



Θεωρίες Μάθησης και Εκπαιδευτικό Λογισμικό

Ενότητα 9: Αρχιτεκτονική Λογισμικού, Όψεις
Αρχιτεκτονικής Λογισμικού, Διαγράμματα UML

Σταύρος Δημητριάδης
Τμήμα Πληροφορικής



Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο



ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ ΚΑΙ ΘΡΗΣΚΕΥΜΑΤΩΝ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ

Άδειες Χρήσης

- Το παρόν εκπαιδευτικό υλικό υπόκειται σε άδειες χρήσης Creative Commons.
- Για εκπαιδευτικό υλικό, όπως εικόνες, που υπόκειται σε άλλου τύπου άδειας χρήσης, η άδεια χρήσης αναφέρεται ρητώς.



Χρηματοδότηση

- Το παρόν εκπαιδευτικό υλικό έχει αναπτυχθεί στα πλαίσια του εκπαιδευτικού έργου του διδάσκοντα.
- Το έργο «Ανοικτά Ακαδημαϊκά Μαθήματα στο Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης» έχει χρηματοδοτήσει μόνο τη αναδιαμόρφωση του εκπαιδευτικού υλικού.
- Το έργο υλοποιείται στο πλαίσιο του Επιχειρησιακού Προγράμματος «Εκπαίδευση και Δια Βίου Μάθηση» και συγχρηματοδοτείται από την Ευρωπαϊκή Ένωση (Ευρωπαϊκό Κοινωνικό Ταμείο) και από εθνικούς πόρους.





ΑΡΙΣΤΟΤΕΛΕΙΟ
ΠΑΝΕΠΙΣΤΗΜΙΟ
ΘΕΣΣΑΛΟΝΙΚΗΣ

ΑΝΟΙΚΤΑ
ΑΚΑΔΗΜΑΪΚΑ
ΜΑΘΗΜΑΤΑ



Αρχιτεκτονική Λογισμικού, Ώψεις Αρχιτεκτονικής Λογισμικού, Διαγράμματα UML



Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο



ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ
ΕΚΠΑΙΔΕΥΣΗ ΚΑΙ ΔΙΑ ΒΙΟΥ ΜΑΘΗΣΗ
επένδυση στην κοινωνία της γνώσης

ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ ΚΑΙ ΘΡΗΣΚΕΥΜΑΤΩΝ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ
πρόγραμμα για την ανάπτυξη

Περιεχόμενα ενότητας

1. Αρχιτεκτονική Λογισμικού (Software Architecture)
2. Όψεις Αρχιτεκτονικής Λογισμικού (Software Architecture Views)
3. Διαγράμματα UML (UML Diagrams) - Διαγράμματα Περίπτωσης Χρήσης (Use Case Diagrams)





ΑΡΙΣΤΟΤΕΛΕΙΟ
ΠΑΝΕΠΙΣΤΗΜΙΟ
ΘΕΣΣΑΛΟΝΙΚΗΣ

Αρχιτεκτονική Λογισμικού

Αρχιτεκτονική λογισμικού

Software architecture

- Η αρχιτεκτονική λογισμικού είναι η **οργάνωση** και **περιγραφή** της δομής ενός λογισμικού σε **υψηλότερο επίπεδο αφαίρεσης**, τονίζοντας:
- **Δομή**: τα επιμέρους **στοιχεία λογισμικού** από τα οποία αποτελείται το σύστημα
- **Επικοινωνία**: τις μεταξύ τους **αλληλεπιδράσεις** (πχ. ροή δεδομένων, αποφάσεις, κλπ.)
- **Χαρακτηριστικά**: καθώς και **πρότυπα** και **περιορισμούς** που αφορούν τα στοιχεία αυτά.



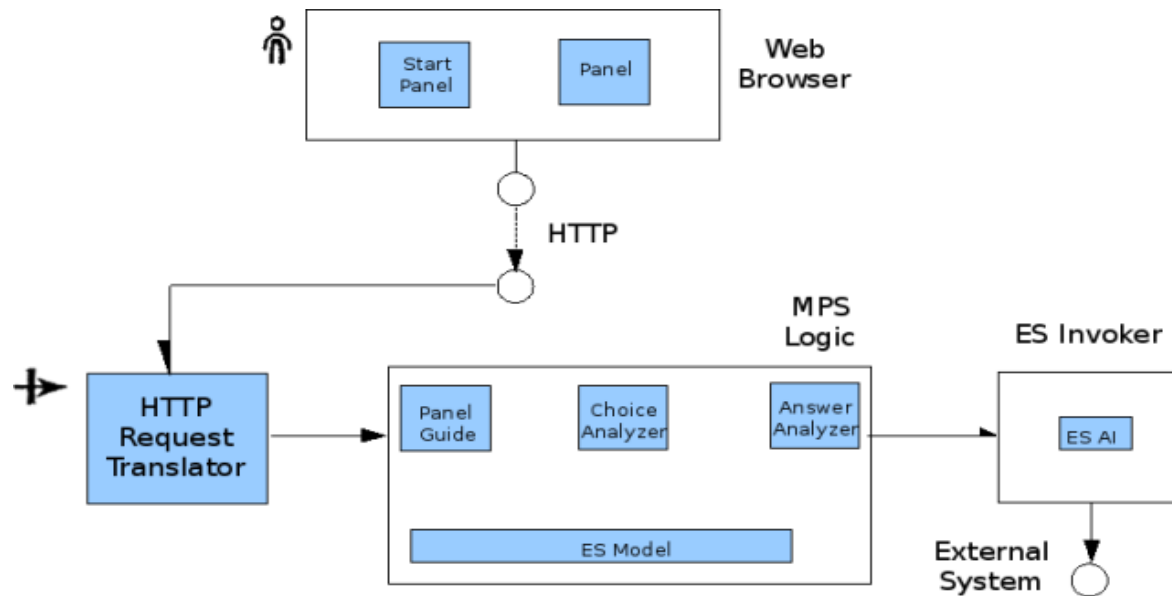
Γιατί σχεδιάζουμε την Αρχιτεκτονική ενός λογισμικού;

- Αναπαριστούμε τα **σημαντικά στοιχεία** του λογισμικού σε υψηλό επίπεδο αφαίρεσης **χωρίς** να ανησυχούμε για λεπτομέρειες
- Τονίζουμε την **επικοινωνία** των στοιχείων μεταξύ τους (είσοδος, έξοδος, κλπ.)
- Κατανοούμε τη **γενική λειτουργία** του λογισμικού και τις **υπηρεσίες** που προσφέρει
- Τονίζουμε τις βασικές **αποφάσεις** που παίρνουν οι δημιουργοί του λογισμικού κατά τη σχεδιάσή του



Τι σχεδιάζουμε όταν σχεδιάζουμε την αρχιτεκτονική ενός λογισμικού;

- Συνηθισμένη πρακτική είναι να σχεδιάζουμε **άτυπα (informally)**:
- **Blocks**: που δηλώνουν κάποιες «οντότητες» του λογισμικού όπως **Τμήματα** (components) λογισμικού, **Κλάσεις** (classes), κλπ....
- **Συνδέσεις (connectors)**: που δηλώνουν **επικοινωνία** (μεταφορά δεδομένων) και γενικά σχέσεις **αλληλεπίδρασης** μεταξύ των στοιχείων του λογισμικού

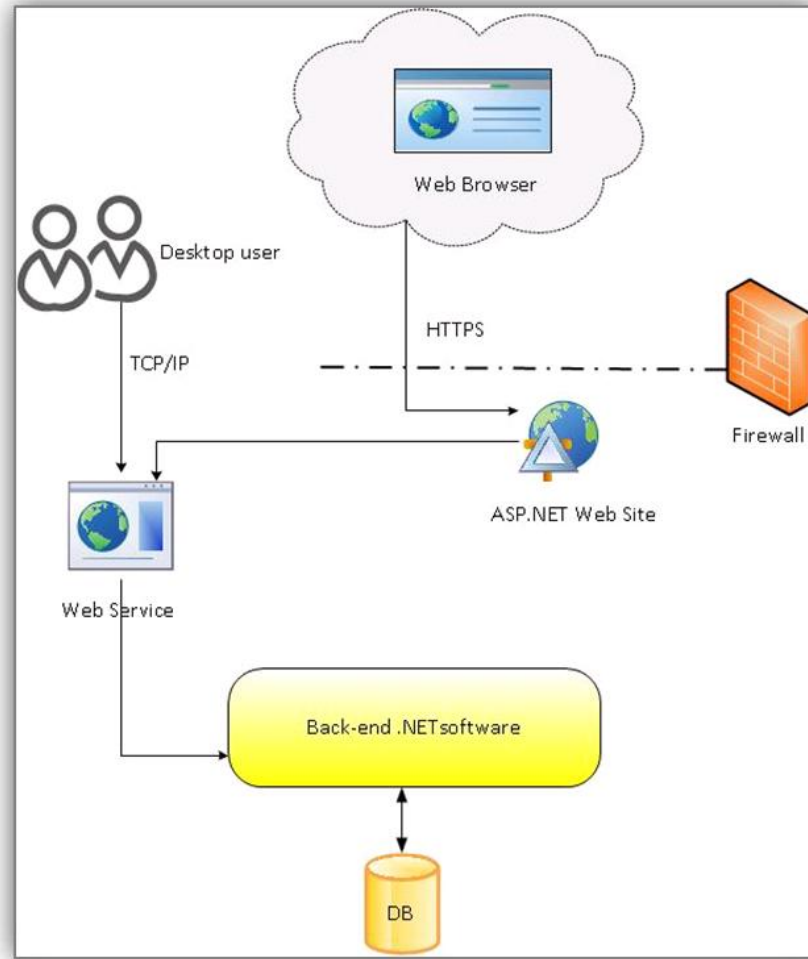


Εικόνα 1



Τι σχεδιάζουμε όταν σχεδιάζουμε την αρχιτεκτονική ενός λογισμικού;

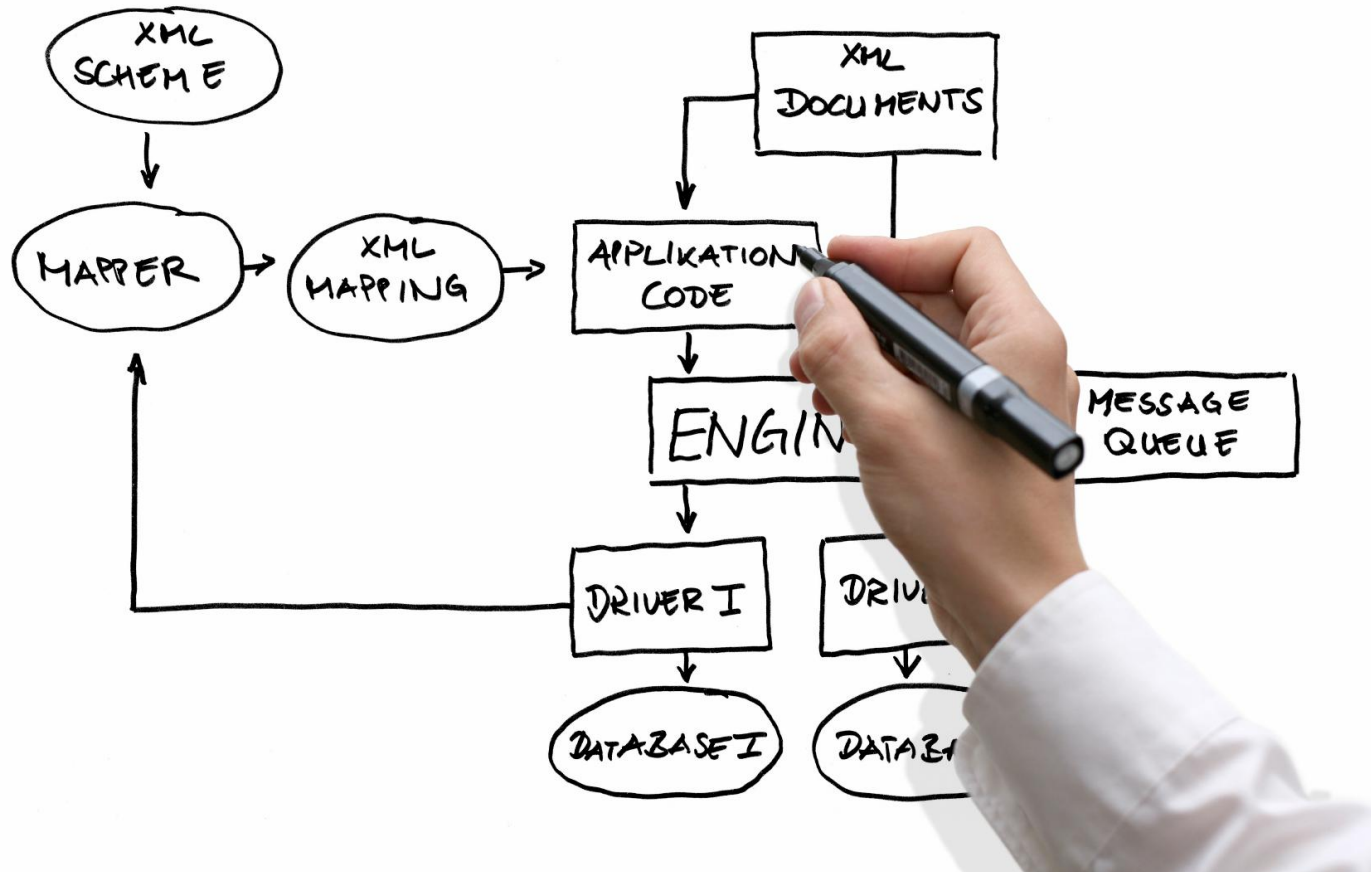
- Μπορεί να χρησιμοποιούμε
- Επίπεδα
- Χρώμα
- Εικονίδια
- Βέλη, Γραμμές
-



Εικόνα 2



‘Η απλά με το χέρι...

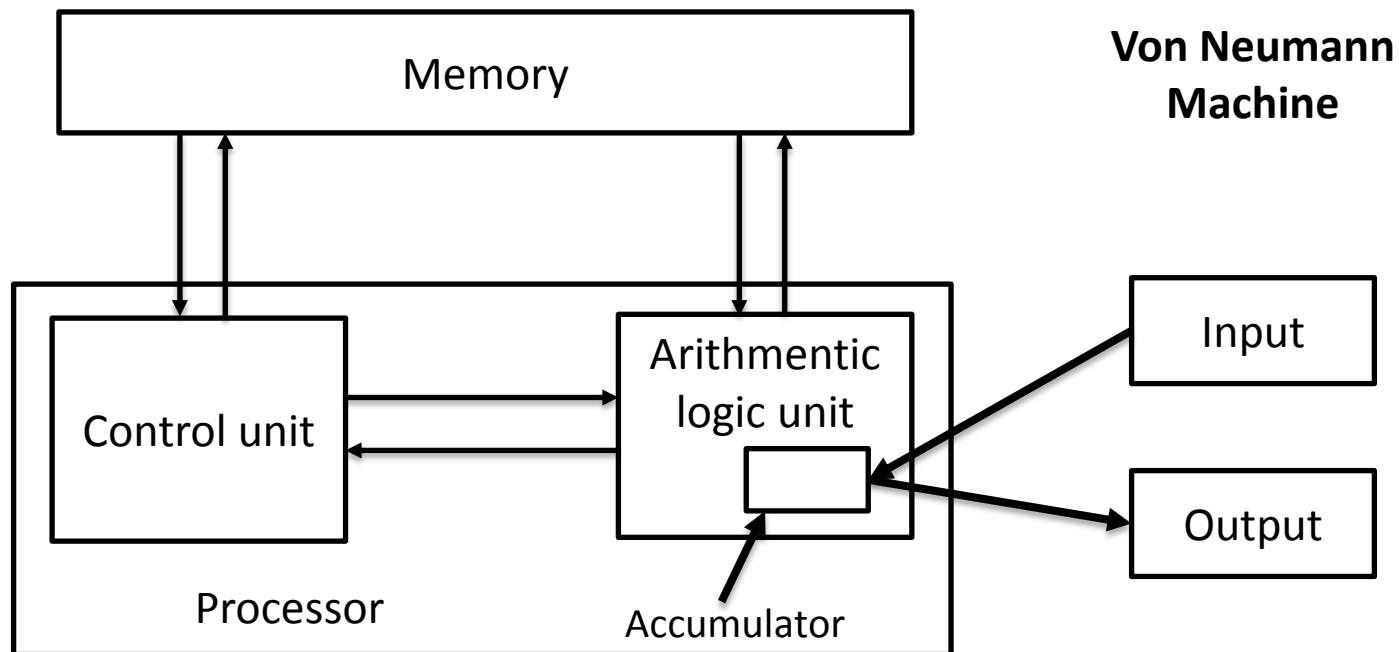


Εικόνα 3

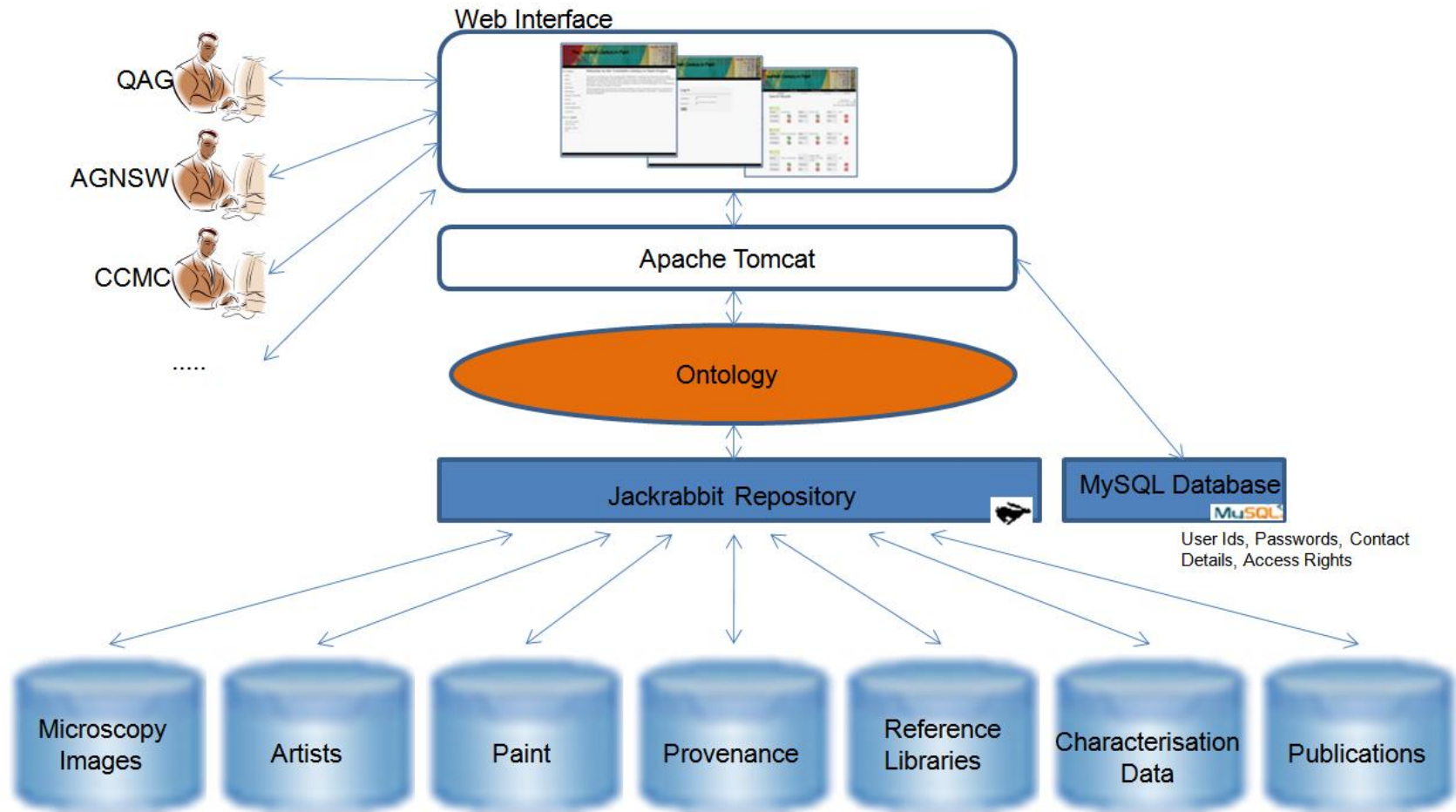


Μπορεί να σχεδιάζουμε απλά αναπαριστώντας λίγα στοιχεία

- Εφόσον απεικονίζουμε στοιχεία υλικού ή συνδυάζουμε υλικό και λογισμικό που συναποτελούν ένα τεχνολογικό σύστημα, μιλάμε γενικότερα για **αρχιτεκτονική συστήματος** (system architecture)



Η' μπορεί να σχεδιάζουμε σύνθετα περιλαμβάνοντας πλήθος στοιχείων



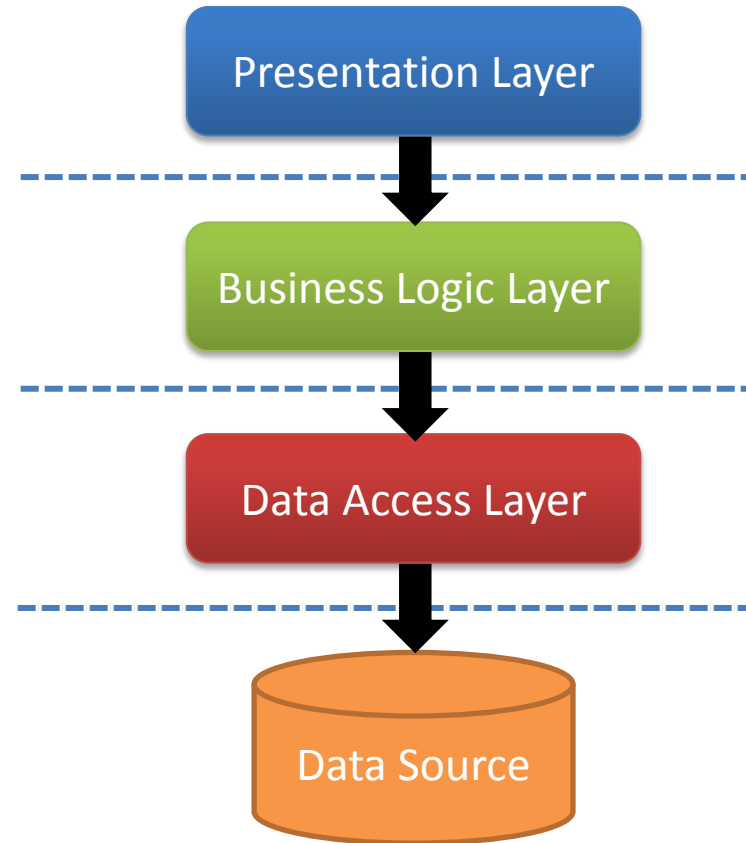
Εικόνα 4

Άτυπη σχεδίαση

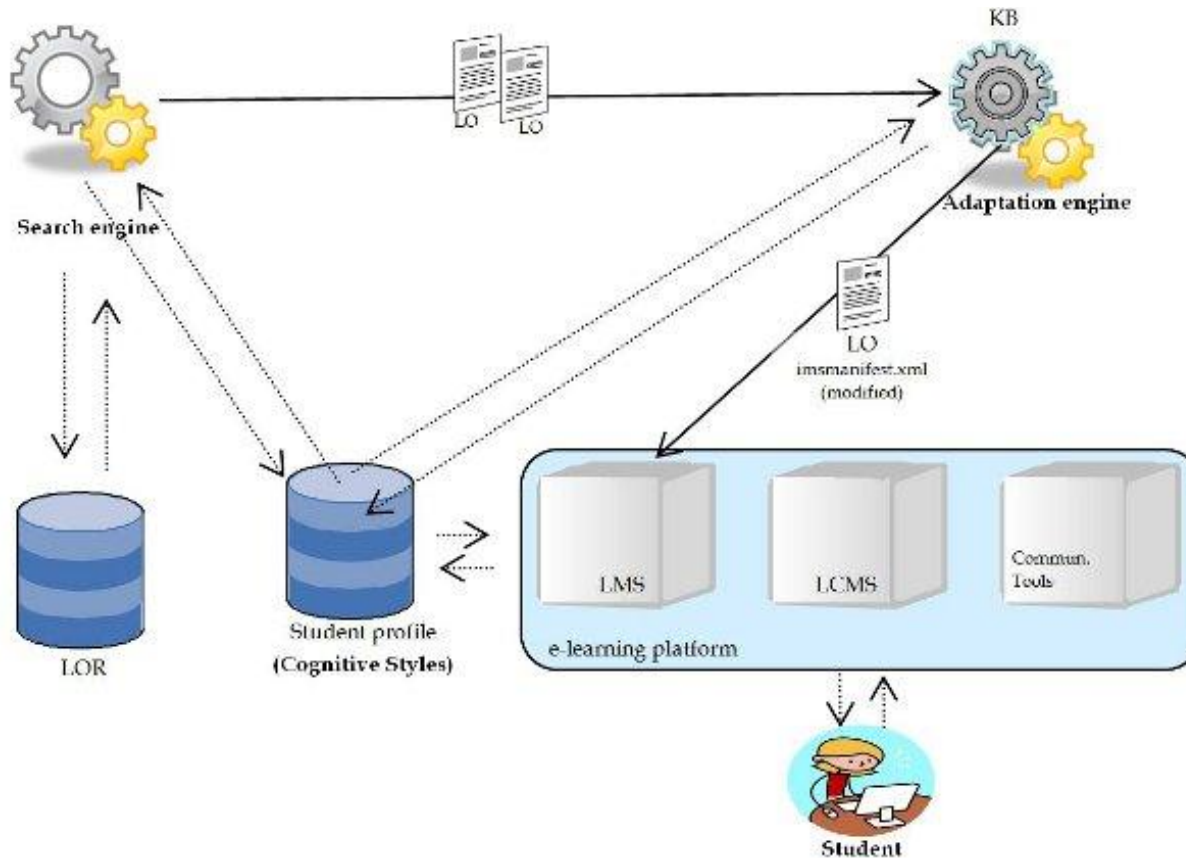
- Η αρχιτεκτονική ενός συστήματος μπορεί να σχεδιαστεί άτυπα, δηλ. χωρίς τη χρήση κάποιου τυπικού (formal) τρόπου («γλώσσας») αναπαράστασης
- **Κόμβοι (blocks):** συνήθως αναφέρονται σε **τμήματα (components)** του λογισμικού ή **επίπεδα (layers)** της αρχιτεκτονικής
- **Σύνδεσμοι (βέλη):** δηλώνουν **ροή δεδομένων** μεταξύ των κόμβων/επιπέδων που συνδέουν
- Μπορεί ακόμη να χρησιμοποιούνται διάφορα εικονίδια για να δηλωθούν χρήστες, βάση δεδομένων, κλπ.



Άτυπη σχεδίαση

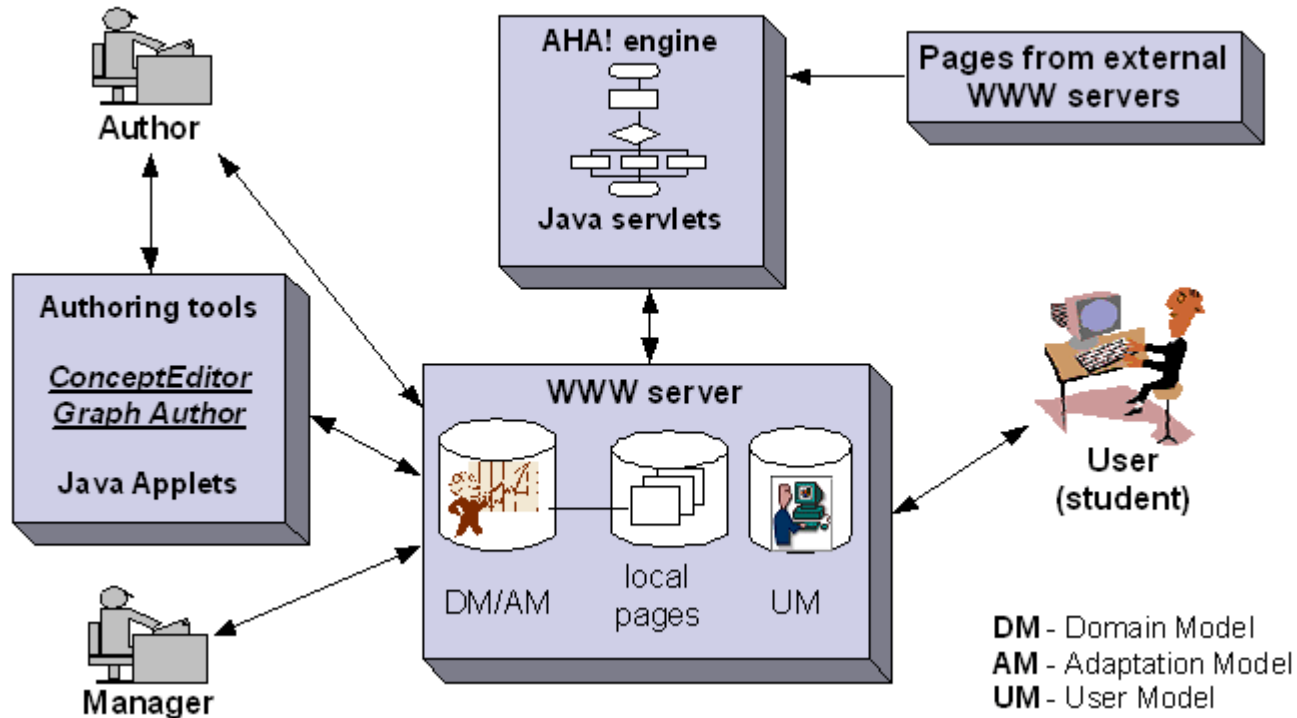


Άτυπη σχεδίαση



Εικόνα 5

Άτυπη σχεδίαση



Εικόνα 6



ΑΡΙΣΤΟΤΕΛΕΙΟ
ΠΑΝΕΠΙΣΤΗΜΙΟ
ΘΕΣΣΑΛΟΝΙΚΗΣ

Όψεις Αρχιτεκτονικής Λογισμικού

Τυπική αναπαράσταση αρχιτεκτονικής - Περιλαμβάνει:

- (α) **Γλώσσα τυπικής μοντελοποίησης**, δηλ. αναπαράστασης της αρχιτεκτονικής
 - ...ώστε να χρησιμοποιούμε και να κατανοούμε μια κοινή γλώσσα
 - **UML, Unified Modeling Language**
- (β) Ένα **μοντέλο σχετικά με τις διάφορες όψεις** της αρχιτεκτονικής ενός συστήματος
 - ...ώστε να έχουμε μια κοινή κατανόηση για τις διαφορετικές **όψεις (views)** αρχιτεκτονικής ενός συστήματος που μπορούμε να σχεδιάσουμε
 - Πχ. λογική αρχιτεκτονική, φυσική αρχιτεκτονική, κλπ.
 - **Μοντέλο 4+1**



Τι είναι η UML (1/2)

- **UML: Unified Modeling Language**
- Ενοποιημένη Γλώσσα Σχεδιασμού / Μοντελοποίησης
- Είναι μια **γραφική γλώσσα** για την **οπτική** παράσταση, τη διαμόρφωση **προδιαγραφών** και την **τεκμηρίωση** συστημάτων που βασίζονται σε λογισμικό.
- Η UML στοχεύει στο σχεδιασμό αντικειμενοστρεφών συστημάτων. Το σχέδιο είναι μια απλοποιημένη παράσταση της πραγματικότητας. Σχεδιάζουμε για να μπορέσουμε να **καταλάβουμε** το σύστημα που αναπτύσσουμε.



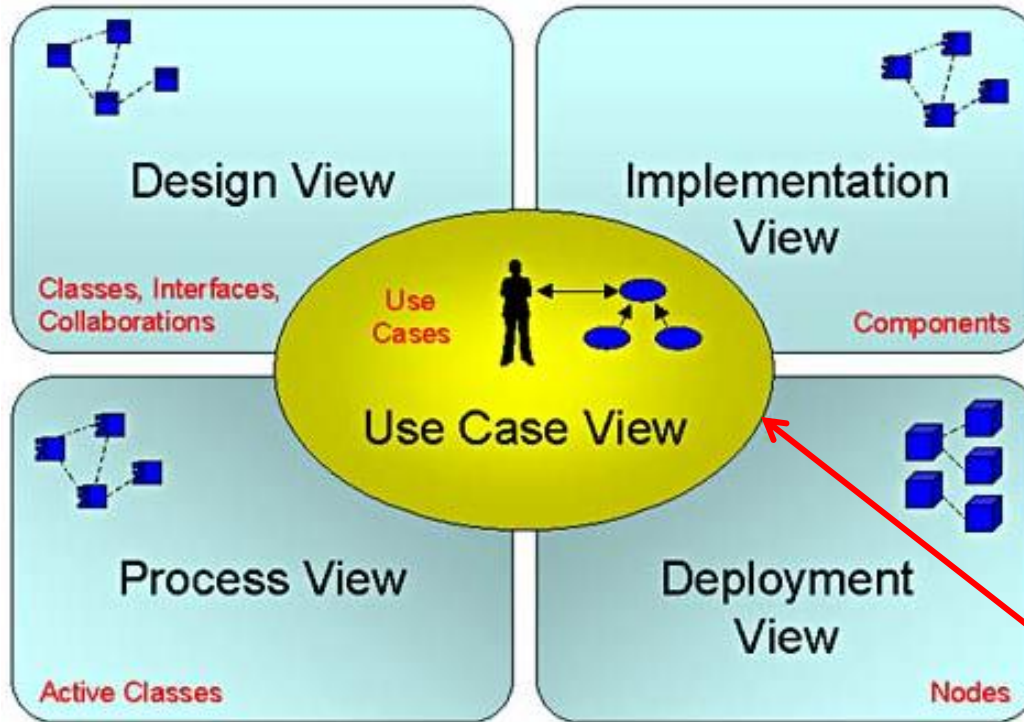
Τι είναι η UML (2/2)

- Δημιουργώντας ένα σχέδιο επιτυγχάνουμε **τέσσερις** στόχους:
 1. παριστάνουμε **οπτικά** το σύστημα που έχουμε ή θέλουμε να κατασκευάσουμε,
 2. προσδιορίζουμε τη **δομή** και τη **συμπεριφορά** του συστήματος,
 3. δημιουργούμε ένα **πρότυπο** για να βασίσουμε την κατασκευή του συστήματος,
 4. **τεκμηριώνουμε** τις αποφάσεις που λάβαμε.
- Η UML περιλαμβάνει τρία βασικά στοιχεία:
- **Οντότητες** (, δηλ. στοιχεία που περιλαμβάνει το σύστημά μας)
- **Σχέσεις** (μεταξύ των οντοτήτων του συστήματος)
- **Διαγράμματα** (που αναπαριστούν οντότητες & τις σχέσεις τους)



Όψεις αρχιτεκτονικής: το μοντέλο 4+1

- Υπάρχουν διάφοροι τρόποι (όψεις) να αναπαρασταθεί η αρχιτεκτονική ενός συστήματος λογισμικού
- Κάθε όψη (view) παρουσιάζει και τονίζει στοιχεία της αρχιτεκτονικής από διαφορετική οπτική εξυπηρετώντας διαφορετικές ανάγκες αναπαράστασης



Εικόνα 7

- Σχεδίασης ή Λογική
 - Design or Logical
- Υλοποίησης
 - Implementation
- Διεργασιών
 - Process
- Ανάπτυξης ή Φυσική
 - Deployment or Physical
- +1 → **Use Cases**
- (Περιπτώσεις Χρήσης)



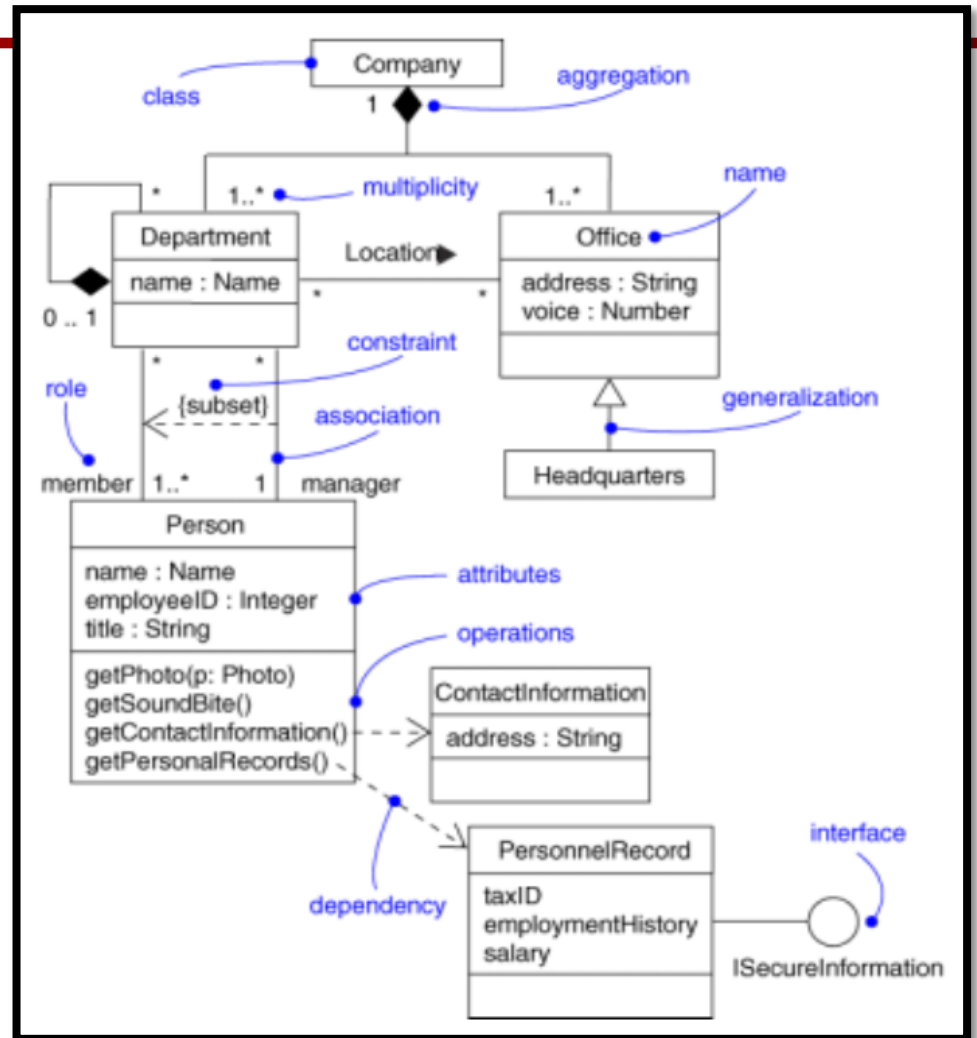
Σχεδίασης ή Λογική όψη

- Σχεδίασης ή Λογική όψη (Design or Logical view)
- Αποτυπώνει τις λειτουργίες που το σύστημα προσφέρει στον τελικό χρήστη
- Διαγράμματα UML που χρησιμοποιούνται για τη λογική όψη: [Class diagram](#), [Communication diagram](#), [Sequence diagram](#)



Παράδειγμα

- Διάγραμμα UML κλάσεων για την αναπαράσταση μιας λογικής όψης της αρχιτεκτονικής ενός συστήματος
- Απεικονίζει τις κλάσεις του λογισμικού



Εικόνα 8

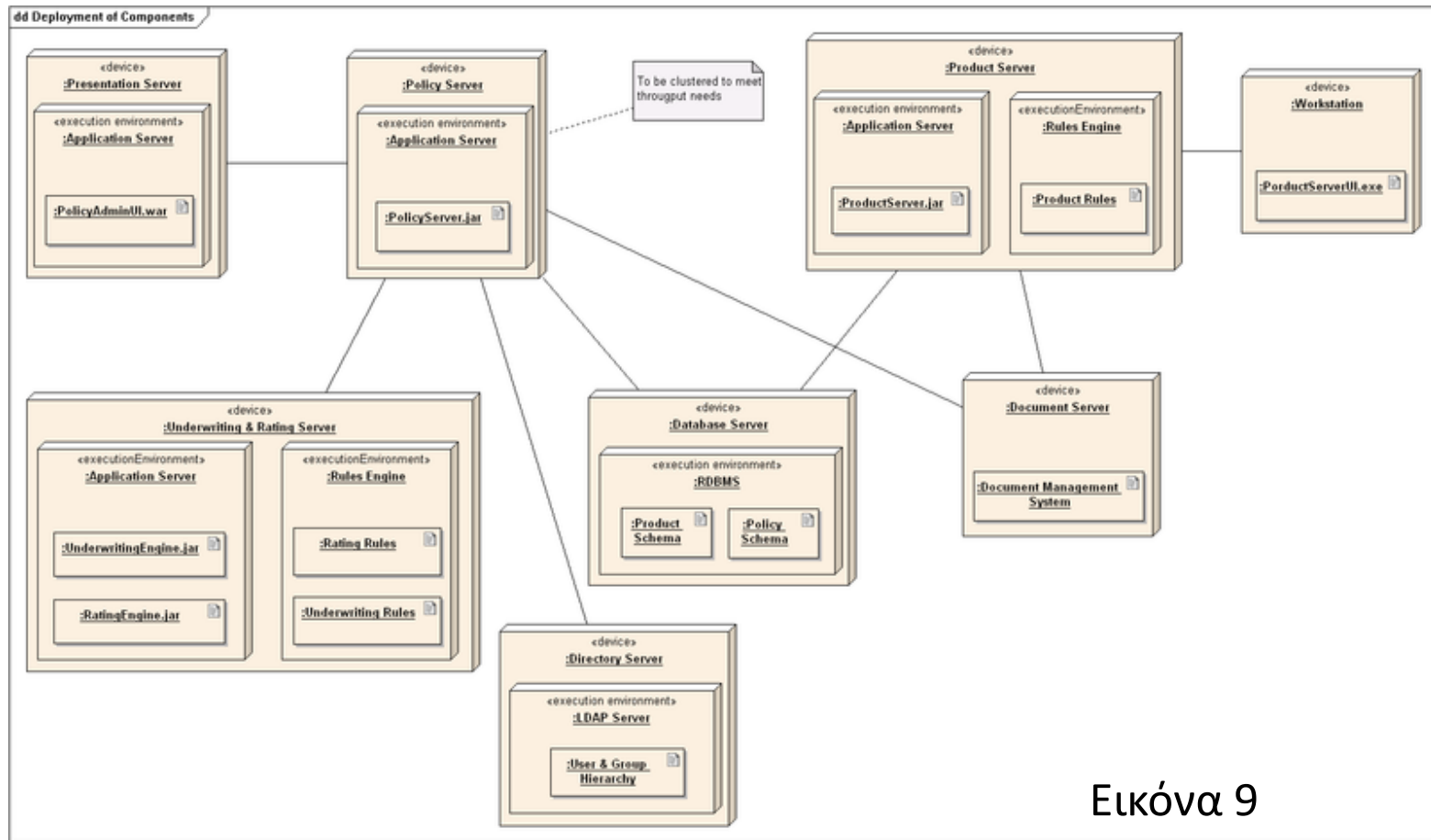
Ανάπτυξης ή Φυσική όψη

- **Ανάπτυξης ή Φυσική όψη** (Deployment or Physical view)
- Αποτυπώνει το σύστημα από την πλευρά του μηχανικού (system engineer).
- Αναπαριστά την τοπολογία των τμημάτων λογισμικού σε φυσικό επίπεδο και τις μεταξύ τους συνδέσεις
- UML διαγράμματα: [Deployment diagram](#)



Παράδειγμα

- Διάγραμμα UML ανάπτυξης (deployment)
- Για την αναπαράσταση της φυσικής όψης της αρχιτεκτονικής ενός συστήματος



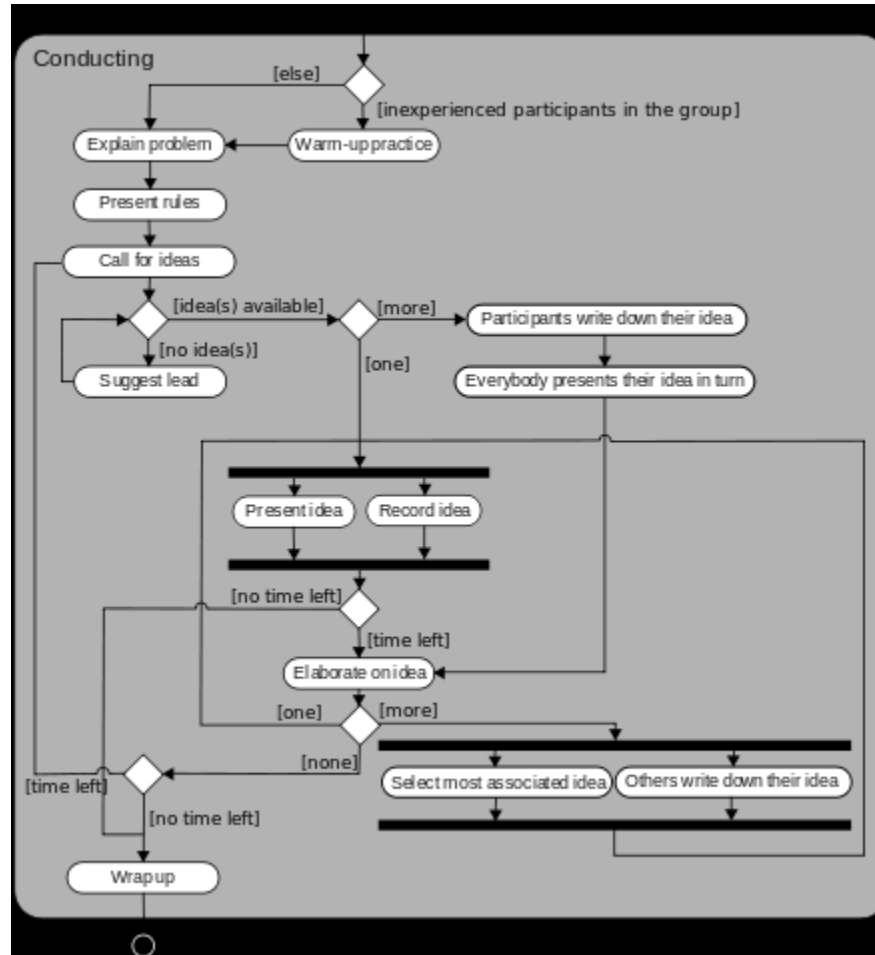
Εικόνα 9

Όψη διεργασιών

- **Όψη διεργασιών (Process view)**
- Αφορά τη δυναμική συμπεριφορά του συστήματος (δηλ. συμπεριφορά κατά την ώρα λειτουργίας του) (runtime behavior)
- Αναπαριστά τις διεργασίες (processes) του συστήματος και την μεταξύ τους επικοινωνία
- UML διαγράμματα: [Activity diagram](#)



Παράδειγμα



Εικόνα 10

Όψη Υλοποίησης

- Παρουσιάζει το σύστημα από την **οπτική του προγραμματιστή** και εστιάζει στη διαχείριση (management) του λογισμικού
- Συνηθισμένος τρόπος παρουσίασης με χρήση UML διαγράμματος Τμημάτων ([Component diagram](#)) για την παρουσίαση των τμημάτων του συστήματος

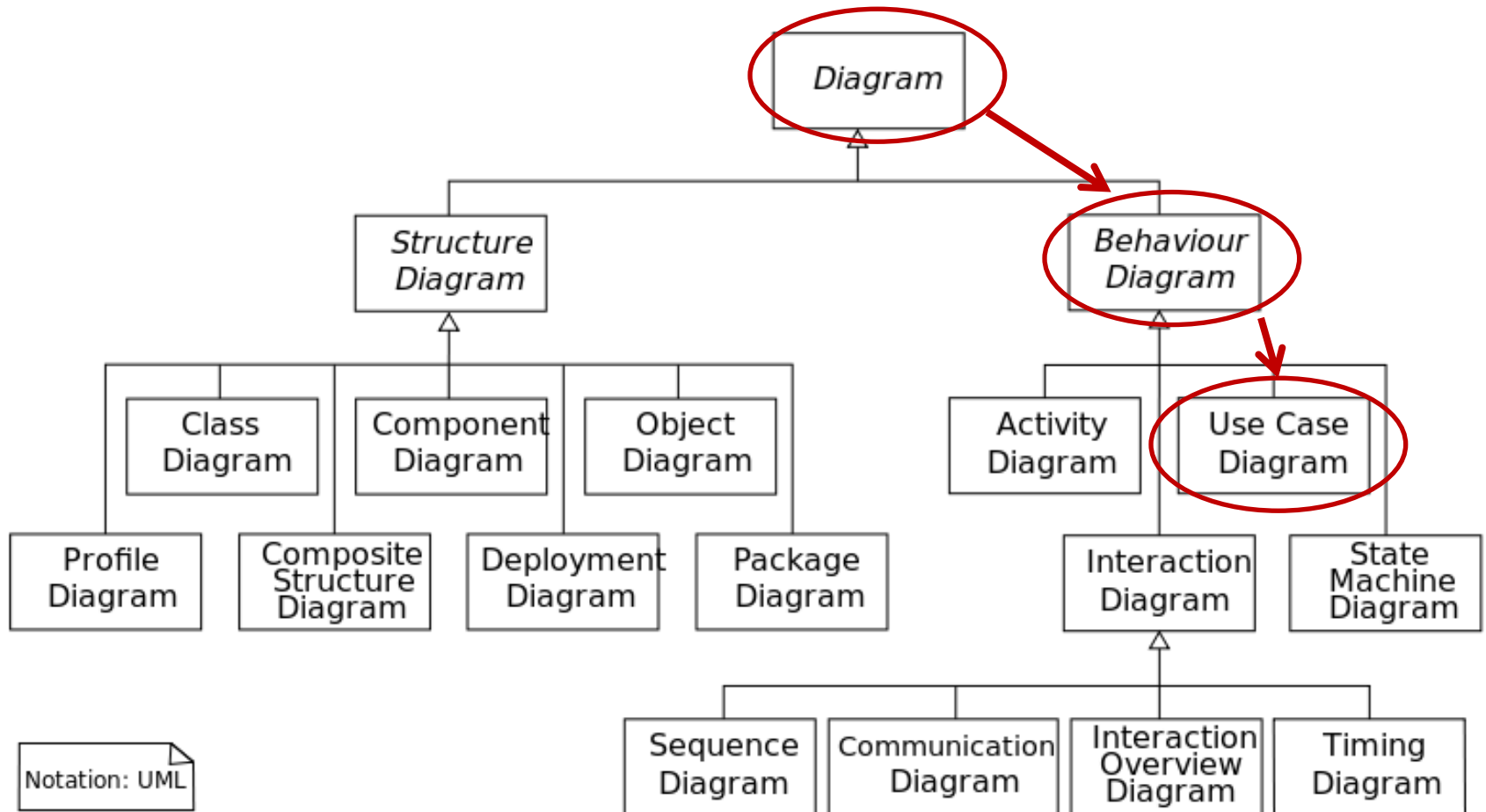




Διαγράμματα UML

Διαγράμματα Περιπτώσεων Χρήσης Use case Diagrams

UML 2.x Διαγράμματα



Εικόνα 10

Γενικές κατηγορίες UML διαγραμμάτων

- **Διαγράμματα δομής (Structure diagrams)**
- Παρουσιάζουν τα σημαντικά στοιχεία της δομής του λογισμικού (ή γενικότερα του τεχνολογικού συστήματος).
- Χρησιμοποιούνται συχνότατα για την **αναπαράσταση της αρχιτεκτονικής** ενός λογισμικού.

- **Διαγράμματα συμπεριφοράς (Behavior diagrams)**
- Τονίζουν το **τι πρέπει να συμβαίνει στο λογισμικό** (τεχνολογικό σύστημα) κατά τη διάρκεια της λειτουργίας του.
- Χρησιμοποιούνται συνηθέστατα για την **περιγραφή των λειτουργιών** ενός συστήματος





ΑΡΙΣΤΟΤΕΛΕΙΟ
ΠΑΝΕΠΙΣΤΗΜΙΟ
ΘΕΣΣΑΛΟΝΙΚΗΣ

Διαγράμματα Περιπτώσεων Χρήσης

Use case Diagrams

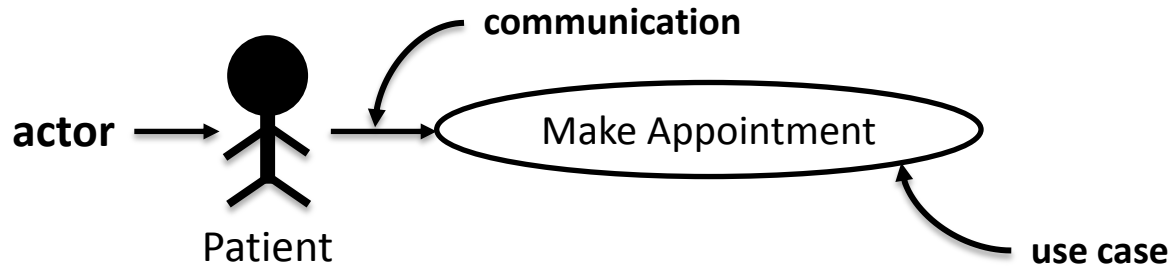
Διαγράμματα Περιπτώσεων Χρήσης (Use Case Diagrams)

- Τα διαγράμματα περιπτώσεων χρήσης αναπαριστούν τι υπηρεσίες προσφέρει το σύστημα σε κάποιον χρήστη του συστήματος (δηλ. τι κάνει το σύστημα και όχι το πώς το κάνει).
- **Περίπτωση Χρήσης:** Μια Περίπτωση Χρήσης (Use Case) είναι μια αλληλουχία ενεργειών που εκτελεί το λογισμικό αλληλεπιδρώντας με το χρήστη ή με εξωτερικά συστήματα, προκειμένου να ικανοποιήσει μία λειτουργική απαίτηση
- Ένα διάγραμμα περίπτωσης χρήσης προκύπτει **αναλύοντας σενάρια χρήσης** του συστήματος, δηλ. αναλύοντας περιπτώσεις για το **πώς χρησιμοποιείται** το σύστημα από τους πιθανούς χρήστες.



Παράδειγμα-1: Διάγραμμα Περιπτώσεων Χρήσης για ένα σύστημα διαχείρισης συναντήσεων σε κλινική

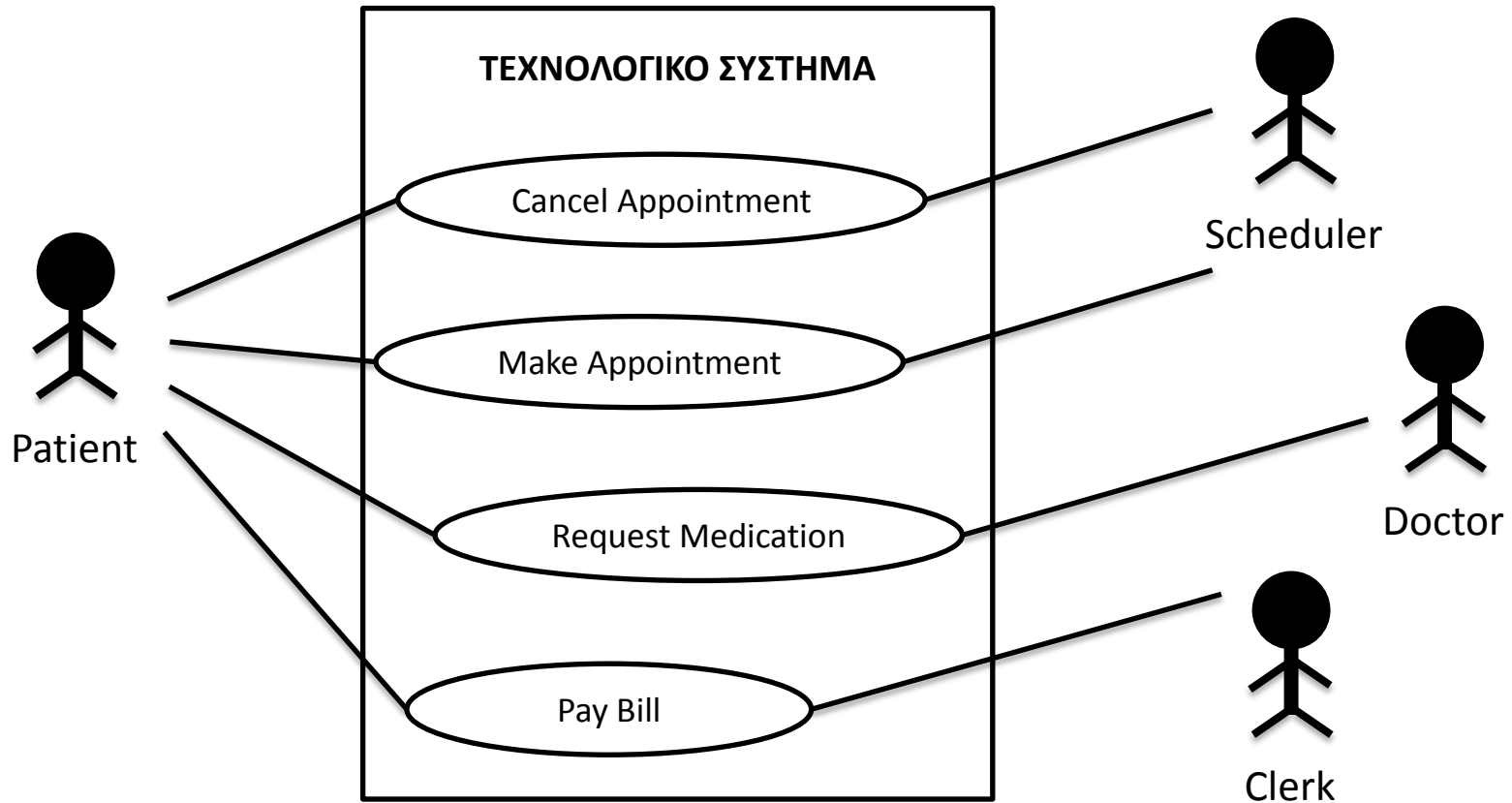
- Σενάριο: Ένας ασθενής ζητά ένα ραντεβού για εξετάσεις σε μια κλινική. Ο χειριστής του συστήματος βρίσκει την πλησιέστερη διαθέσιμη ημερομηνία και προγραμματίζει την επίσκεψη
- Αναπαράσταση σε διάγραμμα περιπτώσεων χρήσης:
- Τι περιλαμβάνει το διάγραμμα:



- **Actor (δράστης, χειριστής, χρήστης):** ο χρήστης του συστήματος
- **Σχέση επικοινωνίας (communication):** δηλώνει τη διάδραση χρήστη-συστήματος
- **Περίπτωση χρήσης (Use case):** δηλώνει τη λειτουργία που προσφέρει το σύστημα στον χρήστη



Παράδειγμα-2: Διάγραμμα Περιπτώσεων Χρήσης

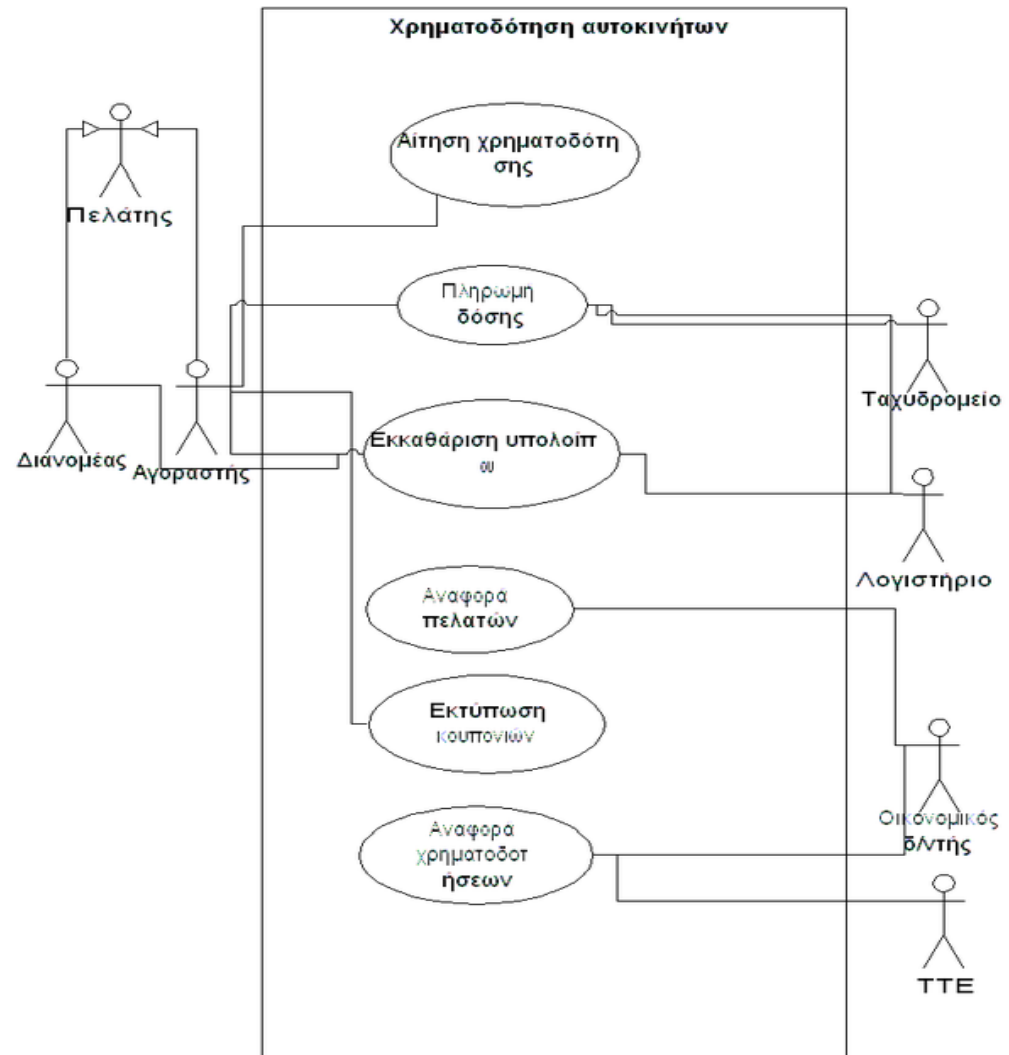


- Ποιοι είναι οι χρήστες;
- Ποιες είναι οι περιπτώσεις χρήσης για τον κάθε χρήστη;



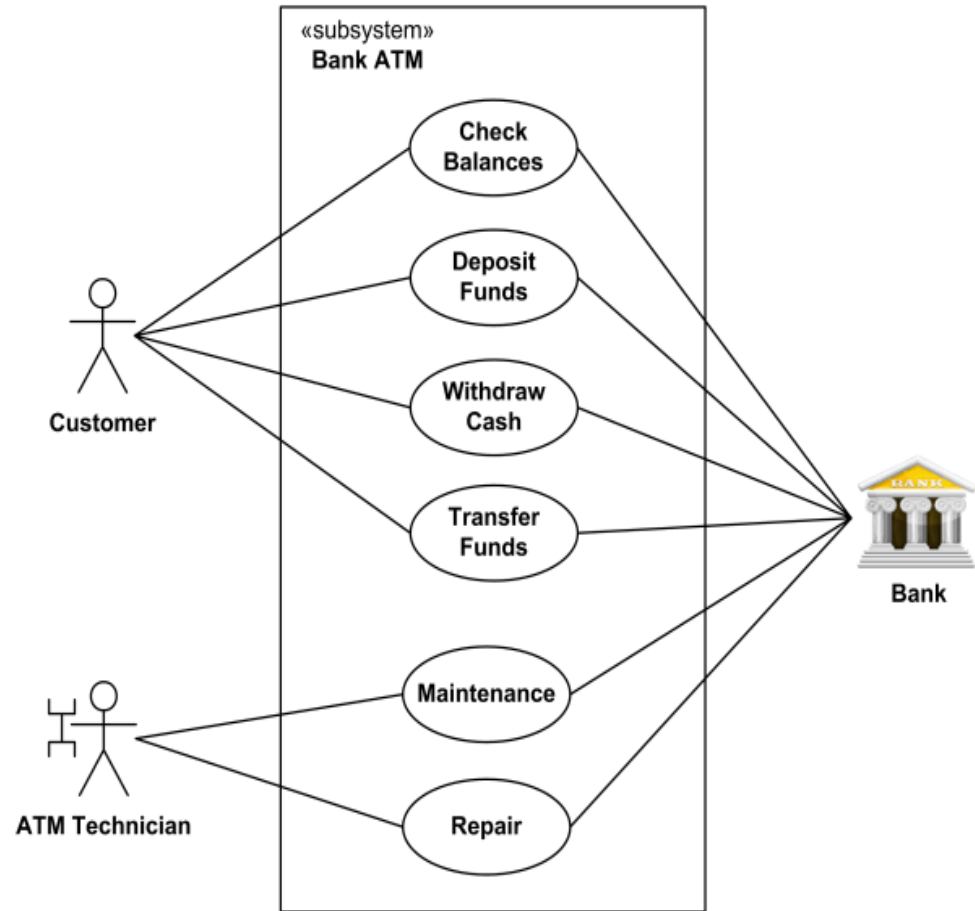
Παράδειγμα-3: Διάγραμμα Περιπτώσεων Χρήσης

- Ποιοι είναι οι χρήστες;
- Ποιες είναι οι περιπτώσεις χρήσης για τον κάθε χρήστη;



Παράδειγμα-4: Διάγραμμα Περιπτώσεων Χρήσης ATM

- Ποιοι είναι οι χρήστες;
- Ποιες είναι οι περιπτώσεις χρήσης για τον κάθε χρήστη;

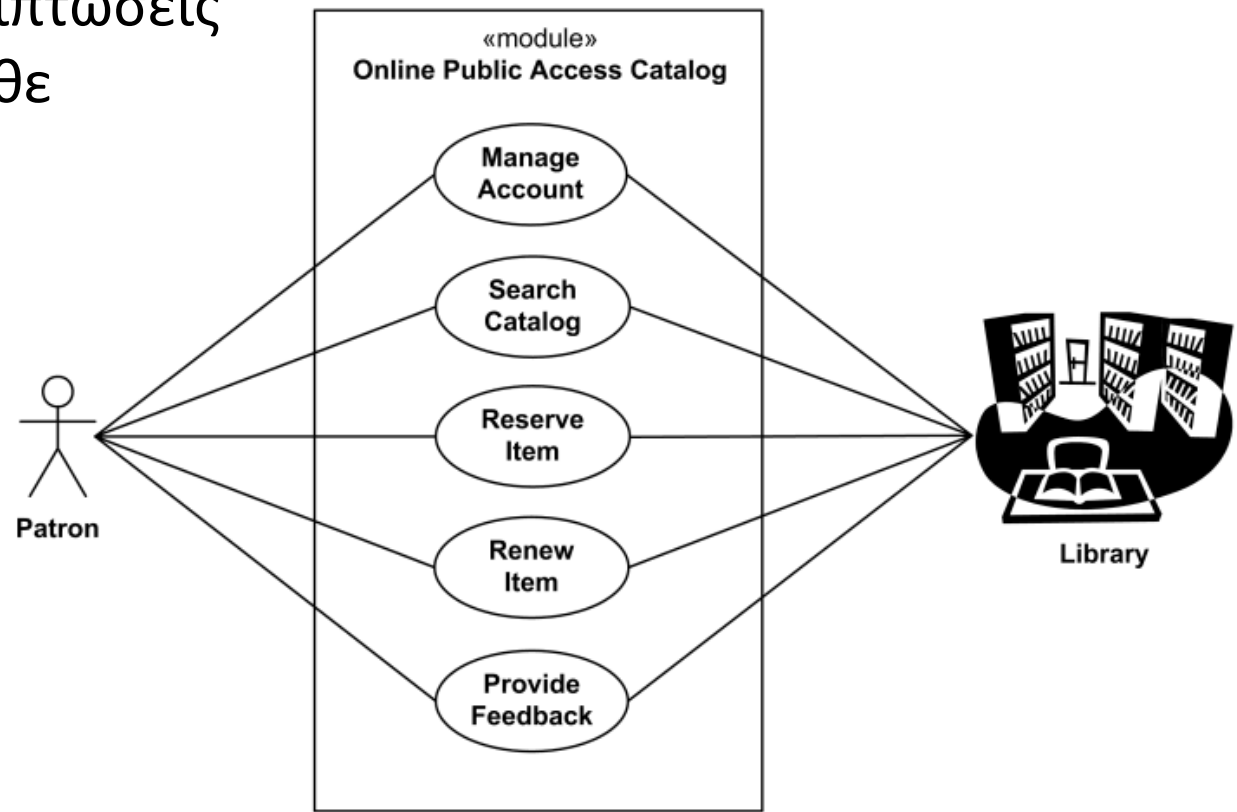


Εικόνα 11



Παράδειγμα-5: Διάγραμμα Περιπτώσεων Χρήσης Online Βιβλιοθήκης

- Ποιοι είναι οι χρήστες;
- Ποιες είναι οι περιπτώσεις χρήσης για τον κάθε χρήστη;



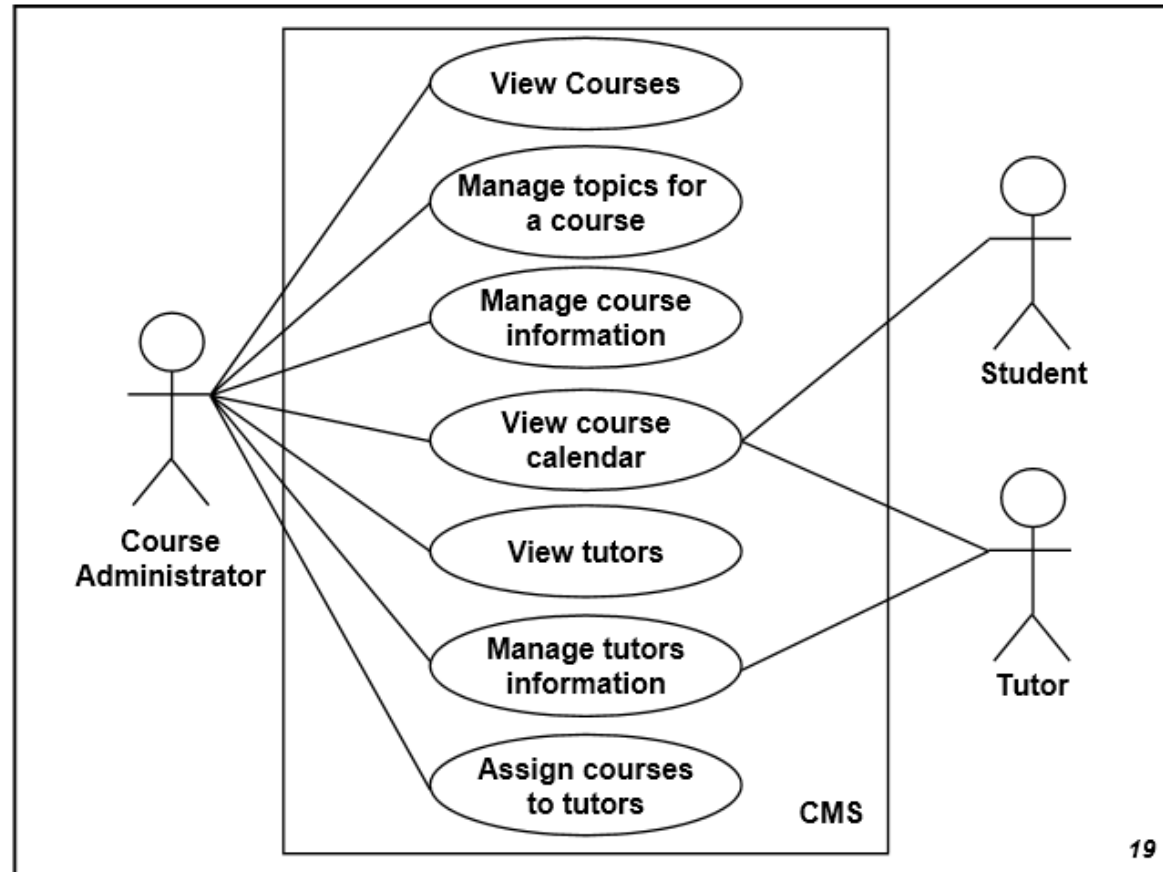
Εικόνα 12



Παράδειγμα-6: Διάγραμμα Περιπτώσεων Χρήσης CMS (Content Management System)

- Ποιοι είναι οι χρήστες;
- Ποιες είναι οι περιπτώσεις χρήσης για τον κάθε χρήστη;

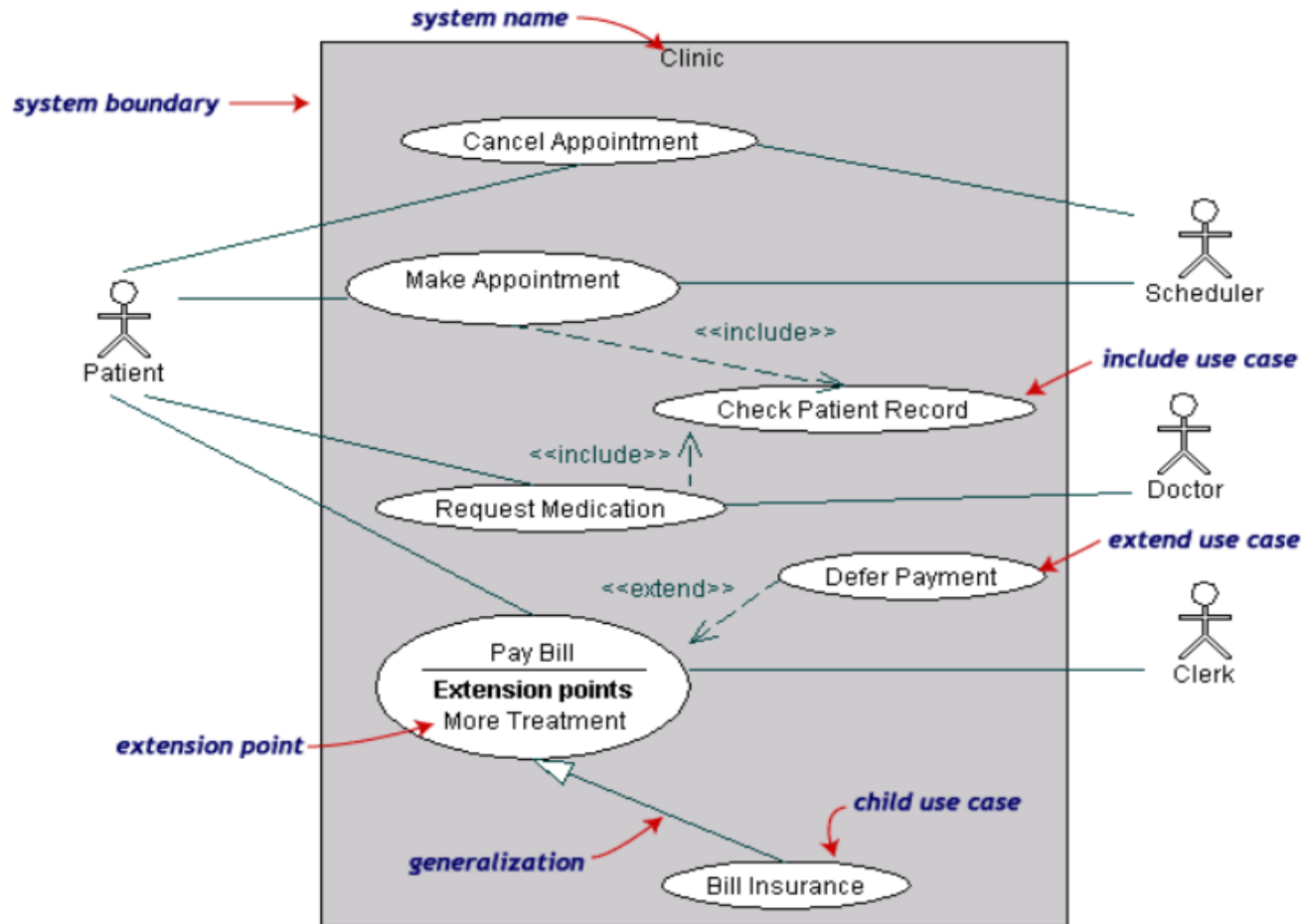
Εικόνα 13



19



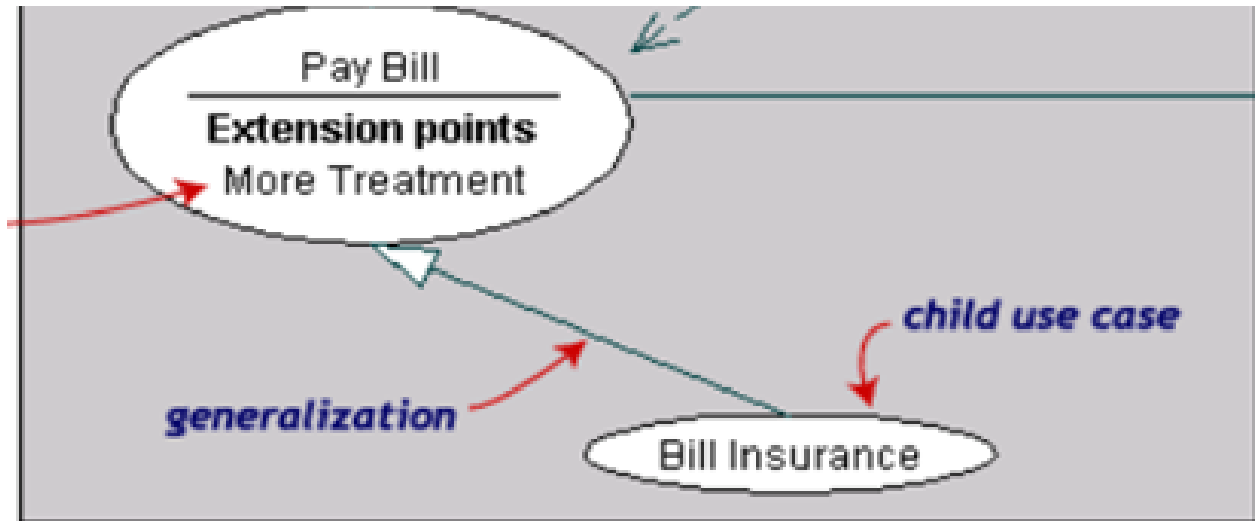
Πρόσθετες σχέσεις σε Διάγραμμα Περιπτώσεων Χρήσης



Εικόνα 14

Γενίκευση - Generalization

- **Generalization** (γενίκευση): δηλώνει ότι κάποια περίπτωση χρήσης (ο «κόμβος-παιδί») είναι απλά **ειδική περίπτωση μιας γενικότερης** (του «κόμβου-γονέα»)– Δήλωση με: συνεχή γραμμή με λευκό τριγωνικό βέλος προς το κόμβο-γονέα



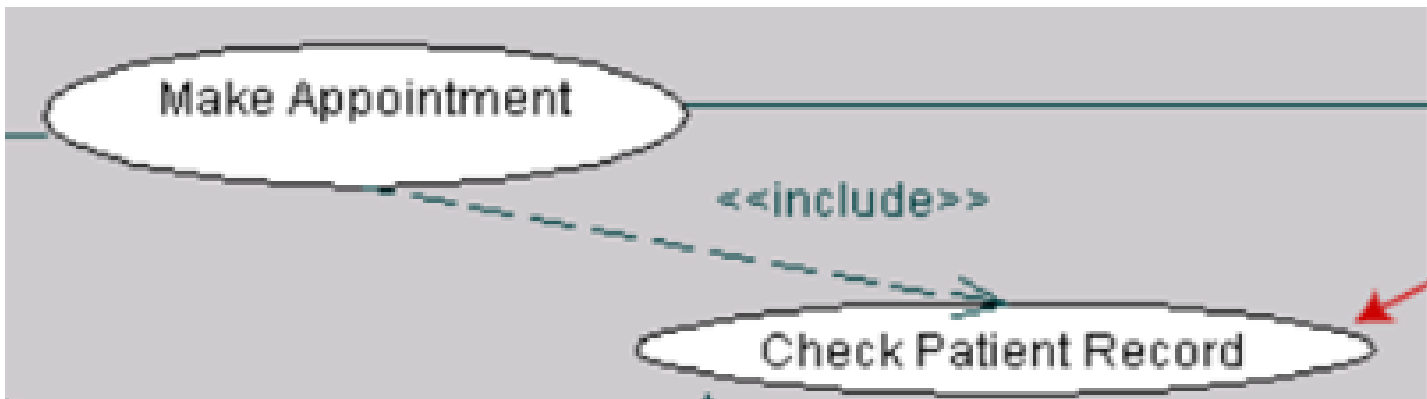
Εικόνα 14

- Εξηγείστε τη σχέση γενίκευσης στο παραπάνω διάγραμμα



Συμπερίληψη - Include

- **Include** (συμπερίληψη): δηλώνει ότι μια περίπτωση χρήσης περιλαμβάνεται μέσα σε μια άλλη (και επομένως εκτελείται πάντα όταν εκτελείται η δεύτερη)
 - Δήλωση με: διακεκομμένη γραμμή βέλους που δείχνει στην περίπτωση χρήσης που περιλαμβάνεται
 - Χρησιμοποιείται η λέξη-ένδειξη `<<include>>`
 - ...στα ελληνικά `<<περιλαμβάνεται>>`



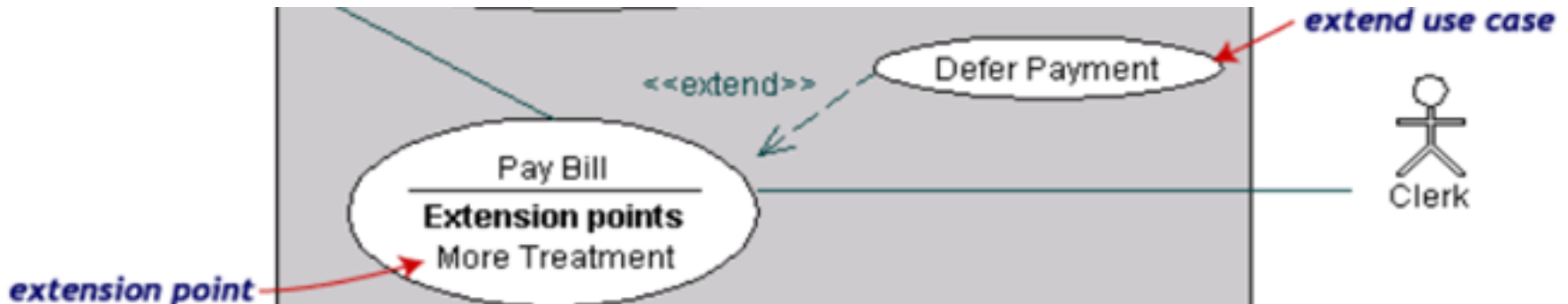
Εικόνα 14

Εξηγείστε τη σχέση συμπερίληψης στο παραπάνω διάγραμμα



Επέκταση - Extend

- **Extend** (επέκταση): δηλώνει ότι μια περίπτωση χρήσης είναι διαφορετική εκδοχή μιας άλλης (και μπορεί να υλοποιηθεί στη θέση της άλλης, δηλ. επεκτείνει την άλλη)
 - Δήλωση: με τη λέξη <<extend>>
 - Το διακεκομμένο βέλος δείχνει από την περίπτωση επέκτασης προς την βασική περίπτωση
 - Μέσα στη βασική περίπτωση γράφουμε πότε γίνεται η αντικατάσταση με την περίπτωση επέκτασης



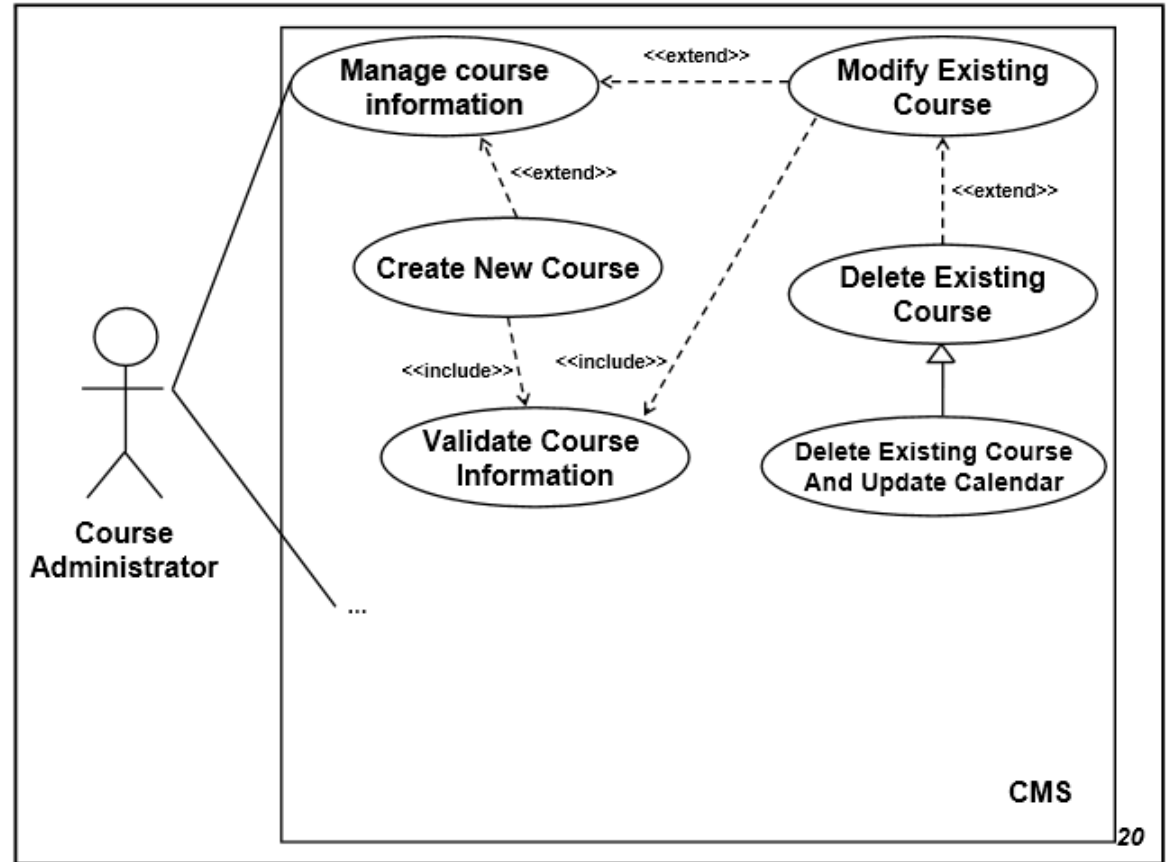
Εικόνα 14

- Εξηγείστε τη σχέση επέκτασης στο παραπάνω διάγραμμα



Παράδειγμα-7: Πρόσθετες σχέσεις σε Διάγραμμα Περιπτώσεων Χρήσης

- Ποιες είναι οι πρόσθετες περιπτώσεις χρήσης και σχέσεις στο διάγραμμα αυτό;
- Τι δηλώνουν;

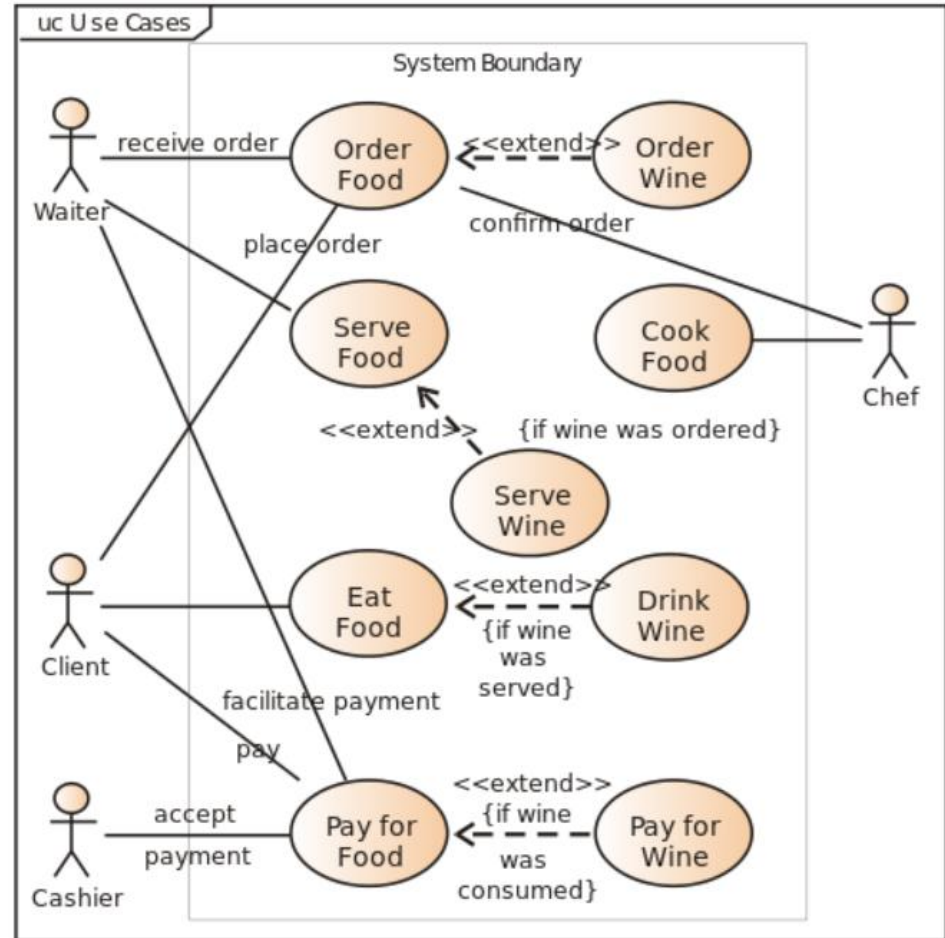


Εικόνα 15



Παράδειγμα-8: Πρόσθετες σχέσεις σε Διάγραμμα Περιπτώσεων Χρήσης

- Ποιες είναι οι πρόσθετες περιπτώσεις χρήσης και σχέσεις στο διάγραμμα αυτό;
- Τι δηλώνουν;



Εικόνα 16



Άσκηση-1

- Δημιουργήστε διαγράμματα περιπτώσεων χρήσης (σκεφθείτε και σχεδιάστε και πρόσθετες σχέσεις) για το λογισμικό με τις παρακάτω λειτουργίες:
- **Λογισμικό εξάσκησης με ερωτήσεις πολλαπλών επιλογών**
 - **Μαθητής:**
 - Επιλέγει την ενότητα που τον ενδιαφέρει
 - Απαντά στο ερωτηματολόγιο
 - Βλέπει τα αποτελέσματά του (σκορ)
 - **Δάσκαλος:**
 - Εισάγει στη ΒΔ τις ερωτήσεις
 - Επιλέγει κάποιες από αυτές και τις οργανώνει σε ερωτηματολόγιο
 - Βλέπει τα αποτελέσματα (σκορ) των μαθητών



Άσκηση-2

- Δημιουργήστε διαγράμματα περιπτώσεων χρήσης (σκεφθείτε και σχεδιάστε και πρόσθετες σχέσεις) για το λογισμικό με τις παρακάτω λειτουργίες:
- **Εκπαιδευτικό λογισμικό πολυμέσων**
 - **Μαθητής:**
 - Επιλέγει την ενότητα που θα παρακολουθήσει
 - Διαδρά με το σύστημα απαντώντας σε ερωτηματολόγια ελέγχου γνώσης
 - Τυπώνει σελίδες
 - Αποθηκεύει εικόνες ή άλλο υλικό κλπ.
 - **Δάσκαλος:**
 - Επιλέγει ενότητα
 - Οργανώνει ερωτηματολόγια
 - Δημιουργεί δικούς του κόμβους προσθέτοντας υλικό στη ΒΔ



Περιεχόμενα Διαλέξεων

- Αρχιτεκτονική Λογισμικού (Software Architecture)
- Όψεις Αρχιτεκτονικής Λογισμικού (Software Architecture Views)
- Διαγράμματα UML (UML Diagrams) -
Διαγράμματα Περίπτωσης Χρήσης (Use Case Diagrams)



Σημείωμα Χρήσης Έργων Τρίτων

- Το Έργο αυτό κάνει χρήση των ακόλουθων έργων:
- Εικόνες/Σχήματα/Διαγράμματα/Φωτογραφίες
- Εικόνα 1. Software Architecture: Execution View - Coronet, http://coronet.iicm.tugraz.at/sa/s5/sa_execution.html
- Εικόνα 2. What is a Software Architecture Document and how would you build it?, <https://aspGuy.wordpress.com/>
- Εικόνα 3. How to Prepare for (and Ace) the Technical Interview, <http://www.cio.com/article/2383000/careers-staffing/how-to-prepare-for--and-ace--the-technical-interview.html>
- Εικόνα 4. An architecture diagram for the proposed system, <http://www.20thcpaint.org/activities.jsp?menu=5&subPage=team4>
- Εικόνα 5. Di Bitonto, P. (2010). Multi-agent architecture for retrieving and tailoring LOs in SCORM compliant distance learning environments.
- Εικόνα 6. De Bra, P., Aroyo, L., & Chepegin, V. (2006). The next big thing: Adaptive web-based systems. *Journal of Digital Information*, 5(1).
- Εικόνα 7. P. Kruchten (1995). The 4+1 View Model of Architecture. In IEEE Software, vol. 12, no. 6, November 1995, pp. 42-50, IEEE.
- Εικόνα 8. Introduction to UML, <http://www.openloop.com/softwareEngineering/uml/overview/>
- Εικόνα 9. Deployment diagram, https://en.wikipedia.org/wiki/Deployment_diagram
- Εικόνα 10. Activity diagram, https://en.wikipedia.org/wiki/Activity_diagram
- Εικόνα 11. Hierarchy of UML 2.2 Diagrams, shown as a class diagram, https://en.wikipedia.org/wiki/Unified_Modeling_Language
- Εικόνα 12. UML Use Case Diagram Examples, <http://www.uml-diagrams.org/use-case-diagrams-examples.html>
- Εικόνα 13. UML Use Case Diagram Examples, <http://www.uml-diagrams.org/use-case-diagrams-examples.html>
- Εικόνα 14. Mengenal Use Case Diagram, <http://nickizoner.blogspot.gr/2013/06/mengenal-use-case-diagram.html>
- Εικόνα 15. UML Use Case Diagram Examples, <http://www.uml-diagrams.org/use-case-diagrams-examples.html>
- Εικόνα 16. Use Case Diagram, https://en.wikibooks.org/wiki/Introduction_to_Software_Engineering/UML/Introduction



Σημείωμα Αναφοράς

Copyright Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης, Σταύρος Δημητριάδης.
«Θεωρίες Μάθησης. Βασικές Έννοιες & Ορισμοί». Έκδοση: 1.0. Θεσσαλονίκη
2015. Διαθέσιμο από τη δικτυακή διεύθυνση:
<http://eclass.auth.gr/courses/OCRS416/>



Σημείωμα Αδειοδότησης

Το παρόν υλικό διατίθεται με τους όρους της άδειας χρήσης Creative Commons Αναφορά - Μη Εμπορική Χρήση - Όχι Παράγωγα Έργα 4.0 [1] ή μεταγενέστερη, Διεθνής Έκδοση. Εξαιρούνται τα αυτοτελή έργα τρίτων π.χ. φωτογραφίες, διαγράμματα κ.λ.π., τα οποία εμπεριέχονται σε αυτό και τα οποία αναφέρονται μαζί με τους όρους χρήσης τους στο «Σημείωμα Χρήσης Έργων Τρίτων».



Ο δικαιούχος μπορεί να παρέχει στον αδειοδόχο ξεχωριστή άδεια να χρησιμοποιεί το έργο για εμπορική χρήση, εφόσον αυτό του ζητηθεί.

Ως **Μη Εμπορική** ορίζεται η χρήση:

- που δεν περιλαμβάνει άμεσο ή έμμεσο οικονομικό όφελος από την χρήση του έργου, για το διανομέα του έργου και αδειοδόχο
- που δεν περιλαμβάνει οικονομική συναλλαγή ως προϋπόθεση για τη χρήση ή πρόσβαση στο έργο
- που δεν προσπορίζει στο διανομέα του έργου και αδειοδόχο έμμεσο οικονομικό όφελος (π.χ. διαφημίσεις) από την προβολή του έργου σε διαδικτυακό τόπο

[1] <http://creativecommons.org/licenses/by-nc-nd/4.0/>





Τέλος ενότητας

Επεξεργασία: <Τέγος Στέργιος>
Θεσσαλονίκη, <Εαρινό εξάμηνο 2014-2015>



Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο



ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ
ΕΚΠΑΙΔΕΥΣΗ ΚΑΙ ΔΙΑ ΒΙΟΥ ΜΑΘΗΣΗ
επένδυση στην κοινωνία της γνώσης
ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ ΚΑΙ ΘΡΗΣΚΕΥΜΑΤΩΝ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



ΕΣΠΑ
2007-2013
πρόγραμμα για την ανάπτυξη
ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ



ΑΡΙΣΤΟΤΕΛΕΙΟ
ΠΑΝΕΠΙΣΤΗΜΙΟ
ΘΕΣΣΑΛΟΝΙΚΗΣ

Σημειώματα

Διατήρηση Σημειωμάτων

Οποιαδήποτε αναπαραγωγή ή διασκευή του υλικού θα πρέπει να συμπεριλαμβάνει:

- το Σημείωμα Αναφοράς
- το Σημείωμα Αδειοδότησης
- τη δήλωση Διατήρησης Σημειωμάτων
- το Σημείωμα Χρήσης Έργων Τρίτων (εφόσον υπάρχει)

μαζί με τους συνοδευόμενους υπερσυνδέσμους.

