

DIS501 ΑΝΑΛΥΣΗ ΚΑΙ ΣΧΕΔΙΑΣΗ ΠΛΗΡΟΦΟΡΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

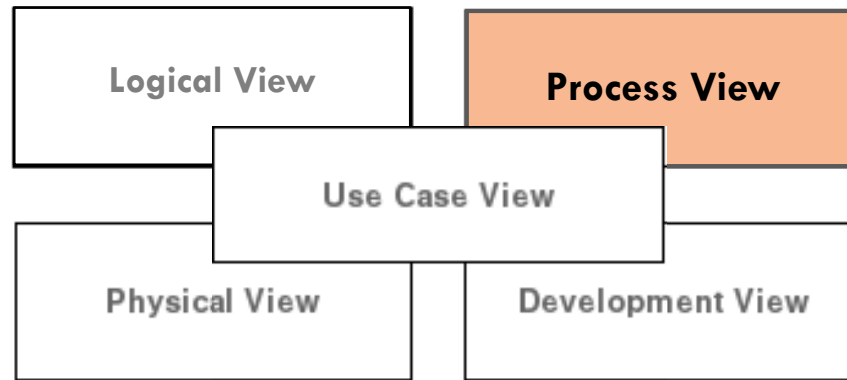
Lecture 9: Διαγράμματα κατάστασης (State transition diagrams/State charts)

By: Δρ. Αυγούστα Κυριακίδου-Ζαχαρούδιου

Μαθησιακά αποτελέσματα

- Εμβαθύνση στη δυναμική μοντελοποίηση (dynamic modelling)
- Εισαγωγή στις έννοιες της μηχανής κατάστασης (state machine concepts)
- Κατανόηση αντικειμένων και καταστάσεων (objects and states)
- Διαγράμματα κατάστασης αντικειμένων και σημειογραφία της UML
- Κατασκευή διαγραμμάτων κατάστασης αντικειμένων (state charts)

Μαθησιακός έλεγχος (learning check)



- **Δυναμικά διαγράμματα (Dynamic modelling)**
 - Μια σειρά διαγραμμάτων που περιγράφουν τη δυναμική συμπεριφορά ενός αντικειμενοστραφούς συστήματος.
 - Διαγράμματα ακολουθίας αντικειμένων (object sequence diagrams), μετάβασης καταστάσεων (state transition diagrams) και δραστηριοτήτων (activity diagrams)

Object Model vs Dynamic Model

❑ Μοντέλο αντικειμένων (Object model)

- Ενθυλάκωση (encapsulation) δεδομένων και μεθόδων μέσα σε κλάσεις/αντικείμενα
- Σχέσεις μεταξύ κλάσεων/αντικειμένων
- Ένα "στιγμιότυπο" σε μία μόνο χρονική στιγμή

❑ Δυναμικό μοντέλο

- Οι πτυχές ενός συστήματος που αφορούν το χρόνο και τις αλλαγές συνθέτουν το *δυναμικό* μοντέλο
- Πώς τα αντικείμενα συμπεριφέρονται και συνεργάζονται για μια συγκεκριμένη περίπτωση χρήσης (use case) (Object sequence diagrams)
- Αλλαγές σε αντικείμενα με την πάροδο του χρόνου και σε πολλές περιπτώσεις χρήσης (state transition diagrams)

**ΑΛΛΑ... ΤΙ ΠΡΟΚΑΛΕΙ ΈΝΑ
ΑΝΤΙΚΕΙΜΕΝΟ ΝΑ ΑΛΛΑΞΕΙ Η
ΝΑ ΔΡΑΣΕΙ;**

**BUT... WHAT TRIGGERS AN
OBJECT TO CHANGE OR ACT?**



Διαγράμματα κατάστασης

- ❑ Ορισμένα αντικείμενα μπορούν να βρεθούν σε ένα διακριτό σύνολο καταστάσεων
 - κάθε κατάσταση επηρεάζει την συμπεριφορά τους.
- ❑ Τα διαγράμματα κατάστασης μοντελοποιούν τη συμπεριφορά του συστήματος ως απόκριση σε εξωτερικά και εσωτερικά γεγονότα
- ❑ Εκφράζουν
 - Τις καταστάσεις του συστήματος ως κόμβους
 - Τις ενέργειες ως ακμές μεταξύ των κόμβων
- ❑ Όταν συμβεί κάποια ενέργεια το σύστημα (και επομένως τα αντικείμενα/κλάσεις του συστήματος) μεταβαίνουν από μια κατάσταση σε μια άλλη

1. Διάρκεια ζωής και κύκλοι ζωής: .

- Μια παραγγελία (Purchase Order) ξεκινά τη ζωή της ως έγγραφο αίτησης, εγκρίνεται σε διάφορα επίπεδα, αποστέλλεται σε έναν προμηθευτή, εκτελείται με την παραλαβή των αγαθών και, τέλος, αρχειοθετείται για μελλοντική αναφορά.
- Κάθε φορά που κάτι συμβαίνει στο αντικείμενο και κάτι αλλάζει σε αυτό, λέμε ότι τώρα το αντικείμενο υπάρχει σε διαφορετική κατάσταση (State).
- Μπορούμε επίσης να πούμε ότι κάθε φορά που το χαρακτηριστικό ενός αντικειμένου αλλάζει την τιμή του (an object's attribute changes its value), το αντικείμενο υπάρχει πλέον σε μια νέα κατάσταση (state).
- Ορίζουμε ως μετάβαση κατάστασης (state transition) την πράξη της αλλαγής από μια κατάσταση σε μια άλλη.

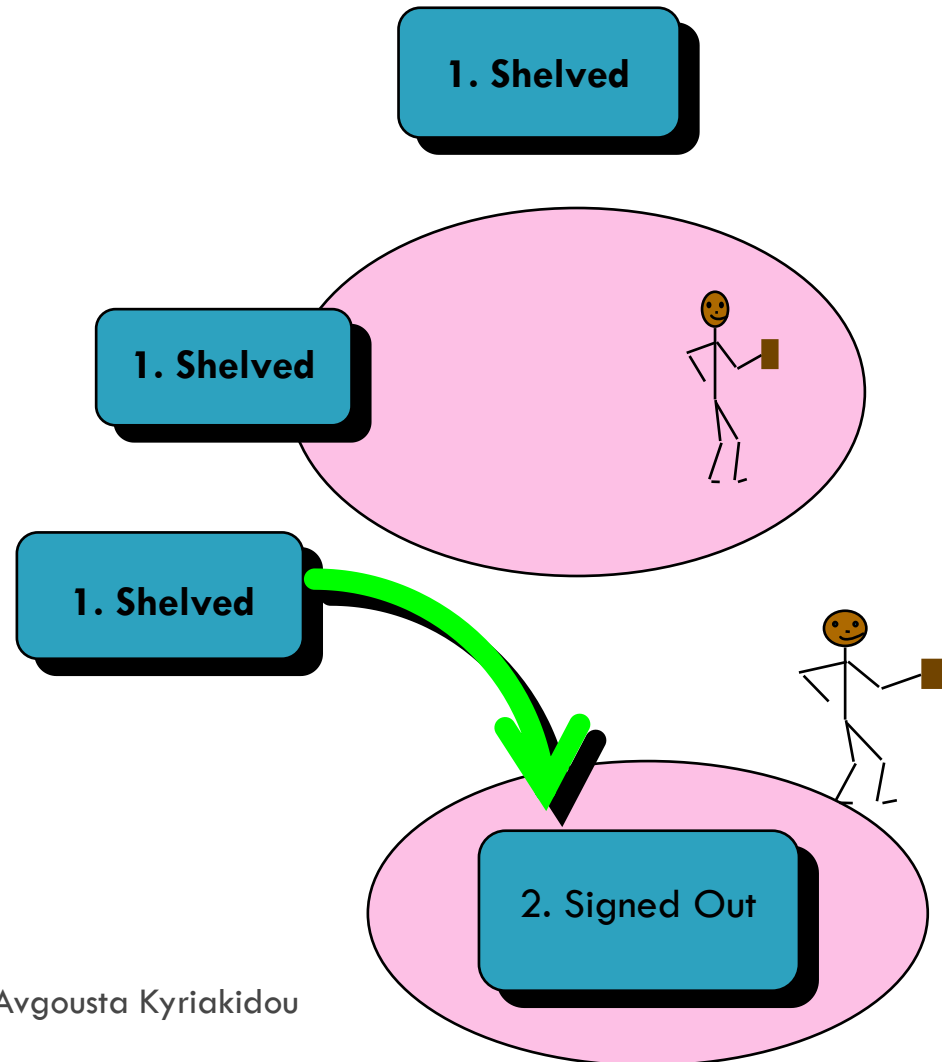
Ακολουθεί ένα παράδειγμα χρησιμοποιώντας ένα βιβλίο της βιβλιοθήκης . . .

1. Lifetimes and Lifecycles: Example

Το τυπικό βιβλίο (book) της
βιβλιοθήκης

Ξεκινά την ημέρα του στο ράφι
(being shelved). .

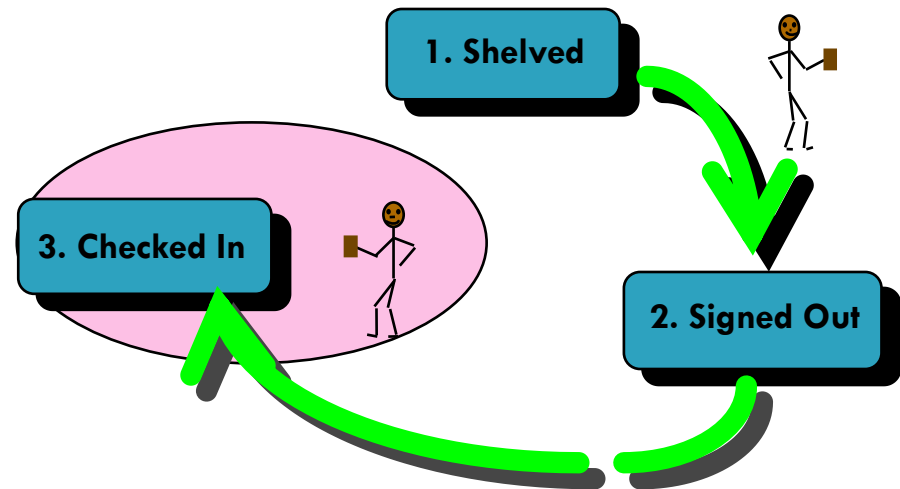
Όταν ένας Συνδρομητής έρθει και
δανειστεί το Βιβλίο, η κατάσταση
(State) του Βιβλίου αλλάζει σε
"Signed Out".



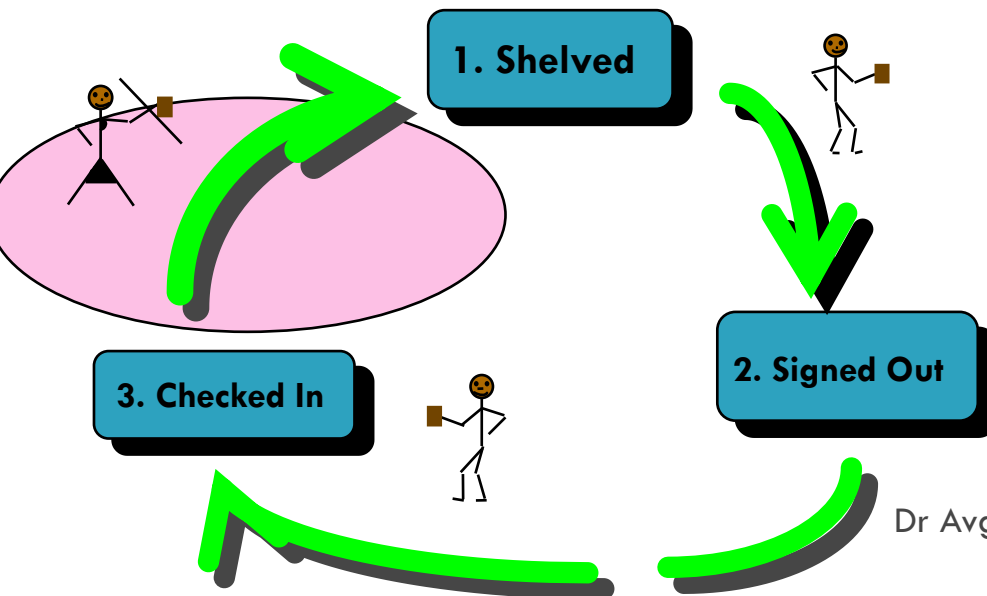
Όταν ο συνδρομητής το επιστρέψει (When the Subscriber brings it back. . .)

Όταν ο συνδρομητής το φέρει πίσω. . .

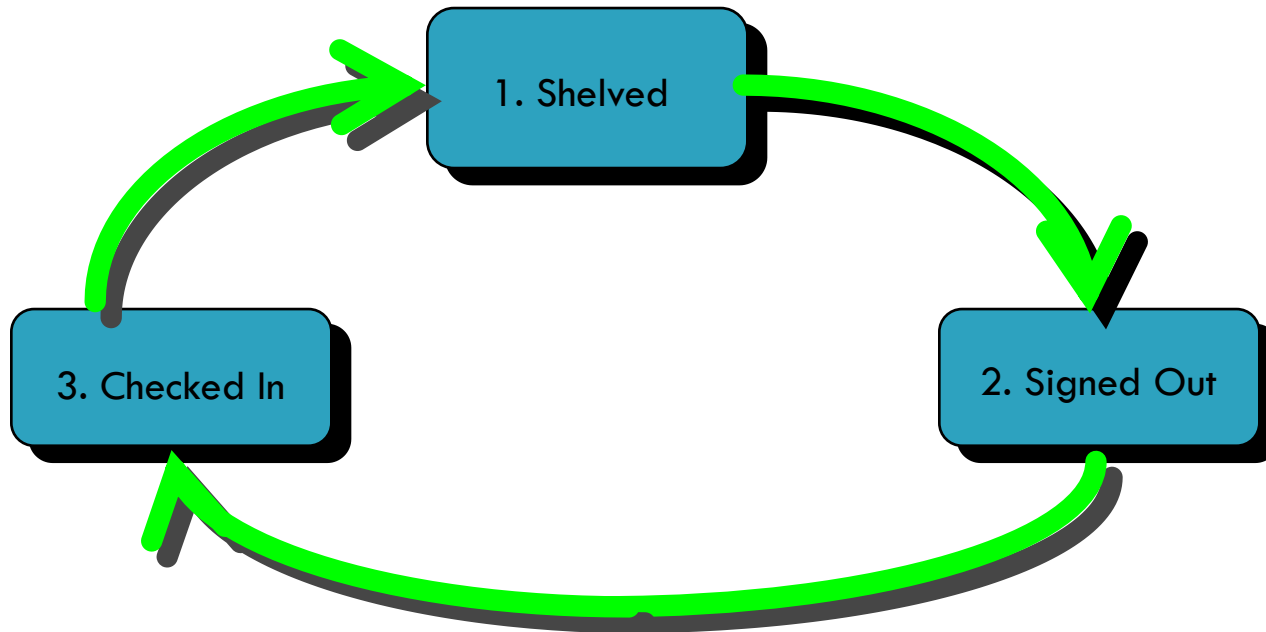
Η κατάσταση (state) του βιβλίου αλλάζει σε "Checked In".



Και όταν ο βιβλιοθηκάριος το βάζει πίσω στο ράφι, η κατάσταση (state) του βιβλίου αλλάζει και πάλι σε "Shelved." (πίσω στο ράφι)



State transition diagram - Διάγραμμα μετάβασης κατάστασης



- Ένα τέτοιο Loop ονομάζεται **λειτουργικός κύκλος ζωής (Operational life cycle)**
- Πολλά αντικείμενα έχουν μια ζωή που **δεν επαναλαμβάνεται**, αλλά περνάει από στάδιο σε στάδιο, από τη "γέννηση - birth" μέχρι το "θάνατο - death" (δηλαδή τη διαγραφή, την αρχειοθέτηση ή οτιδήποτε άλλο).
- Αυτό το είδος ονομάζεται **Lifetime**

2. Events and states

Όλα τα αντικείμενα έχουν μια κατάσταση.

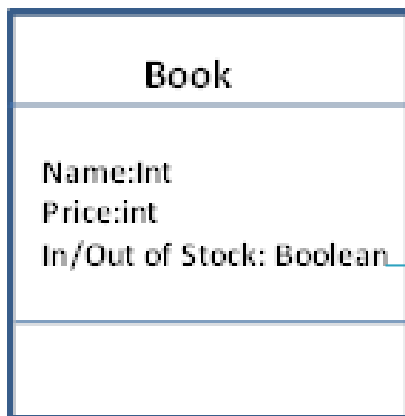
Εμείς το λέμε αυτό:

- Ένα **event** (συμβάν) **λαμβάνεται από** ένα αντικείμενο, το οποίο στην πραγματικότητα σημαίνει ότι το συμβάν αυτό **διεγείρει** αυτό το αντικείμενο και επιφέρει αλλαγές.
- Και μόλις του συμβεί αυτό το γεγονός, λέμε γενικά ότι αυτό το αντικείμενο βρίσκεται σε **μια νέα κατάσταση (new state)**.
- Η απάντηση ενός αντικειμένου σε ένα **event** μπορεί να οδηγήσει σε άλλα πράγματα εκτός από την αλλαγή της κατάστασής του και αυτό συνήθως εξαρτάται από την ήδη υπάρχουσα κατάστασή του:
 - ▣ Δημιουργήσει ένα νέο **event** και να το στείλει πίσω στον αποστολέα
 - ▣ Δημιουργήσει ένα νέο **event** και να το προωθήσει σε ένα τρίτο αντικείμενο
- **Μια κατάσταση (state)** περιγράφει μια συγκεκριμένη κατάσταση την οποία μπορεί να καταλαμβάνει ένα αντικείμενο για ένα χρονικό διάστημα, ενώ περιμένει κάποιο γεγονός ή έναυσμα.

2. Events and states (συνέχεια...)

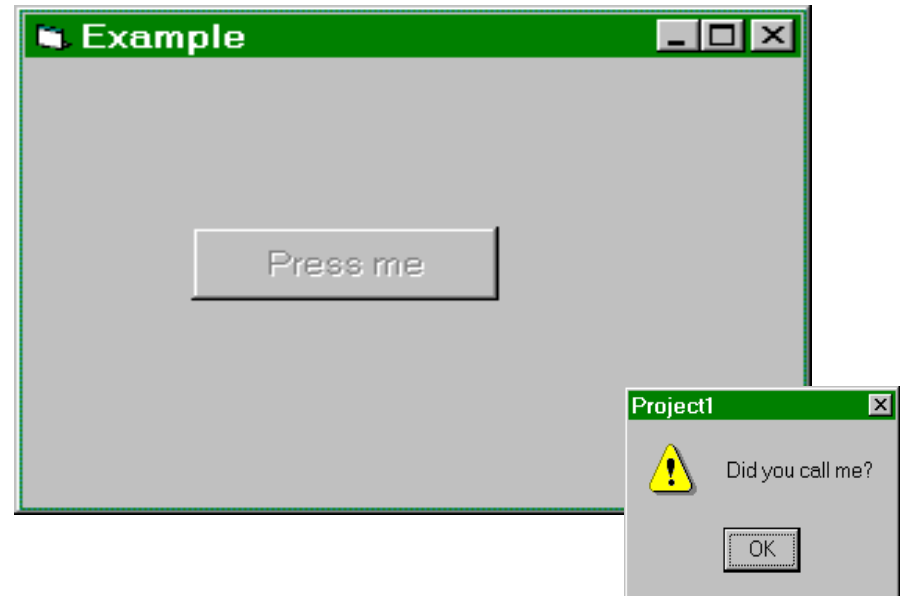
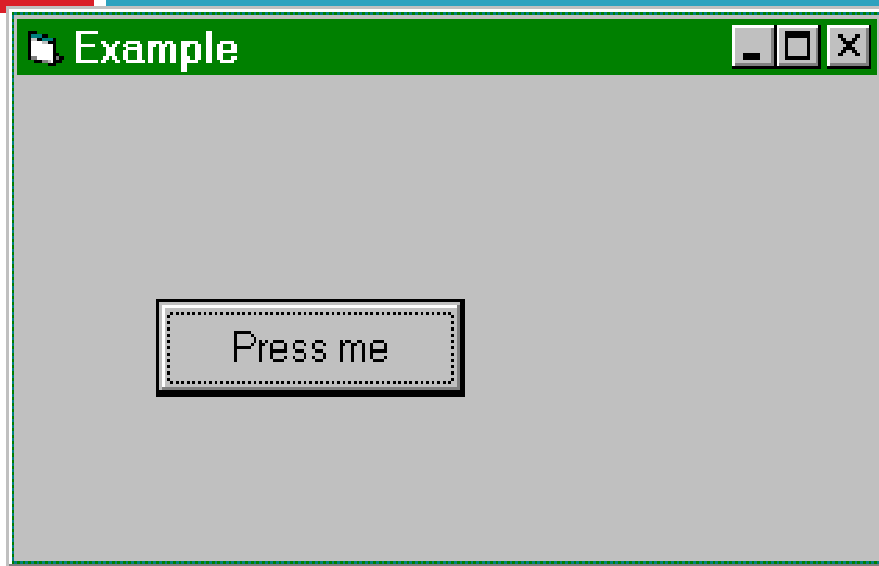
Γενικά,

- Ορισμένα από τα χαρακτηριστικά (attributes) του αντικειμένου σχετίζονται με την κατάστασή (state) του, ενώ άλλα όχι.
- Για να καταγράψουμε ένα επιχειρηματικό συμβάν, πρέπει να **αλλάξουμε την τιμή τουλάχιστον ενός χαρακτηριστικού του αντικειμένου.** (*change the value of at least one attribute of that object.*)
- Δεδομένου ότι οι τιμές των χαρακτηριστικών μπορούν να θεωρηθούν ότι **αντιπροσωπεύουν** την κατάσταση του αντικειμένου, αυτό προκαλεί μια **μετάβαση κατάστασης (state transition).**
- State transition: μετάβαση από μια κατάσταση σε μια άλλη.



Λαμβάνουμε κυρίως υπόψη τις τιμές **Boolean** των χαρακτηριστικών. Αυτές είναι εκείνες που μπορούν να αλλάξουν την κατάσταση ενός αντικειμένου. Ωστόσο, πρέπει να λάβουμε υπόψη μας και τα άλλα χαρακτηριστικά, ώστε να ξέρουμε τι βιβλίο θέλουμε.

Παράδειγμα "Πριν" και "Μετά"



- Αλλαγή κατάστασης
- Δημιουργεί event σε άλλο αντικείμενο

Παράδειγμα: Πιθανές καταστάσεις του space shuttle



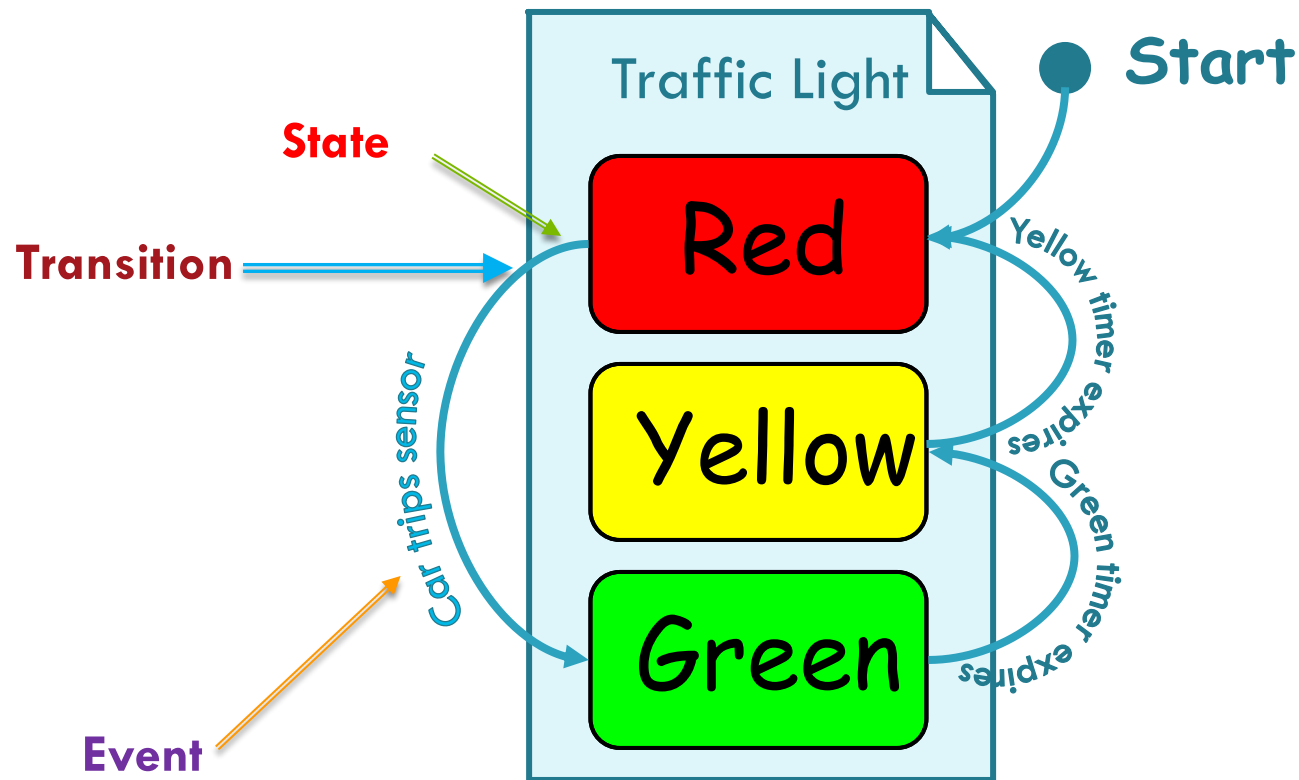
Κατάσταση
"Pre-launch"

Κατάσταση
"Πτήση"

Event που ενεργοποιεί το αντικείμενο space shuttle το οποίο βρίσκεται στην **κατάσταση pre-launch** να αλλάξει κατάσταση και ενώ βρίσκεται στον αέρα να βρεθεί σε **κατάσταση Flight**.

Θα μπορούσαμε να εμφανίσουμε πρόσθετες καταστάσεις όπως **Landed, Checkout κ.λπ. εάν το όριζαν οι απαιτήσεις.

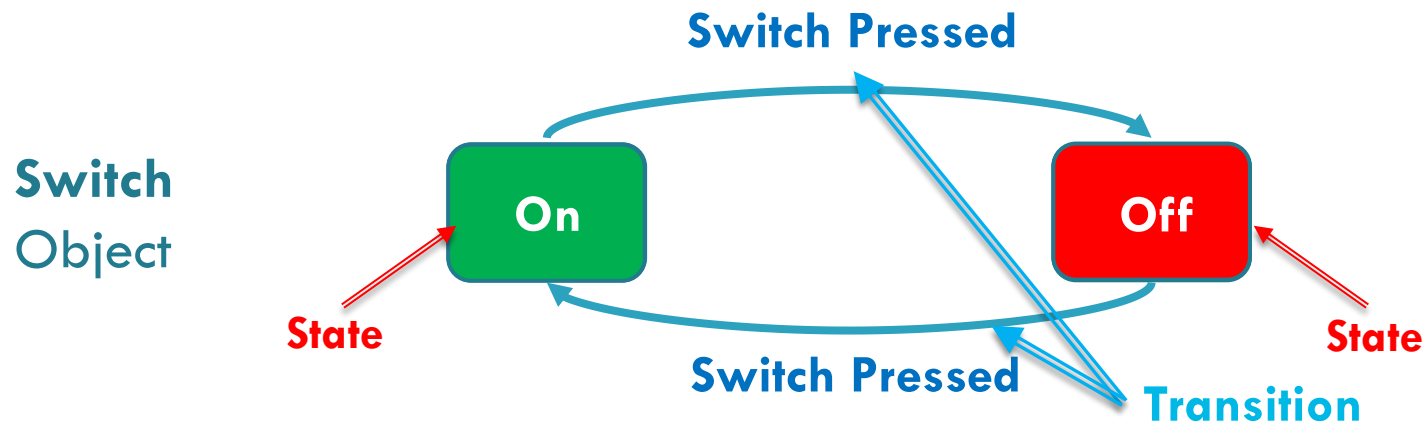
Example: states of a Traffic Light Object



Dynamic Modelling

State transition diagram

- Περιγράφει τη δυναμική συμπεριφορά ενός αντικειμένου
- Δείχνει τις *καταστάσεις (states)* και *μεταβάσεις (state transitions)* από μια κατάσταση σε άλλη.
- Δημιουργούμε ένα state transition diagram για κάθε αντικείμενο (object) με σημαντική δυναμική συμπεριφορά



State transition diagrams

- Διαγράμματα μηχανών καταστάσεων (aka state charts / finite state / state transition / state diagrams)
- Εστίαση στις αλλαγές κατάστασης αντικειμένων
- Μοντελοποιεί την σειρά των καταστάσεων (states) στις οποίες βρίσκεται ένα αντικείμενο κατά την διάρκεια του κύκλου ζωής του σαν απάντηση σε events, μαζί με τις αντιδράσεις και τις ενέργειες του
- Βοηθά να αναγνωρίσουμε όλες τις πιθανες συμπεριφορες ενός αντικειμένου.
- Το διάγραμμα περιλαμβάνει:
 - Το σύνολο καταστάσεων (states) (συμπεριλαμβανομένης μιας αρχικής κατάστασης εκκίνησης)
 - Μεταβάσεις (state transitions) μεταξύ καταστάσεων

**Απεικονίζει τη ροή ελέγχου του λογισμικού χρησιμοποιώντας
καταστάσεις και μεταβάσεις**

State chart/transition Diagrams

Which objects do we model?

- **Κάθε αντικείμενο/κλάση έχει καταστάσεις (states), αλλά δεν πρέπει να μοντελοποιείται κάθε αντικείμενο/κλάση με ένα διάγραμμα καταστάσεων.**
 - Τα state transition diagrams πρέπει να χρησιμοποιούνται μόνο για αντικείμενα (objects) με σημαντική δυναμική συμπεριφορά (dynamic behaviour).
 - δηλαδή, κλάσεις με δύο ή περισσότερες πιθανές καταστάσεις (States) κατά τη διάρκεια της δραστηριότητας του συστήματος
- **!!Χρησιμοποιήστε το Object sequence diagram για να βρείτε κλάσεις (classes) που έχουν περισσότερες από 2 καταστάσεις (states). Συνήθως αυτές που έχουν μηνύματα που εισέρχονται και εξέρχονται.**

OBJECT SEQUENCE VS STATE TRANSITION DIAGRAMS

- Υπάρχει ένας σημαντικός σύνδεσμος μεταξύ των object sequence diagrams και των state transition diagrams (state charts).
- Ένα **state transition diagram (state chart)** αποτυπώνει όλες τις πιθανές συμπεριφορές και καταστάσεις στις οποίες μπορεί να βρίσκεται το αντικείμενο κατά την διάρκεια της ζωής του και επομένως σε όλες τις περιπτώσεις χρήσης στις οποίες εμπλέκεται.
- Αντίθετα, ένα **object sequence diagram** αποτυπώνει διαφορετικά αντικείμενα που συνεργάζονται και αλληλεπιδρούν για να επιτευχθεί ένα συγκεκριμένο use case → Δείχνει συνεργασία
- Ένα **state transition diagram (state chart)** μπορεί να θεωρηθεί ως μια περιγραφή όλων των πιθανών κύκλων ζωής που μπορεί να ακολουθήσει ένα αντικείμενο μιας κλάσης.
- Μπορεί επίσης να θεωρηθεί ως μια πιο λεπτομερής παρουσίαση μιας κλάσης.

When should we use State chart/transition Diagrams (SCD)

- Χρησιμοποιήστε ένα **state transition diagram (state chart)** για να παρουσιάσετε τη συμπεριφορά ενός αντικειμένου μέσω πολλών περιπτώσεων χρήσης του συστήματος.
- Χρησιμοποιήστε **state transition diagrams** μόνο για κλάσεις (classes) όπου είναι απαραίτητο να κατανοήσετε τη συμπεριφορά του αντικειμένου σε ολόκληρο το σύστημα.
 - ▣ τα διαγράμματα κατάστασης δεν είναι χρήσιμα για την περιγραφή της συνεργασίας όλων των αντικειμένων σε μια περίπτωση χρήσης.
 - Γιατί;
 - Αυτό αποτελεί μέρος του object sequence diagram και όχι του state transition diagram.

3. Object State transition Diagrams

Key constructs

□ Objects

ένα state transition diagram ανά αντικείμενο/κλάση

□ States

- Η κατάσταση (State) στην οποία βρίσκεται το αντικείμενο κάθε στιγμή
- Εννοιολογική περιγραφή των δεδομένων (attributes) του αντικειμένου - **αναπαρίσταται από τις τιμές των boolean χαρακτηριστικών του αντικειμένου**
- Internal Activities: each state can include activities inside it

□ Events

Γεγονότα που ενεργοποιούν τη μετάβαση από μια κατάσταση στην άλλη. Η μετάβαση γίνεται όταν ικανοποιείται και κάποια προαιρετική συνθήκη (Guard Condition)

□ Transition

Μετάβαση μεταξύ καταστάσεων (from one state to another)

3. State Transition Diagram: Basic Notations

□ States

▣ Initial State (σημείο εκκίνησης) ●

▣ Final State (σημείο λήξης) ⬢

▣ State

State Name

Activity: Κάθε state μπορεί να περιλαμβάνει ένα activity μέσα του. Τα **activities** είναι **methods/operations του αντικειμένου/κλάσης** τα οποία ενεργοποιούνται (invoked) κατά την είσοδο στην κατάσταση, κατά την έξοδο από την κατάσταση και κατά την διάρκεια που το αντικείμενο βρίσκεται σε μια κατάσταση.

Μέσα στην κατάσταση αναφέρουμε τι ξεκινάει το activity, τι το σταματάει και τι γίνεται κατά την διάρκεια.

Entry/

Exit/

Do/

State Name

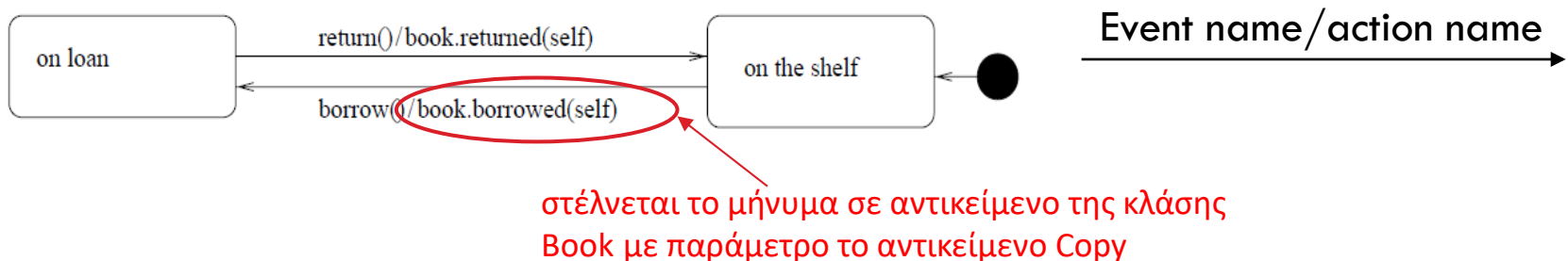
Entry/activity expression

Exit/activity expression

Do/activity expression

3. State Transition Diagram: Basic Notations

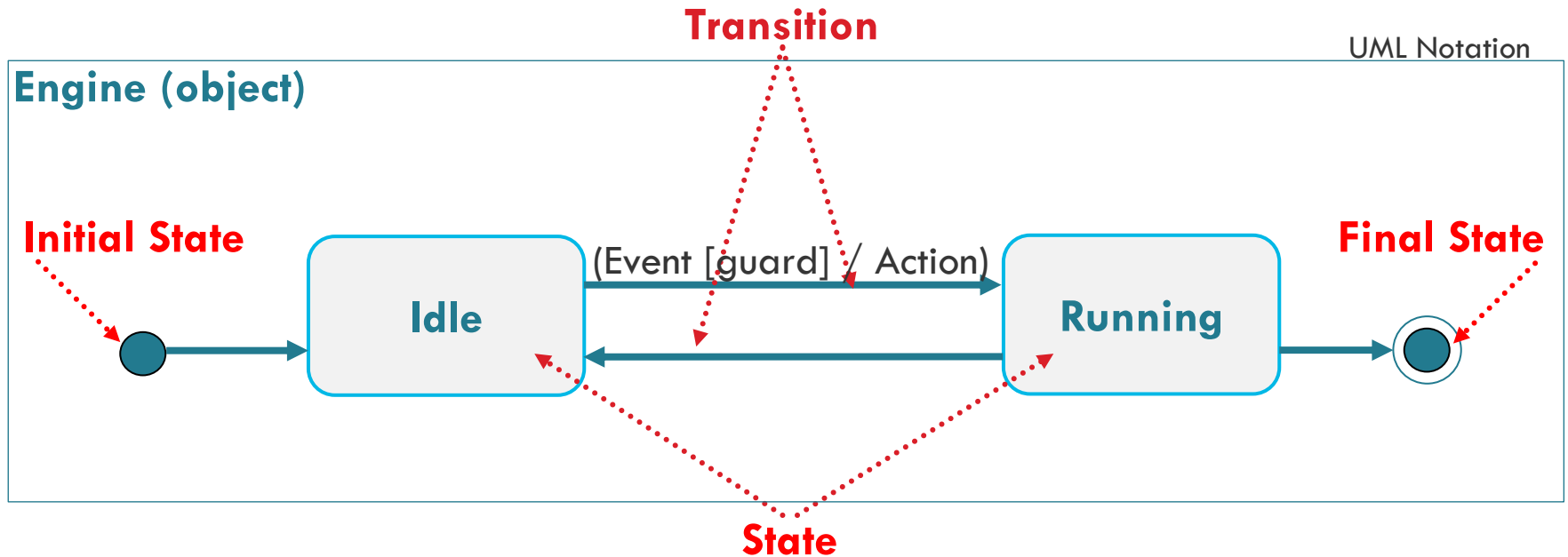
- **Transitions** (μετάβαση από ένα state στο άλλο)
 - ▣ Events προκαλούν αυτά τα transitions
- **Actions:** πάνω στα transitions, δίπλα από το event name, συνήθως ενεργοποιούν μεθόδους ενός αντικειμένου. Δείχνουν την αντίδραση του αντικειμένου σε ένα συμβάν (event)



- **Guard Conditions:** μια έκφραση boolean που αξιολογείται τη στιγμή που πυροδοτείται η ενεργοποίηση ενός event. Αν η έκφραση είναι true, τότε το transition μπορεί να πραγματοποιηθεί.

Object State transition Diagrams

Key constructs

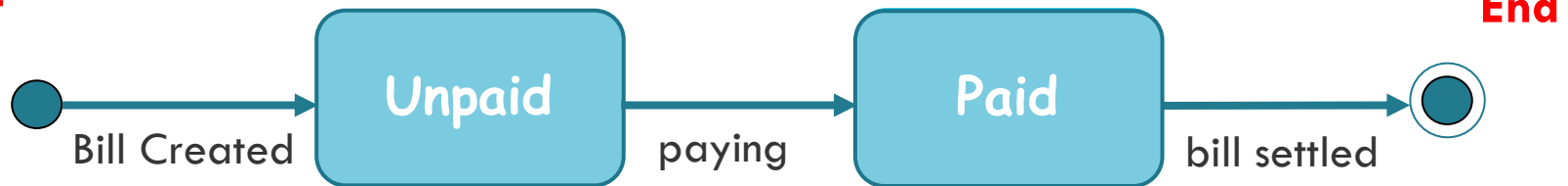


Object State Diagrams

Simple billing example

Bill Object

Start



States

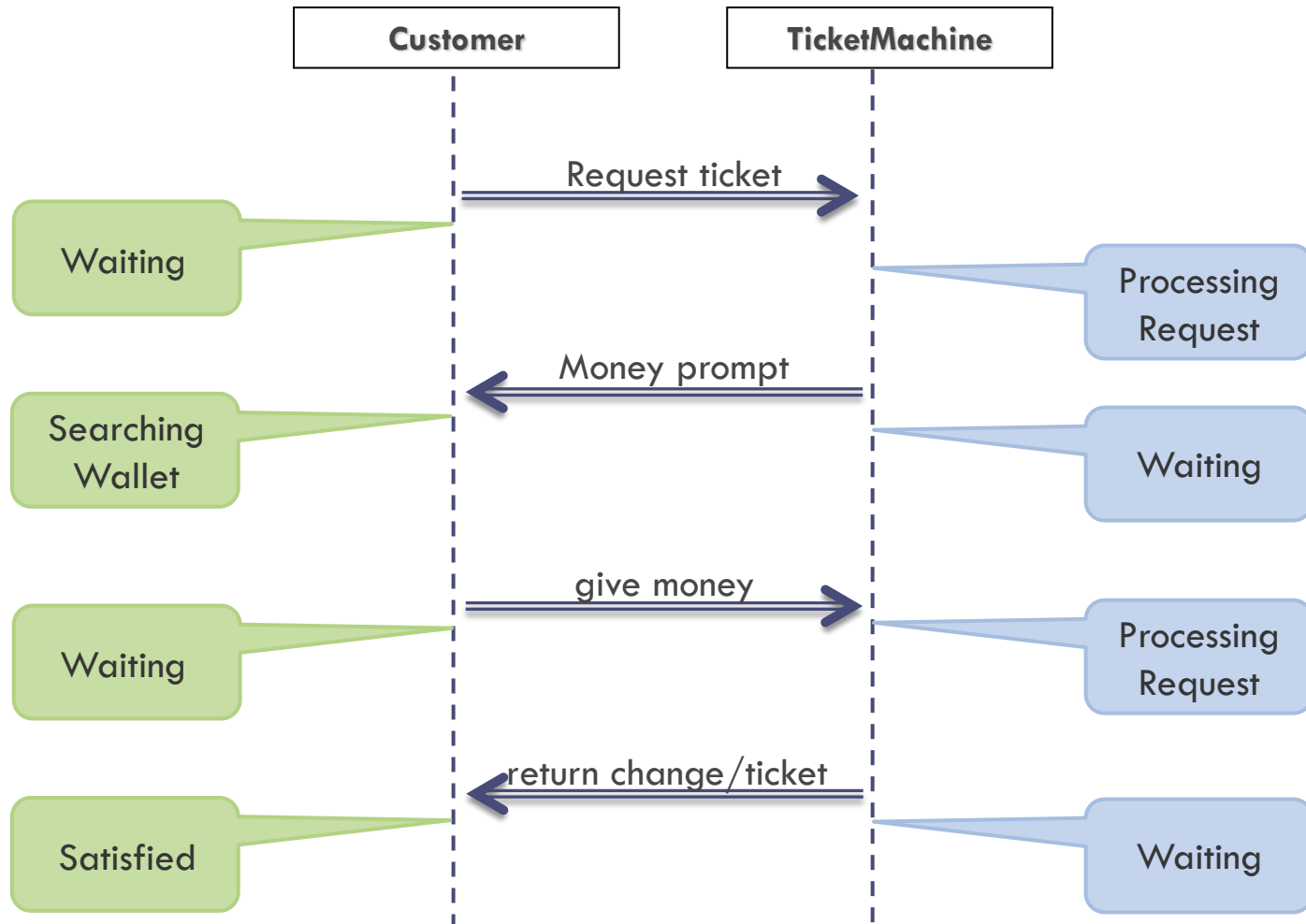
- Μια κατάσταση (state) αντιστοιχεί στο διάστημα μεταξύ δύο γεγονότων (event) που λαμβάνει ένα αντικείμενο.
- Για παράδειγμα ένα **account** είναι **overdrawn** μεταξύ του event το οποίο προκάλεσε το overdrawn και του event το οποίο αντιστοιχεί στην εξόφληση της οφειλής.
- **Τα States βρίσκονται μεταξύ των μηνυμάτων που ανταλλάσσονται μεταξύ των κλάσεων σε ένα Object Sequence diagram.**
- Το state ενός αντικειμένου καθορίζεται από το σύνολο των τιμών οι οποίες είναι φυλαγμένες στα attributes του E.g.
 - Switch is **on**; Bill total **is being calculated**; Invoice is **paid**

States

- Μια κατάσταση μπορεί να αντιστοιχεί/περιγράφει μια συγκεκριμένη ιδιότητα
 - ▣ Π.χ. an account object is in the state of “overdrawn” when the account balance is negative.
- Η κατάσταση ενός αντικειμένου μπορεί να αναπαρασταθεί από μια τιμή.
 - ▣ Π.χ. an engine can be **“on”** or **“off”**.

Events and States

Ticket Machine Scenario



States are between the messages that are exchanged in an Object Sequence diagram between objects.

States

- Τα state Names πρέπει να είναι απλά αλλά περιγραφικά
 - Όχι ουσιαστικό!
 - Όχι ενεργό ρήμα
 - Θα πρέπει να υποδεικνύει μια ενεργή ενέργεια, π.χ In process
 - Examples: Proposed, Full, Closed, Scheduled, On, Off

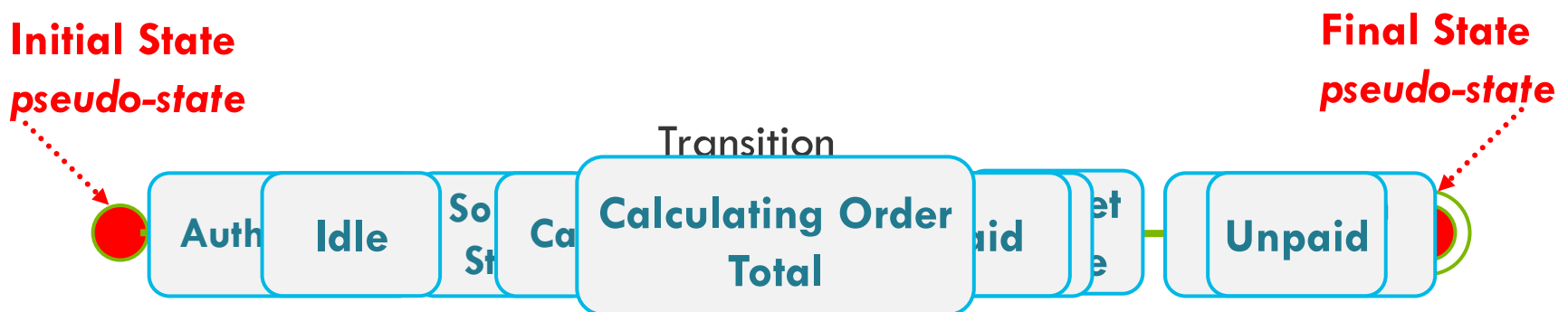
At any moment in time, an object exists in a certain manner or condition, which we say is a state

A state has a duration – a finite time

- Question “Black Hole” States.
 - A black hole state είναι ένα state που έχει transitions προς αυτήν αλλά κανένα να φεύγει προς τα έξω.
- Question “Miracle” States.
 - A miracle state είναι ένα state που έχει transitions αν φεύγουν προς τα έξω αλλά τίποτα προς αυτήν.

State Types

- Dynamic (processing) (e.g. authorising, calculating, dispatching)
- Static (waiting until an event happens) (e.g. idle, wait, paid, unpaid)
- Combined (processing until some event happens) (e.g. calculating order total)
- Initial and Final states (pseudo-states)
- “Black Hole” (a state with inward transitions but none out)
- “Miracle” (a state with outward transitions but none in)



Events

- Κάτι που συμβαίνει σε μια χρονική στιγμή
- Ερέθισμα από το ένα αντικείμενο στο άλλο
 - ▣ Ένα event λαμβάνεται από ένα αντικείμενο, π.χ. το event διεγείρει αυτό το αντικείμενο.
 - ▣ Και προκαλεί την ενεργοποίηση μιας μετάβασης (triggering of a transition)
- Μπορεί να οδηγήσει σε αλλαγή κατάστασης **Ring [ringcount = 6]
setMode(autoAnswer)** 
- Μπορεί να οδηγήσει σε αποστολή ενός event σε άλλο αντικείμενο

Event Types

- **External Event - Εξωτερικό συμβάν (επίσης γνωστό ως συμβάν συστήματος)**
 - ▣ προκαλείται από κάτι εκτός των ορίων του συστήματος
 - ▣ π.χ. όταν ένας ταμίας πατάει το κουμπί "εισαγωγή στοιχείου" σε ένα σύστημα, τότε λέμε ότι έχει συμβεί ένα εξωτερικό συμβάν.

- **Internal Event – Εσωτερικό συμβάν**
 - ▣ προκαλείται από κάτι εντός των ορίων του συστήματός μας.
 - ▣ προκύπτει όταν μια μέθοδος καλείται μέσω ενός μηνύματος που αποστέλλεται από ένα άλλο εσωτερικό αντικείμενο. (Τα μηνύματα στα Object sequenced diagrams υποδηλώνουν εσωτερικά γεγονότα)

- **Χρονικό γεγονός - Temporal Event**
 - ▣ προκαλείται από την επέλευση μιας συγκεκριμένης ημερομηνίας και ώρας ή το πέρασμα του χρόνου.

Transitions

- **Transition:** μετακίνηση από μια κατάσταση σε μια άλλη
 - Πραγματοποιείται ουσιαστικά σε μηδενικό χρόνο, ανεξάρτητα από το πόσο περίπλοκη είναι η μετάβαση.
- Είναι η σχέση μεταξύ δύο καταστάσεων που δείχνει ότι ένα αντικείμενο στην πρώτη κατάσταση θα:
 - ▣ εκτελέσει ορισμένες ενέργειες και
 - ▣ εισέλθει στη δεύτερη κατάσταση όταν συμβεί το καθορισμένο event και ικανοποιηθεί η καθορισμένη συνθήκη.

Guard Conditions

- **Transition Guard conditions**

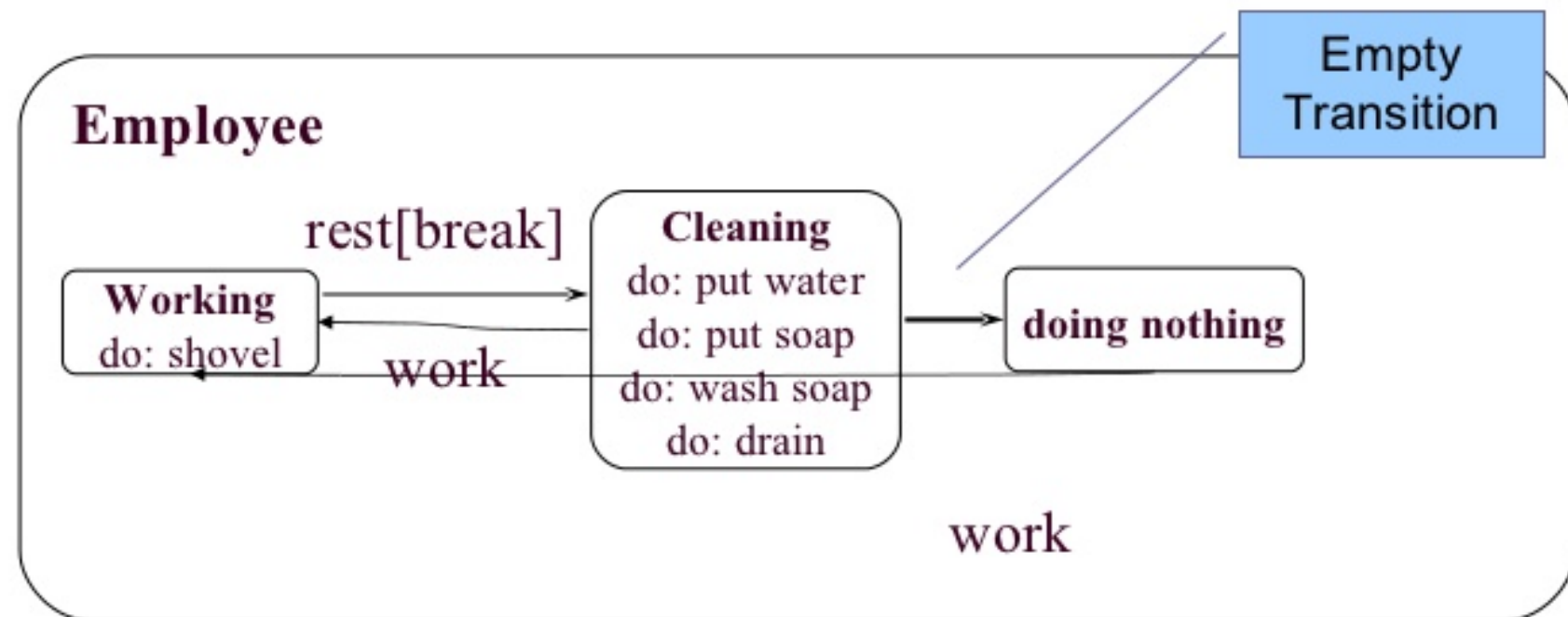
- Μια μετάβαση (transition) μπορεί επίσης να έχει μια συνθήκη προστασίας
 - Μια boolean έκφραση που αξιολογείται όταν το transition ενεργοποιείται μετά από τη λήψη ενός event.
 - Η μετάβαση πραγματοποιείται μόνο εάν η έκφραση boolean αξιολογηθεί ως True.

- Ισχύει για ένα χρονικό διάστημα

- **To Guard condition** εμφανίζεται σε παρένθεση, μετά το όνομα του event

Empty Transitions

- A transition can have any combination (including none) of the events, guards and actions
- When a transition does not have an event, it is taken after all the activities were ended



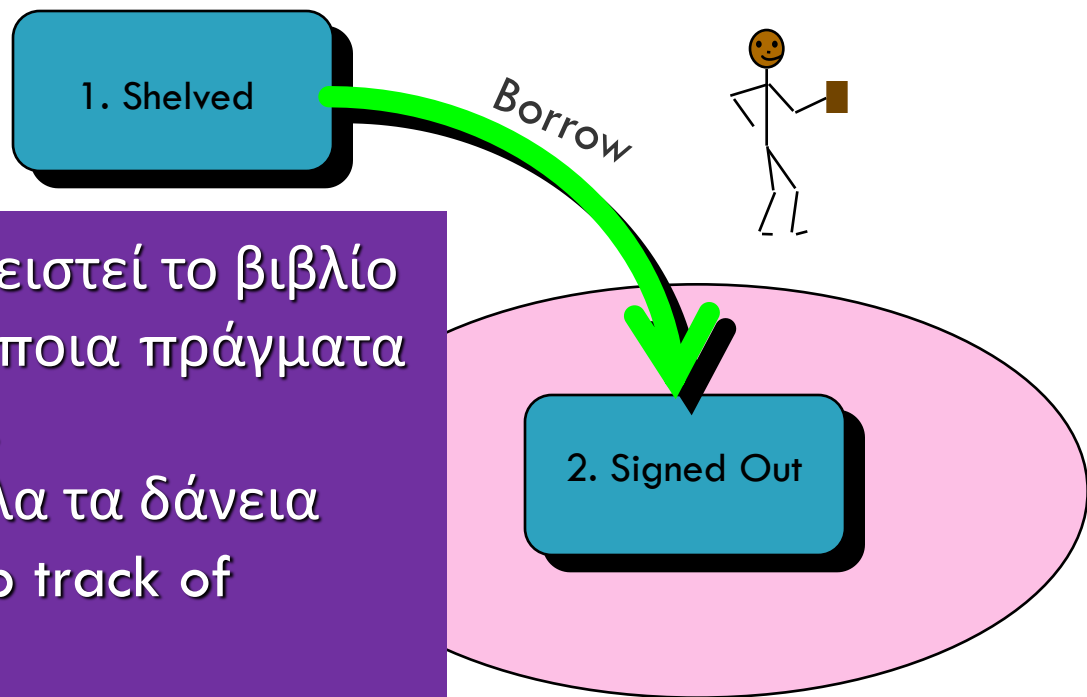
Actions

- Ένα Action είναι μια συμπεριφορά-behaviour (μια μέθοδος του αντικειμένου)
 - ▣ Βρίσκεται πάνω στο transition
 - ▣ Ένα action εκτελείται όταν το transition από μια κατάσταση σε άλλη συμβεί.
 - ▣ Ένα transition μπορεί να προκαλέσει την πυροδότηση/ενεργοποίηση ενός action.
 - ▣ Επομένως γίνεται άμεση ενεργοποίηση/κλήση της μεθόδου/operation του αντικειμένου (object).
- Ο στόχος του State Transition Diagram
 - ▣ Να αναγνωρίσει τα Actions
 - ▣ Για να μπορέσουμε να *αναγνωρίσουμε* με λεπτομέρεια την *συμπεριφορά* και τις *μεθόδους ενός αντικειμένου*.
- ▣ **Always look at the Object Sequence diagram and conceptual class diagram in order to help identify methods and therefore Actions in the State chart.**

Activities

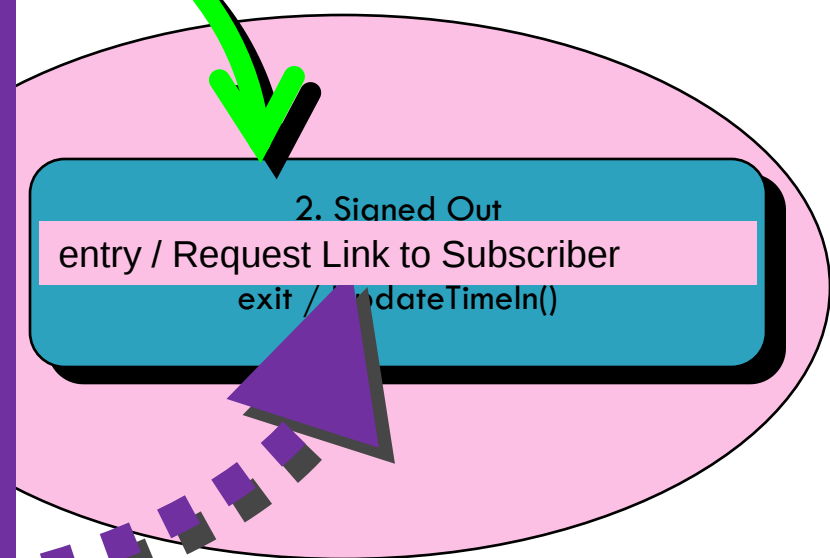
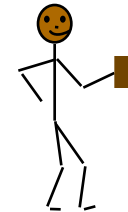
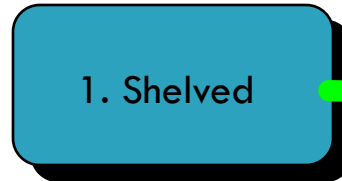
- Συνδεδεμένα με τα states
- Μεθόδοι/operations του αντικειμένου που χρειάζονται κάποια ώρα για να ολοκληρωθούν
- Notation **“do: A”** within a state box
 - Δείχνει ότι ένα activity A ενεργοποιείται στην είσοδο σε ένα state και ολοκληρώνεται στην έξοδο από ένα state -- επομένως το activity A είναι active καθώς είναι μέσα στο state (while in the state).
 - Τα **Activities είναι methods/operations** τα οποία γίνονται activated on entry to the state, on exit from the state, or whilst in the state
- **On Entry and On exit activities** επιτρέπουν στην ίδια μέθοδο να ενεργοποιείται κάθε φορά που το αντικείμενο εισέρχεται ή εξέρχεται από μια συγκεκριμένη κατάσταση.
 - Τα activities επιτρέπουν να γίνει αυτό χωρίς να χρειάζεται να τοποθετούνται ρητά τα actions σε κάθε εισερχόμενη ή εξερχόμενη μετάβαση (transition).
 - Μπορούν επίσης να έχουν arguments or guard conditions.

Object Book example: When the state of the Book changes to “Signed Out.”



Όταν ο πελάτης δανειστεί το βιβλίο πρέπει να γίνουν κάποια πράγματα για να μπορούμε να παρακολουθούμε όλα τα δάνεια των βιβλίων (to keep track of all the loans).

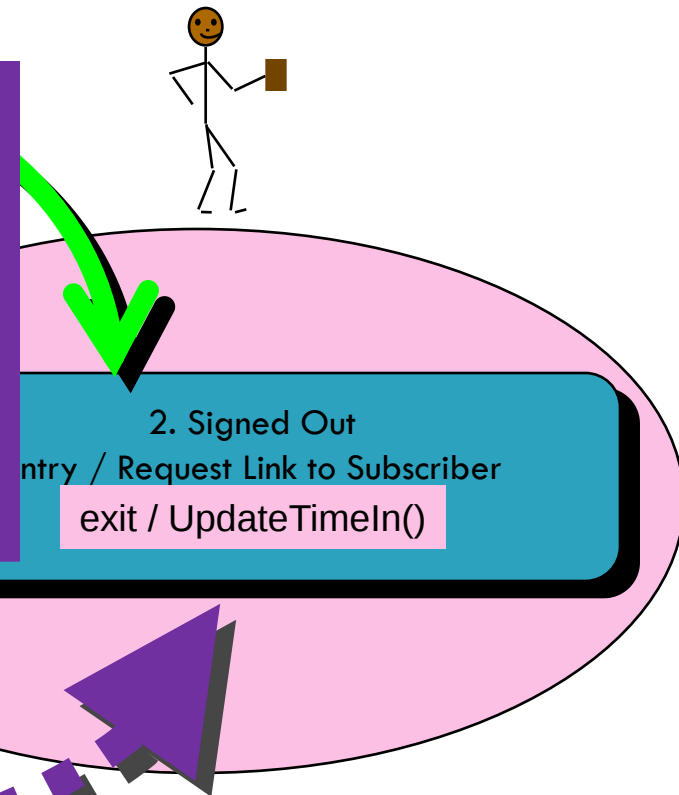
Object Book example: When the state of the Book changes to “Signed Out.”



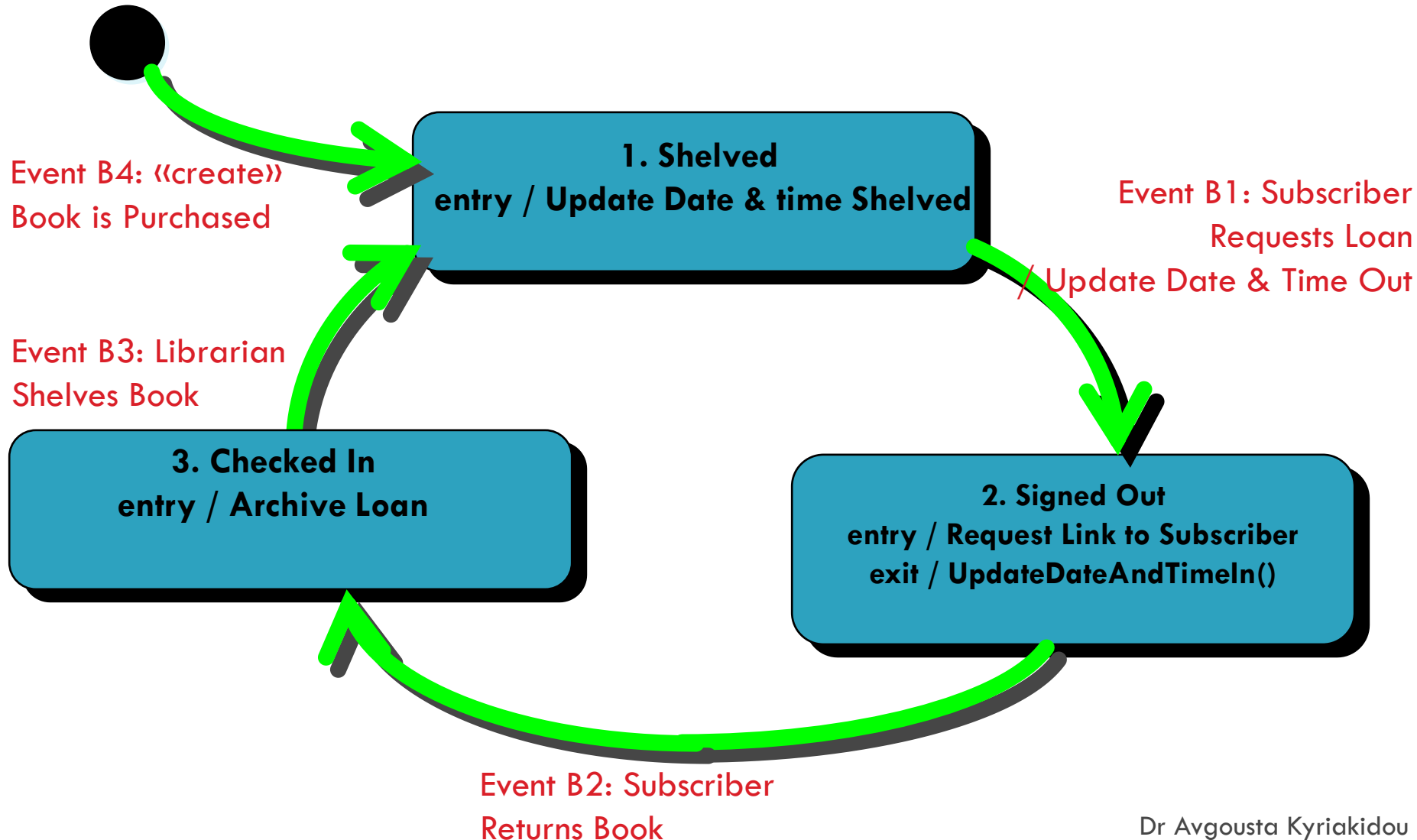
Όταν δανείζεται, το Βιβλίο πρέπει να συνδέεται με κάποιο τρόπο με τον δανειολήπτη - the borrower/subscriber

Object Book example: When the state of the Book changes to “Signed Out.”

Και όταν φεύγει από την κατάσταση "Signed Out", πρέπει να ενημερώνετε η Ημερομηνία και η Ώρα.



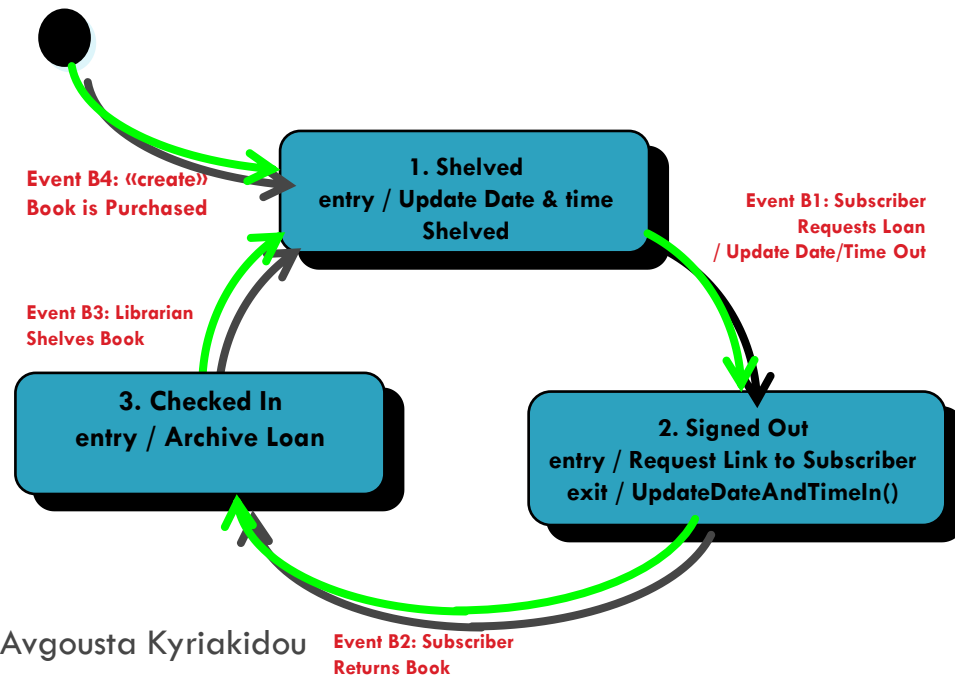
3. State Transition Diagram: Library Book



3. State Transition Diagram: Notation

Events are shown with **arrows (transitions)**.

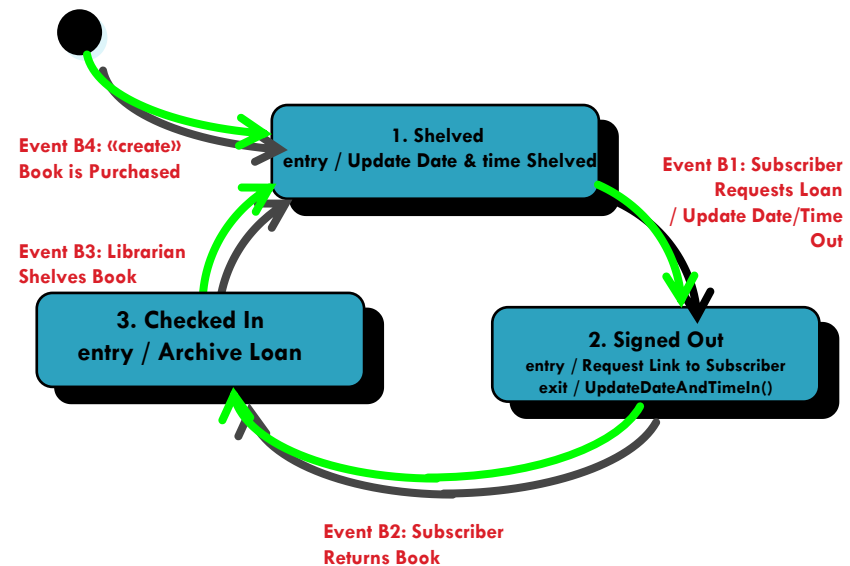
- Το event που προκαλεί τη δημιουργία του αντικειμένου εμφανίζεται να προέρχεται από μια μεγάλη μαύρη κουκκίδα και έχει το "στερεότυπο" "create" (π.χ. Event B4).



3. State Transition Diagram: Notation

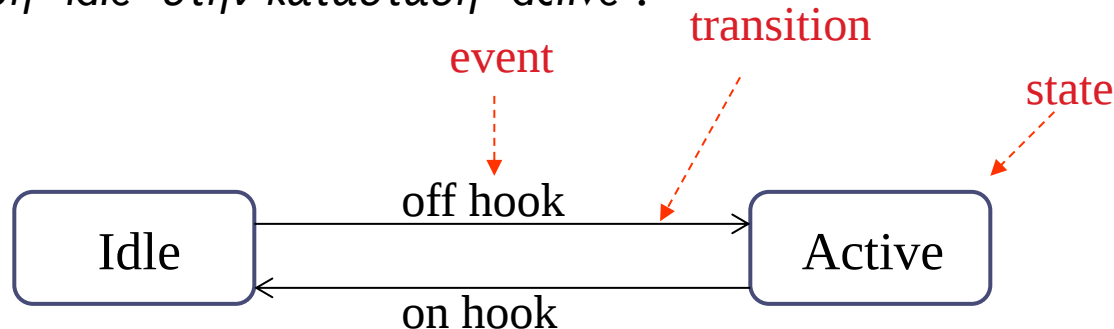
Ο αριθμος ενός **State** είναι μοναδικός *μέσα στο State Diagram* e.g., “**2: Signed Out**”

- Event numbers είναι μοναδικά μέσα στο project
- **Event** numbers έχουν ένα prefix το οποίο αναγνωρίζει το class στο οποίο ανήκουν, e.g.:
 - “**B3**: Librarian Shelves Book” ανήκει στην κλάση **Book**.
 - “**L1**: Create Loan Instance” βρίσκεται στην κλάση **Loan**.



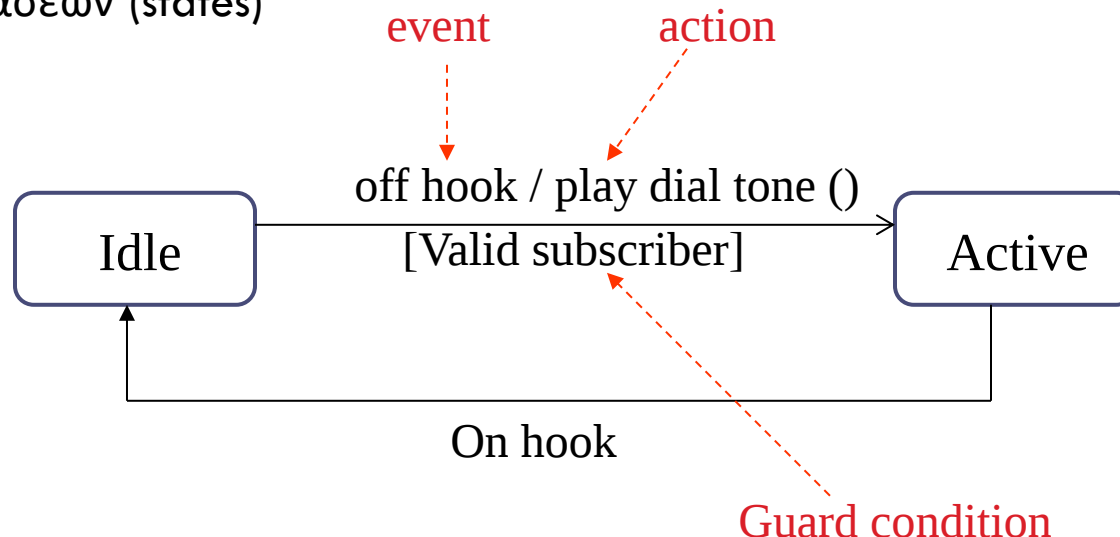
4. Telephone System Example

- **Event** - είναι ένα σημαντικό ή αξιοσημείωτο γεγονός.
 - ▣ π.χ. σηκώνουμε το ακουστικό του τηλεφώνου από την κύρια συσκευή
- **State** - είναι η κατάσταση ενός αντικειμένου σε μια χρονική στιγμή.
 - ο χρόνος μεταξύ των events.
 - ▣ π.χ. ένα τηλέφωνο βρίσκεται σε **κατάσταση "αδράνειας- idle"** μετά την τοποθέτηση του ακουστικού στην συσκευή και μέχρι να το ξανασηκώσουν από την συσκευή.
- **Transition**- είναι μια σχέση μεταξύ δύο states που υποδηλώνει ότι όταν συμβαίνει ένα γεγονός (event), το αντικείμενο (object) μετακινείται από την προηγούμενη κατάσταση (state) στην επόμενη.
 - ▣ π.χ. όταν συμβεί το event "off hook", το object τηλέφωνο μετακινείται από την κατάσταση "idle" στην κατάσταση "active".



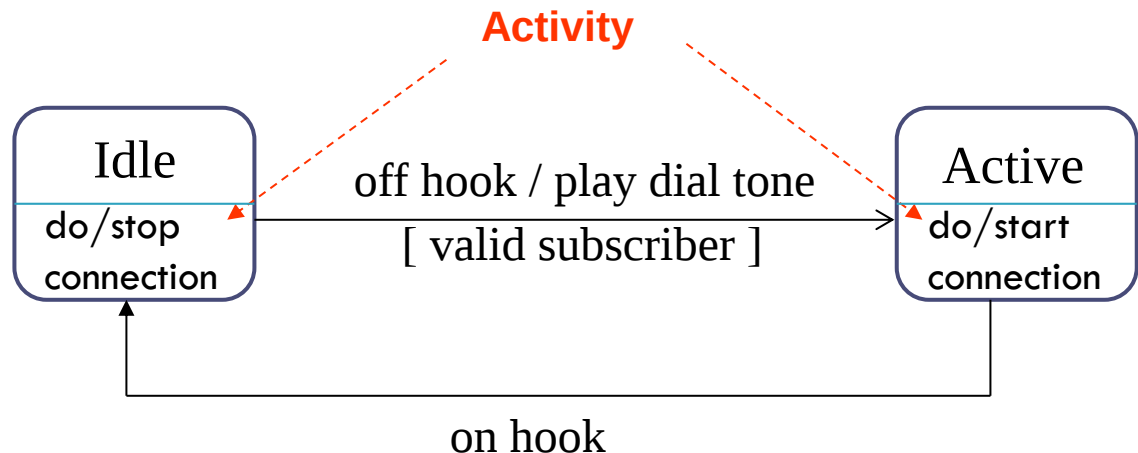
4. Telephone System Example: Actions and Guard Conditions

- **Actions:** συνήθως μέθοδοι που εκτελούνται όταν συμβεί το event και γίνει το transition.
- **Guard Conditions:** χρησιμοποιούνται ως φρουροί στις μεταβάσεις (transitions). Εμφανίζονται σε αγκύλες, μετά το όνομα του συμβάντος
- **Εάν** το **event** συμβεί **και** το guard **είναι αληθές, τότε** η μετάβαση ενεργοποιείται, το **action** (εάν υπάρχει) εκτελείται και το object τηλέφωνο πηγαίνει στην ενεργή κατάσταση.
- Το **action** μπορεί να προκαλέσει άλλα events, τα οποία προκαλούν transitions μεταξύ άλλων καταστάσεων (states)

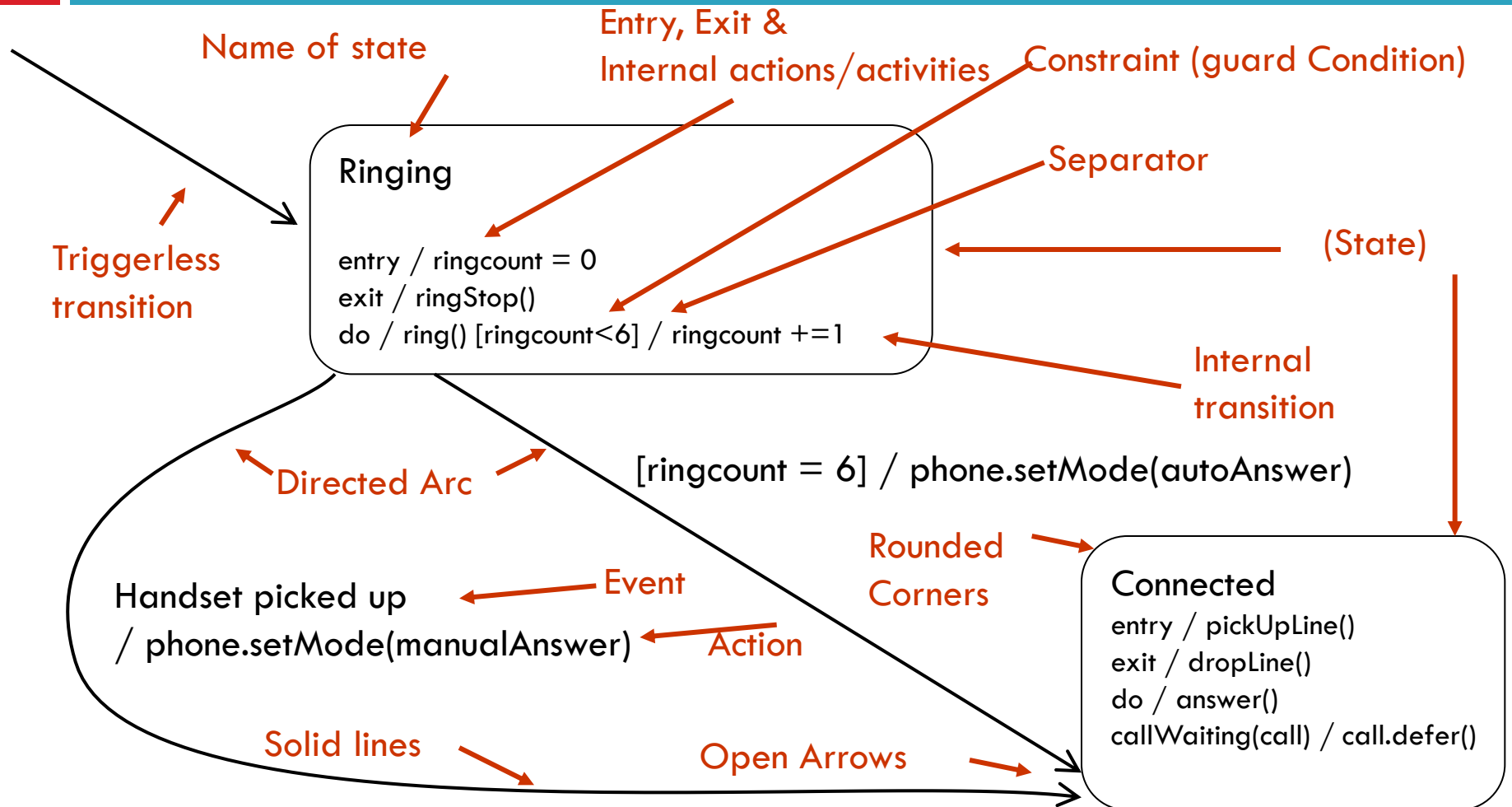


4. Telephone System Example: Activities

- Βρίσκονται μέσα σε ένα State και ενεργοποιούνται On Entry, on Exit and whilst on that state.
- Μέθοδοι/λειτουργίες που απαιτούν χρόνο για να ολοκληρωθούν
- Συμβολισμός "do: A" μέσα σε ένα state
 - υποδεικνύει ότι το activity A αρχίζει κατά την είσοδο στο state και τελειώνει κατά την έξοδο από το state
 - Η μέθοδος είναι ενεργή όσο βρίσκεται σε αυτή την κατάσταση.



4. Transitions, Actions, Guard Conditions example



State transition diagrams:

Composite State (nested state)

- Τα State diagrams μπορούν να γίνουν πολύπλοκα
- Composite state (Σύνθετη κατάσταση) είναι μια κατάσταση που *έχει μια γραφική αποσύνθεση*
 - ▣ αναλύεται σε δύο ή περισσότερες ταυτόχρονες υποκαταστάσεις (sub states)
- Για καλύτερη κατανόηση και διαχείριση
 - ▣ Μια κατάσταση (State) σε ένα state transition diagram μπορεί να επεκταθεί σε ένα state transition diagram σε άλλο επίπεδο
- Κληρονομικότητα των transitions

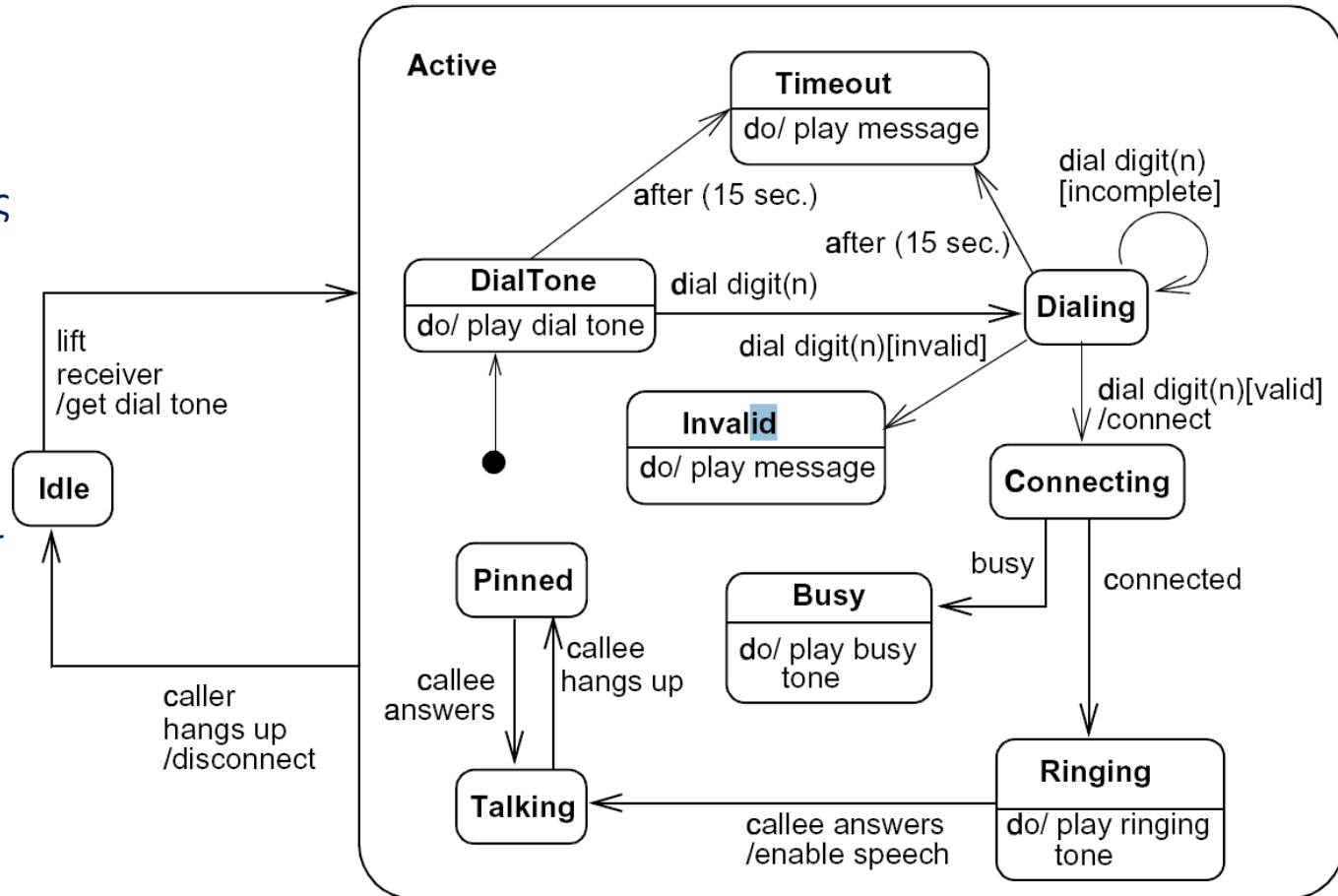
Composite state: Telephone System example

Version 2

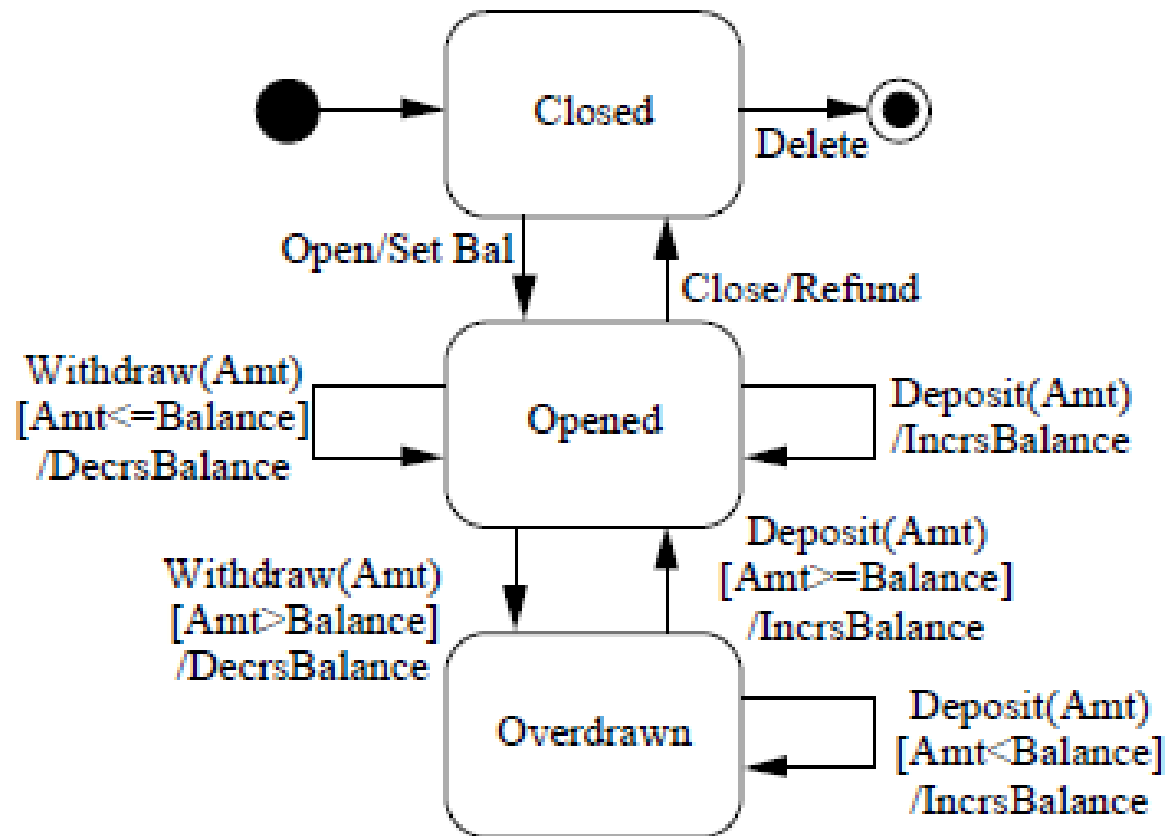
Σημειώστε ότι ένα διάγραμμα καταστάσεων μπορεί να είναι ένας συνεχής βρόχος (Loop):

- δεν υπάρχει κατάσταση εκκίνησης
- δεν ενδιαφέρεται για το πώς αρχίζει ή τελειώνει ο βρόχος

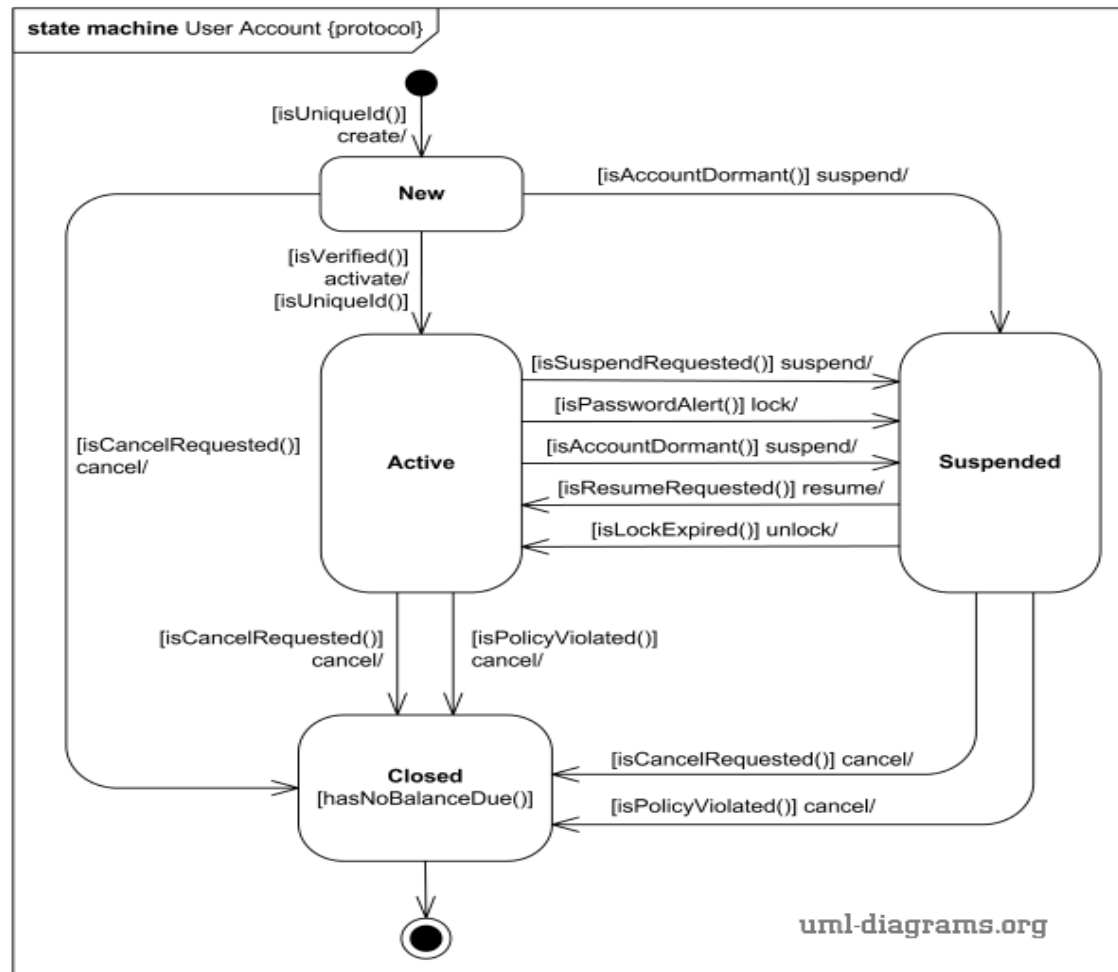
Γενικά όμως είναι χρήσιμο να έχετε μια αρχή και ένα τέλος



Account Object example



Account Object example 2

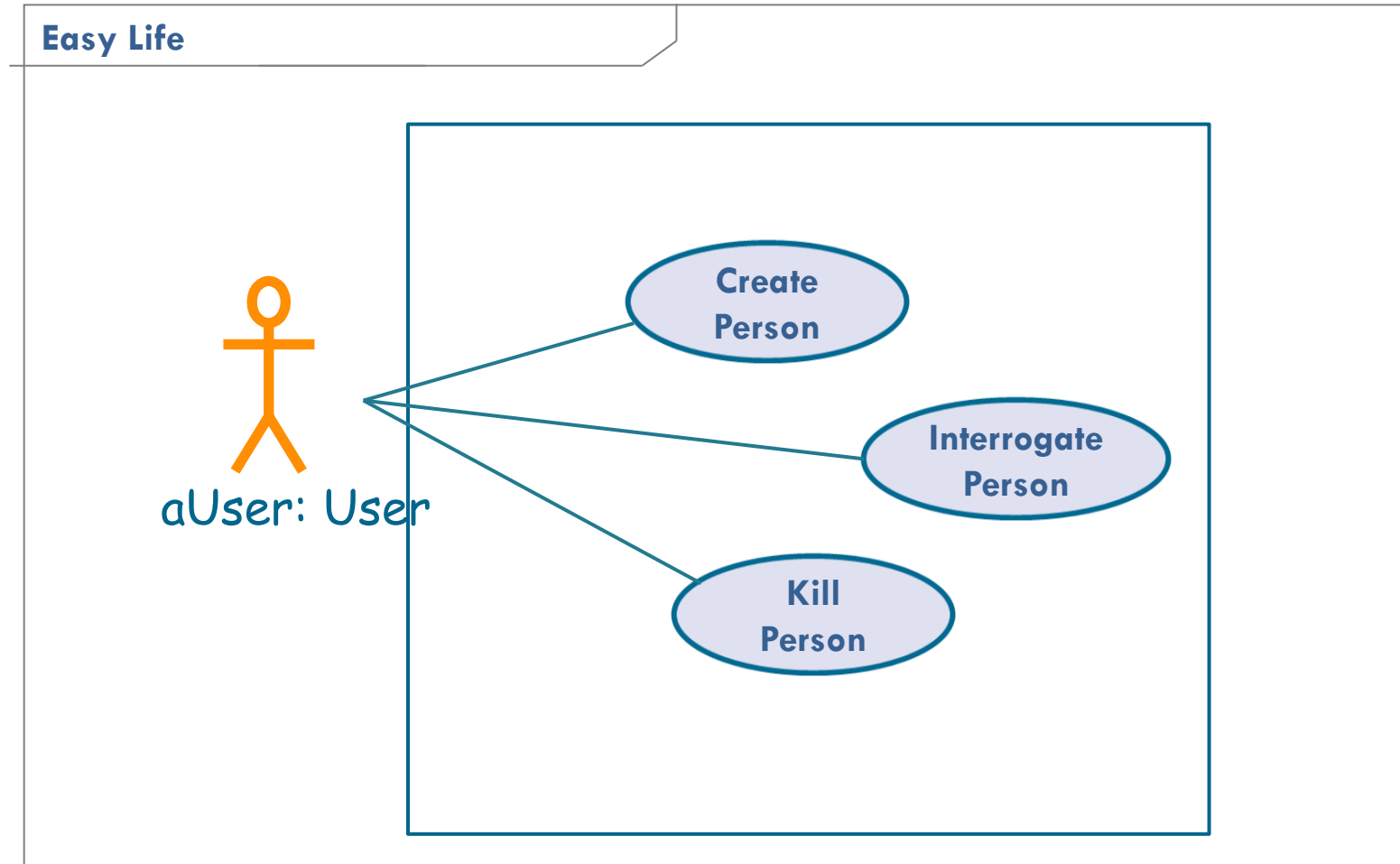


Back to Easy Life Example

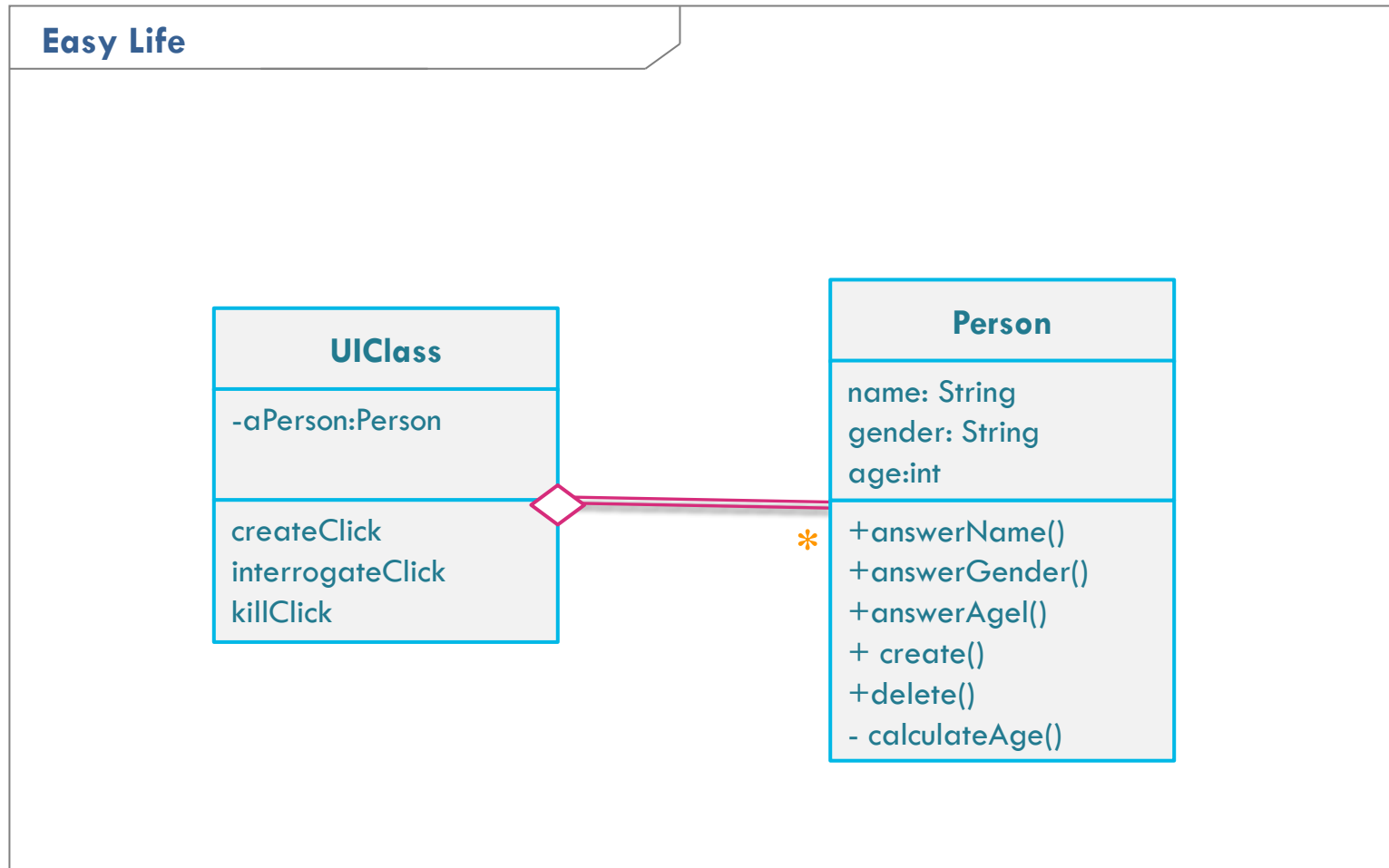
- **The Scenario**

- Create a game where personas are created to live a virtual existence in Second Life.
- It will allow us to **create** a virtual person, **interrogate** them about their existence (name, age and gender) .
- The application should allow users to **kill** a person (if their hearts let them 😊).

Use Case Diagram

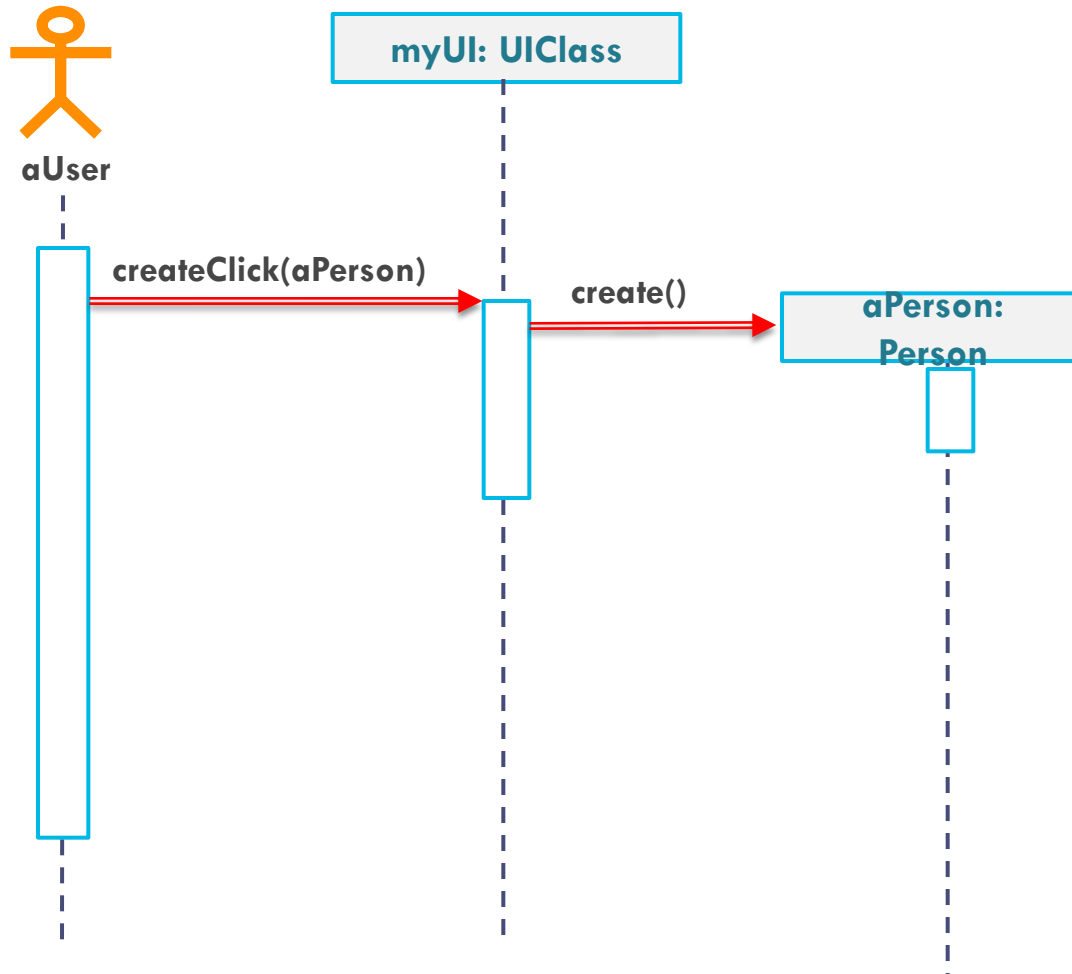


Conceptual Class Diagram

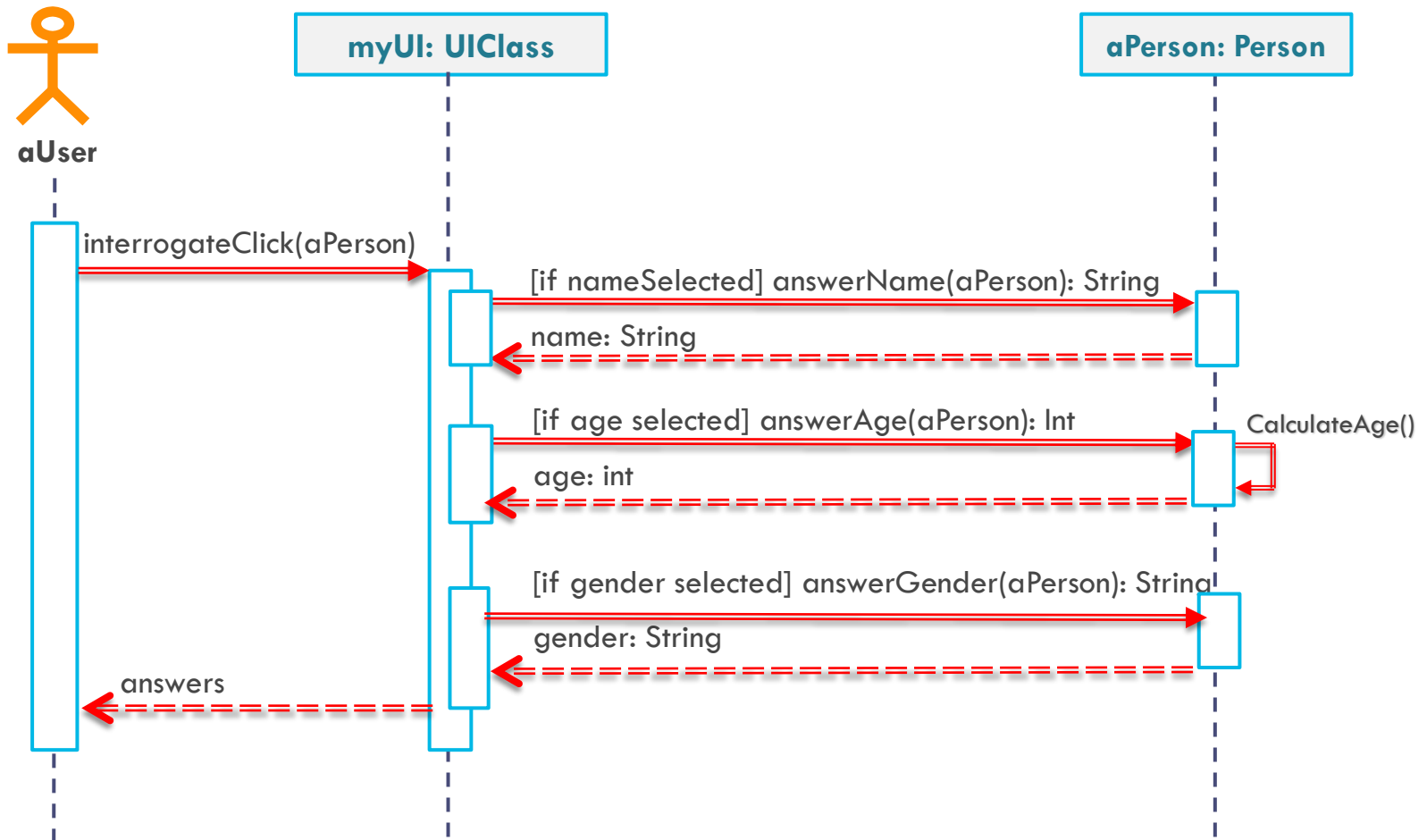


Create Person Sequence Diagram

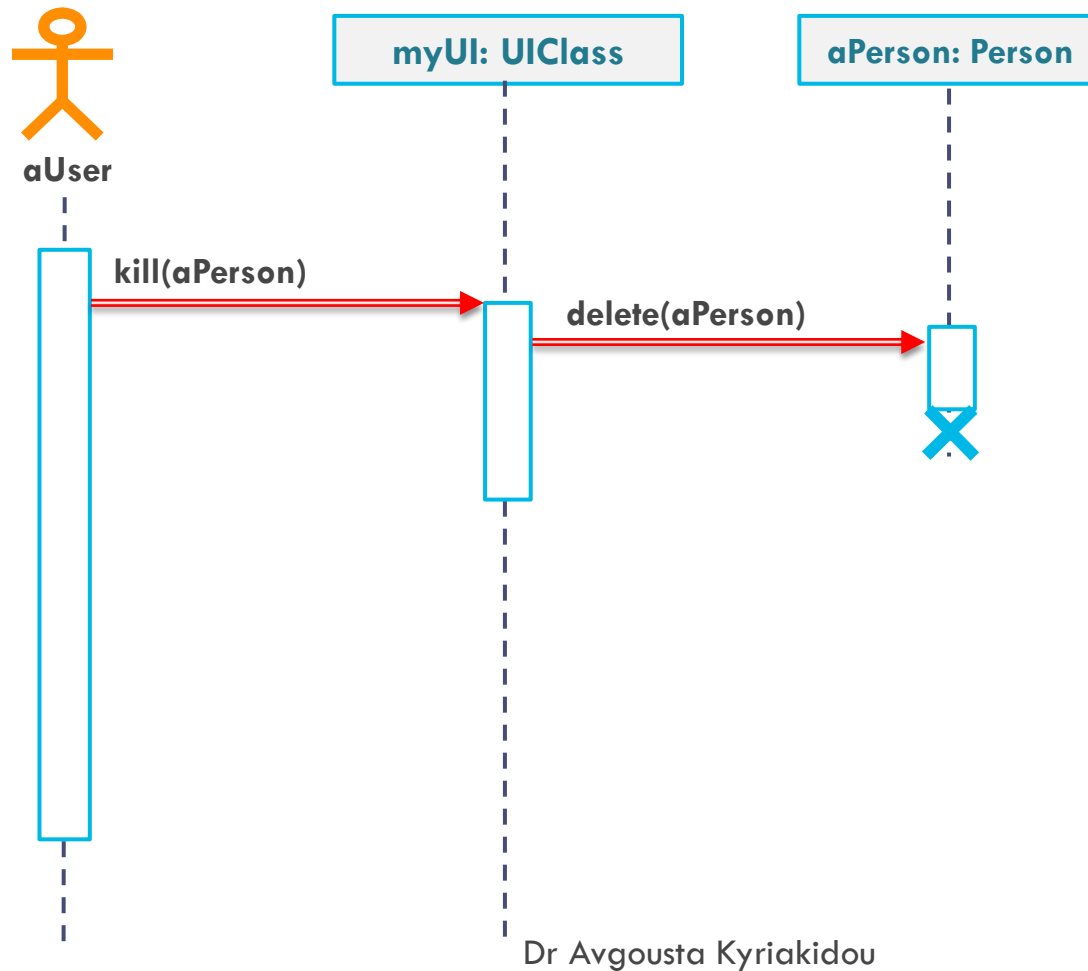
□ Example 1 - An Easy Life



Interrogate Person Sequence Diagram



Kill Person Sequence Diagram



What next?

- Πρέπει να βρούμε τα classes που έχουν ενδιαφέρων states.
- Για κάθε πιθανή κλάση, δημιουργούμε ένα state chart diagram.
- Person?

Solution ?

- SCD for class Person

Solution ?



SCD for class Person

[if name selected] AnswerName ()

State transition diagrams are useful when...

- ❑ Μια κλάση έχει έναν ενδιαφέροντα ή πολύπλοκο κύκλο ζωής, π.χ. κλάσεις που δημιουργούν ή διαγράφουν αντικείμενα ή associations
- ❑ Ένα αντικείμενο μπορεί να ενημερώνει τα attributes του με διάφορους τρόπους καθώς διανύει τον κύκλο ζωής του.
- ❑ Εάν δύο κλάσεις εξαρτώνται η μία από την άλλη, με την έννοια ότι η μία από αυτές μπορεί να ξεκινήσει τον κύκλο ζωής της άλλης ή να αλλάξει τη σειρά με την οποία αυτή μεταβαίνει από κατάσταση σε κατάσταση.
- ❑ Αν διαπιστώσετε ότι η τρέχουσα συμπεριφορά του αντικειμένου εξαρτάται από το τι του συνέβη στο παρελθόν, δηλαδή από το παρελθόν του.

Constructing a State Transition Diagram

- ❑ Προσδιορισμός καταστάσεων (states) κατά τη διάρκεια ζωής ενός αντικειμένου
 - ▣ Ακολουθήστε πιθανά σενάρια χρήσης (use case scenario)
 - ▣ Συγκέντρωση υποψήφιων states
 - ▣ Εξάλειψη των περιττών states (ίδιες απαντήσεις στα ίδια events)
- ❑ Προσδιορισμός μεταβάσεων (transitions)
- ❑ Προσδιορισμός γεγονότων (events) που προκαλούν μεταβάσεις (events/actions)
- ❑ Προσδιορίστε τα guard conditions όπου είναι δυνατόν.
- ❑ Προσδιορισμός activity (μέθοδοι που θα κληθούν κατά την είσοδο/έξοδο/πράξη)
- ❑ Διασταύρωση με άλλα διαγράμματα (όπως object sequence diagrams and class diagram) και στη συνέχεια επιστροφή για περισσότερες λεπτομέρειες.

Wrap up!

- State transition diagrams: present the life of an object within many use cases.
- States, events, transitions, actions, activities, guard conditions, composite states.
- Link Object sequence diagrams to state chart to help identify objects which have states and the actions (methods) attached to the transitions.
- **Common mistake:** Often students draw DFDs rather than State chart diagrams!
- State charts are for **ONE CLASS** only!
- DFD show flow of data in the system and involves more than one entities!

Readings for this week's lecture



State transition diagrams

- ❑ Kung, D.C (2024) Software Engineering, 2nd Edition, McGraw Hill – Chapter 9
- ❑ Pressman, R.S. (2020). Software Engineering: A practitioner's approach, 9th Edition, Mc-Graw Hill – Chapter 8