεν διαλειτουργούν** με άλλες εφαρμογές.
- Εφαρμογές που **διαθέτουν σταθερές (fixed) διεπαφές** με τον χρήστη (user interfaces - UI). Οι διεπαφές αυτές είναι δύσκολο να αντικατασταθούν από διεπαφές φυλλομετρητή (web based UI).
- Εφαρμογές που διαθέτουν **εσωτερικές δηλώσεις σταθερών και μεταβλητών** (hard coded data definitions). Οι δηλώσεις αυτές είναι -συνήθως- συγκεκριμένες για την εφαρμογή και δεν συμμορφώνονται με προσεγγίσεις μοντέλων δεδομένων της επιχείρησης (enterprise data model approach).

Παραδοσιακές Εφαρμογές (Legacy Applications) [2/2]

- Εφαρμογές που **διαθέτουν εσωτερικές δηλώσεις επιχειρησιακών κανόνων** (hard coded business rules). Οι εφαρμογές αυτές είναι δύσκολο να επικαιροποιηθούν όταν οι επιχειρησιακοί κανόνες μεταβάλλονται λόγω μεταβολών στις επιχειρησιακές διεργασίες.
- Εφαρμογές που **αποθηκεύουν οι ίδιες τα στοιχεία αυθεντικοποίησης των χρηστών** (user credentials). Οι εφαρμογές αυτές δυσκολεύουν τη μετάβαση μιας επιχείρησης σε πληροφοριακά συστήματα που επιτρέπουν την πρόσβαση των χρηστών σε πόρους, με τη χρήση μοναδικής αυθεντικοποίησης (single sign-on) και με μεθόδους διαχείρισης ταυτότητας (identity management).

Αρχιτεκτονική πελάτη/εξυπηρετητή (client/server)

Σε αυτή την αρχιτεκτονική υπάρχουν δύο βασικές οντότητες:

- **Του πελάτη (client)**, δηλαδή ένας σταθμός εργασίας που διαθέτει μια εφαρμογή που εκτελείται σε αυτόν και στην οποία έχει πρόσβαση ένας χρήστης και
- **Του εξυπηρετητή (server)**, δηλαδή μια συσκευή εξυπηρετητή (server) όπου εκτελείται μια άλλη εφαρμογή, που εξυπηρετεί την εφαρμογή πελάτη. Συχνά, οι δύο εφαρμογές (πελάτη και εξυπηρετητή) μπορεί να βρίσκονται στην ίδια ή σε διαφορετικές «μηχανές». Στην αρχιτεκτονική αυτή, οι απαιτούμενοι πόροι για την εκτέλεση μιας επιχειρησιακής διαδικασίας μοιράζονται μεταξύ πελάτη και εξυπηρετητή.

Αρχιτεκτονική πελάτη/εξυπηρετητή (client/server)

Η αρχιτεκτονική αυτή δεν χρησιμοποιείται πλέον, καθώς:

- **Τα κόστη συντήρησης και αναβάθμισης είναι σημαντικά.** Η συντήρηση πολλαπλών σταθμών εργασίας έχει σημαντικό κόστος. Επίσης, κάθε αναβάθμιση της εφαρμογής πελάτη, απαιτεί την επικαιροποίηση της εφαρμογής αυτής σε όλους τους σταθμούς εργασίας, γεγονός που επιφέρει επίσης σημαντικά κόστη σε χρήμα και ανθρωπο-προσπάθεια.
- **Η εφαρμογή πελάτη μπορεί να συντηρεί ανομοιογενή δεδομένα στους σταθμούς εργασίας.**
- **Η κεντρική διαχείριση πολλαπλών εφαρμογών πελάτη/εξυπηρετητή είναι μια δύσκολη διαδικασία λόγω της κατανομής της σε πολλούς σταθμούς εργασίας.**

Αρχιτεκτονική πελάτη/εξυπηρετητή (client/server)

Η αντιμετώπιση των μειονεκτημάτων αυτού του τύπου εφαρμογών μπορεί να επιτευχθεί με τις ακόλουθες μεθόδους:

- Εγκατάσταση της εφαρμογής πελάτη σε έναν εξυπηρετητή και **μετατροπή των σταθμών εργασίας σε «τυφλά τερματικά» (dumb terminals)**. Η διαδικασία αυτή είναι εφικτή σε κάθε πληροφοριακό σύστημα που βασίζεται σε λειτουργικά συστήματα με λειτουργίες πολλαπλών χρηστών (multi-user) και δίνει τη δυνατότητα κεντρικής διαχείρισης των εφαρμογών πελάτη/εξυπηρετητή.
- **Εξαγωγή των λειτουργιών της εφαρμογής σε μια σειρά διεπαφών**. Η εξαγωγή μπορεί να επιτευχθεί με τη μετατροπή της εφαρμογής πελάτη/εξυπηρετητή σε μια **σειρά λειτουργικών οντοτήτων (components)**. Μερικές από τις οντότητες αυτές μπορούν να ενσωματωθούν σε τρίτες εφαρμογές. Με τη διαδικασία αυτή μπορεί να επιτευχθεί η σταδιακή αντικατάσταση της εφαρμογής πελάτη/εξυπηρετητή. Η αρχιτεκτονική συστημάτων που διαχειρίζεται τέτοιου είδους μεθόδους είναι η «**αρχιτεκτονική προσανατολισμένη σε υπηρεσίες**» (Service Oriented Architecture – SOA).
- **Μετάπτωση των εφαρμογών αυτών σε εφαρμογές φυλλομετρητή (thin-client architecture)**. Η διαδικασία μετάπτωσης σημαίνει την αντικατάσταση της εφαρμογής πελάτη από αντίστοιχη εφαρμογή που εκτελείται από φυλλομετρητή. Η μετάπτωση είναι αναγκαία όταν οι εφαρμογές πελάτη/εξυπηρετητή έχουν πρόσβαση σε σημαντικά δεδομένα της επιχείρησης.

Αρχιτεκτονική thin-client

Οι εφαρμογές thin (thin clients) χρησιμοποιούν την εφαρμογή φυλλομετρητή (browser) για να εκτελεστούν, αντικαθιστώντας την προσέγγιση των τυφλών τερματικών που παρουσιάστηκε ως λύση για την αντικατάσταση των εφαρμογών πελάτη/εξυπηρετητή.

Η αρχιτεκτονική thin-client έρχεται να μειώσει τα ανωτέρω κόστη. Με την εν λόγω αρχιτεκτονική, όλες οι εργασίες που επιτελούνται από την εφαρμογή πραγματοποιούνται στον εξυπηρετητή μενη ενότητα.

Η εν λόγω αρχιτεκτονική έχει ως σημαντικό πλεονέκτημα ότι εκτελείται ανεξαρτήτως συσκευής (device independent) αλλά και λειτουργικού συστήματος συσκευής (operating system independent), καθώς η εκτέλεση της εφαρμογής εξαρτάται αποκλειστικά από τον φυλλομετρητή.

Αρχιτεκτονική thin-client

- Προκαλεί σημαντικό φόρτο στο δίκτυο.
- Η εφαρμογή πρέπει να έχει υλοποιηθεί ώστε να αποκρίνεται το ίδιο καλά σε κάθε γνωστό φυλλομετρητή. (χρήση ενός συγκεκριμένου φυλλομετρητή ?).
- Οι εφαρμογές φυλλομετρητή δύσκολα επιτυγχάνουν υψηλής ποιότητας περιβάλλοντα διεπαφών χρήστη.
- Χρήση γνωστών τεχνολογιών όπως η Javascript, η XML και τα CSS (Cascading Style Sheets) Διαδοχικά Φύλλα Στυλ) στο πλαίσιο της τεχνολογίας AJAX (Asynchronous Java and XML), υπάρχει η δυνατότητα κατασκευής διεπαφών φυλλομετρητή που προσομοιάζουν αυτών των εφαρμογών σταθμών εργασίας (desktop applications).
- Τα δεδομένα που δίνονται από τον χρήστη πρέπει να ελέγχονται (validated) ώστε να αποφεύγονται πολλαπλές ανταλλαγές αιτημάτων, ιδιαίτερα μέσω του Διαδικτύου, αλλά και να επιτυγχάνεται τόσο η ικανοποίηση του χρήστη όσο και η ασφάλεια των ανταλλαγών πληροφοριών.
- Οι υπολογιστικοί πόροι των εξυπηρετητών ιστού που προσφέρουν τις εφαρμογές thin-client μπορούν να εξαντληθούν γρήγορα, όταν τα αιτήματα των χρηστών είναι πολλά

Αρχιτεκτονική «προσανατολισμένη σε υπηρεσίες» (Service Oriented Architecture – SOA)

Η προσανατολισμένη σε υπηρεσίες αρχιτεκτονική (SOA) διαχωρίζει τις υπηρεσίες (λειτουργίες που προσφέρει ένα σύστημα) από τους **αποδέκτες** της (συστήματα που χρειάζονται αυτές τις λειτουργίες). Ο διαχωρισμός αυτός επιτυγχάνεται με μηχανισμούς που ονομάζονται **συμβόλαια υπηρεσιών** (*service contracts*), σε συνδυασμό με μηχανισμούς προς τους παρόχους (που δημοσιεύουν συμβόλαια) και τους πελάτες (που είναι οι αποδέκτες των συμβολαίων). Για την κατανόησή της Θα πρέπει να θεωρήσουμε τις υπηρεσίες ως τις εφαρμογές του συστήματος, ενώ τα συμβόλαια είναι οι διεπαφές (interfaces) μεταξύ των εφαρμογών του συστήματος.

Σε αντίθεση με τις παραδοσιακές αρχιτεκτονικές που συνδυάζουν υπηρεσίες με πελάτες, η SOA διαχωρίζει το συμβόλαιο από την υλοποίησή του. Ο διαχωρισμός αυτός προσφέρει «χαλαρές» σχέσεις ανάμεσα σε υπηρεσίες και πελάτες. Μια υλοποίηση της SOA είναι οι web services.

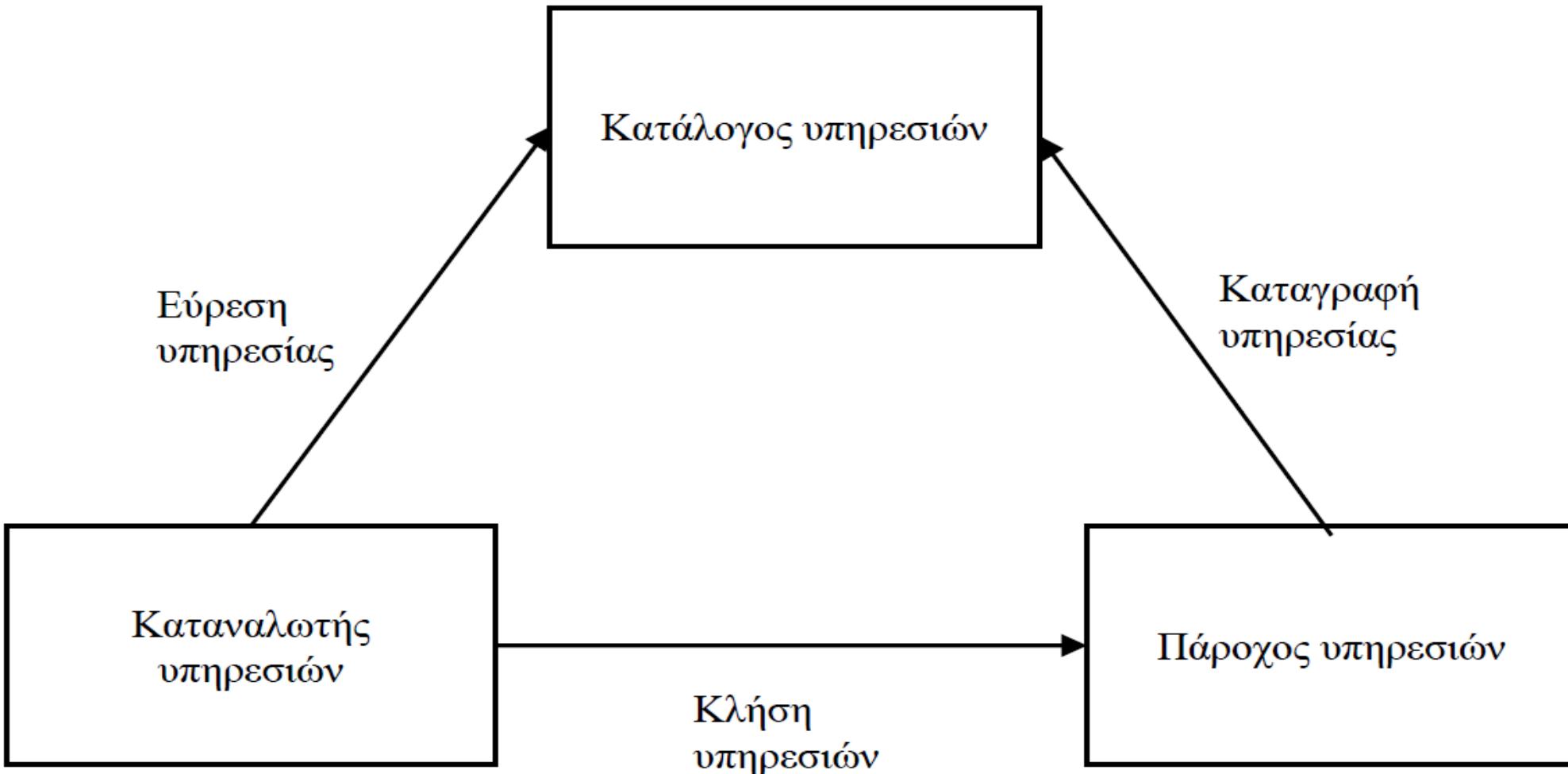
Αρχιτεκτονική «προσανατολισμένη σε υπηρεσίες» (Service Oriented Architecture – SOA)

Επομένως, οι βασικές αρχές της αρχιτεκτονικής SOA είναι:

- Οι υπηρεσίες είναι αυτόνομες. Κάθε υπηρεσία αναπτύσσεται, εγκαθίσταται και συντηρείται ανεξάρτητα.
- Οι υπηρεσίες είναι κατανεμημένες. Μια υπηρεσία μπορεί να βρίσκεται οπουδήποτε στο διαδίκτυο.
- Οι υπηρεσίες χαλαρά συνδεδεμένες (loosely coupled). Κάθε υπηρεσία είναι ανεξάρτητη από τις άλλες και μπορεί να αντικατασταθεί χωρίς να διαταραχθούν οι εφαρμογές που την χρησιμοποιούν με την προϋπόθεση ότι η διεπαφή (service interface) που δίνει είναι συμβατή.
- Οι υπηρεσίες παρέχουν ένα σαφώς ορισμένο συμβόλαιο διεπαφών (interface contract).

Αρχιτεκτονική «προσανατολισμένη σε υπηρεσίες» (Service Oriented Architecture – SOA)

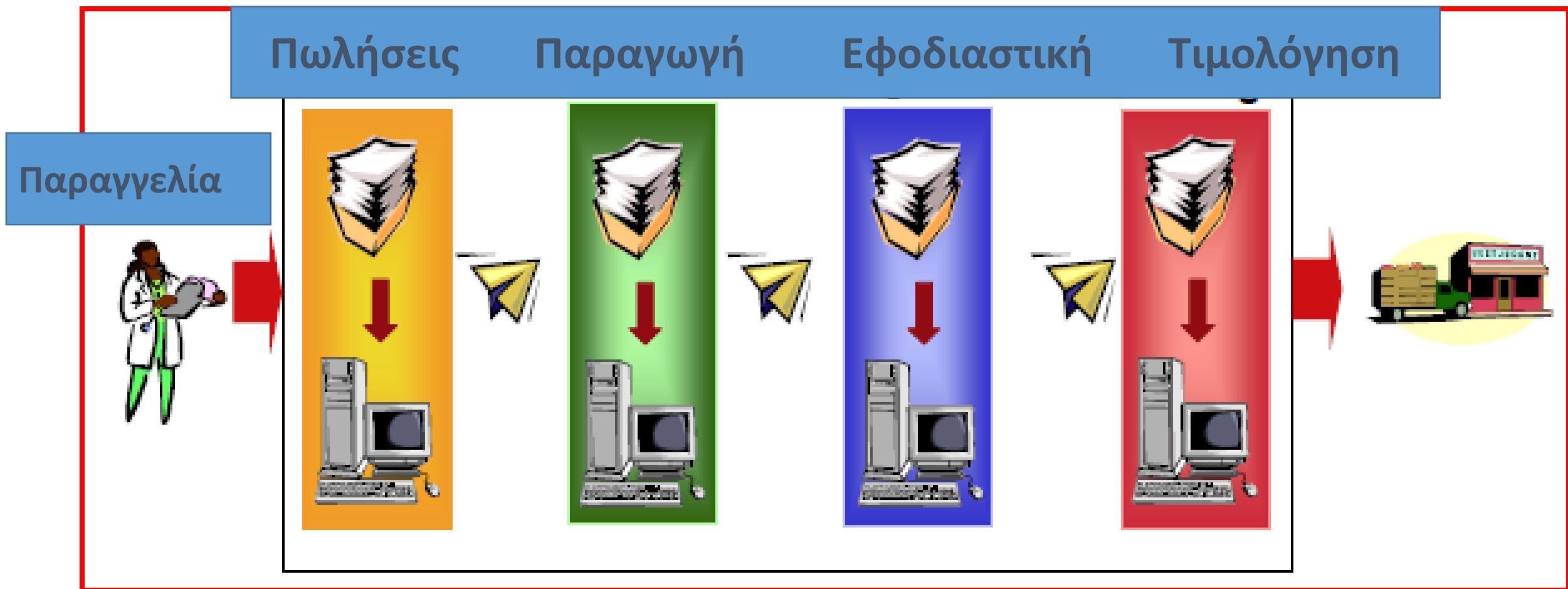
Η φιλοσοφία της αρχιτεκτονικής SOA.



Τι είναι ένα σύστημα ERP

- Είναι ένα πληροφοριακό σύστημα
- Ένα σύστημα ERP
 - αυτοματοποιεί τις καθημερινές εργασίες ενός οργανισμού
 - ολοκληρώνει τις διαδικασίες ενός οργανισμού
- Είναι μια διαδικασία για τη διαχείριση των πόρων σε ένα οργανισμό με συντονισμένο τρόπο
- Συνήθως βασίζεται σε μια βάση δεδομένων

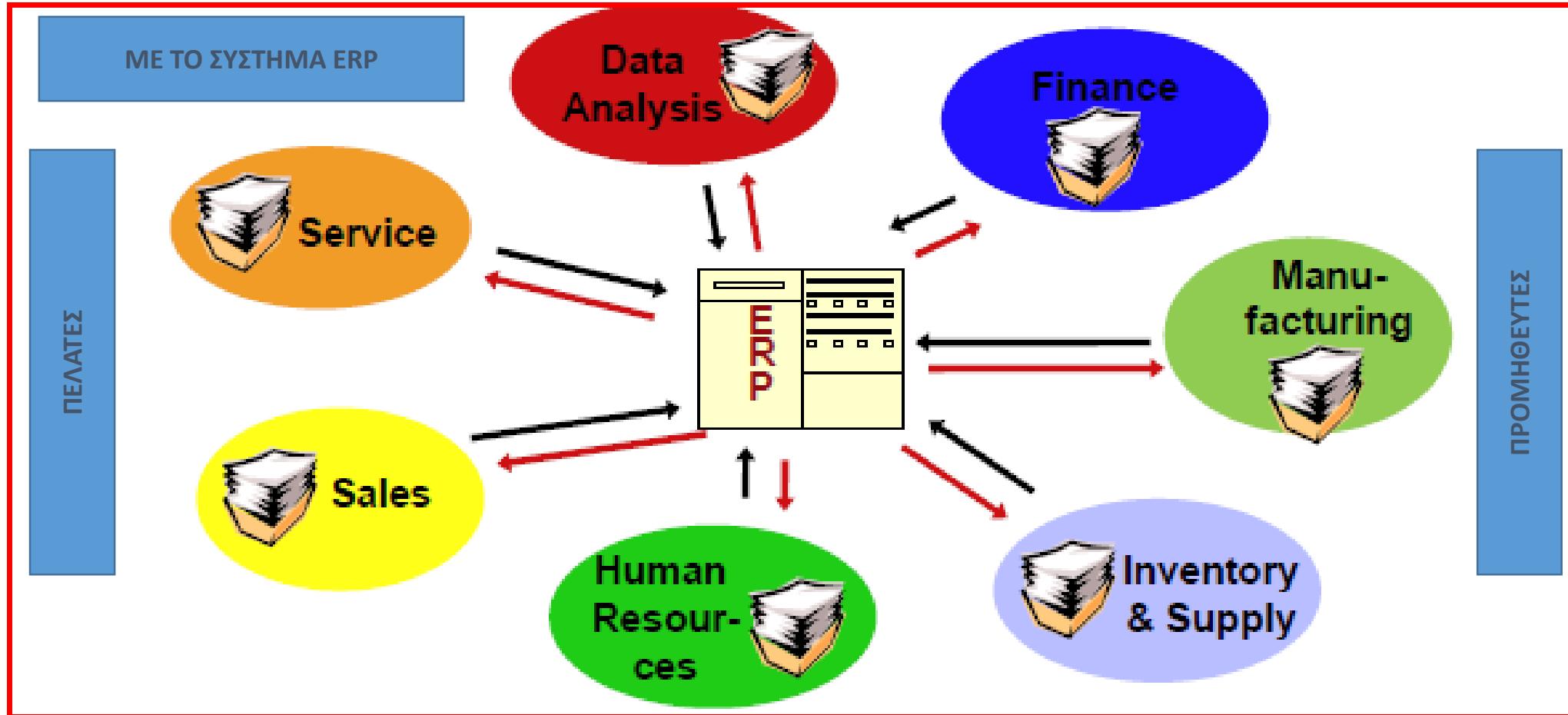
Ένα παράδειγμα χωρίς τη χρήση ERP



Προβλήματα

Καθυστερήσεις, κόστος, επανάληψη εισαγωγής της πληροφορίας

Με τη χρήση συστήματος ERP



Service : Υπηρεσίες

Sales : Πωλήσεις

Human Resources: Ανθρώπινοι Πόροι

Inventory and Supply: Απόθεμα και προμήθειες

Manufacturing-Κατασκευή Προϊόντων

Finance – Οικονομική Διαχείριση

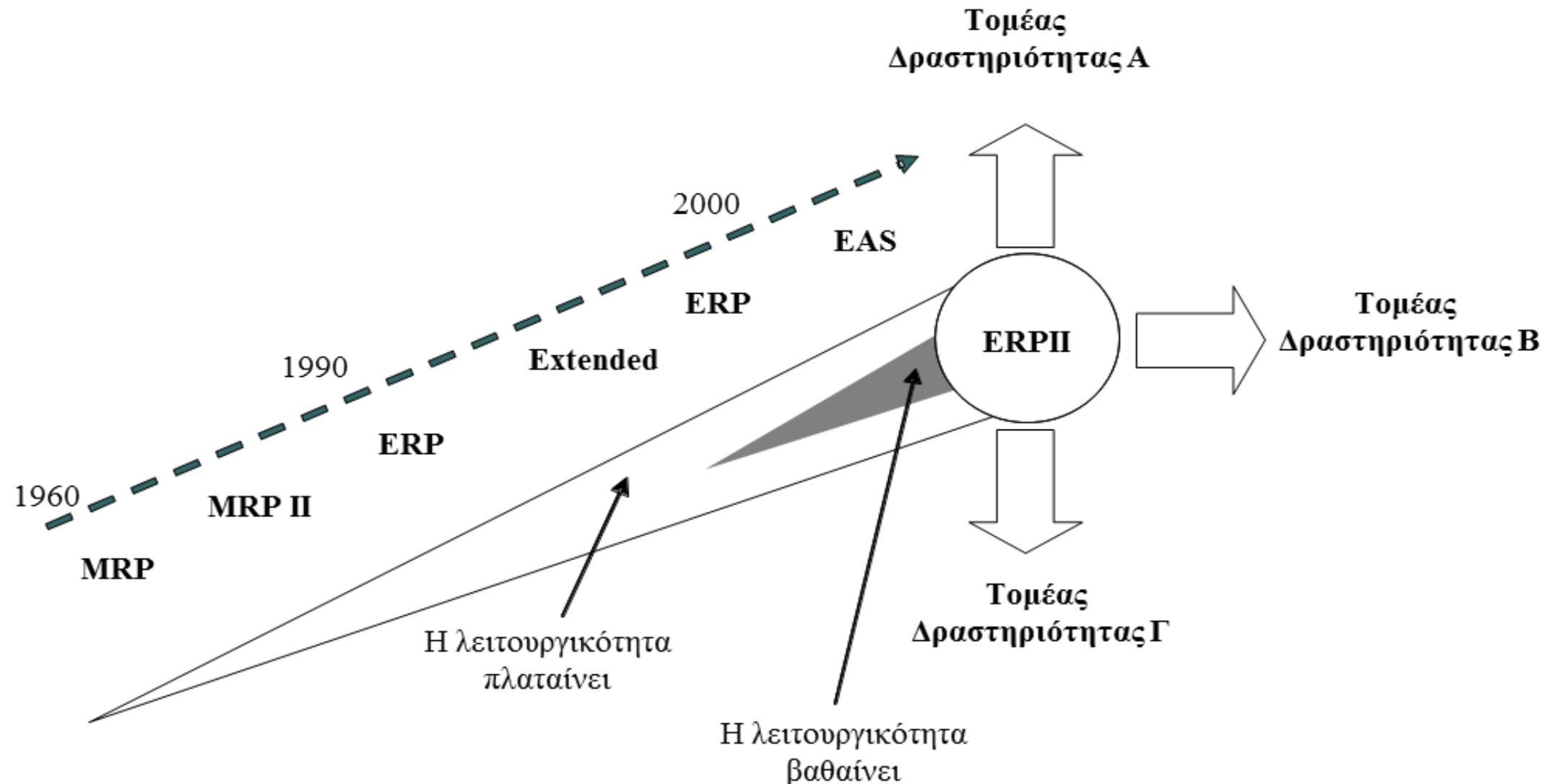
Data Analysis – Ανάλυση δεδομένων

Τα συστήματα ERP προσφέρουν

Ένα σύστημα ERP δίνει τη δυνατότητα σε μια επιχείρηση

- να ισορροπεί την προσφορά και τη ζήτηση προϊόντων και υπηρεσιών,
- να συνδέει τους πελάτες με τους προμηθευτές σχηματίζοντας εφοδιαστικές αλυσίδες,
- να υιοθετεί αποδεδειγμένες επιχειρησιακές διαδικασίες με σκοπό τη βέλτιστη λήψη επιχειρηματικών αποφάσεων,
- να ελαχιστοποιεί το κόστος παραγωγής προϊόντων ή παροχής υπηρεσιών ενώ ταυτόχρονα να βελτιώνει την ικανοποίηση του πελάτη,
- να καταγράφει και να βελτιώνει την κατανόηση των αναγκών της αγοράς και των πελατών,
- να μεγιστοποιεί την αποδοτικότητα των πόρων της επιχείρησης,
- να αυτοματοποιεί την οικονομική διαχείριση της επιχείρησης,
- να συγκεντρώνει και να ολοκληρώνει όλα τα δεδομένα της επιχείρησης με τέτοιο τρόπο ώστε αυτά να είναι άμεσα διαθέσιμα σε όλη την οργάνωση και
- να δίνει τη δυνατότητα στην επιχείρηση να εφαρμόζει νέες πολιτικές και νέες στρατηγικές.

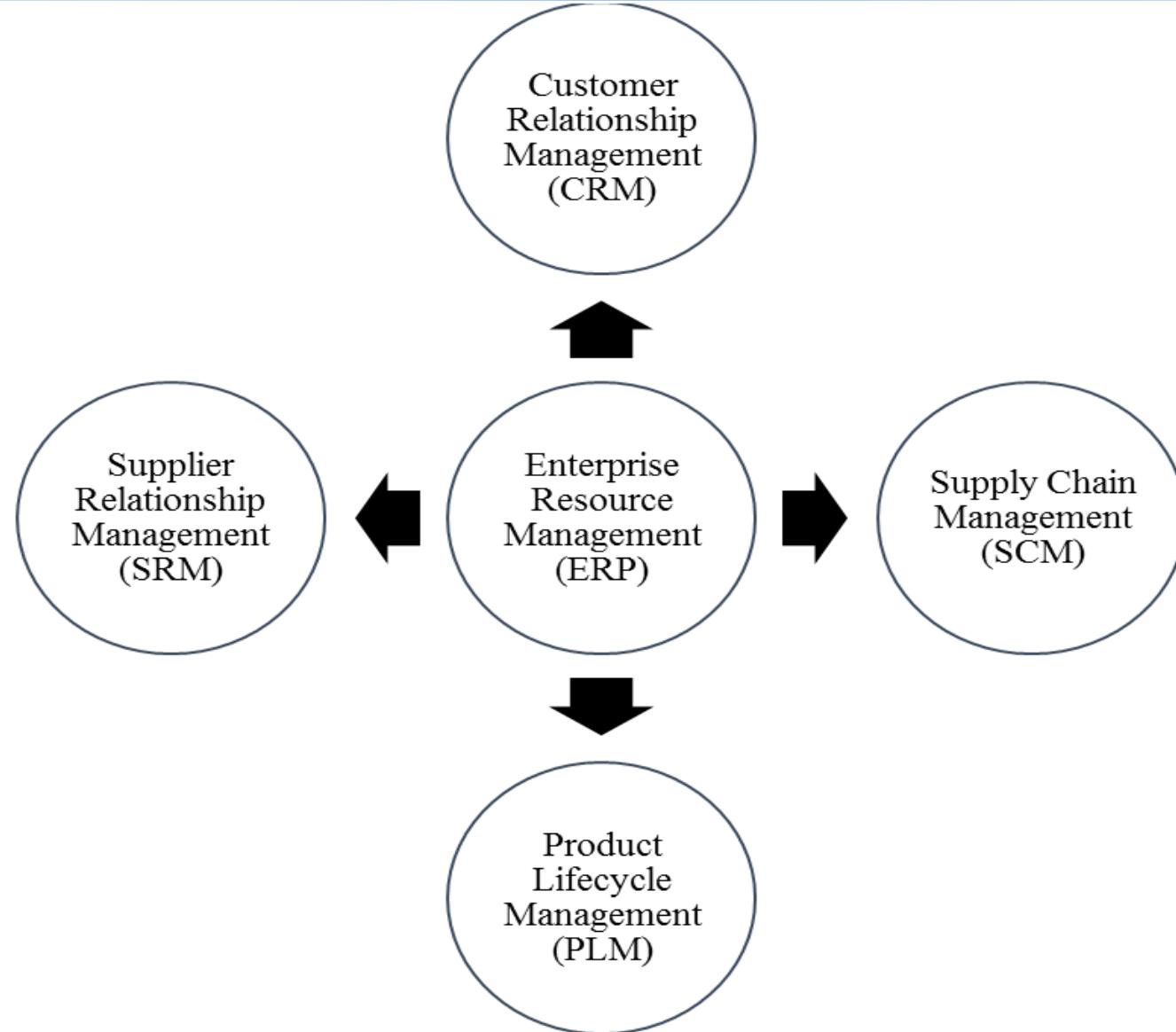
Η εξέλιξη των συστημάτων



SAP Business Suite

- Η εταιρεία SAP στο προϊόν της SAP Business Suite παρουσιάζει ένα Επιχειρηματικό Πλαίσιο Εφαρμογής, το οποίο ολοκληρώνει σε ένα ενιαίο σύνολο τα παρακάτω συστήματα:
 - Σύστημα Προγραμματισμού Επιχειρησιακών Πόρων – Enterprise Resource Planning (ERP)
 - Σύστημα Διαχείρισης Πελατειακών Σχέσεων - Customer Relationship Management (CRM),
 - Σύστημα Διαχείρισης Εφοδιαστικής Αλυσίδας -Supply Chain Management (SCM),
 - Σύστημα Διαχείρισης Κύκλου Ζωής Προϊόντων - Product Lifecycle Management (PLM), και
 - Σύστημα Διαχείρισης Προμηθευτών - Supplier Relationship Management (SRM).

Το επιχειρηματικό πλαίσιο εφαρμογών του συστήματος SAP

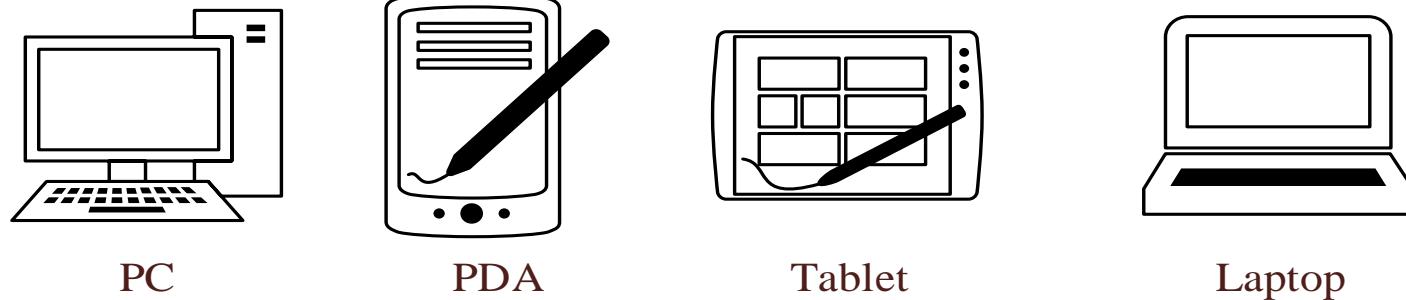


Η αρχιτεκτονική των συστημάτων ERP

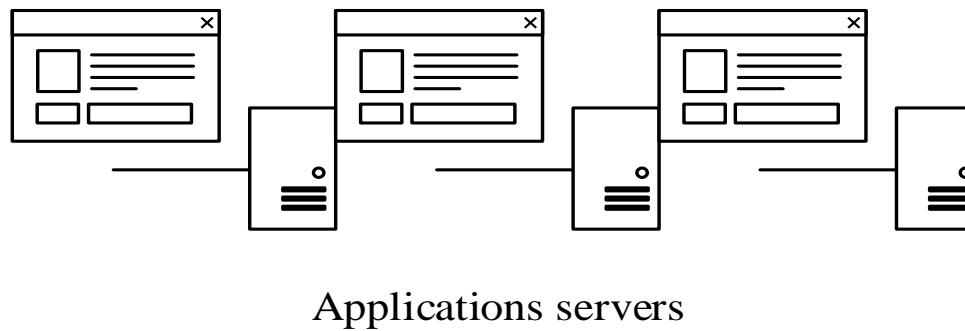
- Δομή που βασίζεται σε πολλές ευδιάκριτες επιχειρησιακές ενότητες (modular design),
- Χρήση κοινής σχεσιακή βάσης δεδομένων (Relational Database Management System - RDBMS) για την αποθήκευση όλων των επιχειρησιακών δεδομένων.
- Ολοκλήρωση των δεδομένων αλλά και διαδικασιών.
- Χρήση τυποποιημένων διεπαφών (interface) και γραφικών διεπαφών χρήστη.
- Δυνατότητα παραμετροποίησης.
- Δυνατότητα ορισμού επιχειρηματικών διαδικασιών (business processes) και ορισμού ροής εργασιών (workflows).

Αρχιτεκτονική 3-επιπέδων

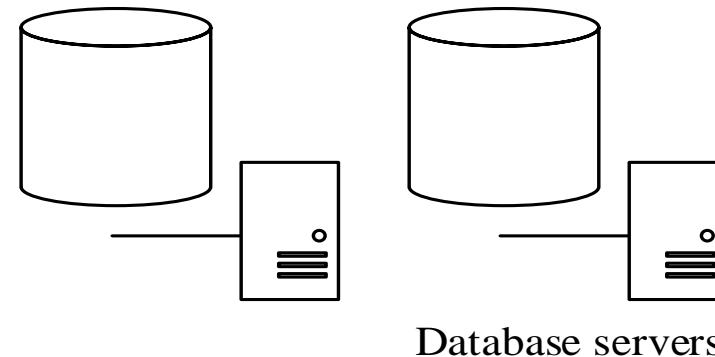
Presentation layer



Business logic layer



Database layer



Επιχειρηματικές διεργασίες

- Μια επιχειρηματική διεργασία (business process) είναι ένα λογικό σύνολο ενεργειών με σκοπό την παραγωγή ενός συγκεκριμένου προϊόντος ή μιας υπηρεσίας. Σε μια επιχείρηση αποτελούν ένα σύνολο αλληλεξαρτώμενων δραστηριοτήτων που παράγουν ένα σύνολο συγκεκριμένων εκροών για ένα πρόσωπο ή μία άλλη διεργασία μέσω συγκεκριμένων εισροών και εργασιών προστιθέμενης αξίας. Επίσης μια επιχειρηματική διαδικασία μπορεί να οριστεί ως ένα σύνολο δραστηριοτήτων και εργασιών που με την ολοκλήρωσή της, επιτυγχάνει έναν οργανωτικό στόχο. Η διαδικασία πρέπει να περιλαμβάνει σαφώς καθορισμένες εισόδους και μία τουλάχιστον έξοδο. Αυτές οι είσοδοι αποτελούνται από όλα αυτά τα στοιχεία που συμβάλλουν (άμεσα ή έμμεσα) στην προστιθέμενη αξία μιας υπηρεσίας ή προϊόντος.

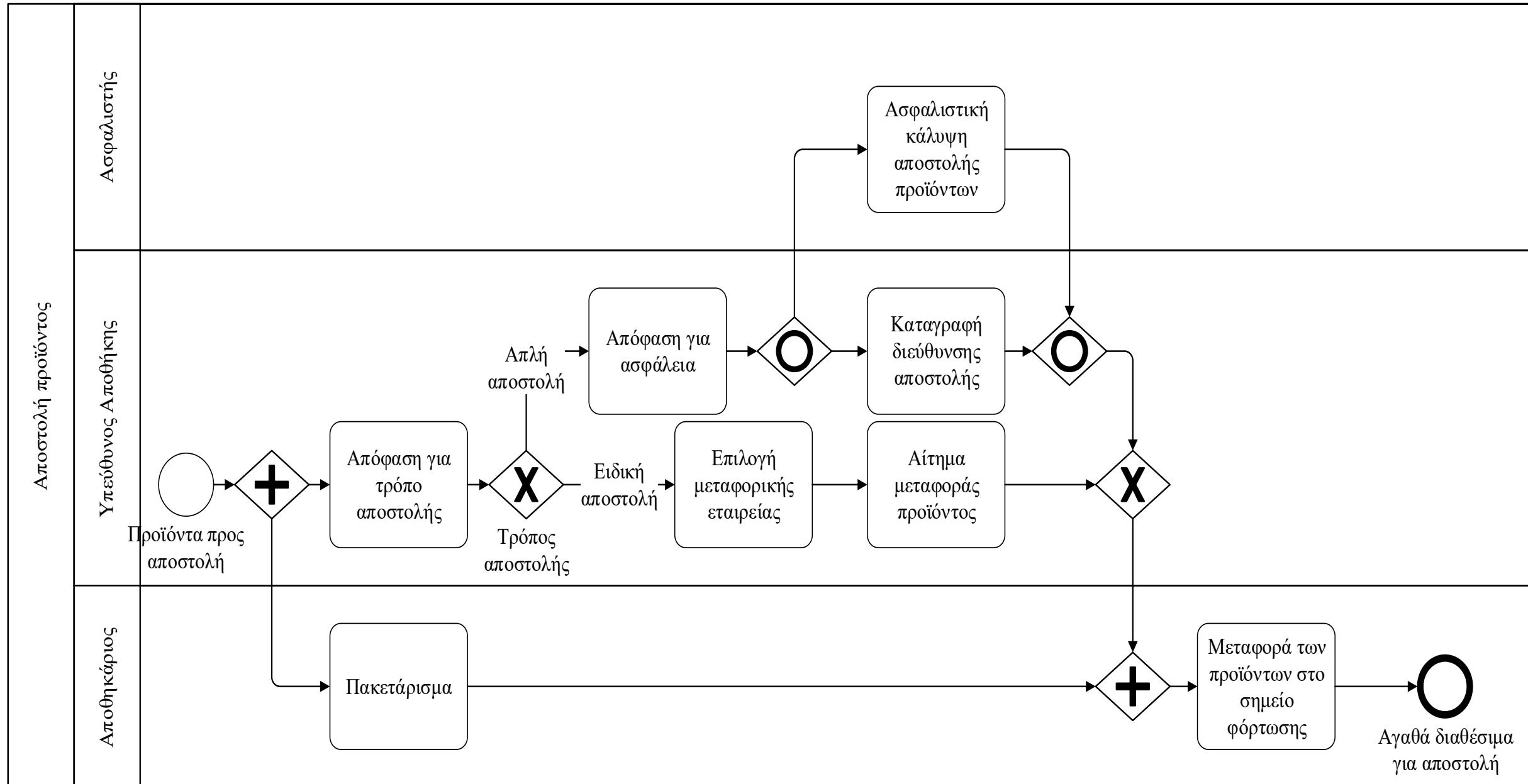
Παραδείγματα επιχειρηματικών διεργασιών

- Πώληση προϊόντων (Fulfillment)
- Αγορά υλικών (Procurement)
- Παραγωγή προϊόντων (Production)
- Διαχείριση Έργου (Project Management)

Μέθοδοι μοντελοποίησης επιχειρηματικών διεργασιών

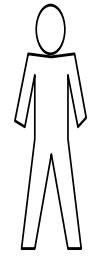
- απλές μεθόδοι περιγραφής επιχειρηματικών διεργασιών, όπως τα διαγράμματα ροής (flowcharts)
- τις ημιτυπικές (semi-formal) τεχνικές όπως τα διαγράμματα διαδικασιών ελεγχόμενων από αλυσίδες γεγονότων (event controlled chain of process) που χρησιμοποιούνται από τα εργαλεία ARIS και το σύστημα ERP SAP. Στην ίδια κατηγορία ανήκουν τα διαγράμματα της γλώσσας UML καθώς και η αναπαράσταση Business Process Modeling Notation (BPMN)
- γλώσσες αναπαράστασης με τη χρήση της XML όπως η Business Process Execution Language (BPEL) που έχει αναπτυχθεί από τον οργανισμό OASIS και αποτελεί την πιο διαδεδομένη γλώσσα αναπαράστασης επιχειρηματικών διεργασιών, Workflow XML (WfXML), Business Process Modeling Language (BPML), κ.α.
- τις πιο αυστηρές και θεωρητικές περιγραφές όπως Pi-calculus και τα Petri Nets

Παράδειγμα διεργασίας με BPMN

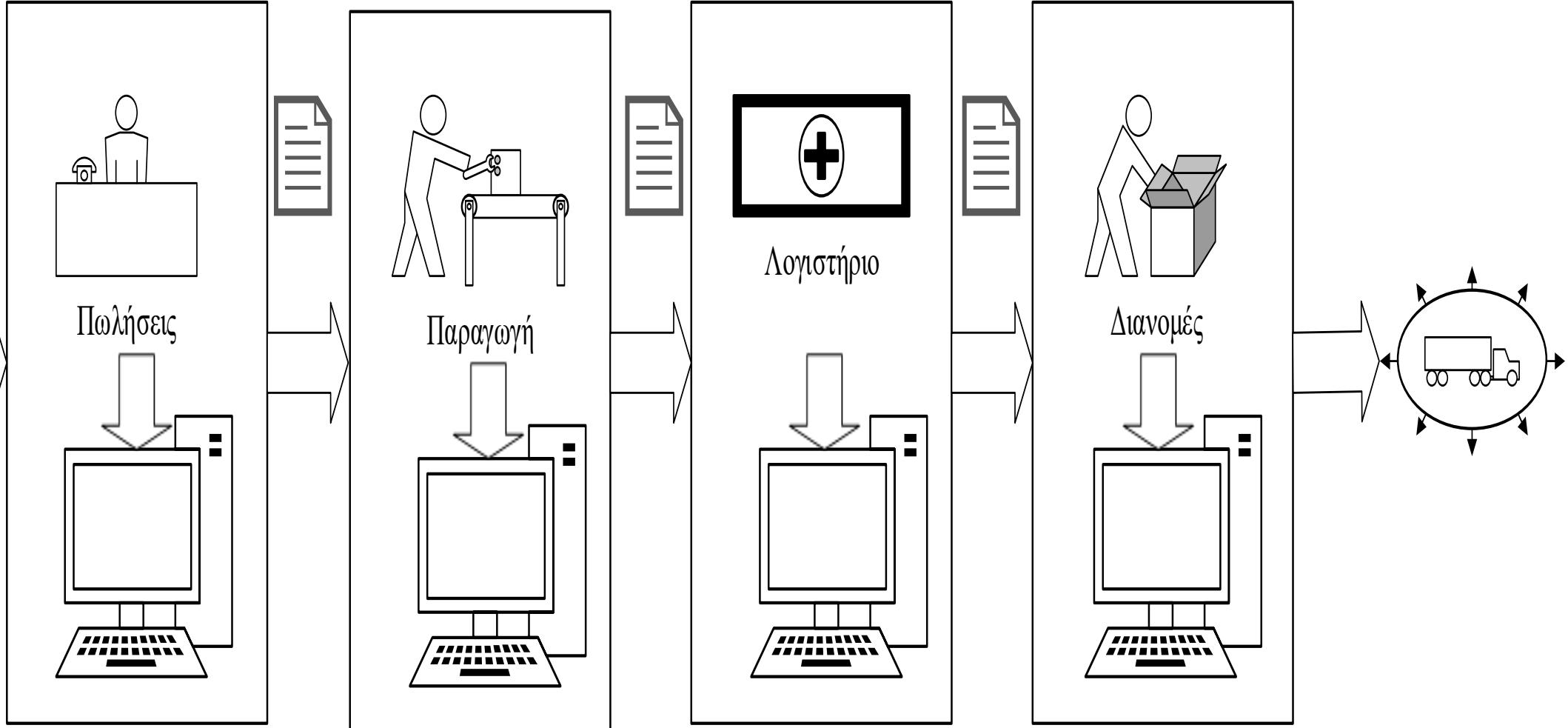


Ο τρόπος εργασίας με μεμονωμένα πληροφοριακά συστήματα

Παραγγελία



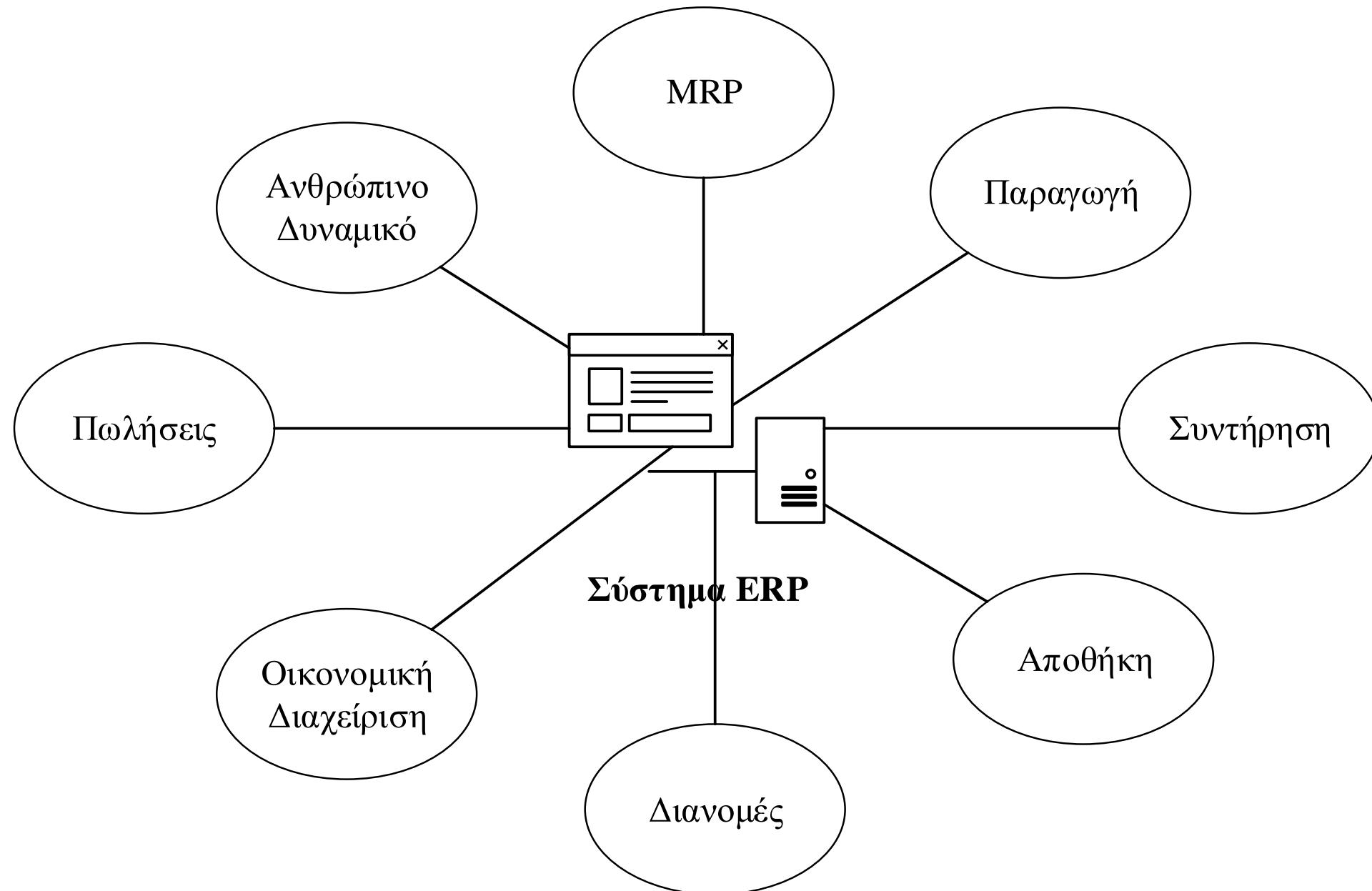
Πελάτης



Οπτικές γωνίες ολοκλήρωσης ενός πληροφοριακού συστήματος

- **Ολοκλήρωση των δεδομένων (data integration)** επιτυγχάνεται όταν όλα τα σχήματα δεδομένων που χρησιμοποιούνται από όλα τα τμήματα της επιχείρησης είναι ενοποιημένα και οι εγγραφές στα αντίστοιχα πεδία έχουν την ίδια τιμή.
- **Λειτουργική ολοκλήρωση (operation integration)** απαιτεί τον ορισμό της ροής εργασίας (workflow), δηλαδή τη σύνδεση των επιμέρους ενεργειών, ή των βημάτων της επιχειρηματικής διεργασίας, με την προηγούμενη ή την επόμενη εργασία, αντίστοιχα.
- **Ολοκλήρωση διεργασιών (process integration)** σημαίνει ότι έχουν αναπτυχθεί διεπαφές (interfaces) μεταξύ των διαφόρων επιχειρηματικών διεργασιών.
- **Ολοκλήρωση λογισμικού (software integration)** επιτυγχάνεται όταν διαφορετικές εφαρμογές λογισμικού μπορούν να χρησιμοποιούν τα δεδομένα και τις λειτουργίες του άλλου.

Κεντρική αποθήκευση πληροφορίας σε ένα σύστημα ERP



Η λειτουργικότητα των συστημάτων ERP

- Μάρκετινγκ και Πωλήσεις,
- Διαχείριση Εφοδιαστικής Αλυσίδας,
- Οικονομική και Λογιστική Διαχείριση και
- Διοίκηση Ανθρώπινου Δυναμικού

Τα modules ενός ERP συστήματος

Διαχείριση Εφοδιαστικής Αλυσίδας	Οικονομική και Λογιστική Διαχείριση	Ανθρώπινοι Πόροι	Μάρκετινγκ και Πωλήσεις
Προγραμματισμός και πρόγραμμα παραγωγής (Production Planning and Scheduling)	Γενική Λογιστική (General ledger)	Διαχείριση Προσωπικού (Personnel planning)	Διαχείριση Πελατών (Customer Management)
MRP (Material Requirements Planning)	Διαχείριση Παγίων (Assets accounting)	Μισθοδοσία (Payroll)	(Διαχείριση Παραγγελιών) Order management
Διαχείριση Υλικών και Διαχείριση Αποθήκης (Inventory management and Warehouse management)	Κέντρα κόστους/κερδών (Cost center accounting/ Profit center accounting)	Απολογισμός Ταξιδίων (Travel expenses)	Πρόβλεψη Πωλήσεων (Forecasting)
Συντήρηση Εργοστασίου (Plant maintenance)	Λογαριασμοί Εισπρακτέοι/ Πληρωτέοι (Accounts receivable and payable)	Διαχείριση Οργανωτικής Δομής (Organisational structure)	Προγραμματισμός Πωλήσεων (Sales planning)
Διαχείριση Ποιότητας (Quality management)	Προϋπολογισμός (Budgeting)	Διαχείριση Χρόνου Εργαζομένων (Time management)	
Προγραμματισμός Διανομών (Distribution planning)	Κοστολόγηση προϊόντων/υπηρεσιών (Costing)	Πρόσληψη προσωπικού	

Οι εξειδικευμένες λύσεις για βιομηχανικούς κλάδους

- Αεροδιαστημικής Τεχνολογίας και Άμυνας (Aerospace and Defence)
- Αυτοκινητοβιομηχανίας (Automotive)
- Τραπεζικό (Banking)
- Χημικών βιομηχανιών (Chemicals)
- Καταναλωτικών προϊόντων (Consumer Products)
- Κατασκευαστικών εταιρειών (Engineering, Construction)
- Υπηρεσιών Υγείας (Healthcare)
- Υψηλής Τεχνολογίας (High Technology)
- Ανώτατης Εκπαίδευσης (Higher Education)
- Κατασκευής μηχανολογικών προϊόντων (Industrial Machinery)
- Ασφαλειών (Insurance)
- Επιστημών ζωής (Life Sciences)
- Εφοδιαστικής (Logistics Service Prod.)
- Μέσων Μαζικής Επικοινωνίας (Media)
- Προϊόντων Άλεσης (Mill Products)
- Ορυχείων (Mining)
- Πετρελαιοειδών και Φυσικού Αερίου (Oil & Gas)
- Φαρμακευτικές (Pharmaceuticals)
- Ταχυδρομείων (Postal Services)
- Συμβουλευτικών Υπηρεσιών (Professional Services)
- Δημοσίου Τομέα (Public Sector)
- Σιδηροδρομικές (Railways)
- Λιανικής Πώλησης (Retail)
- Τηλεπικοινωνιών (Telecommunications)
- Κοινής Ωφελείας (Utilities)
- Χονδρικού Εμπορίου (Wholesale Distribution)

Εξειδικευμένες λύσεις ανά κλάδο - Νοσοκομεία

Εξειδικευμένη κλαδική λύση (Νοσοκομεία)

Διαχείριση
Ασθενών

Διαχείριση
Εισαγωγών/
Εξαγωγών

Διαχείριση Ραντεβού

...

Νοσήλεια

Σύστημα ERP

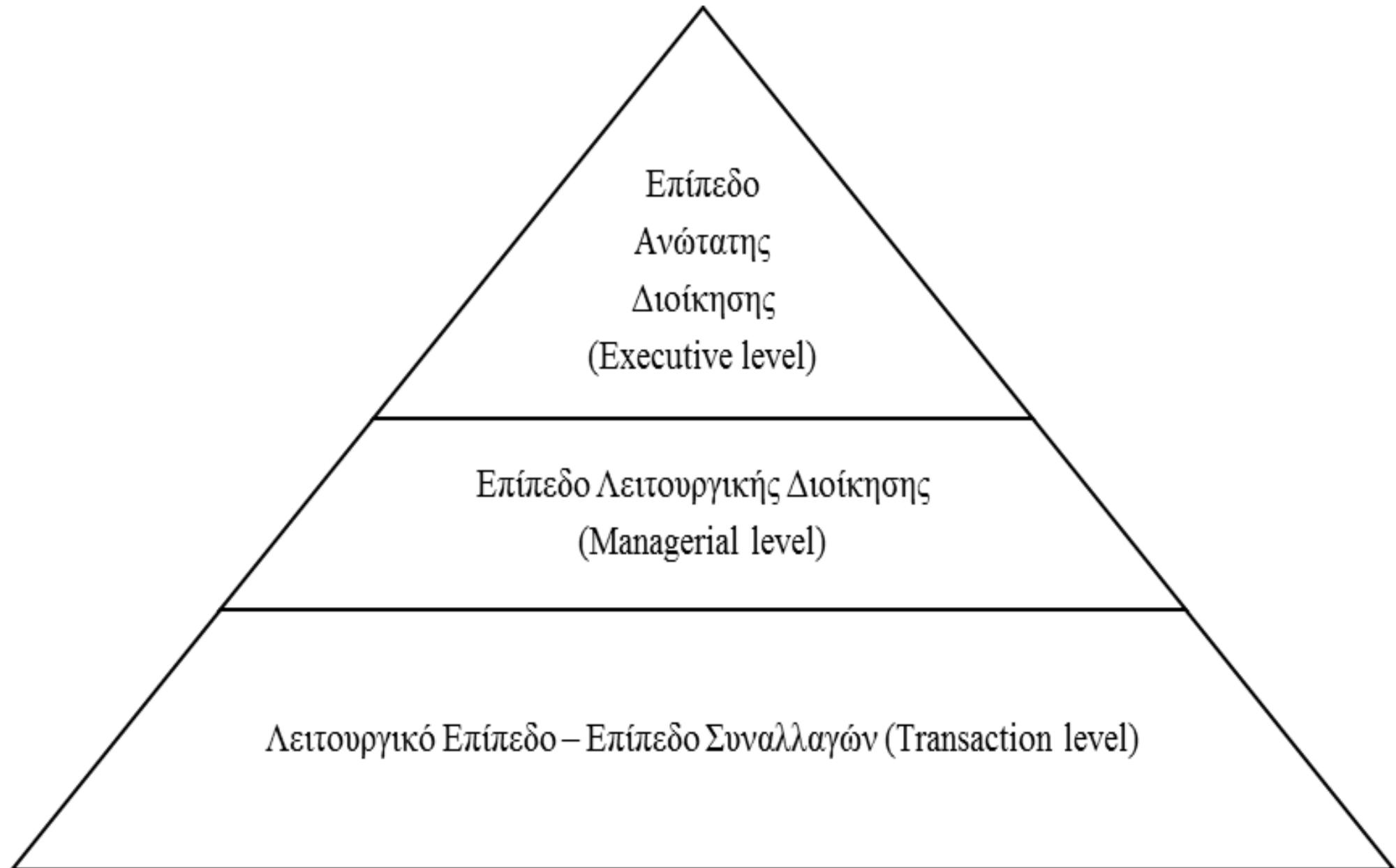
Οικονομική
Διαχείριση

Διοίκηση
Λειτουργίας

Διοίκηση Ανθρώπινου
Δυναμικού

Πληροφοριακή υποδομή

Συστήματα ERP και διοικητικά επίπεδα



Πλεονεκτήματα χρήσης συστημάτων ERP

- Βελτίωση της λειτουργίας της επιχείρησης, το οποίο επιτυγχάνεται μέσω της μείωσης αποθηκευμένων υλικών, του καλύτερου ελέγχου αποθεμάτων, της μείωσης του αναγκαίου προσωπικού για εκτέλεση διαχειριστικών λειτουργιών, κ.α.
- Αύξηση της παραγωγικής δυνατότητας μέσω του ολοκληρωμένου προγραμματισμού παραγωγής,
- Βελτίωση και αποτελεσματικότερη διαχείριση παραγγελιών,
- Ταχύτερη διαχείριση οικονομικών διαδικασιών και κλείσιμο οικονομικών κύκλων και γενικότερα μέσω της βελτίωσης της εκτέλεσης των επιχειρηματικών διαδικασιών,
- Βελτίωση της διαθέσιμης εταιρικής πληροφορίας μέσω της ύπαρξης κεντρικής βάσης δεδομένων όπου αποθηκεύονται όλα τα εταιρικά δεδομένα και
- Βελτίωση της διαθεσιμότητας και της επεκτασιμότητας των πληροφοριακών συστημάτων μέσω της χρήσης ανοικτών και σύγχρονων αρχιτεκτονικών.

Μειονεκτήματα χρήσης συστημάτων ERP

- Δυσκολία προσαρμογής στους γοργούς ρυθμούς εξέλιξης των προϊόντων και των επιχειρηματικών δραστηριοτήτων του οργανισμού. Τα συστήματα ERP είναι εξαιρετικά δαπανηρά τόσο στην εγκατάσταση όσο και στη συντήρησή τους με συνέπεια οι αλλαγές στο περιβάλλον της επιχείρησης να μην απορροφώνται γρήγορα.
- Το σύστημα αδυνατεί να αντεπεξέλθει στους όρους λειτουργίας του αν η συνεργασία με τους προμηθευτές είναι ελλιπής. Οι προμηθευτές αποτελούν παράγοντα ζωτικής σημασίας για τη διατήρηση της αλυσίδας ζωής ενός συστήματος ERP, με συνέπεια μια διαφωνία μαζί τους να θέσει σε κίνδυνο τη ζωτικότητα του συστήματος.
- Το σύστημα αδυνατεί να λειτουργήσει σωστά αν οι χρήστες του δεν είναι άρτια εκπαιδευμένοι. Τα συστήματα αυτά παρέχουν αυτήν την αυτοματοποίηση βασιζόμενα σε πολύπλοκες διαδικασίες, που με έναν εσφαλμένο χειρισμό μπορεί να αποσταθεροποιηθούν.

Ανάλυση Πληροφοριακών Συστημάτων

**Εαρινό Εξάμηνο 2020-2021
Lec03 (Εργαστήριο) – 09/03/2021
Διδάσκων: Γεώργιος Χρ. Μακρής**

Διαλέξεις – παρουσιάσεις

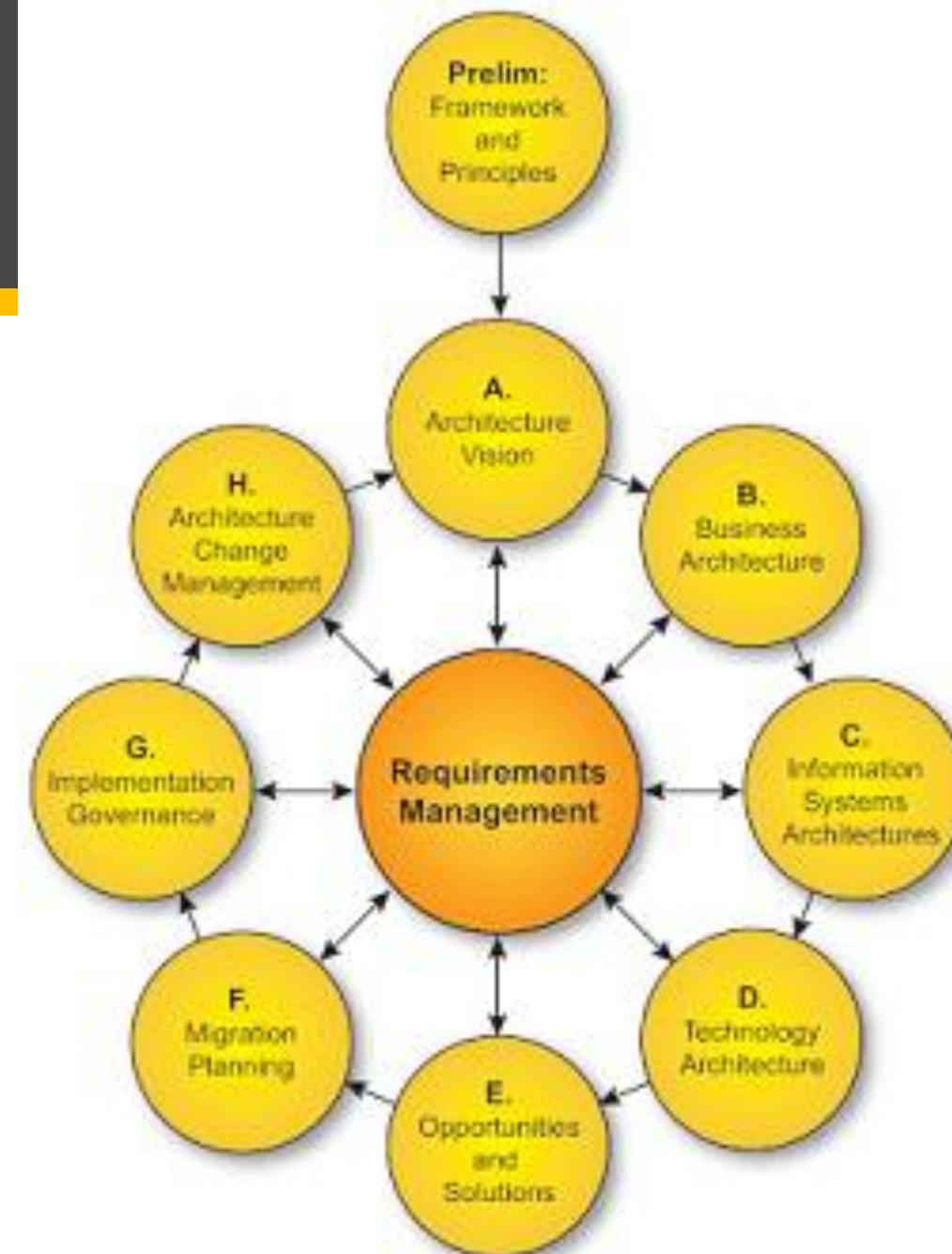
Μέρους του υλικού του μαθήματος στηρίζεται στο υλικό που χρησιμοποίησε ο κ. **Σταμέλος** στις παραδόσεις του για το μάθημα τα προηγούμενα χρόνια.

Επίσης μέρος της παρούσας παρουσίασης προέρχεται από την πτυχιακή της **Όλγας Κοιλαρίδου**

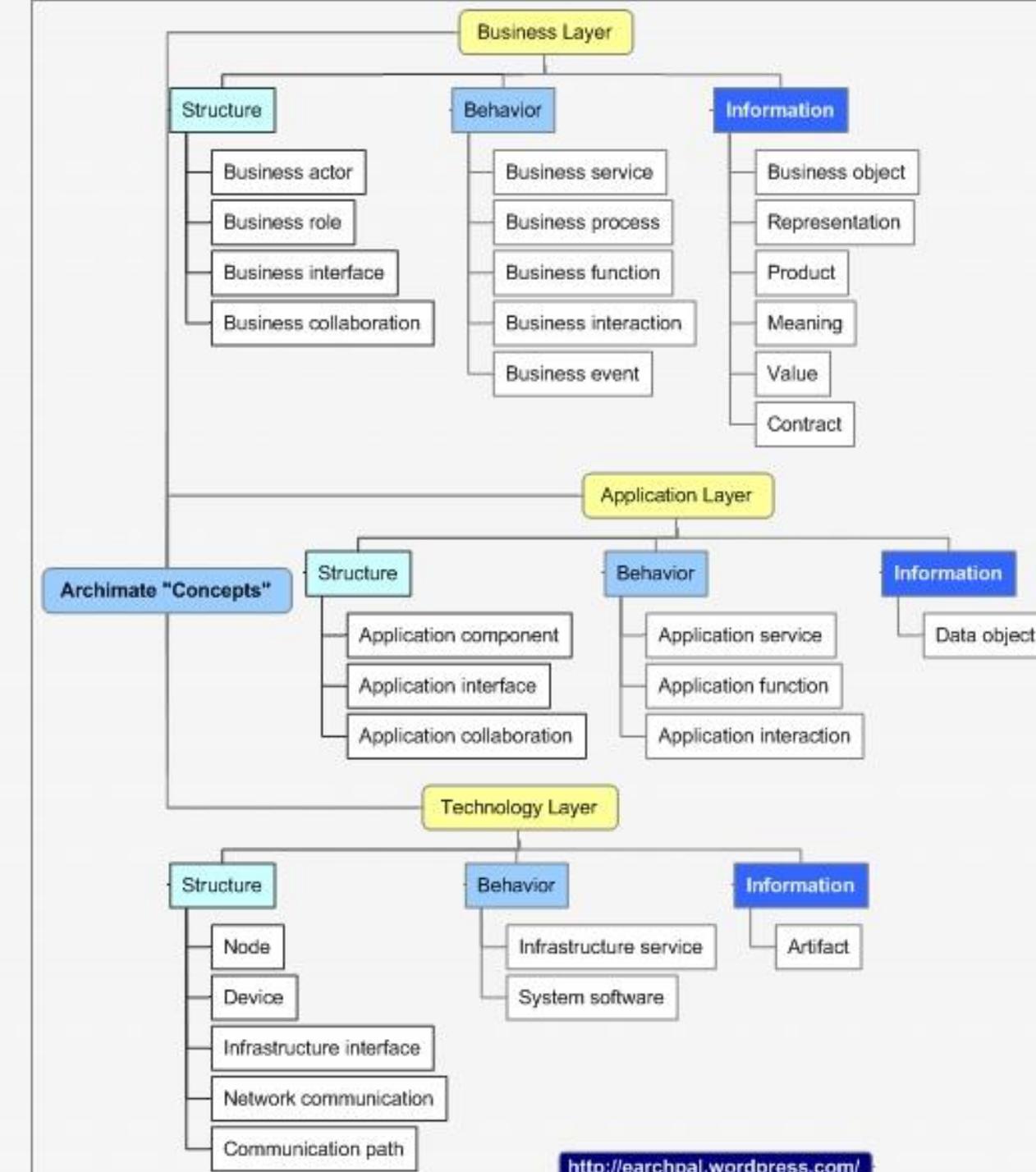
Πλαίσιο Open Group

Εμφανίζεται και ως TOGAF (The Open Group's Architecture Framework) και οργανώνεται σε 3 τομείς:

- **Μέθοδος Ανάπτυξης Αρχιτεκτονικής** (9 αρχιτεκτονικές φάσεις)
- **Επιχειρησιακό Συνεχές** (εικονική αρχιτεκτονική αποθήκη)
- **Βάση άντλησης πόρων** (παροχή βοήθειας στον αρχιτέκτονα για ορθή χρήση των αποτελεσμάτων της ΜΑΑ)



TOGAF





Λογισμικά Μοντελοποίησης

Εμπορικά εργαλεία:

Borland Caliber RM,
Adaptive Enterprise
Architecture Manager,
Envision VIP

Ελεύθερα εργαλεία:

Modelio, Archi, Iteraplan,
Essential Architecture
Manager





Πρόγραμμα Μοντελοποίησης Archi

Γλώσσα μοντελοποίησης αρχιτεκτονικής επιχειρήσεων ArchiMate

- Open source
- Δωρεάν
- Τρέχει σε όλα τα λειτουργικά συστήματα
- Εύκολη εγκατάσταση και χρήση
- Επιλογή τόσο ως εισαγωγικό όσο και ως βασικό εργαλείο στα χέρια ενός αρχιτέκτονα επιχειρήσεων





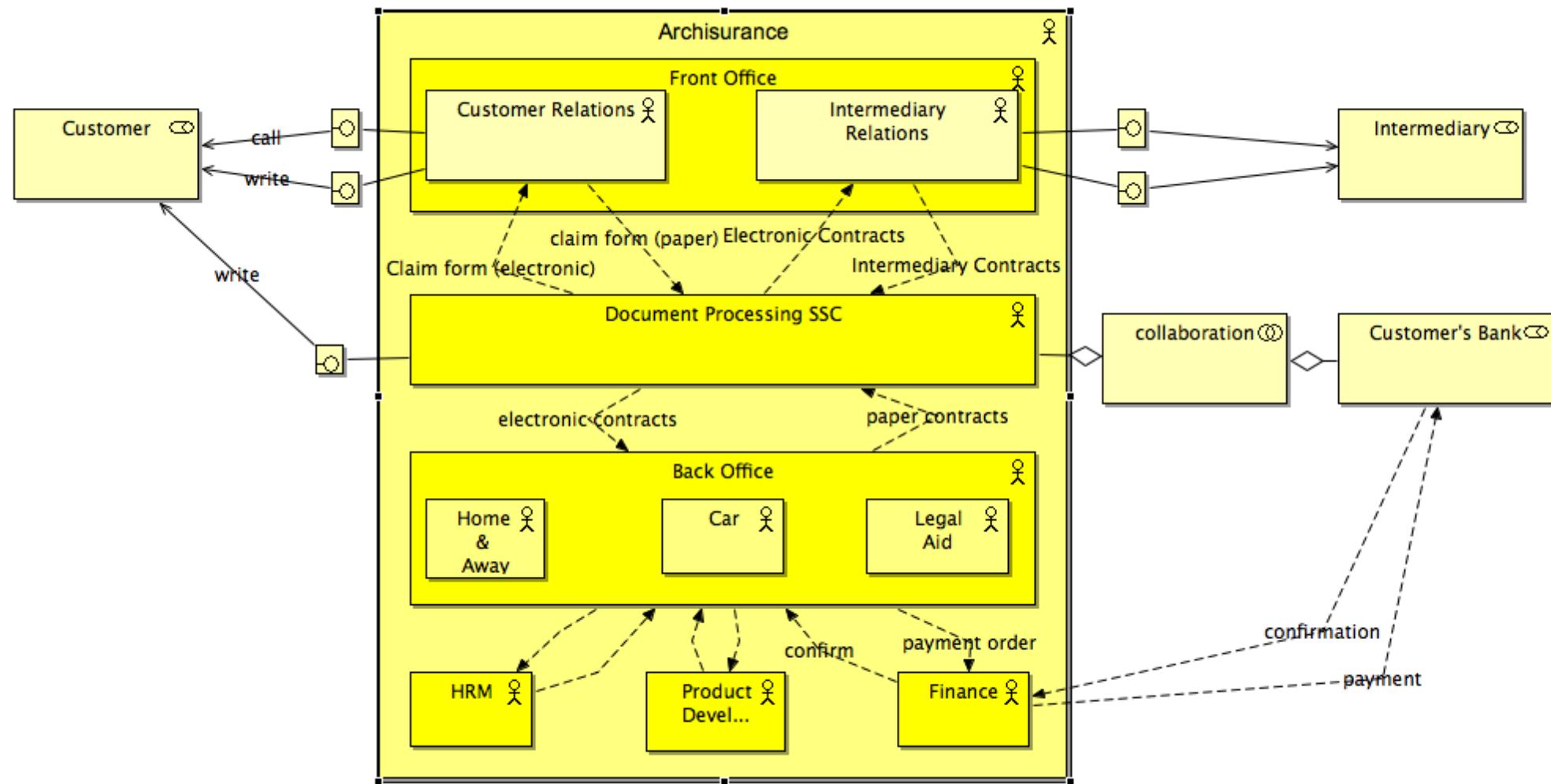
Το παράδειγμα της Archinsurance

Ανάλυση όλων των οπτικών της επιχείρησης

▼ Archinsurance

- Business
- Application
- Technology
- Motivation
- Implementation & Migration
- Connectors
- Relations
- ▼ Views
 - Actor Cooperation view
 - Application Behaviour View
 - Application Cooperation View
 - Application Structure View
 - Archimate View
 - Business Cooperation View
 - Business Function View
 - Business Process View
 - Business Product View
 - Goal and Principle View
 - Implementation and Installation View
 - Information Structure View
 - Layered View
 - Organisation Structure View
 - Organisation Tree View
 - Service Realisation View
 - Technical Infrastructure View

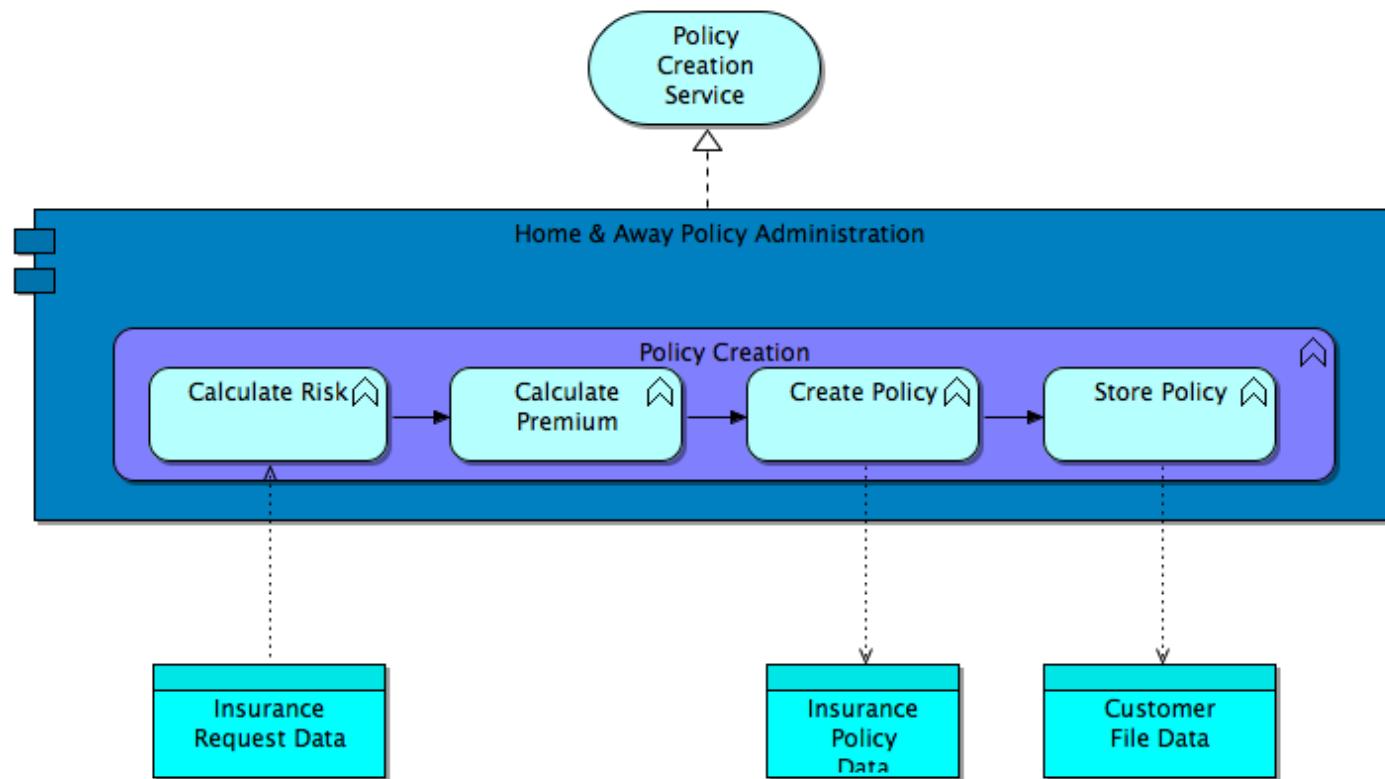
Actor Cooperation View



Actor Cooperation View

- Παρουσιάζει τις σχέσεις των παραγόντων μεταξύ τους αλλά και με το περιβάλλον τους.
- Ο πελάτης έχει επαφή με το γραφείο εξυπηρέτησης της εταιρίας και συγκεκριμένα με το τμήμα εξυπηρέτησης πελατών. Εσωτερικοί παράγοντες που συνεργάζονται και αλληλεπιδρούν είναι τα γραφεία εξυπηρέτησης και οργανωτικής υποστήριξης, η διοίκηση ανθρωπίνου δυναμικού, το οικονομικό τμήμα καθώς και το υπεύθυνο για την παράδοση των προϊόντων τμήμα. Έπειτα παρουσιάζεται η απαραίτητη συνεργασία με ενδιάμεσους παράγοντες αλλά και με την τράπεζα του πελάτη η οποία θα καθορίσει και την επιβεβαίωση της αγοράς και την πληρωμή για την υπηρεσία που προσφέρθηκε στον πελάτη.

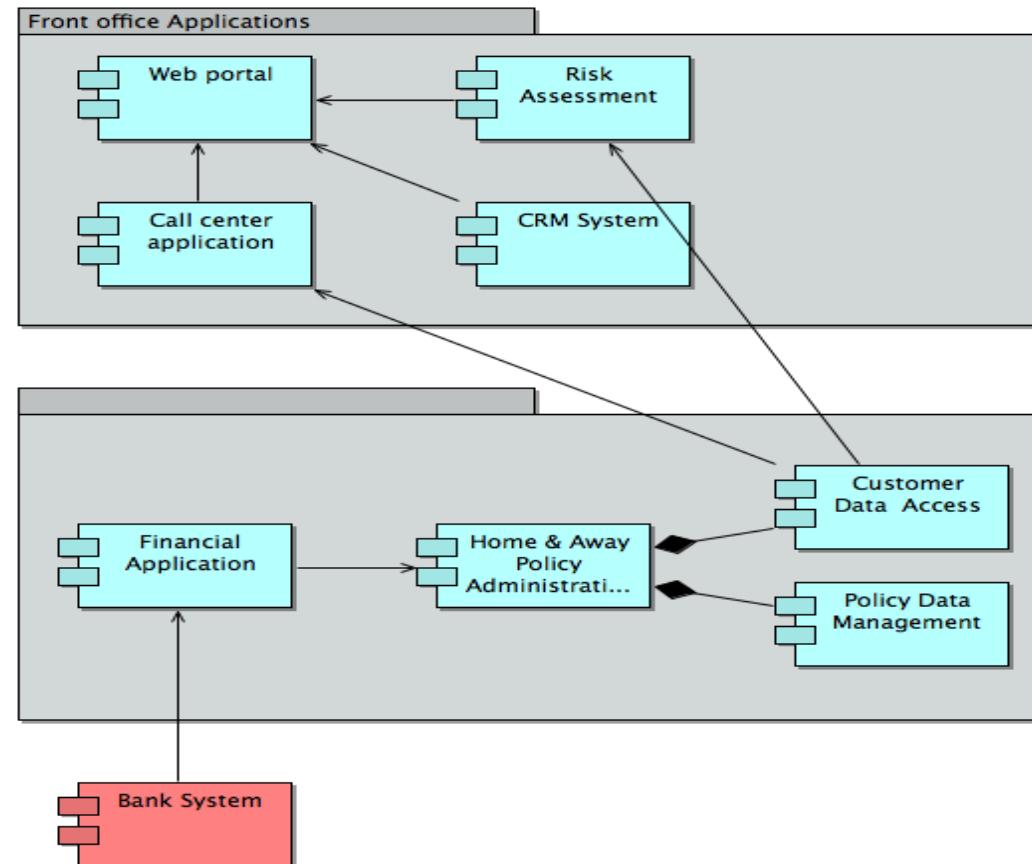
Application Behaviour View



Application Behaviour View

- Παρουσιάζεται ο τρόπος με τον οποίο η εταιρία δημιουργεί την πολιτική της μέσω των κατάλληλων λειτουργιών υπολογισμού ρίσκου, ασφάλιστρων και δημιουργίας και αποθήκευσης της πολιτικής αυτής.

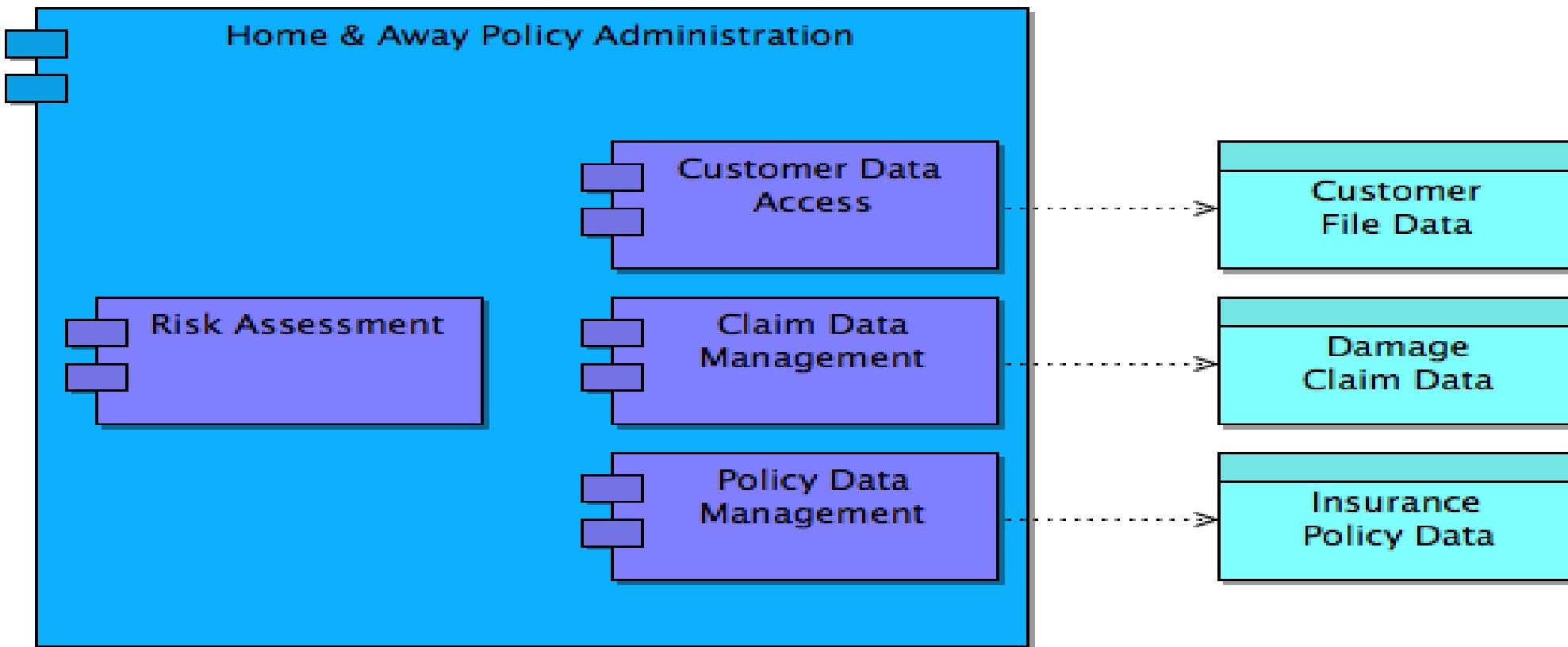
Application Cooperation View



Application Cooperation View

- Το διάγραμμα δείχνει τον τρόπο με τον οποίο συνεργάζονται τα συστατικά του γραφείου εξυπηρέτησης της εταιρίας με συστατικά που αφορούν δεδομένα πελατών, πολιτικών και οικονομικά στοιχεία που προκύπτουν έπειτα και από λήψη στοιχείων από το ανάλογο τραπεζικό σύστημα.

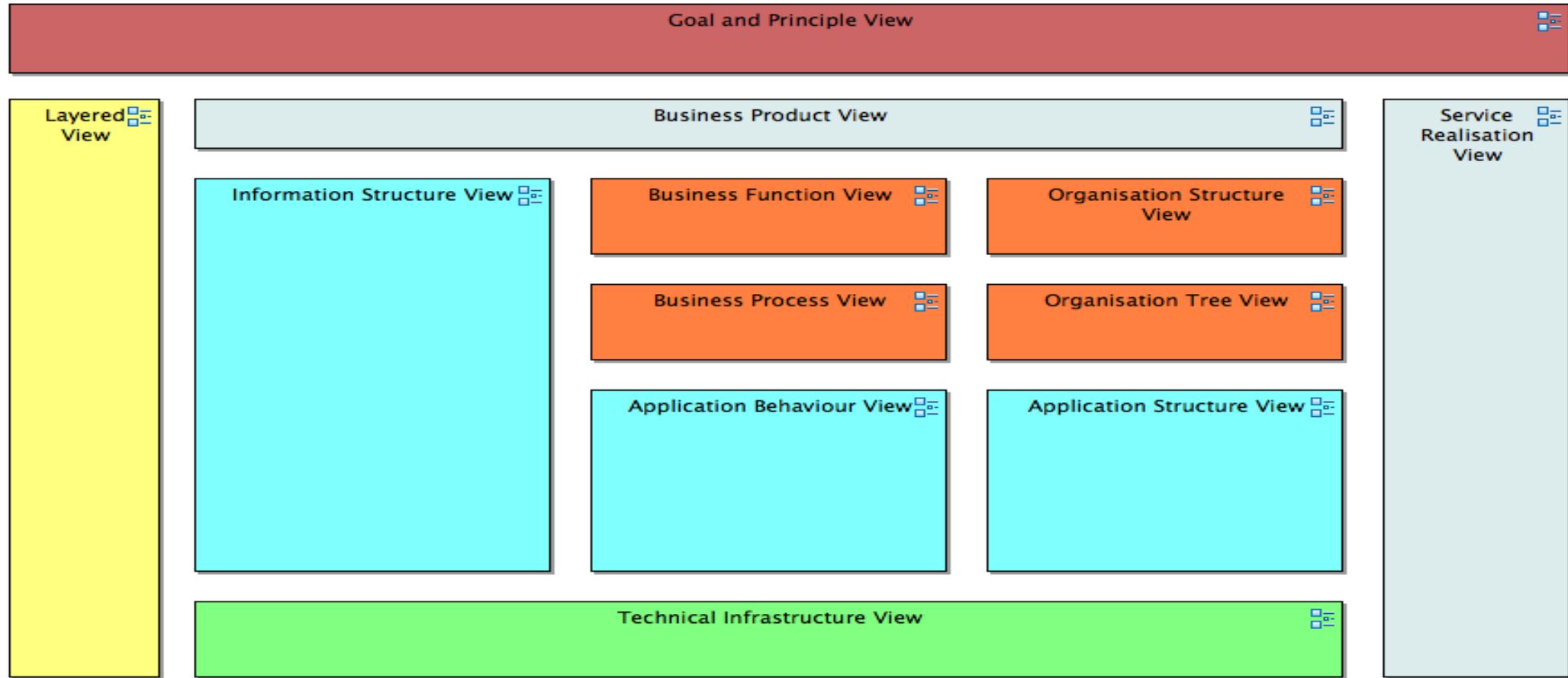
Application Structure View



Application Structure View

- Ασχολείται με τη δομή ενός ή περισσότερων εφαρμογών ή συστατικών.
- Δείχνει τα συστατικά που συνθέτουν τη διαχείριση της πολιτικής του τμήματος Home & Away που ασχολείται με ασφάλειες για ιδιοκτήτες σπιτιών αλλά και για ταξιδιωτικές ασφάλειες. Το συστατικό πρόσβασης σε δεδομένα πελατών έχει πρόσβαση στο αντικείμενο με αρχεία δεδομένων πελατών για τη συλλογή στοιχείων, το συστατικό διαχείρισης αιτήσεων δεδομένων έχει πρόσβαση σε δεδομένα δηλώσεων ζημιών και το συστατικό διαχείρισης δεδομένων πολιτικών έχει πρόσβαση στο αντικείμενο με δεδομένα των πολιτικών της επιχείρησης. Προσθέτοντας σε όλα αυτά τα συστατικά και αυτό της αξιολόγησης ρίσκου προκύπτει το ολικό συστατικό που αφορά τη διαχείριση της πολιτικής Home & Away.

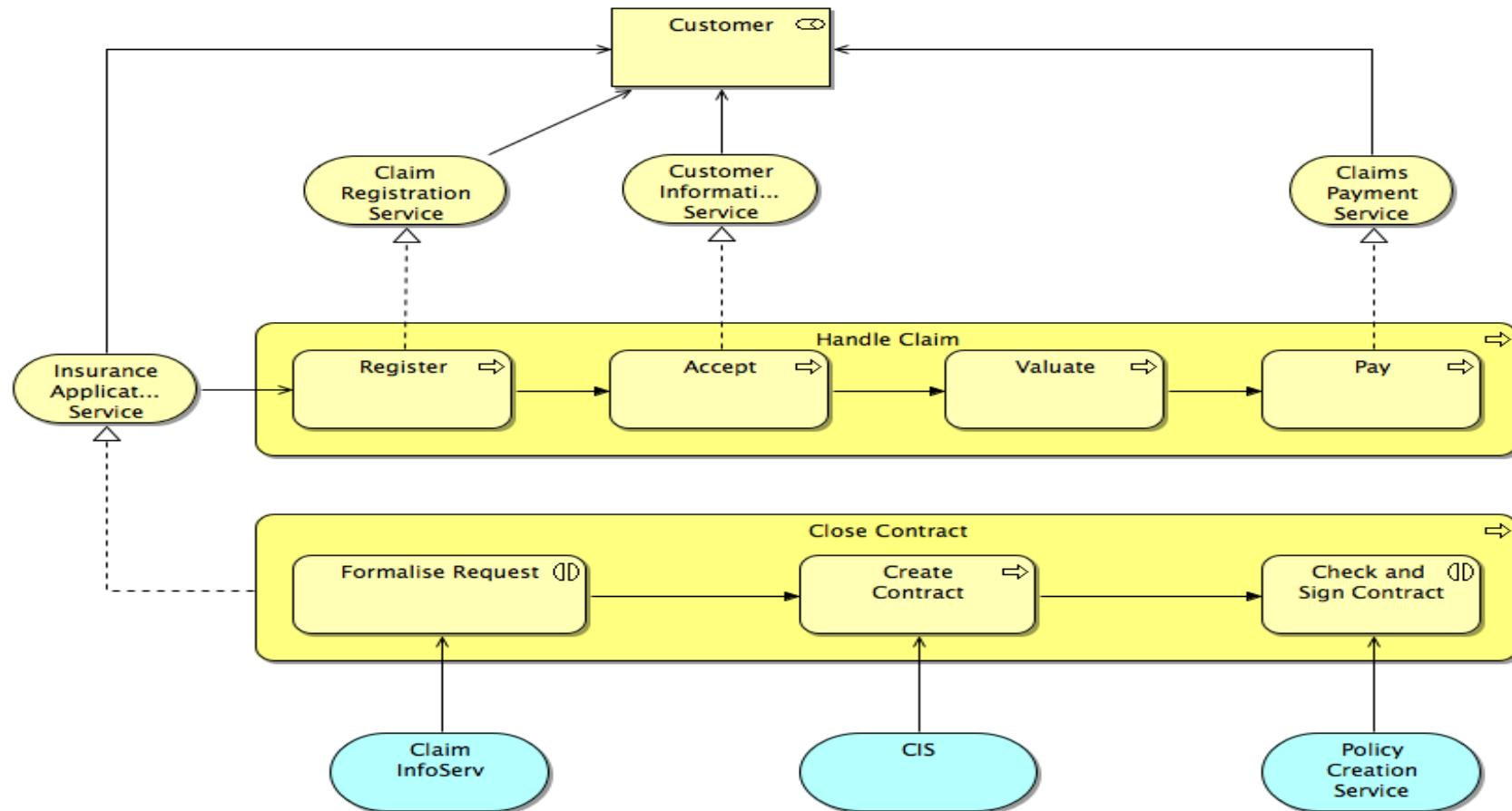
Archimate View



Archimate View

- Παρουσιάζει όλα τα στοιχεία και τα επίπεδα της αρχιτεκτονικής.
- Η εταιρία Archinsurance αναλύεται στους στόχους και τις αρχές της, στο επιχειρηματικό προϊόν που προσφέρει, την πληροφοριακή δομή της, τις επιχειρηματικές λειτουργίες και διαδικασίες από τις οποίες αποτελείται. Ακόμα, το διάγραμμα περιλαμβάνει τη δομή του οργανισμού τόσο με δεντρική μορφή όσο και με παρουσίαση των εμφωλευμένων παραγόντων που την αποτελούν. Δίνεται επίσης η δυνατότητα οργάνωσης της εταιρίας με βάση τη συμπεριφορά ή τη δομή των εφαρμογών της, την τεχνική της υποδομή ή τον τρόπο πραγματοποίησης των υπηρεσιών της. Τέλος, χάρη στην πολυεπίπεδη προβολή απεικονίζονται οι διαφορετικές πτυχές της επιχείρησης όλες μαζί σε ένα συνολικό διάγραμμα.

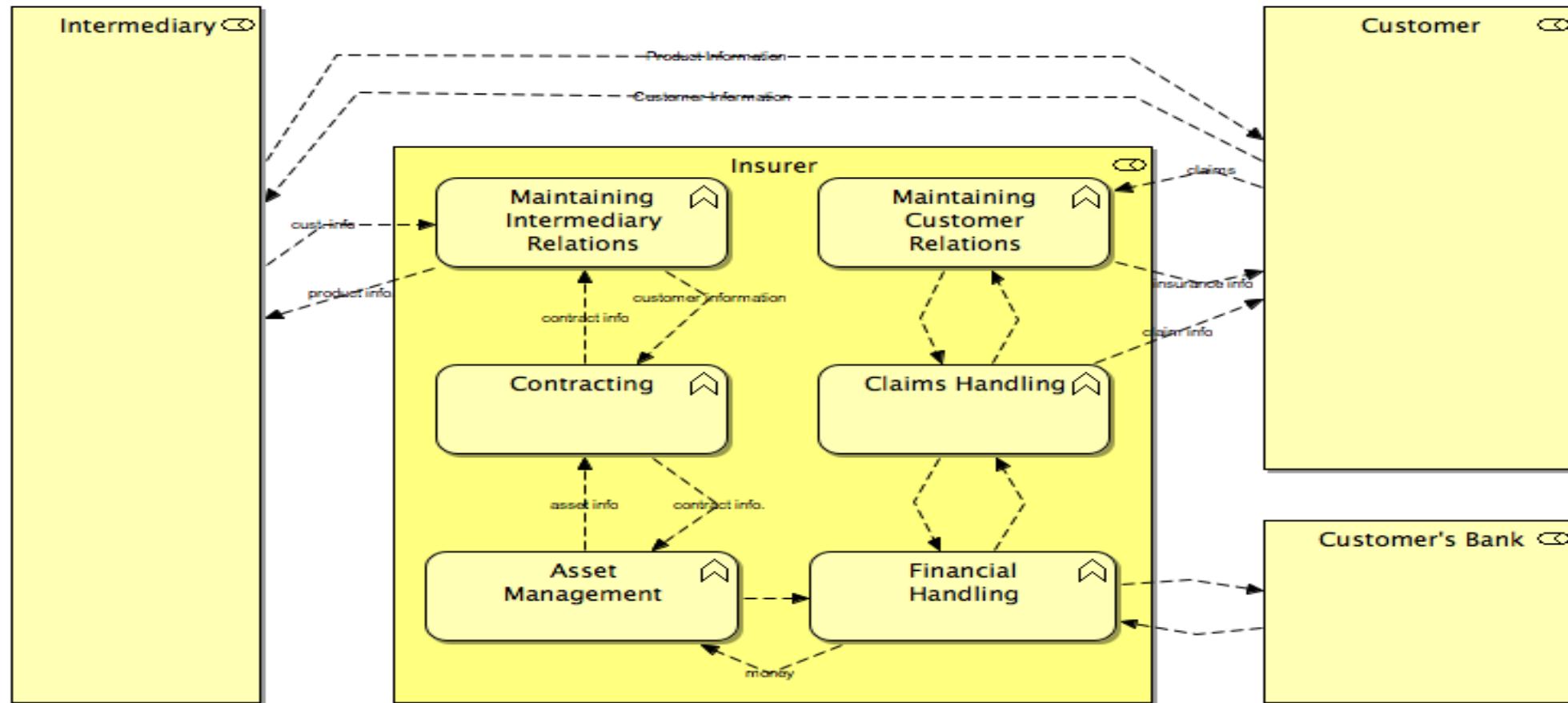
Business Cooperation View



Business Cooperation View

- Καταδεικνύει σχέσεις ενός ή περισσότερων επιχειρηματικών διαδικασιών μεταξύ τους ή και με το περιβάλλον τους.
- Παρουσιάζεται ο τρόπος με τον οποίο ο πελάτης γίνεται τελικός αποδέκτης της υπηρεσίας που επιθυμεί.
- Για να γίνει αυτό όμως έχουν προηγηθεί σχέσεις και αλληλεπιδράσεις πολλών διαδικασιών και υπηρεσιών.

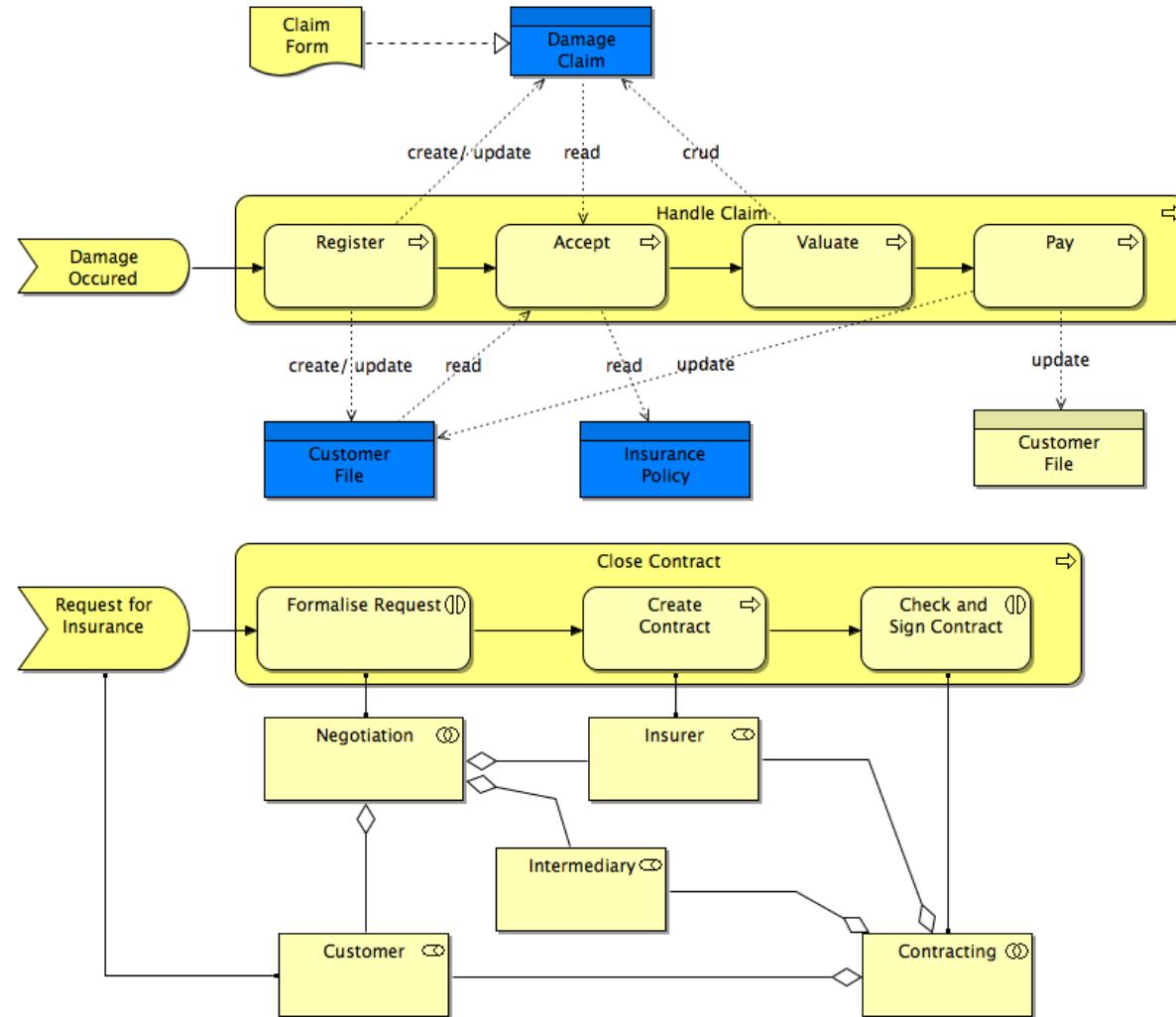
Business Function View



Business Function View

- Δείχνει τις κύριες επιχειρηματικές λειτουργίες και τις πιο σταθερές πτυχές ενός οργανισμού.
- Ο ασφαλιστής, ως ο βασικότερος παράγοντας της εταιρίας, είναι υπεύθυνος για τη διατήρηση σχέσεων με τους μεσάζοντες, οι οποίοι χορηγούν τους πελάτες με πληροφορίες για τα προϊόντα και δέχονται πληροφορίες με στοιχεία των πελατών. Πληροφορίες που αυτοί κατέχουν είναι απαραίτητες για τη λειτουργία των συμβολαίων και έπειτα για τη διαχείριση των κεφαλαίων. Στη συνέχεια πραγματοποιείται η λειτουργία οικονομικής διαχείρισης, σε συνεργασία με την τράπεζα του πελάτη, και διαχείρισης αιτήσεων, καταλήγοντας στη διατήρηση των σχέσεων με τους πελάτες και παρέχοντας σε αυτούς την επιθυμητή ασφάλεια.

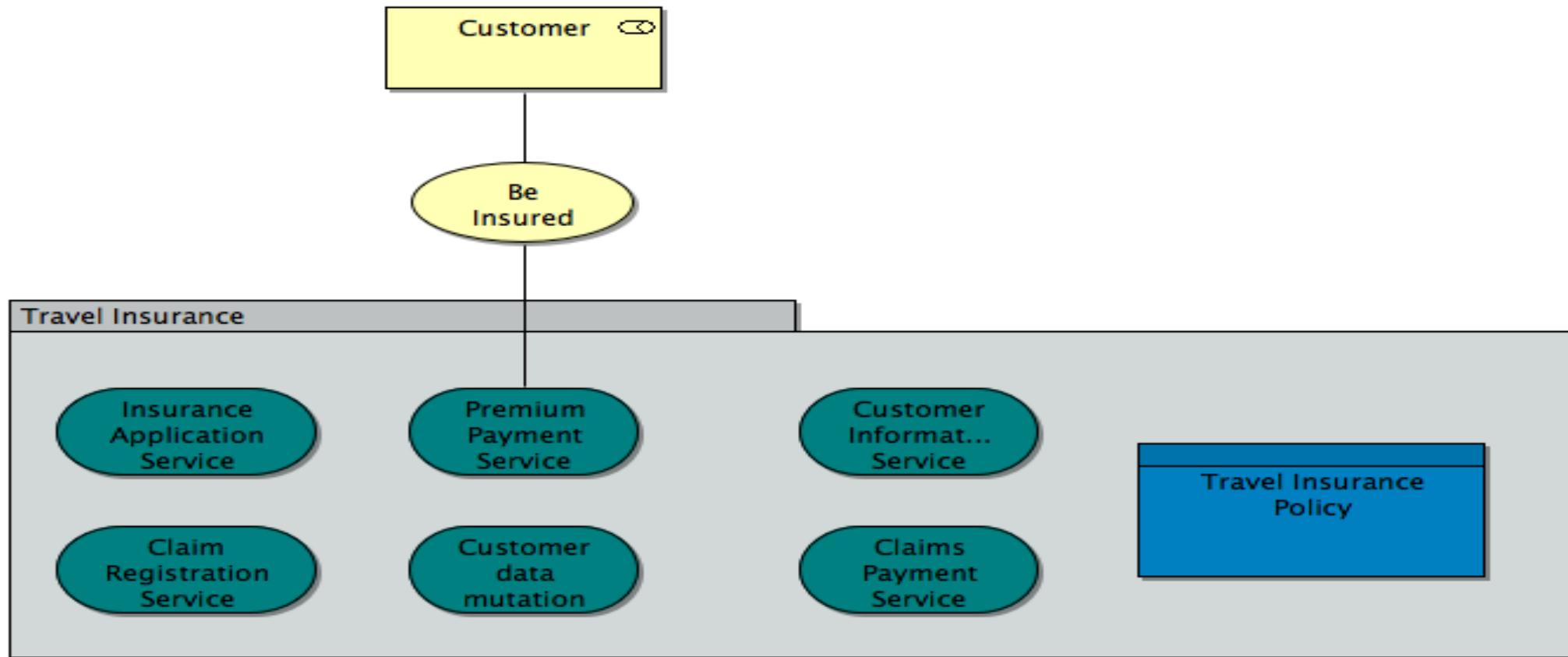
Business Process View



Business Process View

- Παρουσιάζει τη σύνθεση μιας ή περισσότερων επιχειρηματικών διαδικασιών.
- Η παρουσίαση κάποια ζημιάς συνοδεύεται από ένα γεγονός που τη δηλώνει και σηματοδοτεί την έναρξη των απαραίτητων διαδικασιών για το χειρισμό της. Αυτές οι διαδικασίες περιλαμβάνουν την εγγραφή του χρήστη και της ζημιάς με τη δημιουργία ή την ανανέωση του ανάλογου αντικειμένου. Στη συνέχεια και αφού έχει προηγηθεί επικοινωνία με το αρχείο της ζημιάς και του πελάτη επέρχεται η αποδοχή της αίτησης και η ανάγνωση των όρων της πολιτικής που αυτή περιλαμβάνει. Έπειτα, εκτιμάται το επίπεδο της ζημιάς, πραγματοποιείται η πληρωμή και ανανεώνεται το αρχείο του πελάτη.
- Ακόμα, το γεγονός αίτησης για ασφάλεια, το οποίο τροφοδοτεί τη διαδικασία για το κλείσιμο του συμβολαίου. Αυτό περιλαμβάνει τη διάδραση επισημοποίησης της αίτησης, τη δημιουργία του συμβολαίου και τέλος τον έλεγχο και την υπογραφή του συμβολαίου. Όλες αυτές οι διαδικασίες φυσικά συνδέονται και με άλλες διαδράσεις όπως διαπραγματεύσεις και γραφειοκρατεία που συνοδεύει τα συμβόλαια αλλά και με τους ρόλους του ασφαλιστή, των μεσάζοντων και του πελάτη.

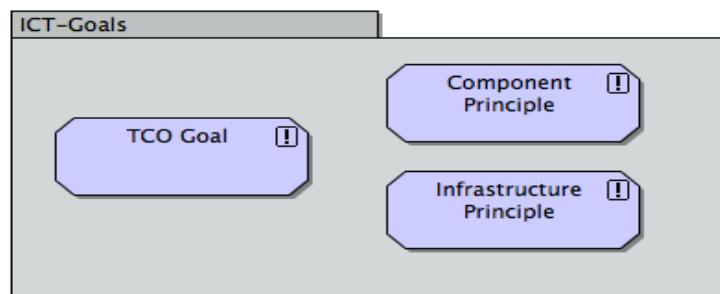
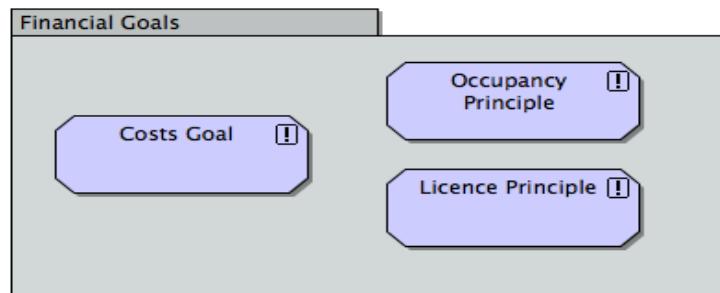
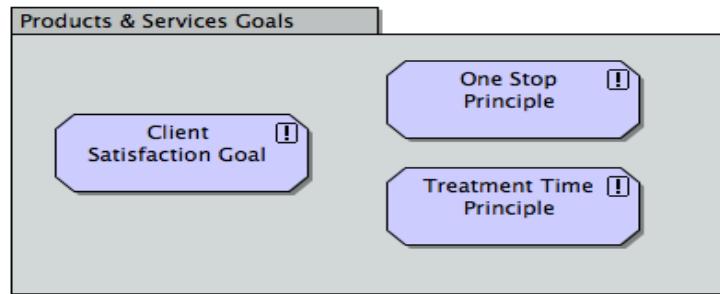
Business Product View



Business Product View

- Απεικονίζει την αξία των προϊόντων που παρέχονται σε πελάτες ή άλλους εξωτερικούς παράγοντες.
- Το διάγραμμα φανερώνει την αξία που κατοχυρώνει ένας ασφαλισμένος πελάτης της εταιρίας κατά τη διάρκεια κάποιου ταξιδιού του. Μια ταξιδιωτική ασφάλιση περιλαμβάνει υπηρεσίες αίτησης για ασφάλιση, αίτημα εγγραφής, υπηρεσίες πριμοδότησης, ασφάλεια των δεδομένων των πελατών και γενικές πληροφορίες για αυτούς και υπηρεσίες διαχείρισης των πληρωμών. Τέλος, η παροχή μιας ταξιδιωτικής ασφάλισης επισφραγίζεται με την ύπαρξη ενός συμβολαίου, το οποίο παρουσιάζει την πολιτική της εταιρίας και περιλαμβάνει τις υποχρεώσεις της ως προς τον πελάτη σε σχέση με το προϊόν που επιλέχθηκε αλλά και τα δικαιώματα του πελάτη.

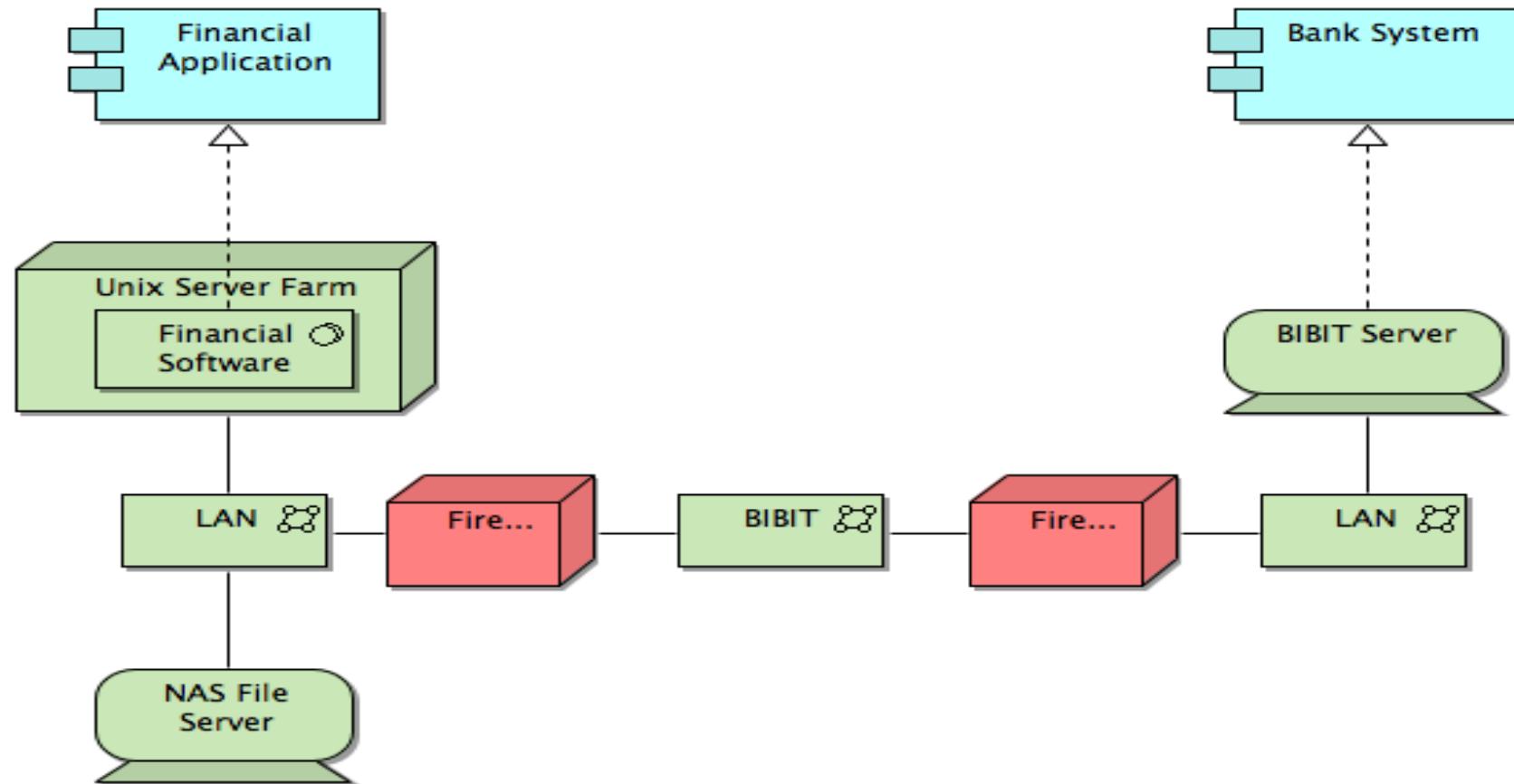
Goal and Principle View



Goal and Principle View

- Μοντελοποιεί την κινητροδότηση χωρίς να εστιάζει σε συγκεκριμένα στοιχεία.
- Η εταιρία Archinsurance θέτει λοιπόν στόχους σε τρεις διαφορετικές κατηγορίες, σχετικά με τα προϊόντα και τις υπηρεσίες που προσφέρει, τα οικονομικά και την τεχνολογική της υποδομή.

Implementation and Installation View



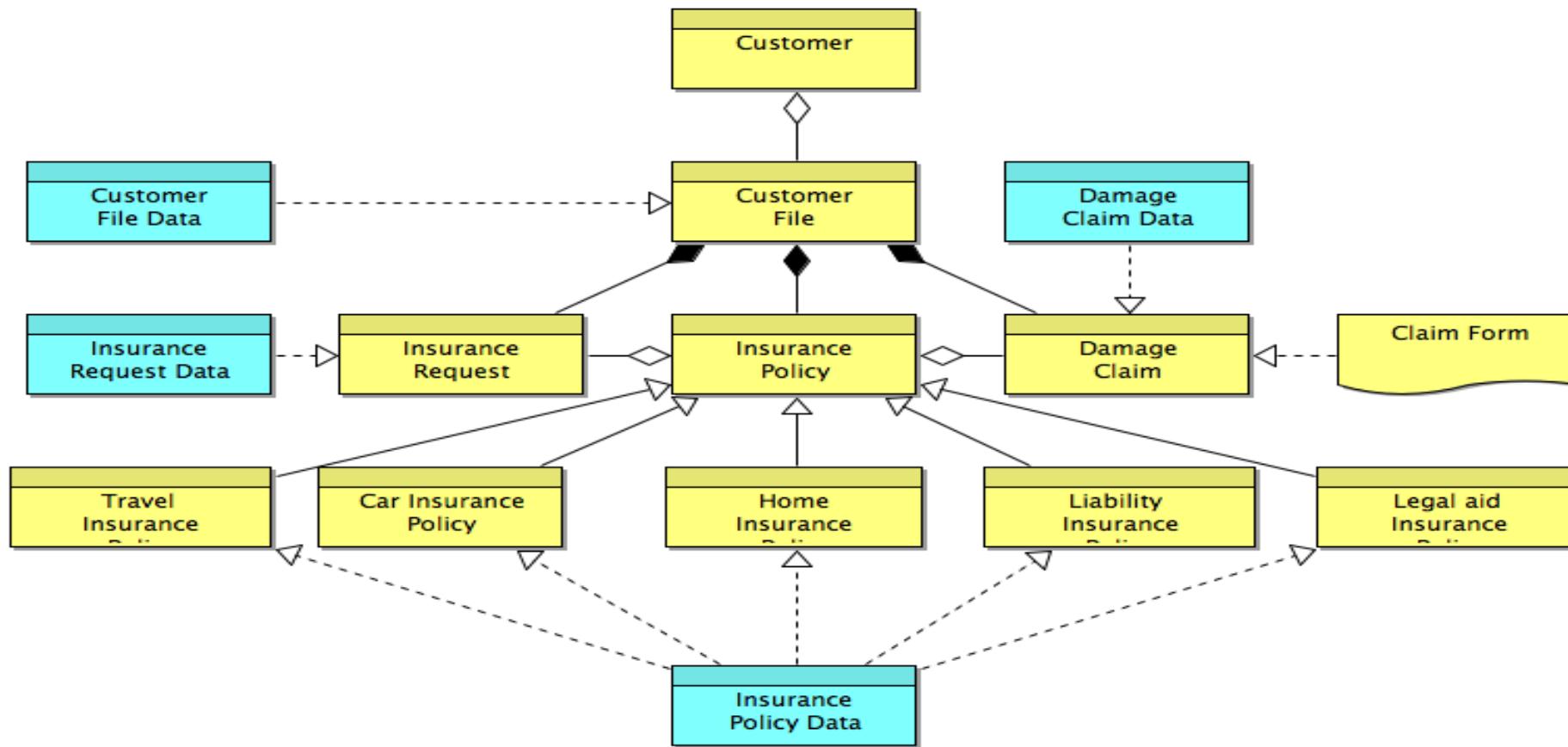
Implementation and Installation View

- Δείχνει πώς μία ή περισσότερες εφαρμογές πραγματοποιούνται στην υποδομή.
- Η τεχνολογική υποδομή της Archinsurance βασίζεται σε ένα δίκτυο πληρωμών BIBIT, το οποίο παρέχει υπηρεσίες πληρωμών για ταχυδρομικές παραγγελίες και διαδικτυακές πωλήσεις. Καθώς περιλαμβάνει ιδιαιτέρως σημαντικές λειτουργίες από τις οποίες κρίνεται σε σημαντικό βαθμό η αξιοπιστία της επιχείρησης προστατεύεται από Firewalls τόσο στη σύνδεσή του μέσω τοπικού δικτύου LAN με τον NAS Server όσο και με τον BIBIT Server που συνεργάζεται με το τραπεζικό σύστημα του πελάτη. Το Firewall που επιλέγεται ελέγχει την εισερχόμενη αλλά και εξερχόμενη κίνηση αναλύοντας πακέτα δεδομένων και καθορίζοντας εάν επιτρέπεται να περάσουν ή όχι, με βάση κάποιο σύνολο κανόνων. Χρησιμοποιείται για να εγκαταστήσει ένα είδος φράγματος ανάμεσα σε αξιόπιστα, ασφαλή εσωτερικά δίκτυα και σε αυτά με τα οποία υπάρχει επικοινωνία.

Implementation and Installation View

- Ο NAS Server (Network-Attached Storage) είναι ένας χώρος αποθήκευσης δεδομένων σε επίπεδο αρχείων ο οποίος συνδέεται με ένα δίκτυο υπολογιστών και έτσι μπορεί να προσφέρει δεδομένα σε ετερογενείς πελάτες. Πρόκειται για συστήματα δικτυωμένων συσκευών που περιέχουν έναν ή περισσότερους σκληρούς δίσκους αφαιρώντας έτσι την ευθύνη του διαμοιρασμού αρχείων από τους υπόλοιπους server του δικτύου. Μέσω του δικτύου LAN συνδέεται με μια Unix Server Farm, μια συλλογή δηλαδή από διακομιστές οι οποίοι ικανοποιούν ανάγκες που ξεφεύγουν των δυνατοτήτων ενός μηχανήματος. Ένα λογισμικό συστήματος που περιλαμβάνει αφορά το οικονομικό σκέλος της επιχείρησης. Αυτό το οικονομικό λογισμικό στη συνέχεια πραγματοποιεί και τη λειτουργικότητα μια οικονιμικής αίτησης.
- Το BIBIT δίκτυο μέσω του firewall σχετίζεται με ένα δίκτυο LAN με τον BIBIT Server. Αυτός περιλαμβάνει όλες τις απαραίτητες πληροφορίες που σχετίζονται με τον τρόπο πληρωμής της υπηρεσίας που παρέχθηκε στον πελάτη και έτσι στη συνέχεια πραγματοποιεί και το τραπεζικό σύστημα με βάση αυτές.

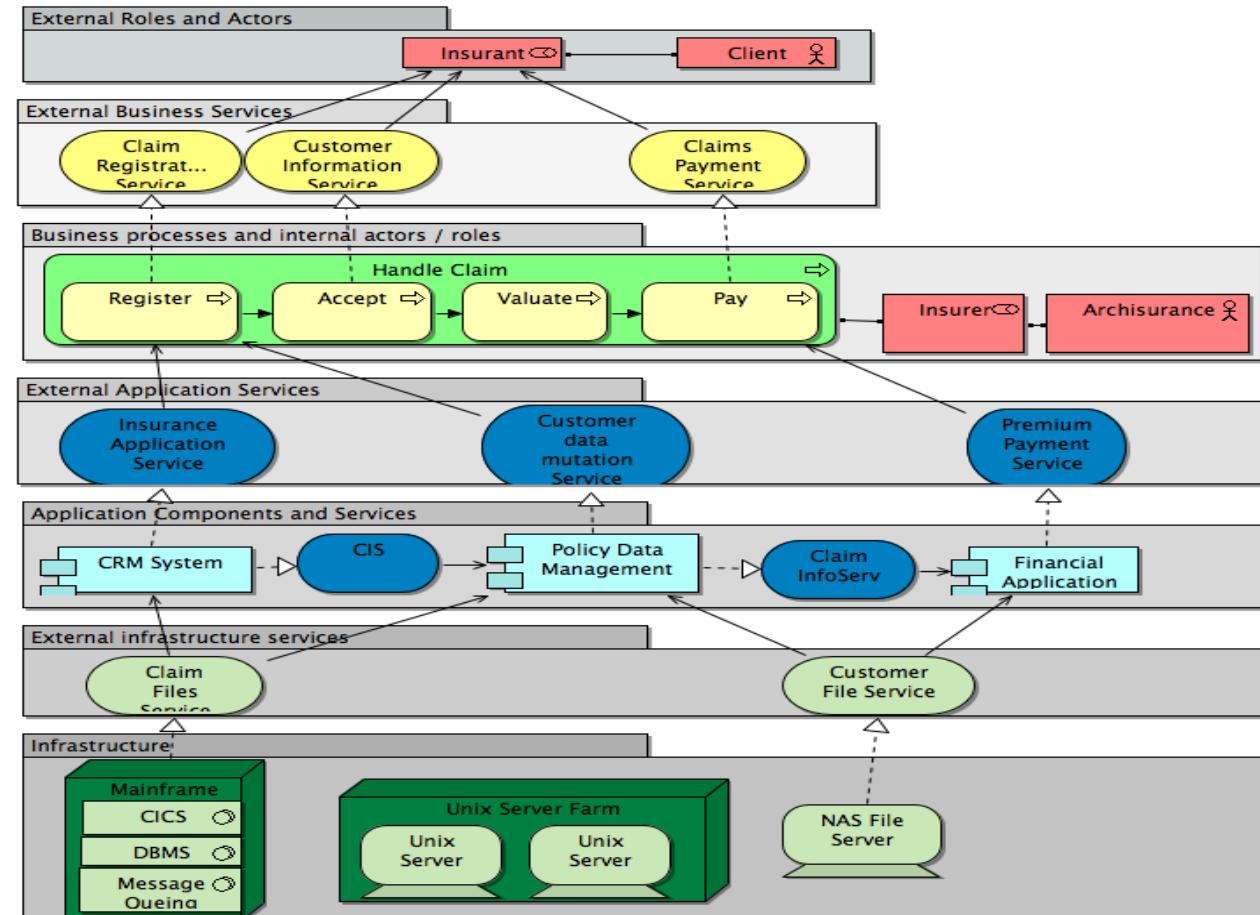
Information Structure View



Information Structure View

- Δείχνει τη δομή των πληροφοριών που χρησιμοποιούνται στην επιχείρηση.
- Το αντικείμενο ‘Πελάτης’ **συμπεριλαμβάνει** τα ‘Άρχεία Πελατών’, τα οποία **αποτελούνται από** την ‘Αίτηση για ασφάλεια’, την ‘Πολιτική ασφάλειας’ και τη ‘Δήλωση ζημιάς’. Τα ‘Άρχεία Πελατών’ **πραγματοποιούν** το αντικείμενο ‘Δεδομένα αρχείων πελατών’, η ‘Αίτηση για ασφάλεια’ **πραγματοποιεί** το αντικείμενο ‘Δεδομένα αίτησης για ασφάλεια’ και η ‘Δήλωση ζημιάς’ **πραγματοποιεί** τόσο το αντικείμενο ‘Δεδομένα δήλωσης ζημιάς’ όσο και την πληροφορία που σχετίζεται με την ‘Αίτηση φόρμας’. Στη συνέχεια παρουσιάζεται η σχέση **εξειδίκευσης** της ‘Πολιτικής ασφάλειας’ στα αντικείμενα ‘Πολιτική ταξιδιωτικής ασφάλειας’, ‘Πολιτική ασφάλειας αυτοκινήτου’, ‘Πολιτική ασφάλειας σπιτιού’, ‘Πολιτική ασφάλειας ευθύνης’ και τέλος ‘Πολιτική νομικής βοήθειας’. Αυτά είναι και τα πιο συγκεκριμένα αντικείμενα που **πραγματοποιούν** τα ‘Δεδομένα πολιτικής ασφάλειας’

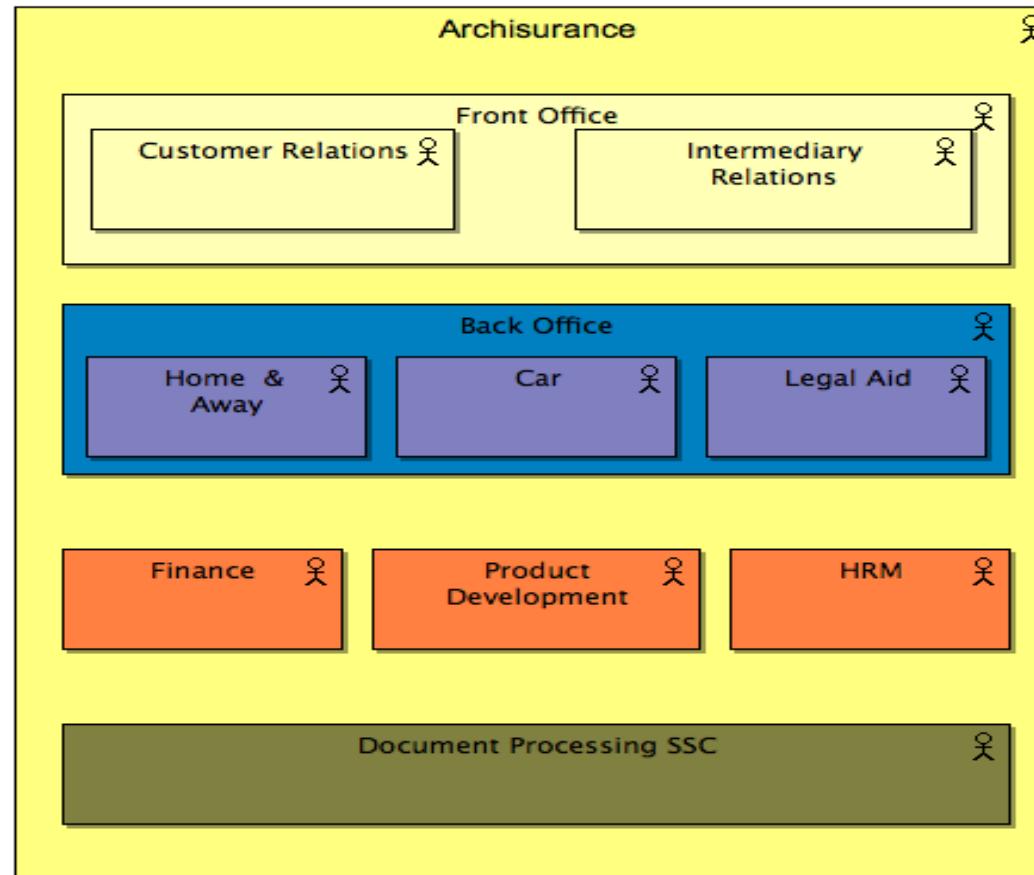
Layered View



Layered View

- Απεικονίζει πολλά επίπεδα και πτυχές της επιχειρηματικής αρχιτεκτονικής σε ένα διάγραμμα.
- Το διάγραμμα που αφορά την Archinsurance αποτελείται από τα διαφορετικά επίπεδα που την απαρτίζουν. Αυτά αναλύονται σε εξωτερικούς ρόλους και παράγοντες, εξωτερικές επιχειρηματικές υπηρεσίες, επιχειρησιακές διαδικασίες και εσωτερικούς παράγοντες και ρόλους, εξωτερικές υπηρεσίες εφαρμογών, συστατικά και υπηρεσίες εφαρμογών, εξωτερικές υπηρεσίες υποδομής και την απαραίτητη τεχνολογική υποδομή.

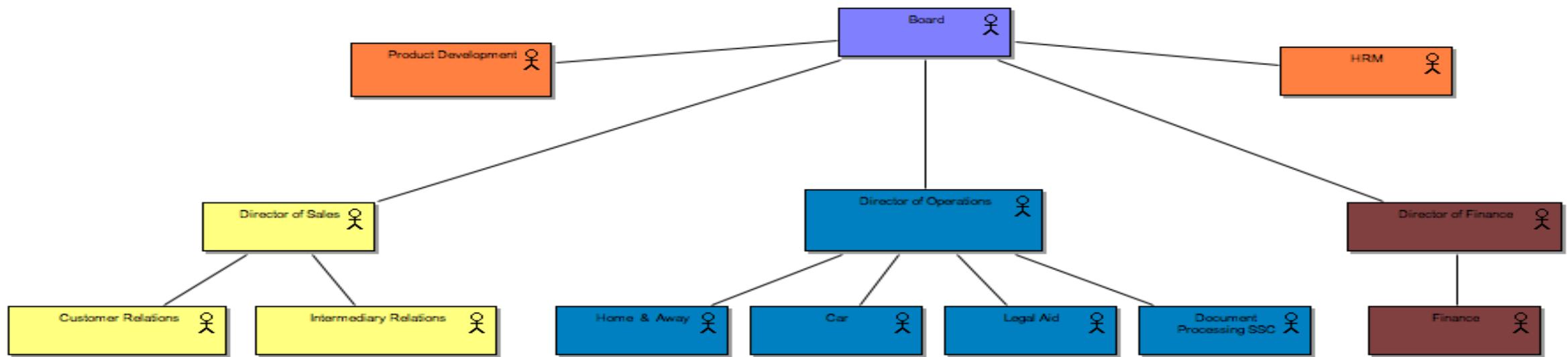
Organisation Structure View



Organisation Structure View

- Εστιάζει στην εσωτερική οργάνωση μιας εταιρίας.
- Παρουσιάζεται η ίδια η εταιρία Archinsurance ως ένας γενικός επιχειρηματικός παράγοντας ο οποίος περικλείει πολλούς υπο-παράγοντες που συνθέτουν μια λειτουργική επιχείρηση. Πρώτα εμφανίζεται το μπροστινό γραφείο εξυπηρέτησης με τα τμήματά του να περιλαμβάνουν τις σχέσεις με τους πελάτες και σχέσεις με μεσάζοντες. Έπειτα, υπάρχει το γραφείο οργανωτικής υποστήριξης το οποίο αναλύεται στους παράγοντες οι οποίοι είναι υπεύθυνοι για το τμήμα Home & Away, Car και Legal Aid. Τους επιχειρηματικούς παράγοντες της Archinsurance συμπληρώνουν ο τομέας οικονομικών, παραγωγής προϊόντων, διαχείρισης ανθρωπίνου δυναμικού και επεξεργασίας εγγράφων.

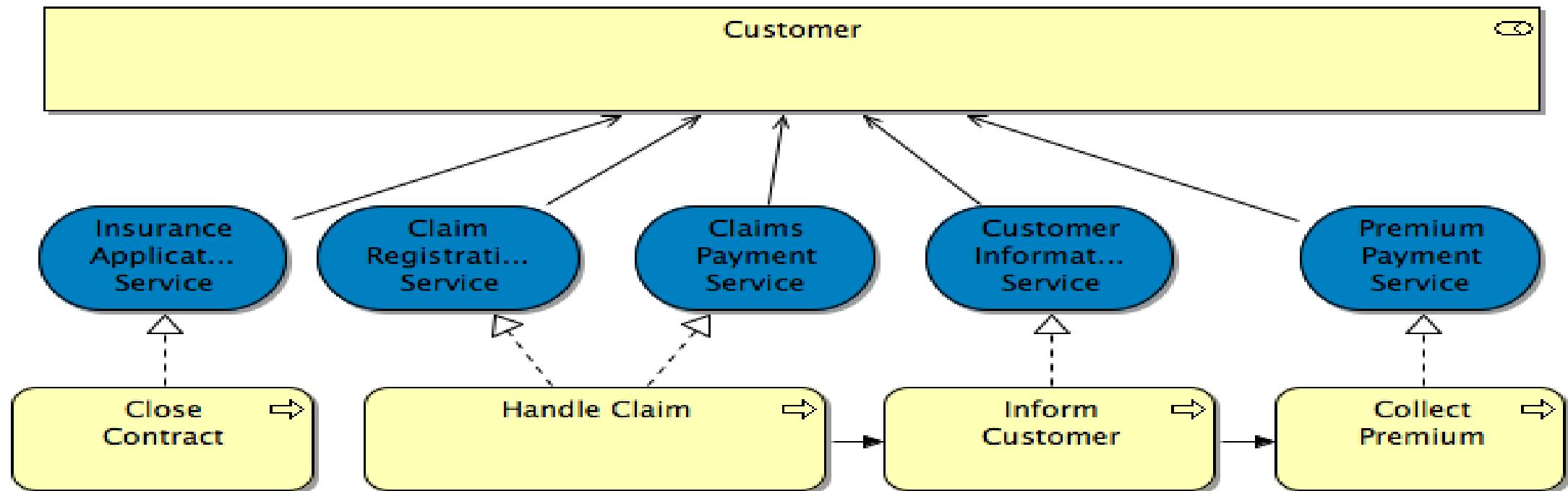
Organisation Tree View



Organisation Tree View

- Στη δενδρική μορφή του βλέπουμε στη ρίζα τον παράγοντα της επιτροπής διεύθυνσης και αμέσως μετά τον υπεύθυνο δημιουργίας προϊόντων και τον υπεύθυνο ανθρωπίνου δυναμικού. Στη συνέχεια θέση έχουν ο διευθυντής πωλήσεων, που σχετίζεται με τα τμήματα σχέσεων με πελάτες και μεσάζοντες, ο επιχειρησιακός διευθυντής που σχετίζεται με τα τμήματα Home & Away, Car, Legal Aid αλλά και με το τμήμα επεξεργασίας εγγράφων και τέλος ο διευθυντής οικονομικών που σχετίζεται αποκλειστικά με το οικονομικό τμήμα της επιχείρησης.

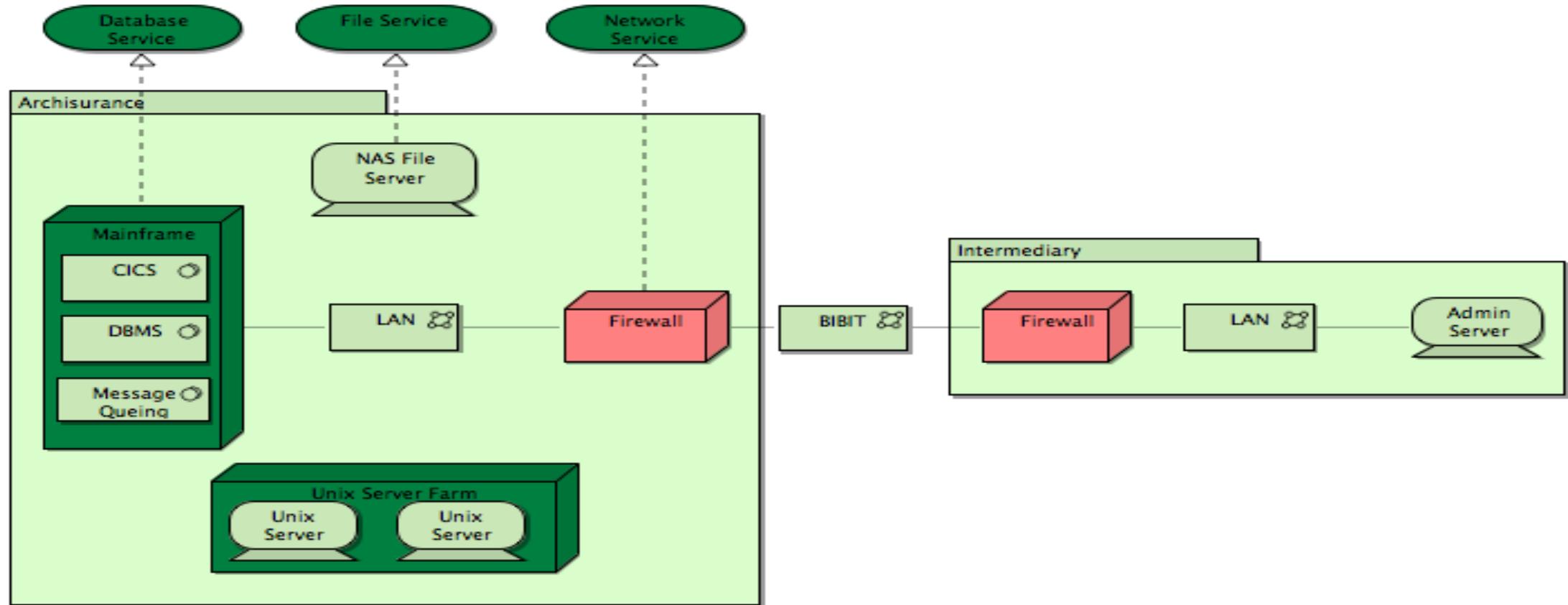
Service Realisation View



Service Realisation View

- Δείχνει τον τρόπο που μια ή περισσότερες επιχειρηματικές υπηρεσίες πραγματοποιούνται από τις βασικές διαδικασίες.
- Η διαδικασία “κλείσιμο συμβολαίου” πραγματοποιεί την υπηρεσία εφαρμογής ασφάλειας η οποία χρησιμοποιείται από τον πελάτη. Η διαδικασία “χειρισμός αίτησης” πραγματοποιεί τόσο την υπηρεσία εγγραφής του πελάτη όσο και της πληρωμής της παρεχόμενης υπηρεσίας και πυροδοτεί την έναρξη της διαδικασίας “πληροφόρηση πελάτη” που πραγματοποιεί την ανάλογη υπηρεσία πληροφόρησης. Τέλος, ενεργοποιείται η διαδικασία συλλογής των ασφάλιστρων με τη δημιουργία της υπηρεσίας πληρωμής την οποία υποχρεούται να ικανοποιήσει ο πελάτης.

Technical Infrastructure View



Technical Infrastructure View

- Περιλαμβάνει στοιχεία υποδομής λογισμικού και υλικού που υποστηρίζουν το επίπεδο εφαρμογών.
- **Λογισμικά CICS** (Customer Information Control System): ένα online πρόγραμμα επεξεργασίας συναλλαγών που κάνοντας χρήση ενός συνόλου εργαλείων κατασκευάζει εφαρμογές συναλλαγής με πελάτες
- **Λογισμικά DBMS** (Database Management System): ειδικά σχεδιασμένες εφαρμογές που αλληλεπιδρούν με το χρήστη και άλλες εφαρμογές ή με την ίδια τη βάση δεδομένων για να συλλέξουν και να αναλύσουν δεδομένα
- **Message Queuing**: λογισμικά που χρησιμοποιούνται για την ενδοεπικοινωνία μέσα σε μια διαδικασία είτε πρόκειται για πέρασμα ελέγχου ή περιεχομένου. Παρέχουν ένα ασύγχρονο πρωτόκολλο επικοινωνίας, που σημαίνει ότι ο αποστολέας και ο παραλήπτης ενός μηνύματος δε χρειάζεται να αλληλεπιδράσουν με την ουρά μηνυμάτων ταυτόχρονα. Τα μηνύματα τοποθετούνται σε ουρά και αποθηκεύονται μέχρι να τα ανακτήσει ο παραλήπτης.

Technical Infrastructure View

- Η επιχείρηση αποτελέσται από ένα κόμβο κεντρικού υπολογιστή, εξοπλισμένου με λογισμικά CICS, DBMS και Message Queuing.
- Τμήμα της τεχνολογικής υποδομής της εταιρίας είναι ένα σύμπλεγμα δύο εξυπηρετητών Unix και ενός NAS ο οποίος συνδέεται με ένα δίκτυο υπολογιστών και έτσι μπορεί να προσφέρει δεδομένα σε ετερογενείς πελάτες.
- Η τεχνολογική υποδομή της Archinsurance βασίζεται σε ένα δίκτυο πληρωμών BIBIT, το οποίο παρέχει υπηρεσίες πληρωμών για ταχυδρομικές παραγγελίες και διαδικτυακές πωλήσεις. Προστατεύεται από Firewalls τόσο στη σύνδεσή του μέσω LAN με το κεντρικό σύστημα υπολογιστών της επιχείρησης, όσο και με το σύνολο των διαμεσολαβητών. Το Firewall που επιλέγεται ελέγχει την εισερχόμενη αλλά και εξερχόμενη κίνηση.
- Όλη αυτή η τεχνολογική υποδομή έχει ως αποτέλεσμα την ανάπτυξη και δημιουργία υπηρεσιών βάσεων δεδομένων που προκύπτουν από το σύστημα κεντρικού υπολογιστή της επιχείρησης, υπηρεσιών αρχείων παραγόμενων από τον NAS File Server και υπηρεσίας δικτύου όπως προκύπτει από το τείχος προστασίας που έχει επιλεγεί.



Δομικά Στοιχεία Μοντελοποίησης

Η παλέτα για τη δημιουργία μοντέλων ArchiMate

Palette

- Magic Connector
- Specialisation relation
- Composition relation
- Aggregation relation
- Assignment relation
- Realisation relation
- Triggering relation
- Flow relation
- Used By relation
- Access relation
- Association relation
- Influence relation
- Junction

- Actor
- Role
- Collaboration
- Interface
- Function
- Process
- Event
- Interaction
- Product
- Contract
- Service
- Value
- Meaning
- Representation
- Object
- Location

- Component
- Collaboration
- Interface
- Service
- Function
- Interaction
- Data Object

- Artifact
- Communication Path
- Network
- Infrastructure Interface
- Infrastructure Function
- Infrastructure Service
- Node
- System Software
- Device

- Stakeholder
- Driver
- Assessment
- Goal
- Principle
- Requirement
- Constraint

- Work Package
- Deliverable
- Plateau
- Gap

Επίπεδο Εφαρμογών [1/6]

 Component

 Collaboration

 Interface

 Service

 Function

 Interaction

 Data Object

Component: Ένα συστατικό εφαρμογής ορίζεται ως ένα σπονδυλωτό, ικανό να αναπτυχθεί και να αντικατασταθεί, μέρος ενός συστήματος που ενσωματώνει το περιεχόμενό του και εκθέτει τη λειτουργικότητά του μέσω ενός συνόλου διεπαφών. Ένα συστατικό εφαρμογής είναι μια αυτόνομη μονάδα λειτουργικότητας του επιπέδου εφαρμογών και μπορεί να εκτελέσει μία ή περισσότερες λειτουργίες εφαρμογών. Είναι επαναχρησιμοποιούμενο και αντικαταστάσιμο και υπάρχει πρόσβαση σε αυτό μόνο μέσω ενός συνόλου διεπαφών εφαρμογών. Συνεργαζόμενα συστατικά εφαρμογών είναι συνδεδεμένα μέσω ενός συνόλου συνεργασιών εφαρμογών. Ένα συστατικό μπορεί να ανατεθεί σε μία ή περισσότερες λειτουργίες εφαρμογών, επιχειρηματικές διαδικασίες ή επιχειρηματικές λειτουργίες. Έχει μία ή περισσότερες διεπαφές εφαρμογών μέσω των οποίων εκθέτει τη λειτουργικότητά του. Ανήκει στην κατηγορία δομικών στοιχείων του επιπέδου εφαρμογών.

Τυπικές σχέσεις που το αφορούν:

Used by -- application component

Composition -- application component

Realisation -- application service

Assignment -- application function, business process, business function

Επίπεδο Εφαρμογών [2/6]

 Component

 Collaboration

 Interface

 Service

 Function

 Interaction

 Data Object

Collaboration: Μια συνεργασία εφαρμογών ορίζεται ως μια (προσωρινή) διαμόρφωση δύο ή περισσότερων συστατικών που συνεργάζονται για να εκτελέσουν από κοινού αλληλεπιδράσεις εφαρμογών. Μια συνεργασία καθορίζει ποια συστατικά συνεργάζονται για να εκτελεστεί κάποια εργασία. Η συνεργατική συμπεριφορά, συμπεριλαμβανομένου για παράδειγμα του μοτίβου επικοινωνίας των συστατικών, μοντελοποιείται από μια αλληλεπίδραση εφαρμογών. Ανήκει στην κατηγορία δομικών στοιχείων του επιπέδου εφαρμογών.

Τυπικές σχέσεις που το αφορούν:

Assignment -- application interaction,

business interaction Used by -- application interface

Composition -- application interface

Interface: Μια διεπαφή εφαρμογής δηλώνει τον τρόπο με τον οποίο ένα συστατικό συνδέεται με το περιβάλλον του. Καθορίζει το πώς η λειτουργικότητα ενός συστατικού μπορεί να προσεγγιστεί από άλλα συστατικά (παρεχόμενη διεπαφή) και ποια λειτουργικότητα απαιτεί από το περιβάλλον (απαιτούμενη διεπαφή). Μια διεπαφή εφαρμογής αποκαλύπτει μια υπηρεσία εφαρμογής στο περιβάλλον. Η υπηρεσία αυτή μπορεί να αποκαλυφθεί στο περιβάλλον από διαφορετικές διεπαφές. Η διεπαφή μπορεί να ανατεθεί σε υπηρεσίες εφαρμογών ή επιχειρήσεων. Ανήκει στην κατηγορία δομικών στοιχείων του επιπέδου εφαρμογών.

Τυπικές σχέσεις που το αφορούν:

Assignment -- application service, business service

Επίπεδο Εφαρμογών [3/6]

 Component

 Collaboration

 Interface

 Service

 Function

 Interaction

 Data Object

Service: Μια υπηρεσία εφαρμογής ορίζεται ως μια εξωτερικά ορατή μονάδα λειτουργικότητας, παρεχόμενη από ένα ή περισσότερα συστατικά, εκτιθέμενη μέσω καλά ορισμένων διεπαφών και με νόημα για το περιβάλλον. Μια υπηρεσία εφαρμογής εκθέτει τη λειτουργικότητα των συστατικών στο περιβάλλον τους, στην οποία στη συνέχεια υπάρχει πρόσβαση μέσω μίας ή περισσότερων διεπαφών εφαρμογής. Μια υπηρεσία υλοποιείται από μία ή παραπάνω λειτουργίες εφαρμογής οι οποίες εκτελούνται από το συστατικό. Μπορεί να απαιτήσει, να χρησιμοποιήσει και να παράγει αντικείμενα δεδομένων. Υπηρεσίες εφαρμογής χρησιμοποιούνται από επιχειρηματικές διαδικασίες, επιχειρηματικές λειτουργίες, αλληλεπιδράσεις ή λειτουργίες εφαρμογής. Μπορούν να υλοποιήσουν μια υπηρεσία εφαρμογής, να τους ανατεθεί μια διεπαφή εφαρμογής και έχουν πρόσβαση σε αντικείμενα δεδομένων. Μια υπηρεσία εφαρμογής ανήκει στην κατηγορία στοιχείων συμπεριφοράς του επιπέδου εφαρμογών.

Τυπικές σχέσεις που την αφορούν:

Used by -- business process, business function, business interaction, application function

Realisation -- application function

Assignment -- application interface

Access -- data object

Επίπεδο Εφαρμογών [4/6]

 Component

 Collaboration

 Interface

 Service

 Function

 Interaction

 Data Object

Function: Μια λειτουργία εφαρμογής ορίζεται ως μια αναπαράσταση μιας συνεκτικής ομάδας εσωτερικής συμπεριφοράς ενός συστατικού εφαρμογής. Η εσωτερική συμπεριφορά της λειτουργίας εφαρμογής δεν είναι ορατή στο χρήστη ενός συστατικού. Η συμπεριφορά αυτή εξωτερικεύεται μέσω μίας ή περισσότερων υπηρεσιών. Μια λειτουργία εφαρμογής μπορεί να υλοποιήσει υπηρεσίες εφαρμογής και να χρησιμοποιήσει υπηρεσίες άλλων λειτουργιών εφαρμογής καθώς και υπηρεσίες υποδομής. Μια λειτουργία εφαρμογής έχει πρόσβαση σε αντικείμενα δεδομένων και μπορεί να της ανατεθεί κάποιο συστατικό εφαρμογής. Ανήκει στην κατηγορία στοιχείων συμπεριφοράς του επιπέδου εφαρμογών.

Τυπικές σχέσεις που την αφορούν:

Used by -- application service

Triggering -- application function

Access -- data object

Assignment -- application component

Επίπεδο Εφαρμογών [5/6]

 Component

 Collaboration

 Interface

 Service

 Function

 Interaction

 Data Object

Interaction: Μια αλληλεπίδραση εφαρμογής ορίζεται ως η εξωτερικά ορατή μονάδα συμπεριφοράς που εκτελείται από μια συνεργασία δύο ή περισσότερων συστατικών. Μπορεί ακόμα να προσδιορίσει την εξωτερικά ορατή συμπεριφορά που απαιτείται για την πραγματοποίηση μιας υπηρεσίας εφαρμογής. Μια συνεργασία εφαρμογής μπορεί να ανατεθεί σε μια αλληλεπίδραση, η οποία με τη σειρά της μπορεί να υλοποιήσει μια υπηρεσία και έχει πρόσβαση σε αντικείμενα δεδομένων. Υπηρεσίες εφαρμογών και υπηρεσίες υποδομής μπορούν να χρησιμοποιηθούν από μια αλληλεπίδραση εφαρμογής. Ανήκει στην κατηγορία στοιχείων συμπεριφοράς του επιπέδου εφαρμογών.

Τυπικές σχέσεις που την αφορούν:

Assignment -- application interaction

Realisation -- application service

Used by -- application service, infrastructure service

Access -- data object

Επίπεδο Εφαρμογών [6/6]



Component



Collaboration



Interface



Service



Function



Interaction



Data Object

Data Object: Ένα αντικείμενο δεδομένων είναι ένα συνεκτικό, αυτόνομο κομμάτι πληροφορίας, κατάλληλο για αυτοματοποιημένη επεξεργασία και με σαφή σημασία τόσο για το επίπεδο εφαρμογών όσο και για ολόκληρη την επιχείρηση. Μια λειτουργία εφαρμογής δραστηριοποιείται σε ένα αντικείμενο δεδομένων. Ένα αντικείμενο δεδομένων μπορεί να μεταφέρεται μέσω αλληλεπιδράσεων και να χρησιμοποιείται ή να παράγεται από υπηρεσίες εφαρμογών. Ανήκει στην κατηγορία δομικών στοιχείων του επιπέδου εφαρμογών.

Τυπικές σχέσεις που την αφορούν:

Access -- application component, application function, application interaction, application service

Realisation -- business object, artifact

Used by -- application service

Επίπεδο Τεχνολογίας [1/8]

-  **Artifact**
-  **Communication Path**
-  **Network**
-  **Infrastructure Interface**
-  **Infrastructure Function**
-  **Infrastructure Service**
-  **Node**
-  **System Software**
-  **Device**

Artifact: Ένα τεχνούργημα ορίζεται ως ένα φυσικό κομμάτι πληροφορίας που χρησιμοποιείται ή παράγεται από μια διαδικασία ανάπτυξης λογισμικού, ή από την ανάπτυξη και λειτουργία ενός συστήματος. Ένα τεχνούργημα αναπαριστά ένα συγκεκριμένο στοιχείο στο φυσικό κόσμο. Συνήθως χρησιμοποιείται για να μοντελοποιήσει προϊόντα λογισμικού όπως αρχεία, εκτελέσιμα, scripts, πίνακες βάσεων δεδομένων, μηνύματα, έγγραφα, προδιαγραφές. Ένα στιγμιότυπο ενός τεχνουργήματος μπορεί να ανατεθεί και ίσως να αναπτυχθεί σε ένα κόμβο. Ένα συστατικό εφαρμογής ή ένα αντικείμενο δεδομένων, μπορεί να πραγματοποιηθεί με ένα ή περισσότερα τεχνουργήματα. Ανήκει στην κατηγορία ενημερωτικών στοιχείων του επιπέδου τεχνολογίας.

Τυπικές σχέσεις που το αφορούν:

Realisation -- application component, data object

Assignment -- artifact

Επίπεδο Τεχνολογίας [2/8]

-  **Artifact**
-  **Communication Path**
-  **Network**
-  **Infrastructure Interface**
-  **Infrastructure Function**
-  **Infrastructure Service**
-  **Node**
-  **System Software**
-  **Device**

Communication Path: Μία διαδρομή επικοινωνίας ορίζεται ως μια σύνδεση μεταξύ δύο ή περισσότερων κόμβων, μέσω της οποίας αυτοί οι κόμβοι μπορούν να ανταλλάσσουν πληροφορίες. Η διαδρομή χρησιμοποιείται για να μοντελοποιήσει τις λογικές σχέσεις επικοινωνίας μεταξύ των κόμβων. Υλοποιείται από ένα ή περισσότερα δίκτυα, τα οποία αντιπροσωπεύουν τις φυσικές συνδέσεις επικοινωνίας. Ανήκει στην κατηγορία δομικών στοιχείων του επιπέδου τεχνολογίας.

Τυπικές σχέσεις που την αφορούν:

Composition -- communication path

Aggregation -- communication path

Specialisation -- communication path

Network: Ένα δίκτυο ορίζεται ως ένα φυσικό μέσο επικοινωνίας μεταξύ δύο ή περισσότερων συσκευών. Αντιπροσωπεύει την υλική υποδομή επικοινωνίας. Μπορεί να περιλαμβάνει μία ή περισσότερες σταθερές ή ασύρματες συνδέσεις δικτύου και υλοποιεί μία ή περισσότερες διαδρομές επικοινωνίας. Ένα δίκτυο μπορεί να αποτελείται από επιμέρους δίκτυα. Ανήκει στην κατηγορία δομικών στοιχείων του επιπέδου τεχνολογίας.

Τυπικές σχέσεις που το αφορούν:

Assignment -- node, device

Επίπεδο Τεχνολογίας [3/8]

-  Artifact
-  Communication Path
-  Network
-  Infrastructure Interface
-  Infrastructure Function
-  Infrastructure Service
-  Node
-  System Software
-  Device

Infrastructure Interface: Μια διεπαφή υποδομής ορίζεται ως ένα σημείο πρόσβασης, όπου η λειτουργικότητα που προσφέρεται από έναν κόμβο μπορεί να προσεγγιστεί από άλλους κόμβους και συστατικά εφαρμογής. Ρόλος της είναι να καθορίσει τον τρόπο με τον οποίο οι υπηρεσίες υποδομής ενός κόμβου μπορούν να προσπελαστούν από άλλους κόμβους (παρεχόμενη διεπαφή), ή το ποια λειτουργία ένας κόμβος απαιτεί από το περιβάλλον του (απαιτούμενη διεπαφή). Μια διεπαφή υποδομής εκθέτει μια υπηρεσία υποδομής στο περιβάλλον και μπορεί να είναι μέρος ενός κόμβου μέσω σχέσης σύνθεσης. Ανήκει στην κατηγορία δομικών στοιχείων του επιπέδου τεχνολογίας.

Τυπικές σχέσεις που την αφορούν:
Assignment -- infrastructure service
Composition -- node

Επίπεδο Τεχνολογίας [4/8]

-  Artifact
-  Communication Path
-  Network
-  Infrastructure Interface
-  Infrastructure Function
-  Infrastructure Service
-  Node
-  System Software
-  Device

Infrastructure Function: Μια λειτουργία υποδομής περιγράφει την εσωτερική συμπεριφορά ενός κόμβου. Για τον χρήστη ενός κόμβου που εκτελεί μια λειτουργία υποδομής, η λειτουργία αυτή είναι αόρατη. Αν η συμπεριφορά του κόμβου είναι εκτεθειμένη εξωτερικά, αυτό γίνεται μέσω ενός ή περισσοτέρων υπηρεσιών υποδομής. Η λειτουργία υποδομών αποτελεί μία αφαίρεση του τρόπου που αυτή εφαρμόζεται. Μόνο η αναγκαία συμπεριφορά καθορίζεται. Μια λειτουργία υποδομής μπορεί να πραγματοποιήσει υπηρεσίες υποδομής. Οι υπηρεσίες υποδομής άλλων λειτουργιών υποδομής μπορούν να χρησιμοποιηθούν από λειτουργίες υποδομής. Η λειτουργία υποδομής έχει πρόσβαση σε τεχνουργήματα. Ένας κόμβος μπορεί να εκχωρηθεί σε μια λειτουργία υποδομής (πράγμα που σημαίνει ότι ο κόμβος εκτελεί αυτή τη λειτουργία υποδομής). Ανήκει στην κατηγορία στοιχείων συμπεριφοράς του επιπέδου τεχνολογίας.

Τυπικές σχέσεις που την αφορούν:

Used by -- node

Realisation -- infrastructure service

Assignment -- node

Επίπεδο Τεχνολογίας [5/8]

-  [Artifact](#)
-  [Communication Path](#)
-  [Network](#)
-  [Infrastructure Interface](#)
-  [Infrastructure Function](#)
-  [Infrastructure Service](#)
-  [Node](#)
-  [System Software](#)
-  [Device](#)

Infrastructure Service: Μια υπηρεσία υποδομής ορίζεται ως μια εξωτερικά ορατή μονάδα λειτουργικότητας, που παρέχεται από έναν ή περισσότερους κόμβους και εκτίθεται μέσω καλά ορισμένων και με νόημα για το περιβάλλον διεπαφών. Μια υπηρεσία υποδομής φανερώνει τη λειτουργικότητα ενός κόμβου στο περιβάλλον του. Αυτή η λειτουργικότητα είναι προσβάσιμη μέσω μίας ή περισσότερων διεπαφών υποδομής. Μια υπηρεσία υποδομής μπορεί να περιλαμβάνει ανταλλαγή μηνυμάτων, αποθήκευση, ονομασία και υπηρεσίες ευρετηρίου (directory). Έχει πρόσβαση σε τεχνουργήματα, όπως για παράδειγμα ένα αρχείο που περιλαμβάνει ένα μήνυμα. Μια υπηρεσία υποδομής μπορεί να χρησιμοποιηθεί από συστατικά εφαρμογής ή από κόμβους. Μια υπηρεσία υποδομής εκτίθεται από έναν κόμβο, όταν ανατίθεται στις διεπαφές υποδομής του. Μια υπηρεσία υποδομής μπορεί να αποτελείται από επιμέρους υπηρεσίες. Ανήκει στην κατηγορία στοιχείων συμπεριφοράς του επιπέδου τεχνολογίας.

Τυπικές σχέσεις που την αφορούν:

Used by -- application component, node

Assignment -- infrastructure interface

Access -- artifact

Realisation -- node

Επίπεδο Τεχνολογίας [6/8]

-  Artifact
-  Communication Path
-  Network
-  Infrastructure Interface
-  Infrastructure Function
-  Infrastructure Service
-  Node
-  System Software
-  Device

Node: Ένας κόμβος ορίζεται ως ένας υπολογιστικός πόρος πάνω στον οποίο μπορούν να αναπτυχθούν τεχνουργήματα για εκτέλεση. Οι κόμβοι εκτελούν και κατεργάζονται τεχνουργήματα, τα οποία είναι οι αναπαραστάσεις συστατικών και αντικειμένων δεδομένων. Χρησιμοποιούνται για να μοντελοποιήσουν εξυπηρετητές εφαρμογών (application servers), εξυπηρετητές βάσεων δεδομένων (database servers) ή θυγατρικούς σταθμούς εργασίας. Ένας κόμβος μπορεί να αποτελείται από επιμέρους κόμβους που αναπαριστούν φυσικές συσκευές και περιβάλλοντα εκτέλεσης για τεχνουργήματα. Οι κόμβοι μπορεί να συνδέονται μεταξύ τους με διαδρομές επικοινωνίας και σε αυτούς ανατίθενται τεχνουργήματα. Ανήκει στην κατηγορία δομικών στοιχείων του επιπέδου τεχνολογίας.

Τυπικές σχέσεις που τον αφορούν:

Assignment -- artifact, network

Επίπεδο Τεχνολογίας [7/8]

-  **Artifact**
-  **Communication Path**
-  **Network**
-  **Infrastructure Interface**
-  **Infrastructure Function**
-  **Infrastructure Service**
-  **Node**
-  **System Software**
-  **Device**

System Software: Ένα λογισμικό συστήματος αντιπροσωπεύει ένα περιβάλλον λογισμικού για συγκεκριμένους τύπους συστατικών και αντικειμένων που έχουν αναπτυχθεί σε αυτό με τη μορφή τεχνουργημάτων. Το λογισμικό συστήματος είναι μια εξειδίκευση ενός κόμβου που χρησιμοποιείται για να μοντελοποιήσει το περιβάλλον λογισμικού στο οποίο τρέχουν τεχνουργήματα. Μπορεί ακόμα να χρησιμοποιηθεί για να αναπαραστήσει ενδιάμεσο λογισμικό επικοινωνίας. Συνήθως το λογισμικό συστήματος συνδυάζεται με μια συσκευή που αντιπροσωπεύει το υλικό περιβάλλον (hardware) που απαιτείται για το σχηματισμό ενός γενικού κόμβου. Το λογισμικό συστήματος μπορεί να ανατεθεί σε μια συσκευή και τεχνουργήματα μπορεί να ανατεθούν σε αυτό. Ένας κόμβος μπορεί να περιέχει ένα λογισμικό συστήματος και ένα λογισμικό συστήματος μπορεί να περιέχει ένα άλλο λογισμικό συστήματος, όπως για παράδειγμα στην περίπτωση που ένα λειτουργικό σύστημα περιέχει μία βάση δεδομένων. Ανήκει στην κατηγορία στοιχείων συμπεριφοράς του επιπέδου τεχνολογίας.

Τυπικές σχέσεις που το αφορούν:
Assignment -- device, artifact

Επίπεδο Τεχνολογίας [8/8]

-  Artifact
-  Communication Path
-  Network
-  Infrastructure Interface
-  Infrastructure Function
-  Infrastructure Service
-  Node
-  System Software
-  Device

Device: Μία συσκευή ορίζεται ως ένας φυσικός υπολογιστικός πόρος πάνω στον οποίο μπορούν να αναπτυχθούν τεχνουργήματα για εκτέλεση. Μία συσκευή είναι η εξειδίκευση ενός κόμβου και αντιπροσωπεύει ένα φυσικό πόρο με επεξεργαστική δύναμη. Συνήθως χρησιμοποιείται για να μοντελοποιήσει συστήματα υλικού όπως μεγάλους υπολογιστές (mainframes), PCs ή δρομολογητές (routers). Συνήθως είναι κομμάτι ενός κόμβου μαζί με ένα λογισμικό συστήματος. Οι συσκευές μπορεί να είναι μία σύνθεση πολλών επιμέρους συσκευών. Συσκευές μπορεί να διασυνδέονται μέσω δικτύων και τεχνουργήματα μπορεί να ανατεθούν σε αυτές. Ανήκει στην κατηγορία δομικών στοιχείων του επιπέδου τεχνολογίας.

Τυπικές σχέσεις που την αφορούν:

Composition -- device, node

Assignment -- system software, artifact

Επίπεδο Κινήτρου[1/3]

 Stakeholder

Stakeholder: Ένας συμμετέχοντας ορίζεται ως ο ρόλος ενός ατόμου, μιας ομάδας ή ενός οργανισμού που αντιπροσωπεύει τα συμφέροντα ή τις ανησυχίες αυτών σχετικά με το αποτέλεσμα της αρχιτεκτονικής.

 Driver

Τυπικές σχέσεις που τους αφορούν:

Composition, Aggregation, Specialisation

 Goal

Driver: Ένας οδηγός ορίζεται ως κάτι που δημιουργεί, παρακινεί και τροφοδοτεί την αλλαγή σε έναν οργανισμό. Οι οδηγοί μπορεί να είναι εσωτερικοί, οπότε και συσχετίζονται με τους συμμετέχοντες, ή εξωτερικοί.

 Principle

Τυπικές σχέσεις που τον αφορούν:

Composition, Aggregation, Specialisation, Influence

 Requirement

 Constraint

Assessment: Η αξιολόγηση ορίζεται ως το αποτέλεσμα της ανάλυσης κάποιου οδηγού. Η αξιολόγηση μπορεί να αποκαλύψει πλεονεκτήματα, αδυναμίες, ευκαιρίες και κινδύνους σχετικά με μια περιοχή ενδιαφέροντος. Τα αποτελέσματα αυτά πρέπει να αντιμετωπιστούν με την προσαρμογή των υπαρχόντων στόχων ή θέτοντας νέους, γεγονός που μπορεί να προκαλέσει αλλαγές στην αρχιτεκτονική της επιχείρησης.

Τυπικές σχέσεις που την αφορούν:

Composition, Aggregation, Specialisation, Influence

Επίπεδο Κινήτρου[2/3]

 Stakeholder

Goal: Ένας στόχος ορίζεται ως μια τελική κατάσταση που ένας συμμετέχοντας σκοπεύει να επιτύχει. Αυτή μπορεί να είναι οτιδήποτε, από μία κατάσταση ως κάποια παραγόμενη αξία.

 Driver

 Assessment

Τυπικές σχέσεις που τον αφορούν:

Composition, Aggregation, Specialisation, Influence

 Goal

Principle: Μια αρχή ορίζεται ως μια κανονιστική ιδιότητα όλων των συστημάτων σε ένα δεδομένο πλαίσιο ή ο τρόπος με τον οποίο αυτά έχουν υλοποιηθεί. Οι αρχές ορίζουν τις αποσκοπούμενες ιδιότητες συστημάτων και παρακινούνται από κάποιο στόχο.

 Principle

Τυπικές σχέσεις που τις αφορούν:

Composition, Aggregation, Specialisation, Influence

 Requirement

 Constraint

Επίπεδο Κινήτρου[3/3]

 Stakeholder

 Driver

 Assessment

 Goal

 Principle

 Requirement

 Constraint

Requirement: Μία απαίτηση ορίζεται ως μια δήλωση ανάγκης που πρέπει να πραγματοποιηθεί από ένα σύστημα. Οι απαιτήσεις μοντελοποιούν τις ιδιότητες των στοιχείων που είναι απαραίτητα για την επίτευξη των τελικών στόχων ενός οργανισμού, οι οποίοι γνωστοποιούνται από τους στόχους. Από αυτή την άποψη, οι απαιτήσεις αντιπροσωπεύουν τα «μέσα» με τα οποία υλοποιούνται οι στόχοι.

Τυπικές σχέσεις που τις αφορούν:

Composition, Aggregation, Specialisation, Influence

Constraint: Ένας περιορισμός ορίζεται ως ένα εμπόδιο στον τρόπο με τον οποίο πραγματοποιείται ένα σύστημα. Μπορεί να πρόκειται για εμπόδιο στην υλοποίηση του συστήματος (όπως η ειδική τεχνολογία που πρόκειται να χρησιμοποιηθεί) ή στην υλοποίηση κάποιας διαδικασίας (όπως χρονικοί ή χρηματικοί περιορισμοί).

Τυπικές σχέσεις που τους αφορούν:

Composition, Aggregation, Specialisation, Influence

Επίπεδο Εφαρμογής και Μετανάστευσης [1/2]



Work Package: Ένα πακέτο εργασίας ορίζεται ως μια σειρά ενεργειών που αποσκοπούν στην επίτευξη ενός μοναδικού στόχου εντός καθορισμένης προθεσμίας. Ένα πακέτο εργασίας έχει μια σαφώς καθορισμένη αρχή και τέλος στην ημερομηνία, καθώς και ένα καλά καθορισμένο σύνολο στόχων και αποτελεσμάτων. Μπορεί να χρησιμοποιηθεί για να μοντελοποιήσει έργα.



Τυπικές σχέσεις που το αφορούν:

Composition, Aggregation, Specialisation, Flow, Triggering

Deliverable: Ένα παραδοτέο ορίζεται ως ένα επακριβώς ορισμένο αποτέλεσμα ενός πακέτου εργασίας. Αυτό μπορεί να περιλαμβάνει αποτελέσματα κάθε είδους, από αναφορές, έγγραφα, υπηρεσίες, λογισμικό, φυσικά προϊόντα, έως άυλα αποτελέσματα όπως μια οργανωτική αλλαγή.

Τυπικές σχέσεις που το αφορούν:

Composition, Aggregation, Specialisation, Flow, Triggering

Επίπεδο Εφαρμογής και Μετανάστευσης [2/2]



Plateau: Ένα πλατό ορίζεται ως μια σχετικά σταθερή κατάσταση της αρχιτεκτονικής που υπάρχει κατά τη διάρκεια ενός περιορισμένου χρονικού διαστήματος.



Τυπικές σχέσεις που το αφορούν:
Composition, Aggregation, Specialisation, Triggering



Gap: Ένα κενό ορίζεται ως το αποτέλεσμα μιας ανάλυσης χάσματος (gap analysis) ανάμεσα σε δύο πλατό. Η έννοια κενό συνδέεται με δύο πλατό (π.χ. αρχική και τελική αρχιτεκτονική, δύο σειριακές αρχιτεκτονικές μετάβασης) και αναπαριστά τις διαφορές μεταξύ αυτών.

Τυπικές σχέσεις που το αφορούν:
Composition, Aggregation, Specialisation