



Εισαγωγή στην Ενοποιημένη Γλώσσα Μοντελοποίησης (UML)

Β Εισαγωγή στην Ενοποιημένη Γλώσσα Μοντελοποίησης (UML)

Σκοπός Ύπαρξης UML



Σε μεγάλα έργα ανάπτυξης ΠΣ, υπάρχουν πολλαπλές μονάδες, πολλοί προγραμματιστές, διαφορετικοί χρήστες οι οποίοι επιβάλλουν τη χρήση **κοινής ορολογίας**



Σε άλλα τεχνικά έργα η χρήση κοινής ορολογίας και συμβολισμών είναι αυτονόητη και καθιερωμένη

Η UML δημιουργήθηκε ως μια γλώσσα για τη μοντελοποίηση αντικειμενοστραφών εφαρμογών λογισμικού...

B Εισαγωγή στην Ενοποιημένη Γλώσσα Μοντελοποίησης (UML)

Τί είναι η UML



Η UML είναι μια γλώσσα για:

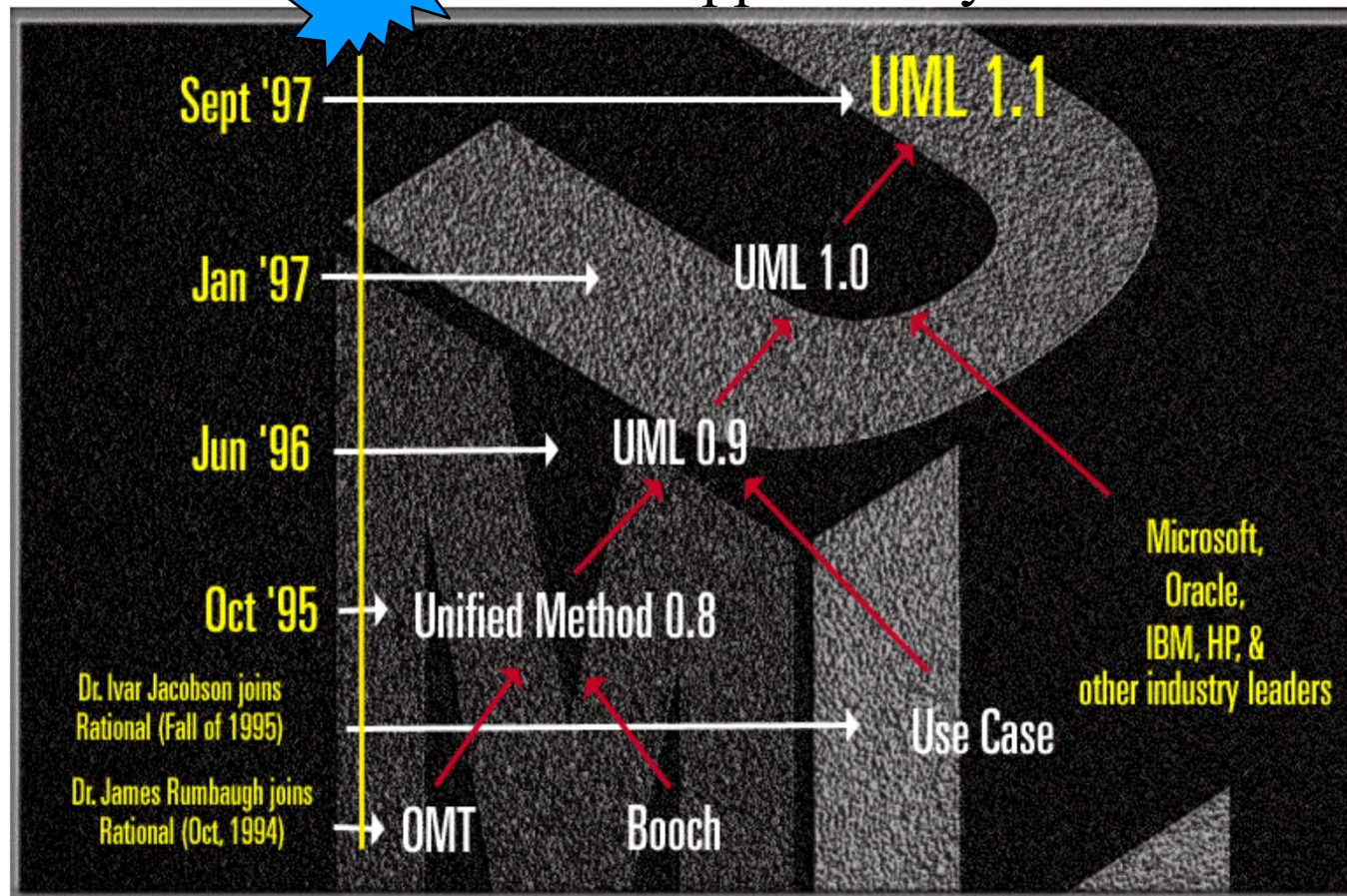
- Κατάρτιση προδιαγραφών λογισμικού και τεκμηρίωση τμημάτων λογισμικού
- Αναπαράσταση με οπτικό τρόπο (visualization) τμημάτων λογισμικού
- Μοντελοποίηση εταιρικών και άλλων συστημάτων που δεν αφορούν λογισμικό

Η UML είναι μία γλώσσα μοντελοποίησης (σύνολο από διαγράμματα). **Δεν είναι** γλώσσα προγραμματισμού !!!

Β Εισαγωγή στην Ενοποιημένη Γλώσσα Μοντελοποίησης (UML)

Η Ιστορία της UML

Nov '97  UML approved by the OMG



Μοντέλα UML

Οι σχεδιαστές χρησιμοποιούν τη UML για δημιουργία μοντέλων, τα οποία είναι χρήσιμα - σωστά όταν είναι:

- **Ακριβή:** περιγράφουν επακριβώς το σύστημα υπό ανάπτυξη
- **Συνεπή:** διαφορετικές απόψεις δεν εκφράζουν πράγματα αλληλοσυγκρουόμενα
- εύκολο να μεταδοθούν σε άλλους
- εύκολο να αλλαχτούν
- **Κατανοητά:** όσο πιο απλά γίνεται αλλά όχι απλοϊκά

B Εισαγωγή στην Ενοποιημένη Γλώσσα Μοντελοποίησης (UML)

Τί πρέπει να θυμόμαστε...

- *Η UML λειτουργεί ανεξάρτητα από τις μεθόδους ανάπτυξης ΠΣ και σε πληθώρα πεδίων εφαρμογής*
- *Είναι ανεξάρτητη από πλατφόρμες και γλώσσες προγραμματισμού*
- *Είναι γλώσσα γραφικής αναπαράστασης με πλούσια σημειολογία*
- *Τα διαγράμματα που προκύπτουν είναι τόσο καλά, όσο καλή είναι η ανάλυση του προβλήματος και η συλλογή απαιτήσεων που έχουμε κάνει*

B Εισαγωγή στην Ενοποιημένη Γλώσσα Μοντελοποίησης (UML)

Μοντελοποίηση με UML (1/4)

Η UML αποτυπώνει τόσο τη στατική δομή, όσο και τη δυναμική συμπεριφορά ενός συστήματος.

Ένα αντικειμενοστραφές σύστημα μοντελοποιείται ως μία συλλογή αντικειμένων που αλληλεπιδρούν για την εκτέλεση μιας λειτουργίας η οποία είναι τελικά αξιοποιήσιμη από τον χρήστη του συστήματος.

Η στατική δομή (ή structural) καθορίζει τα είδη των αντικειμένων που είναι σημαντικά για το σύστημα καθώς και τις συσχετίσεις μεταξύ τους.

Η δυναμική συμπεριφορά (ή behavioral) προσδιορίζει την εξέλιξη των αντικειμένων σε σχέση με τον χρόνο και την επικοινωνία μεταξύ τους.

Β Εισαγωγή στην Ενοποιημένη Γλώσσα Μοντελοποίησης (UML)

Μοντελοποίηση με UML (2/4)

- **Απόψεις:** Δείχνουν από διαφορετικές οπτικές γωνίες το υπό ανάπτυξη σύστημα. Συνδέουν τη γλώσσα μοντελοποίησης με τη διαδικασία που μοντελοποιείται
- **Διαγράμματα:** Είναι γραφήματα που περιγράφουν τα περιεχόμενα μιας άποψης. Η UML έχει εννιά διαφορετικούς τύπους γραφημάτων συνδυασμοί των οποίων συνθέτουν όλες τις απόψεις του συστήματος
- **Στοιχεία των μοντέλων:** Είναι οι ιδέες-σύμβολα που συνθέτουν ένα διάγραμμα. Έχουν πάντα το ίδιο νόημα σε όποιο διάγραμμα και αν χρησιμοποιηθούν
- **Γενικοί μηχανισμοί:** Προσφέρουν πρόσθετη πληροφορία, σχόλια, σημασιολογική πληροφορία για στοιχεία του μοντέλου

- **Άποψη περίπτωσης χρήσης - Use-case view**
Λειτουργία του συστήματος όπως την βλέπουν εξωτερικοί δράστες
- **Λογική άποψη - Logical view**
Στατική δομή και δυναμική συμπεριφορά του συστήματος
- **Άποψη συστατικών του συστήματος - Component view**
Οργάνωση των συστατικών του κώδικα
- **Άποψη συγχρονισμού - Concurrency view**
Επικοινωνία και συγχρονισμός παράλληλων διεργασιών
- **Άποψη ανάπτυξης - Deployment view**
Ανάπτυξη του συστήματος σε φυσικά μέσα (υπολογιστές, άλλες συσκευές)

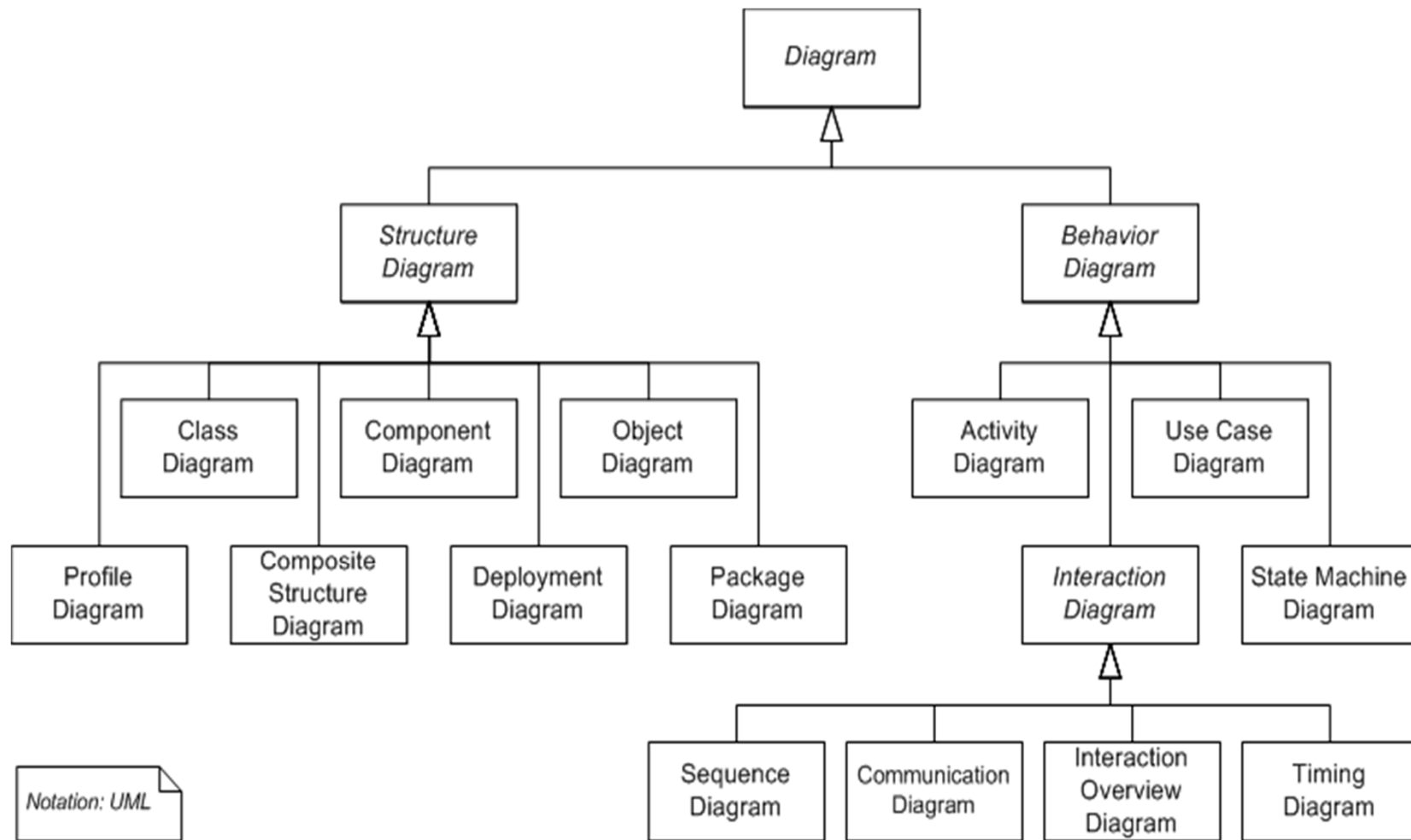
Β Εισαγωγή στην Ενοποιημένη Γλώσσα Μοντελοποίησης (UML)

Μοντελοποίηση με UML (4/4)

	Μοντελοποίηση Απαιτήσεων	Σενάρια, Διαγράμματα Περ.Χρήσης (Scenaria, Use Case Diagrams)
Περιγραφή λειτουργίας από τη σκοπιά του χρήστη	Μοντελοποίηση στατικής δομής του συστήματος	Διαγράμματα Κλάσεων (Class Diagrams) Διαγράμματα Αντικειμένων (Object Diagrams) Διαγράμματα Συστατικών (Component Diagrams) Παραταξιακά Διαγράμματα (Deployment Diagrams)
Στατική Δομή (ποια στοιχεία πρέπει να υπάρχουν στο σύστημα)		Μοντελοποίηση δυναμικών χαρακτηριστικών του συστήματος
Συμπεριφορά (τι πρέπει να συμβεί μέσα στο σύστημα)	Μοντελοποίηση της συμπεριφοράς συγκεκριμένων αντικειμένων	Διαγράμματα Κατάστασης (State Diagrams)
	Μοντελοποίηση των βημάτων εκτέλεσης μιας διαδικασίας	Διαγράμματα Δραστηριότητας (Activity Diagrams)

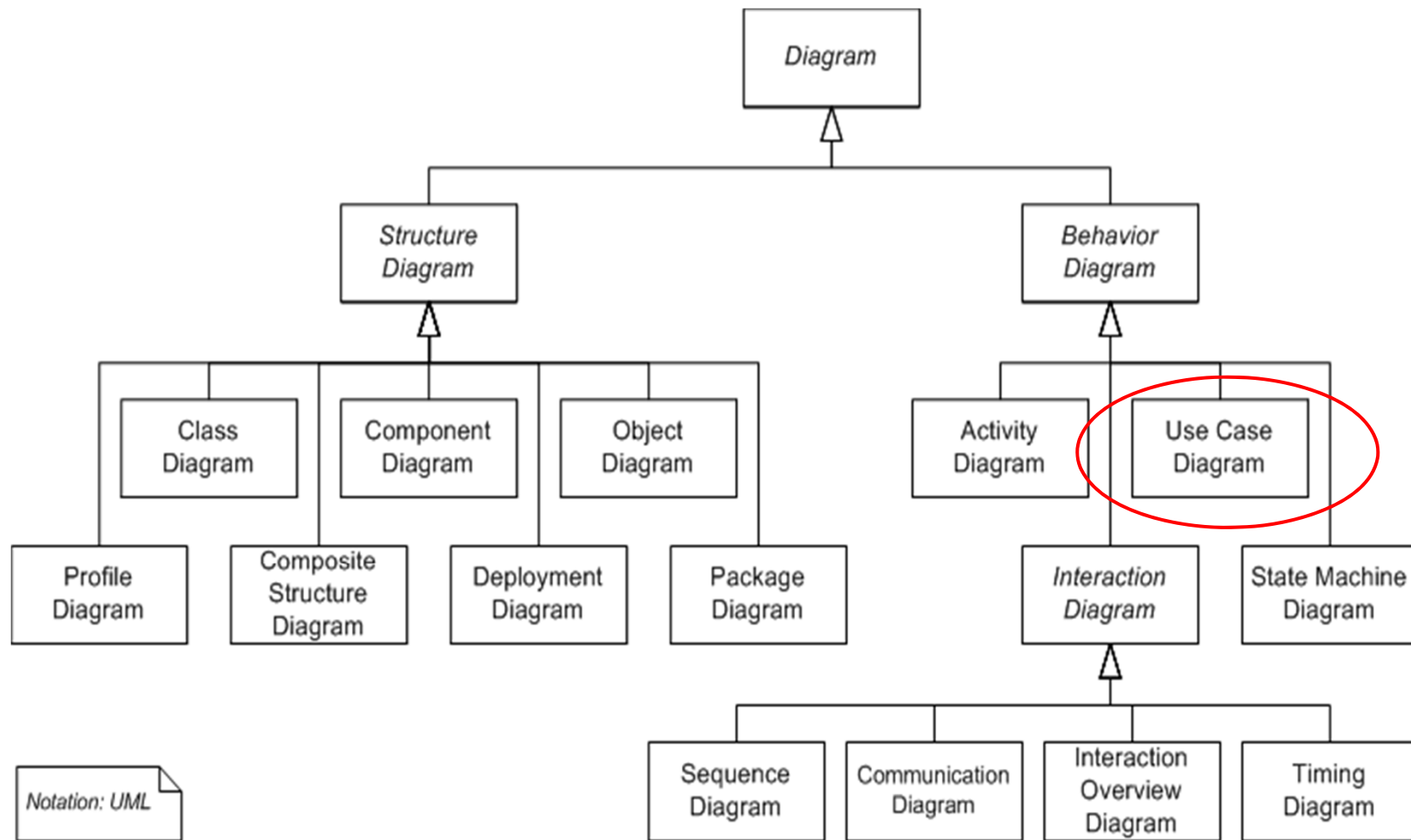
B Εισαγωγή στην Ενοποιημένη Γλώσσα Μοντελοποίησης (UML)

Διαγράμματα UML 2.2



Β Εισαγωγή στην Ενοποιημένη Γλώσσα Μοντελοποίησης (UML)

Διαγράμματα Περιπτώσεων Χρήσης (1/10)



B Εισαγωγή στην Ενοποιημένη Γλώσσα Μοντελοποίησης (UML)


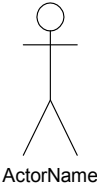

Διαγράμματα Περιπτώσεων Χρήσης (2/10)

- Τα διαγράμματα **περιπτώσεων χρήσης** στη UML χρησιμοποιούνται για τη μοντελοποίηση της συμπεριφοράς ενός συστήματος, υποσυστήματος ή κλάσης, **όπως αυτή γίνεται αντιληπτή από τον εξωτερικό χρήστη.**
- Τα διαγράμματα περιπτώσεων χρήσης διαμερίζουν τη λειτουργικότητα του συστήματος σε **συναλλαγές που έχουν νόημα για τους δράστες** (actors) - ιδανικούς χρήστες του συστήματος. Τα επιμέρους τμήματα της λειτουργικότητας ονομάζονται περιπτώσεις χρήσης.
- Το σύνολο των περιπτώσεων χρήσης συνιστούν τη συμπεριφορά του συστήματος. Ο τυπικός ορισμός μιας περίπτωσης χρήσης είναι **μία ακολουθία συναλλαγών που πραγματοποιείται από το σύστημα για την παραγωγή μετρήσιμων αποτελεσμάτων που έχουν νόημα για τον χρήστη.**

B Εισαγωγή στην Ενοποιημένη Γλώσσα Μοντελοποίησης (UML)




Διαγράμματα Περιπτώσεων Χρήσης (3/10)

Δομικά Στοιχεία Διαγραμμάτων Περιπτώσεων Χρήσης

Construct	Description	Syntax
Περίπτωση χρήσης (use case)	Περιγράφει ποια είναι η λειτουργία ενός συστήματος /κλάσης/ διαπροσωπίας. Είναι ένα σύνολο από δραστηριότητες που επιτελούνται από το σύστημα ώστε να παραχθεί κάποια συμπεριφορά που είναι αισθητή από κάποιο δράστη.	
Δράστης (actor)	Ένα σύνολο από συναφείς ρόλους που παίζουν οι «χρήστες» του συστήματος κατά τη διάρκεια των περιπτώσεων χρήσης. Οι δράστες ενεργούν έξω από το σύστημα.	
Όριο του συστήματος (system boundary)	Ορίζει το όριο το οποίο διαχωρίζει το σύστημα από τους δράστες που αλληλεπιδρούν με το σύστημα.	

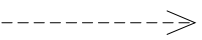
B Εισαγωγή στην Ενοποιημένη Γλώσσα Μοντελοποίησης (UML)

Διαγράμματα Περιπτώσεων Χρήσης (4/10)

Construct	Description	Syntax
Εταιρικότητα (association)	Η συμμετοχή και ο τρόπος επικοινωνίας ενός δράστη με μία περίπτωση χρήσης.	
Γενίκευση (generalization)	Μια σχέση ταξινόμησης ανάμεσα σε μια γενική περίπτωση χρήσης και μια πιο ειδικευμένη περίπτωση χρήσης.	
Επέκταση (extend)	Μια σχέση ανάμεσα σε περιπτώσεις χρήσης που ορίζει εναλλακτικούς τρόπους διαχείρισης της βασικής περίπτωσης χρήσης	

B Εισαγωγή στην Ενοποιημένη Γλώσσα Μοντελοποίησης (UML)

Διαγράμματα Περιπτώσεων Χρήσης (5/10)

Construct	Description	Syntax
Συνοπλογοισμός (include)	Μια σχέση ανάμεσα σε μια βασική και μια ή περισσότερες μερικές περιπτώσεις χρήσης που ορίζει ότι η βασική περίπτωση χρήσης απαιτεί τη λειτουργία των μερικών περιπτώσεων χρήσης.	<code><<include>></code> 

B Εισαγωγή στην Ενοποιημένη Γλώσσα Μοντελοποίησης (UML)

Διαγράμματα Περιπτώσεων Χρήσης (6/10)

- Μια περίπτωση χρήσης περιγράφει κάποια σημαντική και πλήρως κατανοητή λειτουργία του συστήματος (από τη σκοπιά του χρήστη - και των προγραμματιστών)
- Τα σενάρια (περιπτώσεις χρήσεις) ορίζονται με τη χρήση κάποιου κειμένου που περιέχει ουσιαστικά και ρήματα. Τα ουσιαστικά θα μας βοηθήσουν να ορίσουμε τις κλάσεις και τα ρήματα τις δραστηριότητες.
- Παραγοντοποίηση των περιπτώσεων χρήσης που χρησιμοποιούνται και από άλλα σενάρια
 - Εάν η περίπτωση χρήσης απαιτείται τότε χρησιμοποιούμε <<include>>
 - Εάν η περίπτωση χρήσης είναι προαιρετική τότε χρησιμοποιούμε <<extend>>
- Ένα διάγραμμα χρήσης
 - Περιέχει περιπτώσεις στο ίδιο επίπεδο αφαίρεσης
 - Περιέχει μόνο τους απαραίτητους δράστες
- Πολλές και συναφείς περιπτώσεις χρήσεις μπορούν να συμπυκνωθούν σε συσκευασίες (packages)

B Εισαγωγή στην Ενοποιημένη Γλώσσα Μοντελοποίησης (UML)

Διαγράμματα Περιπτώσεων Χρήσης (7/10)

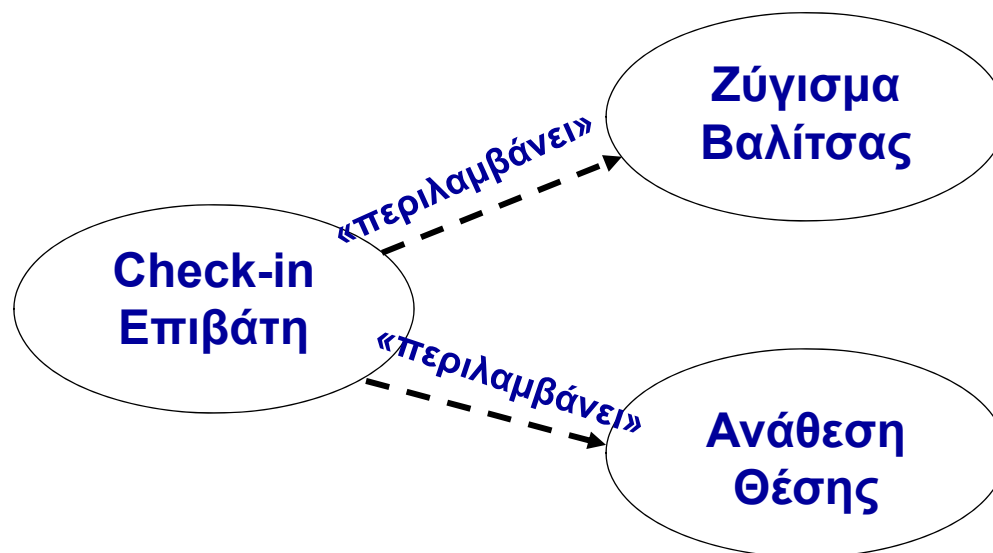
Τα πιθανά Use Cases χωρίζονται σε 2 κατηγορίες:

- Σε αυτά που αποτελούν τους **στόχους** του Χρήστη και είναι όλες οι πιθανές χρήσεις που μπορεί να κάνει στο σύστημα
 - *Παράδειγμα:* μέσω ενός ATM ο πελάτης της τράπεζας (Χρήστης) μπορεί να κάνει ανάληψη, κατάθεση και ερώτηση υπολοίπου
- Στις **ενέργειες** που απαιτούν οι στόχοι του Χρήστη για να υλοποιηθούν
 - *Παράδειγμα:* μέσω ενός ATM ο πελάτης της τράπεζας (Χρήστης) για να μπορεί να κάνει ανάληψη, κατάθεση και ερώτηση υπολοίπου, θα πρέπει να πληκτρολογήσει το PIN

B Εισαγωγή στην Ενοποιημένη Γλώσσα Μοντελοποίησης (UML)

Διαγράμματα Περιπτώσεων Χρήσης (8/10)

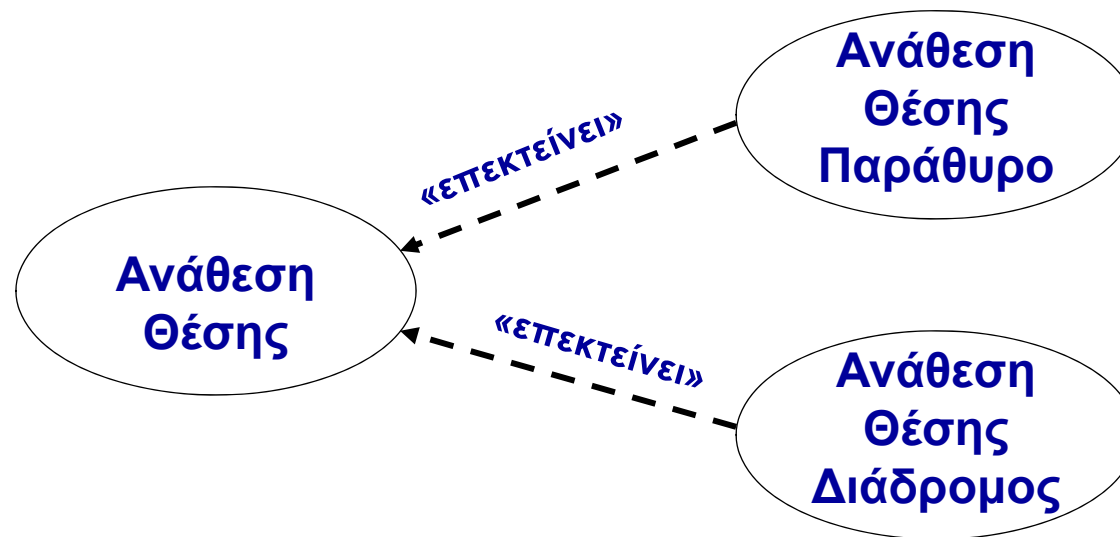
A «περιλαμβάνει» B: Σημαίνει ότι η B είναι μία υπολειτουργία της A η οποία απαιτείται να γίνει προκειμένου να εκπληρωθεί η A



B Εισαγωγή στην Ενοποιημένη Γλώσσα Μοντελοποίησης (UML)

Διαγράμματα Περιπτώσεων Χρήσης (9/10)

A «επεκτείνει» B: Σημαίνει ότι η A είναι μία λειτουργία **ιδίου τύπου** με την B, με τη διαφορά ότι η A είναι μία πιο ειδική/συγκεκριμένη περίπτωση της B.



B Εισαγωγή στην Ενοποιημένη Γλώσσα Μοντελοποίησης (UML)

Διαγράμματα Περιπτώσεων Χρήσης (10/10)

A «γενικεύει» B: Σημαίνει ότι η A είναι μία λειτουργία που κληρονομεί τα χαρακτηριστικά της B, με τη διαφορά ότι η A έχει επιπρόσθετα χαρακτηριστικά σε σχέση με τη B.

Η γενική περίπτωση χρήσης ονομάζεται γονέας ενώ αυτές που την εξειδικεύουν παιδιά

