

Use Cases: μια σύντομη εισαγωγή

Heavily based on “UML & the UP” by
Arlow and Neustadt, Addison Wesley,
2002

(γενικές εισαγωγικές ιδέες)

ΣΥΛΛΟΓΗ ΑΠΑΙΤΗΣΕΩΝ

Ανάλυση απαιτήσεων

- Λειτουργικές απαιτήσεις: τι πρέπει να κάνει το σύστημα
- Μη Λειτουργικές Απαιτήσεις: τι εγγυήσεις / περιορισμούς παρέχει το σύστημα (από πλευράς επίδοσης, αξιοπιστίας, κλπ)

Use cases

- Οι use cases είναι το βασικό εργαλείο με το οποίο καταγράφουμε με δομημένο τρόπο τις λειτουργικές απαιτήσεις του συστήματος
- Σε ένα έργο, παράγουμε ένα σύνολο από use case diagrams & use case documents που μας επιτρέπουν:
 - Να καθορίσουμε ποιοι είναι οι εξωτερικοί χρήστες / συστήματα που αλληλεπιδρούν με το σύστημά μας
 - Ποια τα όρια του συστήματος που πάμε να φτιάξουμε
 - Με ποια «σενάρια» γίνεται αυτή η αλληλεπίδραση
 - Ποιες οι λεπτομέρειες για κάθε use case

Συλλογή και Ανάλυση Απαιτήσεων

- Η διαδικασία της συλλογής και ανάλυσης απαιτήσεων, μετά από συνεντεύξεις με τους εμπλεκόμενους στο σύστημα (χρήστες, διαχειριστές, κλπ) πρέπει να καταλήξει στην οριστικοποίηση και καταγραφή των παρακάτω αποτελεσμάτων
 - Λειτουργικές απαιτήσεις
 - Μη Λειτουργικές απαιτήσεις
 - Ανάθεση προτεραιοτήτων και σημαντικότητας στις απαιτήσεις
 - Συσχέτιση των απαιτήσεων με τις παραχθείσες use-cases

Κοινά λάθη

- Η συλλογή απαιτήσεων είναι μια εργασία που απαιτεί πολλές δεξιότητες (σε τεχνικό, διοικητικό και κοινωνικό επίπεδο), μία όμως από αυτές είναι αξιομνημόνευτη εδώ: η αυστηρότητα
- Κοινά λάθη λόγω έλλειψης αυστηρότητας στην καταγραφή των απαιτήσεων
 - Διαγραφή: λείπει κάτι από το ποιος κάνει τι και πότε (π.χ., ο κωδικός εισάγεται στο σύστημα – από ΠΟΙΟΝ?)
 - Παραμόρφωση: κάποιος περιορισμός έχει εναλλακτικές εκτελέσεις (π.χ., αν ο κωδικός εισαχθεί λάθος, ξαναζητείται από το σύστημα – αντί για «...λάθος μέχρι και 3 φορές ...»)
 - Γενίκευση: κάποιος κανόνας γενικεύεται σε ένα σύνολο χρηστών, ως μη όφειλε (π.χ., ΟΛΟΙ οι χρήστες έχουν ένα κωδικό – ξεχνώντας ότι οι διαχειριστές μπορεί να μοιράζονται ένα κοινό κωδικό)

(ορισμός, στοιχεία, δομή και παραδείγματα)

USE CASES

Use Case Model

- Το Use Case Model περιλαμβάνει:
 - Actors: ρόλοι που εκτελούνται είτε από τους χρήστες του συστήματος ή από άλλα συστήματα εξωτερικά στο υπό μελέτη σύστημα
 - Use cases: σενάρια χρήσης του συστήματος από τους actors
 - Relationships: με ποιους τρόπους οι use cases σχετίζονται με τους actors
 - System boundary: η απόφαση για το τι είναι εντός του συστήματος και τι εκτός (το οποίο δεν είναι τόσο προφανές όσο ακούγεται)

Διαδικασία εκπόνησης

- Βρες τα όρια του συστήματος
- Βρες τους actors
- Βρες τα use cases
 - Ορισμός του use case
 - Κατασκευή σεναρίων μέσα σε ένα use case

System boundary

- System boundary defines what is part of the system and what is external to the system
- External to the system: actors (persons or systems)
- Internal to the system: use cases = what the system offers to the actors
- Notation in diagrams: box with the name of the system, with actors outside the box and use cases inside

Actors

- Actor: ρόλος που υιοθετείται από μια εξωτερική του συστήματος οντότητα για να αλληλεπιδράσει με το σύστημα
- Ένας actor
 - είναι ΠΑΝΤΑ εξωτερικός του συστήματος
 - Αλληλεπιδρά άμεσα με το σύστημα
 - Αναπαριστά ρόλους και όχι συγκεκριμένους ανθρώπους ή συστήματα
 - Χρειάζεται ένα σύντομο όνομα
 - Ζωγραφίζεται με τα γνωστά μπαρμπαδάκια



Και πού να τους βρεις?

- Ποιος χρησιμοποιεί το σύστημα?
- Ποιος διαχειρίζεται το σύστημα?
- Ποια άλλα συστήματα αλληλεπιδρούν με το σύστημα?
- Συμβαίνει κάποιο γεγονός σε τακτά χρονικά διαστήματα? (τότε ο χρόνος είναι ένας actor)

Use Cases

- A use case is “a specification of sequences of actions, including variant sequences and error sequences, that a system, subsystem, or class can perform by interacting with outside actors”
- Use cases:
 - Come with a short name (starting with a verb) and an ID
 - Are **always started by an actor**
 - Are **always written from an actor’s point of view**

Και πού τα βρίσκεις? Ρωτώντας:

- Τι λειτουργικότητα θέλει ένας χρήστης από το σύστημα?
- Ειδοποιείται ένας χρήστης αν αλλάξει η κατάσταση του συστήματος?
- Επηρεάζεται το σύστημα από εξωτερικά γεγονότα?
- Αποθηκεύει / ανακτά πληροφορία το σύστημα, και αν ναι, με ποιο ερέθισμα?

Ζωγραφική για μια use case

- Φούσκα με το όνομά της
- Γραμμή που τη συσχετίζει με κάθε actor που σχετίζεται μαζί της
- Το διάγραμμα των use cases του συστήματος βάζει μαζί system boundary, actors and use cases, all in one

Use case diagram

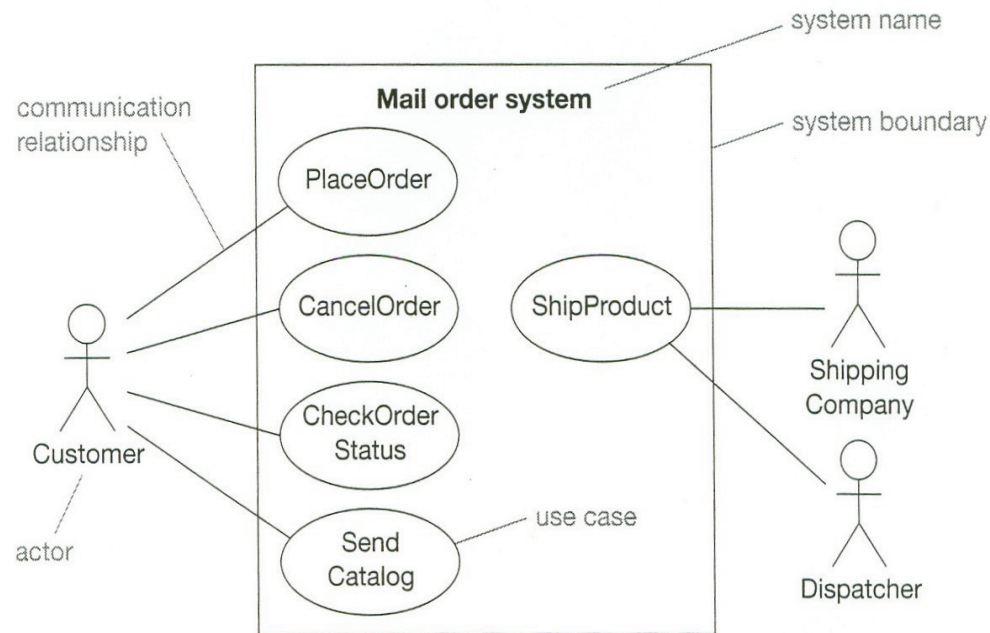


Figure 4.6

Καθορισμός μιας use case

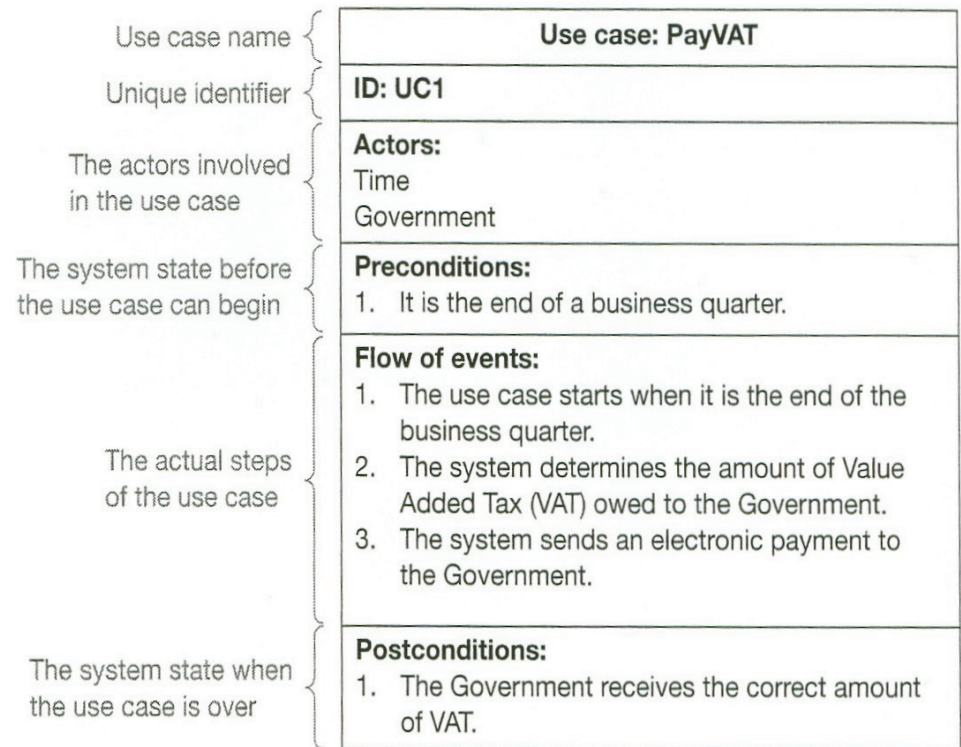
- Το διάγραμμα με τις φούσκες δίνει μόνο μια γενική εποπτεία
- Ο καθορισμός των use cases γίνεται με κείμενα σαν το διπλανό!!!

Use case name	Use case: PayVAT
Unique identifier	ID: UC1
The actors involved in the use case	Actors: Time Government
The system state before the use case can begin	Preconditions: 1. It is the end of a business quarter.
The actual steps of the use case	Flow of events: 1. The use case starts when it is the end of the business quarter. 2. The system determines the amount of Value Added Tax (VAT) owed to the Government. 3. The system sends an electronic payment to the Government.
The system state when the use case is over	Postconditions: 1. The Government receives the correct amount of VAT.

e 4.8

Καθορισμός μιας use case

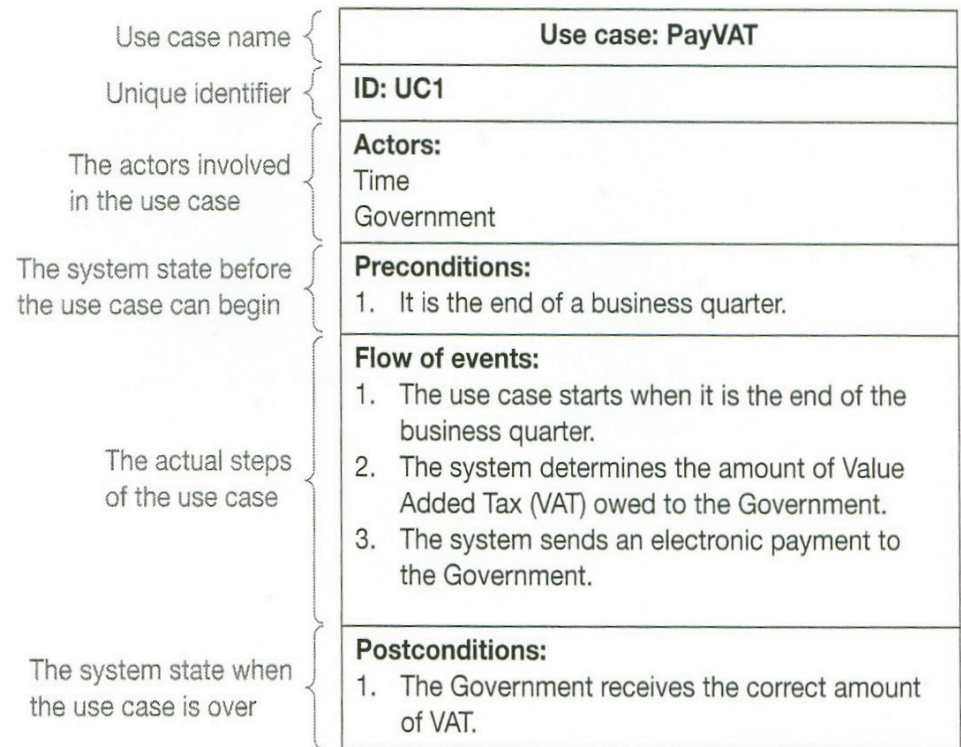
- Όνομα και ID
- Actors (ordered by frequency of usage + highlight the primary actor)
- **Preconditions** = σύνολο συνθηκών που πρέπει να αποτιμάται ΠΑΝΤΑ σε true για να μπορεί να ξεκινήσει η εκτέλεση της use case



e 4.8

Καθορισμός μιας use case

- **Primary flow of events** = sequence of steps to implement the use case
- **Alternative Flows:** για τις περιπτώσεις που υπάρχουν exceptions ή πολλά λογικά μονοπάτια εκτέλεσης
- **Postconditions:** τι εγγυήσεις δίνει η εκτέλεση κάθε flow μετά την ολοκλήρωσή της (πάλι σαν ένα σύνολο από κατηγορήματα που αποτιμώνται σε true)
- **Comments:** σχόλια, ρίσκα, μη λειτουργικές απαιτήσεις, ...



e 4.8

Δομημένη έκφραση για ροές και use cases

- Χαρακτηρισμός του ποιος ξεκινά την use case

1. The use case starts when an <actor> <function>

- Αριθμημένες προτάσεις, με αυστηρή σύνταξη (υποκείμενο = actor / (sub)system, ενεργητικό ρήμα (όχι παθητική φωνή)...

<id> The <actor / (sub)system> <performs function>

- ... και απαντήσεις στα ερωτήματα: Ποιος? Τι? Πότε? Πού? σε σχέση με το αντικείμενο της πρότασης

4. The main engine shall load the records from the log file to the engine's record placeholder

Δομημένη έκφραση

- Reserve **if, else, for, while** as keywords για να μπορείτε να προδιαγράψετε branching & (rarely) loops
- Δύο τρόποι (συμπληρωματικοί) για το branching:
 - Αν είναι μικρό το branching, μέσα στο primary flow (κάποιοι modelers διαφωνούν – θέλουν καθαρό το primary flow)
 - Αλλιώς, χωριστό **alternative flow**

Branching in two ways

Use case: ManageBasket
ID: UC10
Actors: Customer
Preconditions: 1. The shopping basket contents are visible.
Flow of events: 1. The use case starts when the Customer selects an item in the basket. 2. If the Customer selects "delete item" 2.1 The system removes the item from the basket. 3. If the Customer types in a new quantity 3.1 The system updates the quantity of the item in the basket.
Postconditions: 1. The basket contents have been updated.

Use case: DisplayBasket
ID: UC11
Actors: Customer
Preconditions: 1. The Customer is logged on the system.
Flow of events: 1. The use case starts when the Customer selects "display basket". 2. If there are no items in the basket 2.1 The system informs the Customer that there are no items in the basket yet. 2.2 The use case terminates. 3. The system displays a list of all items in the Customer's shopping basket including product ID, name, quantity and item price.
Postconditions:
Alternative flow 1: 1. At any time the Customer may leave the shopping basket screen.
Postconditions:
Alternative flow 2: 1. At any time the Customer may leave the system.
Postconditions:

For and While within use cases (rarely, however)

Use case: FindProduct
ID: UC12
Actors: Customer
Preconditions:
Flow of events: 1. The Customer selects "find product". 2. The system asks the Customer for search criteria. 3. The Customer enters the requested criteria. 4. The system searches for products that match the Customer's criteria. 5. If the system finds some matching products then 5.1. For each product found 5.1.1. The system displays a thumbnail sketch of the product. 5.1.2. The system displays a summary of the product details. 5.1.3. The system displays the product price. 6. Else 6.1. The system tells the Customer that no matching products could be found.
Postconditions:
Alternative flow: 1. At any point the Customer may move to different page.
Postconditions:

Use case: ShowCompanyDetails
ID: UC13
Actors: Customer
Preconditions:
Flow of events: 1. The use case starts when the Customer selects "show company details". 2. The system displays a web page showing the company details. 3. While the Customer is browsing the company details 3.1. The system plays some background music. 3.2. The system displays special offers in a banner ad.
Postconditions:

Σενάρια

Use case: Checkout
ID: UC14
Actors: Customer
Preconditions:
Primary scenario: 1. The use case begins when the Customer selects “go to checkout”. 2. The system displays the customer order. 3. The system asks for the customer identifier. 4. The Customer enters a valid customer identifier. 5. The system retrieves and displays the Customer's details. 6. The system asks for credit card information – name on card, card number and expiry date. 7. The Customer enters the credit card information. 8. The system asks for confirmation of the order. 9. The Customer confirms the order. 10. The system debits the credit card. 11. The system displays an invoice.
Secondary scenarios: InvalidCustomerIdentifier InvalidCreditCardDetails CreditCardLimitExceeded CreditCardExpired
Postconditions:

Use case: Checkout Secondary scenario: InvalidCustomerIdentifier
ID: UC15
Actors: Customer
Preconditions:
Secondary scenario: 1. The use case begins in step 3 of the use case Checkout when the Customer enters an invalid customer identifier. 2. For three invalid entries 2.1. The system asks the Customer to enter the customer identifier again. 3. The system informs the Customer that their customer identifier was not recognized.
Postconditions:

Όταν η δομή γίνεται πολύπλοκη και το branching έχει παραγίνει, βάζουμε **σενάρια**, αναθέτοντας μια ροή σε κάθε ένα εκ των οποίων και χωρίζοντας τη use case σε primary / secondary scenarios