

Ανάλυση απαιτήσεων & Δομημένη ανάλυση

Κεφάλαιο 2

Μέρος 4: Τυπικές Μέθοδοι για Προδιαγραφή
Απαιτήσεων & Αντικειμενοστρεφής ανάλυση

Τυπικές Μέθοδοι για
Προδιαγραφή
Απαιτήσεων
...

Τυπικές Μέθοδοι για Προδιαγραφή Απαιτήσεων

- Οι τυπικές μέθοδοι, διαθέτουν αυστηρό μαθηματικό υπόβαθρο που επιτρέπει την αυστηρή και χωρίς αμφισημίες έκφραση των προδιαγραφών και τον έλεγχό τους
- Διαδεδομένες τυπικές μέθοδοι:
 - γλώσσα Z (γνωστή και ως Z notation) -ίσως η πλέον γνωστή τυπική γλώσσα για τη σύνταξη προδιαγραφών-
 - γλώσσα OCL (Object Constraint Language) που χρησιμοποιείται σε μοντέλα UML
 - διαγράμματα Petri
 - διαγράμματα μετάβασης καταστάσεων



Τυπικές Μέθοδοι για Προδιαγραφή Απαιτήσεων

- Πλεονεκτήματα:
 - Αυστηρό μαθηματικό υπόβαθρο
 - Έλεγχος με αυτοματοποιημένο και διεξοδικό τρόπο
 - Είναι δυνατόν να ελέγξουμε ότι καλύπτονται όλες οι απαιτήσεις
 - Είναι δυνατόν να επιβεβαιώσουμε την ισχύ συγκεκριμένων συνθηκών
 - Είναι εφικτό να διακριβωθεί ότι το σύστημα συμπεριφέρεται με τον επιθυμητό τρόπο
- Μειονεκτήματα:
 - Δυσκολία κατανόησης από πελάτες
 - Αδυναμία επιβεβαίωσης των προδιαγραφών από πελάτες
 - Δυσχερής και κοστοβόρα σύνταξη
 - Ανάγκη για εξειδικευμένο προσωπικό
 - Πολλές εργατοώρες
- Συχνή προτίμηση ημιτυπικών μεθόδων έναντι των τυπικών



Τυπικές Μέθοδοι για Προδιαγραφή Απαιτήσεων

Παράδειγμα:

- έστω ότι για το σύστημα ελέγχου ενός πυρηνικού αντιδραστήρα υπάρχει η εξής απαίτηση:
 - αν αφιχθεί το σήμα διακοπής Δ , τότε η διεργασία B σταματά να εκτελείται.
 - η διεργασία A ξεκινά την εκτέλεσή της μόλις παραληφθεί το σήμα διακοπής Δ .
 - σε κάθε περίπτωση θα πρέπει είτε (α) να εκτελείται η διεργασία A και να μην εκτελείται η B, είτε (β) να εκτελείται η διεργασία B και να μην εκτελείται η A, είτε (γ) να μην εκτελείται καμία από τις διεργασίες A και B.



Τυπικές Μέθοδοι για Προδιαγραφή Απαιτήσεων

Παράδειγμα (συνέχεια):

- Αρχικά διαμορφώνουμε τους μεμονωμένους όρων που εμφανίζονται στην απαίτηση:
 - p : φθάνει το σήμα διακοπής Δ
 - q : εκτελείται η διεργασία B
 - r : εκτελείται η διεργασία A
- Κατόπιν συνδυάζονται οι μεμονωμένοι όροι με λογικούς τελεστές:
 - 1.1 $p \Rightarrow \neg q$
 - 1.2 $p \Rightarrow r$
 - 1.3 $(r \wedge \neg q) \vee (q \wedge \neg r) \vee (\neg q \wedge \neg r)$



Τυπικές Μέθοδοι για Προδιαγραφή Απαιτήσεων

Παράδειγμα (συνέχεια):

- Για να κάνουμε έλεγχο συνέπειας κατασκευάζουμε πίνακα αληθείας και μέσω αυτού επιβεβαιώσουμε ότι υπάρχει τουλάχιστον ένας συνδυασμός τιμών αληθείας των μεμονωμένων όρων που δίνει τιμή «αληθές» σε κάθε συνδυασμένο όρο

A/A	p	q	r	1.1 $p \Rightarrow \neg q$	1.2 $p \Rightarrow r$	1.3 $(r \wedge \neg q) \vee (q \wedge \neg r) \vee (\neg q \wedge \neg r)$
1	T	T	T	F	T	F
2	T	T	F	F	F	T
3	T	F	T	T	T	T
4	T	F	F	T	F	T
5	F	T	T	T	T	F
6	F	T	F	T	T	T
7	F	F	T	T	T	T
8	F	F	F	T	T	T



Δίκτυα Petri – βασικοί ορισμοί

- Ένα δίκτυο Petri αποτελείται από *θέσεις* (places), *μεταβάσεις* (transitions) και *κατευθυνόμενες ακμές* (directed arcs).
 - Οι ακμές συνδέουν θέσεις με μεταβάσεις (στην περίπτωση αυτή οι θέσεις καλούνται *θέσεις εισόδου της μετάβασης*) ή μεταβάσεις με θέσεις (στην περίπτωση αυτή οι θέσεις καλούνται *θέσεις εξόδου της μετάβασης*).
 - Μία ακμή δεν μπορεί να συνδέει όμως άμεσα δύο θέσεις μεταξύ τους ή δύο μεταβάσεις μεταξύ τους.
 - μία θέση μπορεί να είναι θέση εισόδου ή θέση εξόδου για πολλές μεταβάσεις.

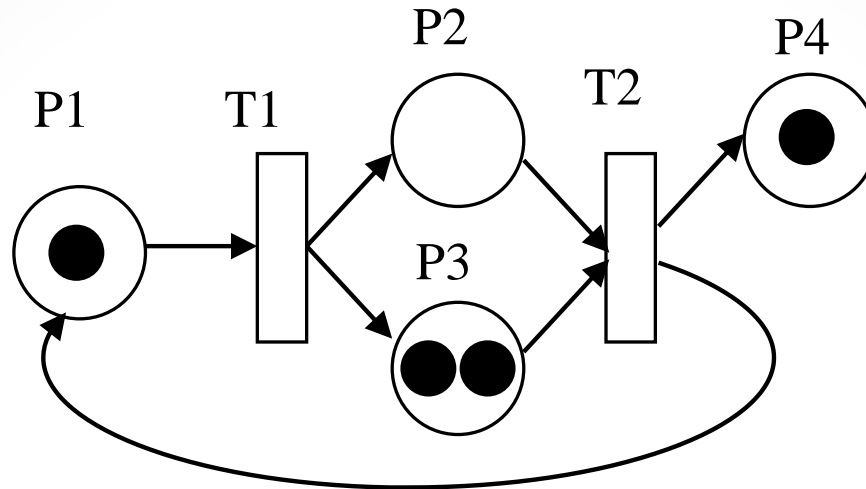


Δίκτυα Petri – βασικοί ορισμοί

- Μία θέση μπορεί να περιέχει έναν ακέραιο αριθμό *διακριτικών* (tokens).
- Μία μετάβαση σε ένα δίκτυο καλείται *ενεργός* όταν εμφανίζεται ένα διακριτικό σε όλες τις θέσεις εισόδου.
- Σε μια ενεργή μετάβαση μπορεί να γίνει *σκανδαλισμός*
 - τότε *καταναλώνεται* ένα διακριτικό από κάθε θέση εισόδου και τοποθετείται ένα διακριτικό σε κάθε θέση εξόδου
 - αν μία θέση εισόδου που περιέχει ένα διακριτικό είναι *συνδεδεμένη* με δύο ή περισσότερες ενεργές μεταβάσεις, τότε δεν είναι *αιτιοκρατικό* (deterministic) το σε ποια από τις μεταβάσεις θα γίνει *σκανδαλισμός*.



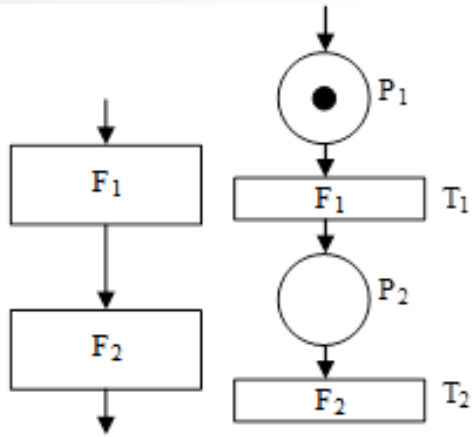
Δίκτυα Petri – Παράδειγμα



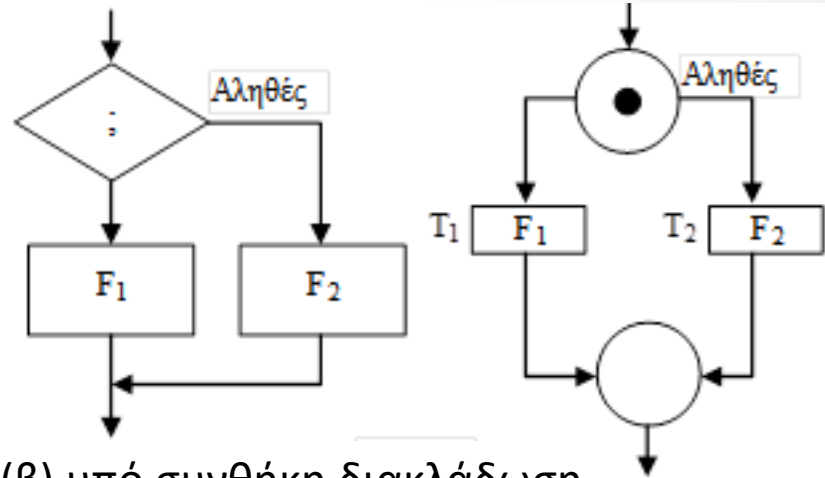
- Θέσεις: P1, P2, P3 και P4
- Μεταβάσεις: T1 και T2
- Οι θέσεις P1, P3 και P4 έχουν 1, 2 και 1 διακριτικά αντίστοιχα
- Μόνο η μετάβαση T1 είναι ενεργός
 - Μετά τον σκανδαλισμό της T1 θα γίνει ενεργός η T2



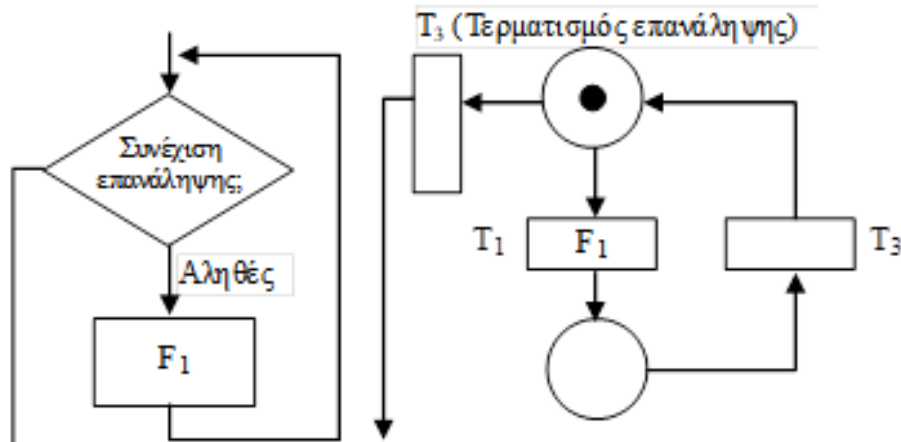
Δίκτυα Petri – Αντιστοιχία με διαδικαστικές γλώσσες προγραμματισμού



(α) ακολουθία



(β) υπό συνθήκη διακλάδωση



(γ) επανάληψη



Αντικειμενοστρεφής ανάλυση

...

Αντικειμενοστρεφής ανάλυση

- Το σύστημα θεωρείται ως ένα σύνολο αντικειμένων που αλληλεπιδρούν για να επιτελέσουν τις λειτουργίες του συστήματος
- Κάθε οντότητα του περιβάλλοντος του προβλήματος αναπαρίσταται με ένα αντικείμενο
 - Η απεικόνιση αυτή είναι πιο «φυσική» και εύληπτη απ' ό,τι οι μέθοδοι που ακολουθούνται σε άλλες προσεγγίσεις
- Το αντικείμενο χαρακτηρίζεται από:
 - Ταυτότητα: κάθε αντικείμενο έχει μοναδική ταυτότητα
 - Κατάσταση: η κατάσταση ενός αντικειμένου αντιστοιχεί στις τιμές όλων των πεδίων του σε μία δεδομένη στιγμή
 - Συμπεριφορά: λειτουργίες που μπορεί να κάνει το αντικείμενο. Μέσω της συμπεριφοράς, ο εξωτερικός κόσμος αλληλεπιδρά με το αντικείμενο, ανακτώντας και τροποποιώντας στοιχεία της κατάστασής του.



Αντικειμενοστρεφής ανάλυση

- Η συμπεριφορά κοινοποιείται στη *διεπαφή* (interface) του αντικειμένου.
 - Μόνο τα ονόματα, οι παράμετροι και οι τύποι αποτελέσματος των λειτουργιών κοινοποιούνται. Οι υλοποιήσεις αποκρύπτονται.
- Τα αντικείμενα αλληλεπιδρούν στέλνοντας μεταξύ τους *μηνύματα* που ουσιαστικά είναι αιτήματα εκτέλεσης λειτουργιών. Ένα μήνυμα μπορεί:
 - Να συνοδεύεται από παραμέτρους
 - Να επιστρέφει αποτέλεσμα



Κύριο εργαλείο αντικειμενοστρεφούς ανάλυσης

- Unified Modeling Language (UML): γλώσσα μοντελοποίησης, μας επιτρέπει να χτίζουμε μοντέλα λογισμικού για να διαχειριστούμε την πολυπλοκότητά του
 - αποτελεί εκ των πραγμάτων (de facto) πρότυπο στη βιομηχανία λογισμικού
 - Είναι ενοποιημένη γιατί συγκεράζει προσεγγίσεις και συμβολισμούς
 - Είναι *γραφική* παρέχοντας διαγράμματα για την περιγραφή του συστήματος
 - Τα οποία συμπληρώνονται από περιγραφές σε μορφή κειμένου
 - Τρέχουσα έκδοση: UML 2.x
- Καλύπτει τους εξής τομείς του λογισμικού:
 - δομή του συστήματος
 - συμπεριφορά του συστήματος



Διαγράμματα της UML

- 13 συνολικά
 - 6 από αυτά περιγράφουν τη δομή του συστήματος (δομικά ή στατικά διαγράμματα)
 - τα υπόλοιπα 7 περιγράφουν τη συμπεριφορά του συστήματος (διαγράμματα συμπεριφοράς ή δυναμικά διαγράμματα)

Κατηγορία	Διάγραμμα
Δομικά	Κλάσεων (class diagrams)
	Αντικειμένων (object diagrams)
	Συστατικών (component diagrams)
	Θέσης σε λειτουργία – εγκατάστασης (deployment)
	Σύνθετης δομής (composite structure diagrams)
	Πακέτων (package diagrams)
Συμπεριφοράς	Περιπτώσεων χρήσης (use case diagrams)
	Μηχανής καταστάσεων (state machine)
	Δραστηριότητας (activity diagrams)
	Επικοινωνίας (communication diagrams)
	Ακολουθίας (sequence diagrams)
	Επισκόπησης αλληλεπίδρασης (interaction overview)
	Χρονισμού (timing)

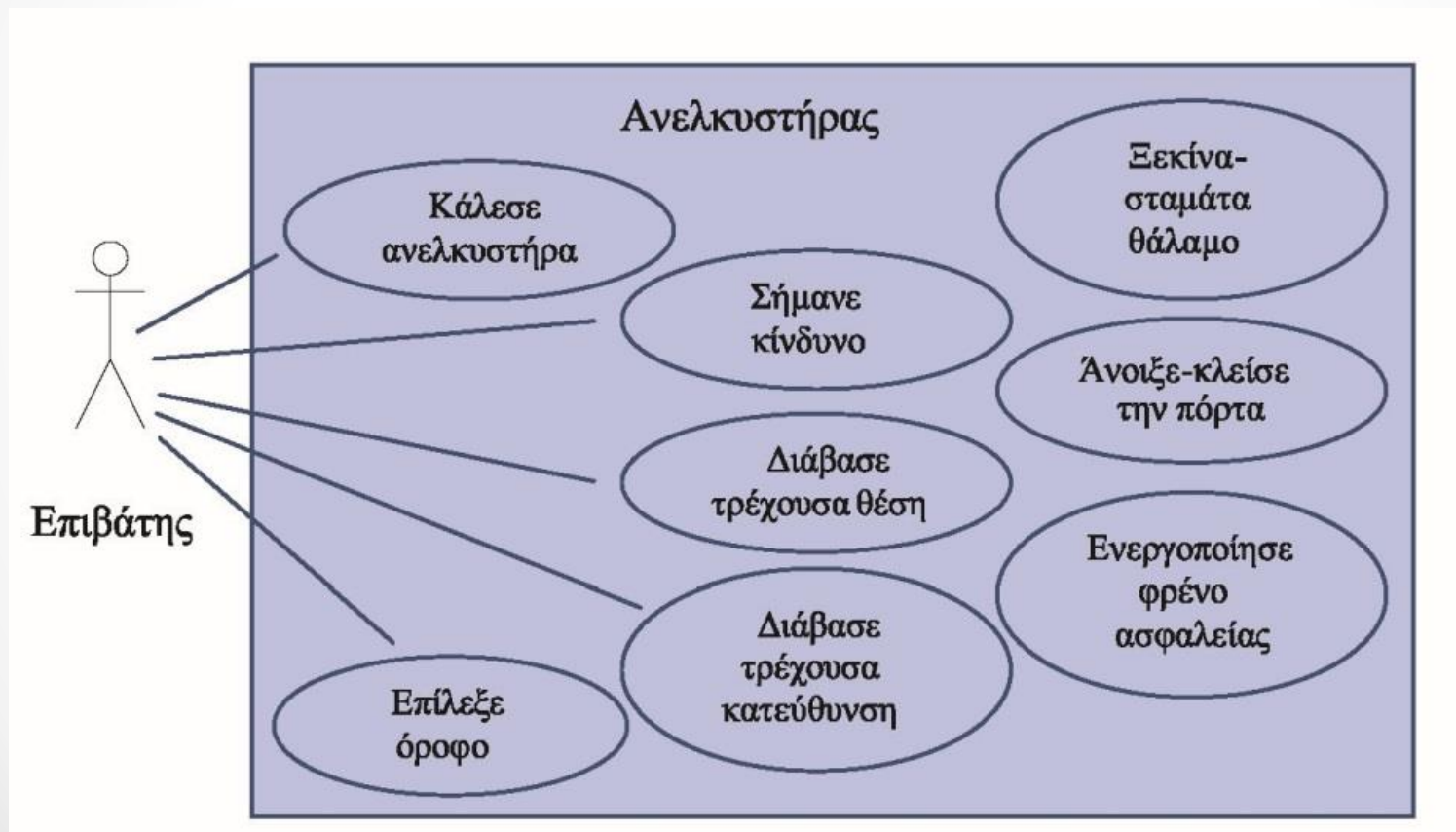
Βασική διαδικασία αντικειμενοστρεφούς ανάλυσης

- Αρχικά προσδιορίζουμε τις ομάδες χρηστών
 - Αυτό περιλαμβάνει ανθρώπους και εξωτερικά συστήματα
 - Οι χρήστες ονομάζονται *actors*
- Για κάθε ομάδα χρηστών αποτυπώνουμε τη λειτουργικότητα που έχει στη διάθεσή της
 - Οι λειτουργίες που είναι διαθέσιμες ονομάζονται στη UML *περιπτώσεις χρήσης (use cases)*
 - Δημιουργείται το *διάγραμμα περιπτώσεων χρήσης (use case diagram)* που περιλαμβάνει όλες τις περιπτώσεις χρήσης
 - Έτσι το διάγραμμα περιπτώσεων χρήσης καθορίζει τα *όρια του συστήματος*: ό,τι αποτυπώνεται ως περίπτωση χρήσης πρέπει να υλοποιηθεί εντός του συστήματος, ενώ πράγματα που δεν αποτυπώνονται θεωρούνται εκτός συστήματος



Παράδειγμα λειτουργίας ανεγκυστήρα

- Actors: επιβάτες, συντηρητές, ομάδες διάσωσης (π.χ. πυροσβέστες)
- Διάγραμμα περιπτώσεων χρήσης για τον actor «επιβάτης ανεγκυστήρα»



Παράδειγμα λειτουργίας ανεγκυστήρα

- Για κάθε περίπτωση χρήσης αποτυπώνονται σε κείμενο:
 - Το όνομά της
 - Ο actor/οι actors που συμμετέχουν
 - Ποιες είναι οι προϋποθέσεις για την εκκίνηση της περίπτωσης χρήσης (συνθήκες εισόδου)
 - Ποια είναι τα αποτελέσματα από την εκτέλεση της περίπτωσης χρήσης (συνθήκες εξόδου)
 - Ποια είναι τα βήματα αλληλεπίδρασης χρήστη-συστήματος για την ολοκλήρωση της περίπτωσης χρήσης
 - Τυχόν απαιτήσεις πραγματικού χρόνου



Ανάδειξη αντικειμένων

- Το επόμενο βήμα είναι να αναδειχθούν οι τύποι αντικειμένων του αντικειμενοστρεφούς μοντέλου που θα απαρτίσουν το σύστημα
 - Αυτό γίνεται με μελέτη των περιπτώσεων χρήσης
- Η αποτύπωση γίνεται στο *διάγραμμα κλάσεων* που περιλαμβάνει τις κλάσεις και τις συσχετίσεις μεταξύ τους
- Κλάση: ένας τύπος αντικειμένων που μοντελοποιεί αντικείμενα με παρόμοια δομή και συμπεριφορά
 - Κάθε κλάση απεικονίζεται ως παραλληλόγραμμο με τρία τμήματα
 - Το πάνω τμήμα δίνει το όνομα της κλάσης
 - Το μεσαίο τμήμα τα χαρακτηριστικά της (attributes)
 - Το χαμηλότερο τμήμα τη συμπεριφορά της
 - Τυπικά, αρχικά σημειώνεται μόνο το όνομα, κατόπιν συμπληρώνεται η συμπεριφορά και τέλος τα χαρακτηριστικά της

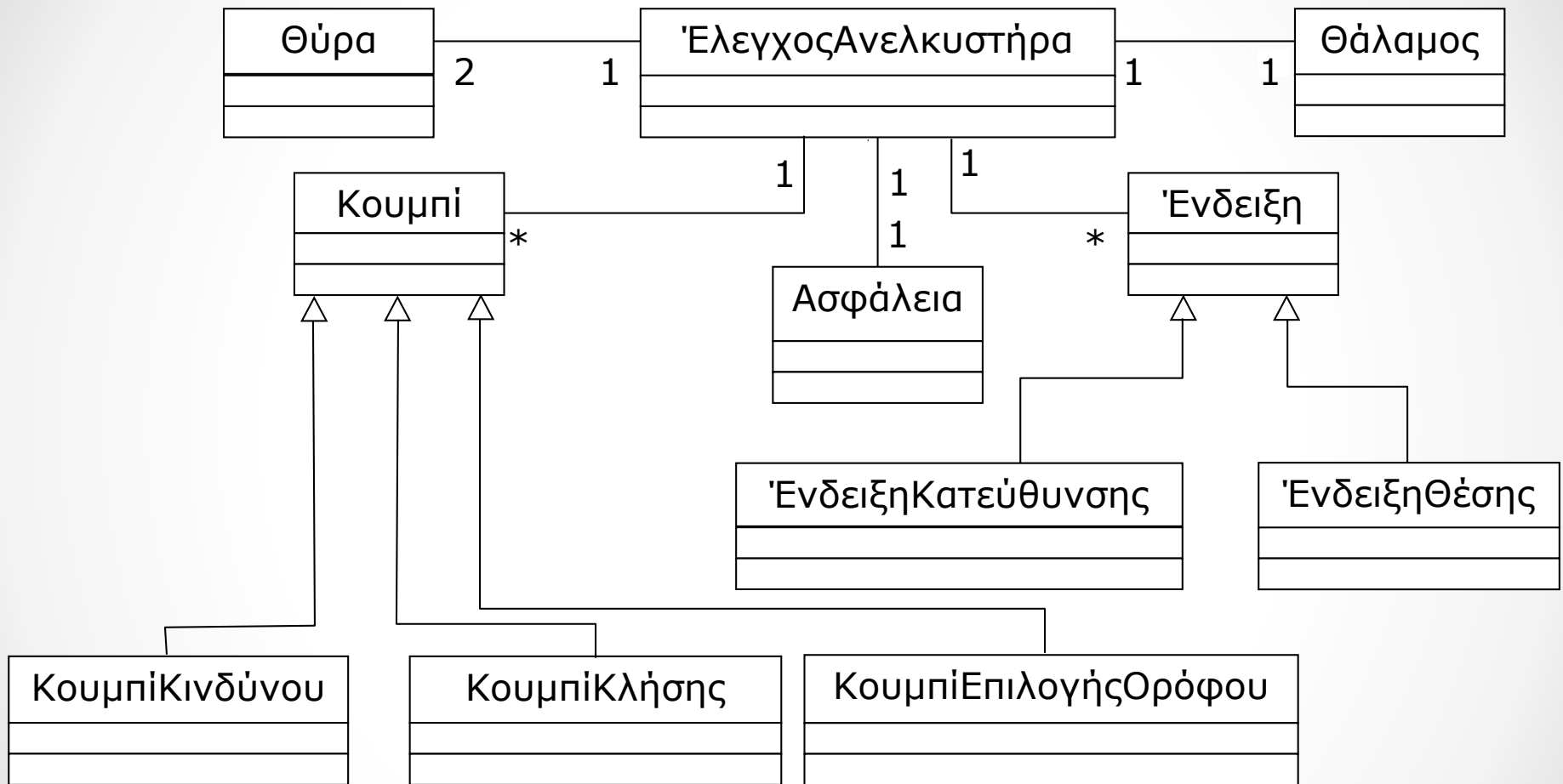


Συσχετίσεις

- Όταν δύο κλάσεις συνδέονται μεταξύ τους, σημειώνουμε στο διάγραμμα κλάσεων μία συσχέτιση μεταξύ τους
 - Η συσχέτιση σημειώνεται με μία γραμμή που ενώνει δύο παραλληλόγραμμα
 - Η συσχέτιση έχει ένα λεκτικό που δίνει τη σημασιολογία της και μπορεί να αναγράφεται και η *πληθικότητά της*
 - πόσα αντικείμενα από την κάθε κλάση μπορούν να σχετίζονται με αντικείμενα της άλλης
 - Υπάρχουν ειδικές συσχετίσεις που αναπαριστούν γενίκευση/εξειδίκευση καθώς και σχέση μέρους-όλου



Διάγραμμα κλάσεων για έλεγχο ανελκυστήρα



↑ Σχέση γενίκευσης/εξειδίκευσης



Διάγραμμα μηχανής καταστάσεων

- Το διάγραμμα μηχανής καταστάσεων μοντελοποιεί πρακτικά μία μηχανή καταστάσεων
- Χρησιμοποιούνται για να απεικονιστεί η συμπεριφορά αντικειμένων που έχουν πολλαπλές διακριτές καταστάσεις και μεταπίπτουν μεταξύ των καταστάσεων ανταποκρινόμενα σε εξωτερικά συμβάντα/ερεθίσματα
- Βασικά συστατικά:
 - Αρχική κατάσταση
 - Τελική κατάσταση
 - Ενδιάμεσες καταστάσεις
 - Μεταβάσεις μεταξύ καταστάσεων
 - Η κάθε μετάβαση μπορεί να έχει *συνθήκη* υπό την οποία πραγματοποιείται



Παράδειγμα διαγράμματος μηχανής καταστάσεων

- Οι μεταβάσεις που αφορούν το *μοτέρ του ανελκυστήρα* δίνονται στο ακόλουθο διάγραμμα

