

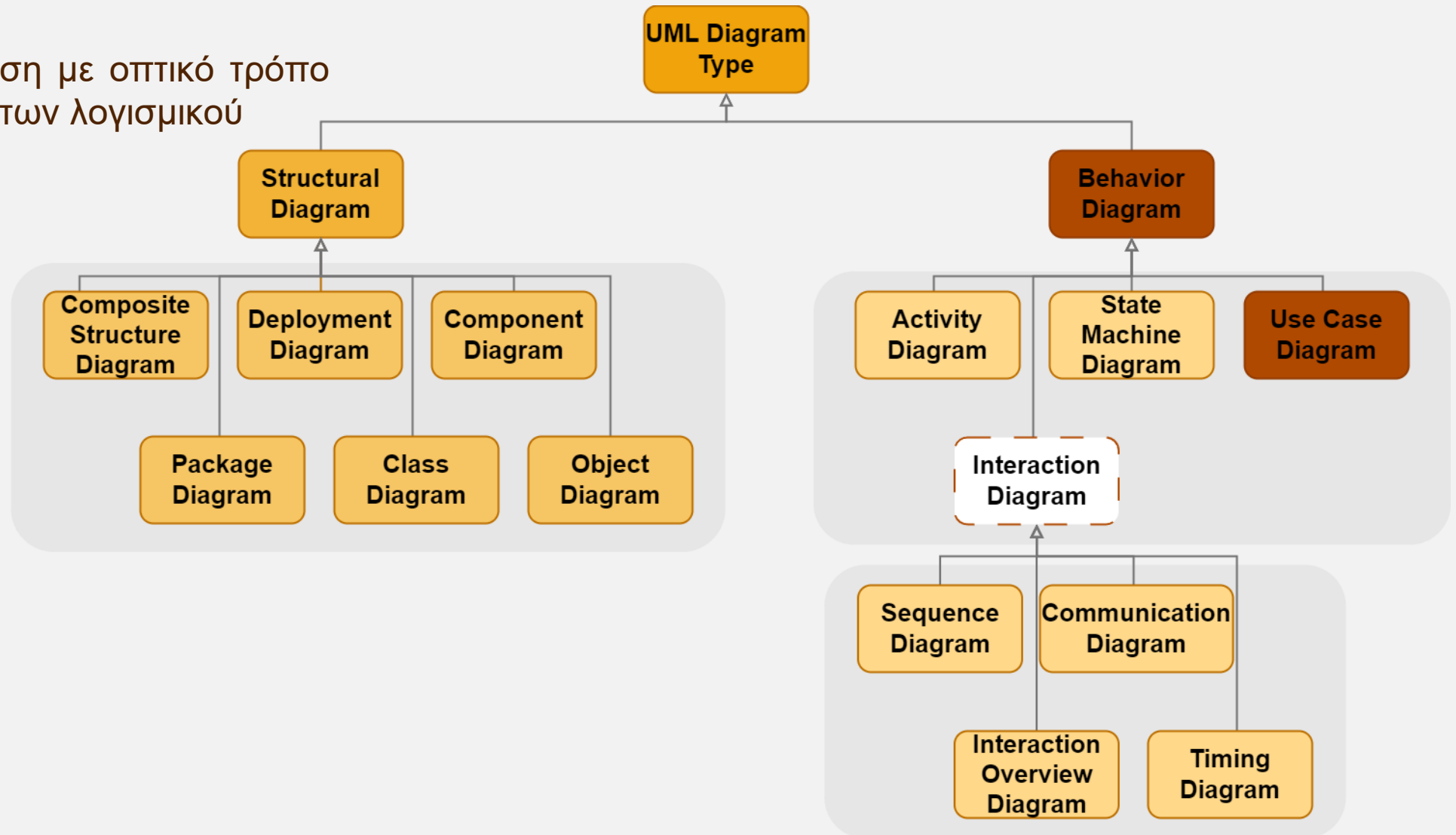
Τεχνολογία Λογισμικού

Εισαγωγή στη UML



UML (Unified Modeling Language)

- **UML** - Αναπαράσταση με οπτικό τρόπο (visualization) τμημάτων λογισμικού



UML (Unified Modeling Language)

Textual Analysis - Ανάλυση Κειμένου

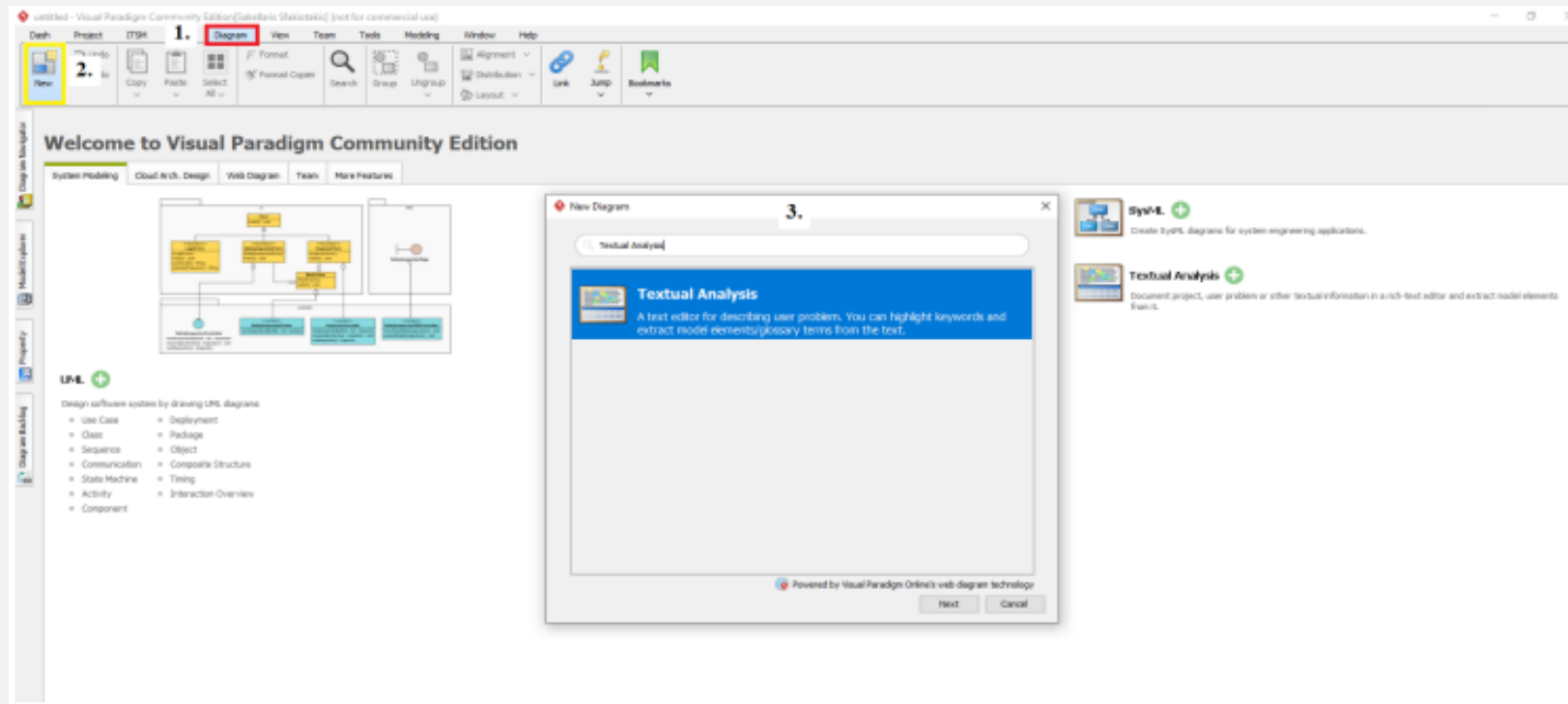


Τι είναι η Ανάλυση Κειμένου?

Πριν αρχίσει τη διαδικασία μοντελοποίησης της επιχειρηματικής διαδικασίας, συνήθως πρέπει να υπάρξει συζήτηση με τους πελάτες σχετικά με τις ανάγκες τους, με σκοπό την εξοικείωση με τις λειτουργίες της εταιρείας τους, καθώς και με τα προβλήματά τους. Κατά τη διάρκεια της συνάντησης, μπορεί να συλλέχθουν χρήσιμες πληροφορίες από τους πελάτες, συμπεριλαμβανομένου του αρχείου συνομιλίας και των εγγράφων. Μπορεί να χρησιμοποιηθεί η **ανάλυση κειμένου (Textual Analysis)**, ένας επεξεργαστής κειμένου για να βοηθήσει στην καταγραφή αυτών των κειμενικών πληροφοριών. Εκτός από έναν απλό επεξεργαστή κειμένου, μπορεί να αναγνωρίσει σημαντικούς όρους ή αντικείμενα (π.χ. κλάση, περιπτώσεις χρήσης) από την περιγραφή του προβλήματος.

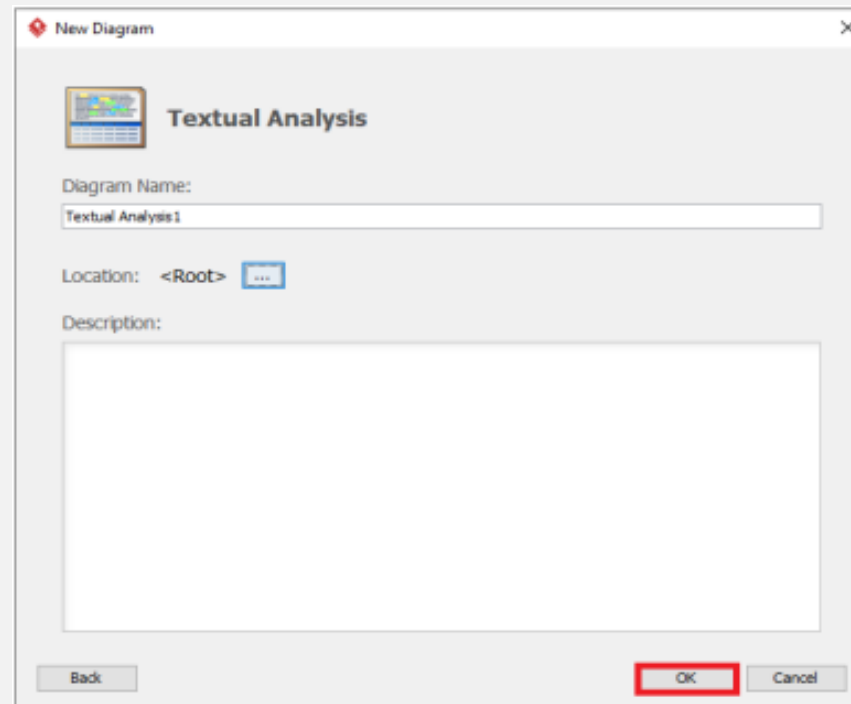
Δημιουργία Textual Analysis

Αφότου ανοίξουμε το Visual Paradigm επιλέγουμε Diagram από την γραμμή εργαλείων, επιλέγουμε New και στο παράθυρο που εμφανίζεται αναζητάμε Textual Analysis.



Δημιουργία Textual Analysis

Επιλέγοντας next δίνουμε όνομα στο διάγραμμα και πατάμε ok για να δημιουργηθεί.



The screenshot shows a 'New Diagram' dialog box with the following fields and options:

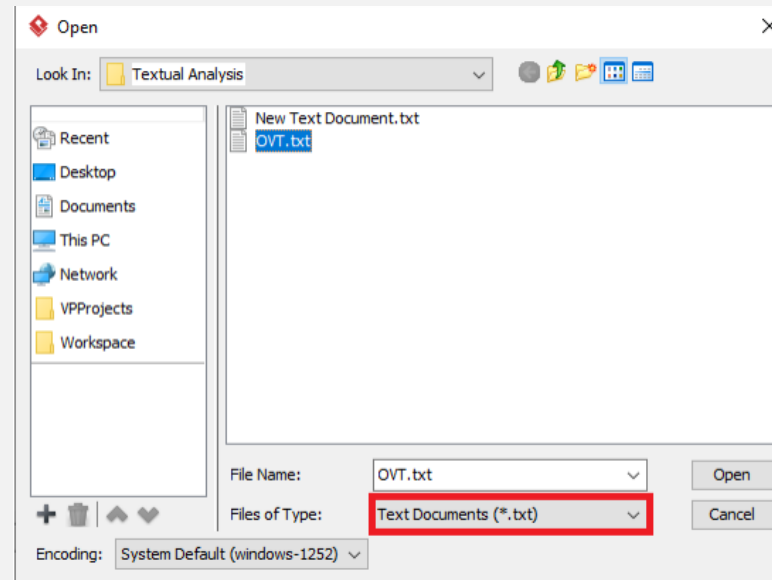
- Diagram Name:** A text input field containing 'Textual Analysis1'.
- Location:** A dropdown menu showing '<Root>' with a '...' button next to it.
- Description:** A large empty text area for entering a description.
- Buttons:** 'Back', 'OK' (highlighted with a red box), and 'Cancel'.

Δημιουργία Textual Analysis

Επιλέγουμε το σύμβολο της εισαγωγής για να εισάγουμε το κείμενο.

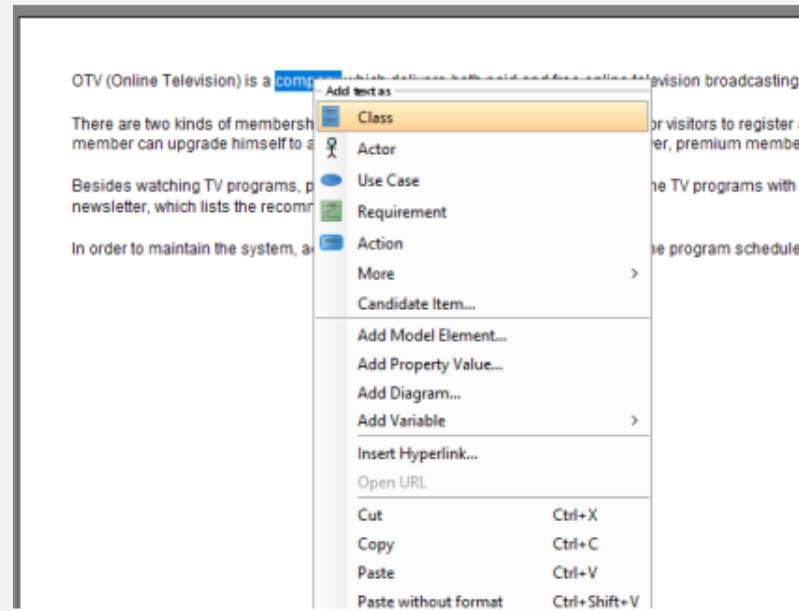


Επιλέγουμε ο τύπος αρχείου να είναι **.txt** και επιλέγουμε το αρχείο μας.



Δημιουργία Textual Analysis

Επιλέγουμε το κείμενο που μας ενδιαφέρει και πατάμε δεξί κλικ ώστε να το ορίσουμε π.χ. σαν Actor.

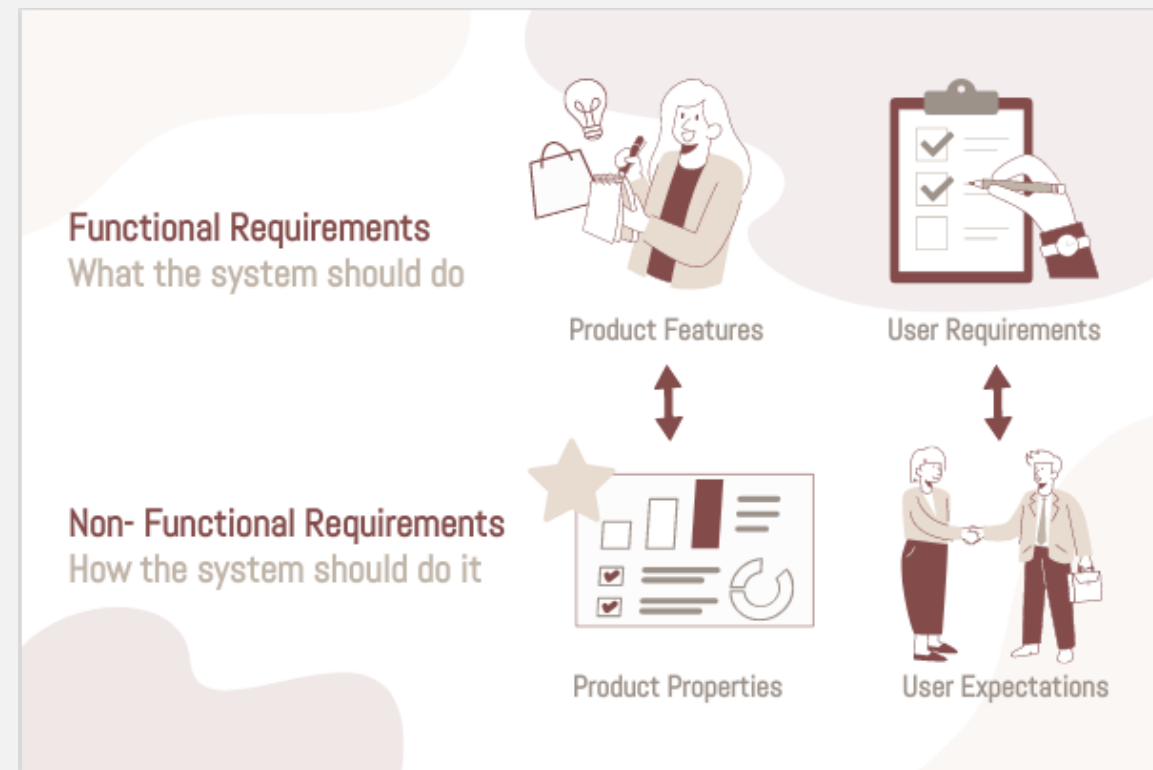
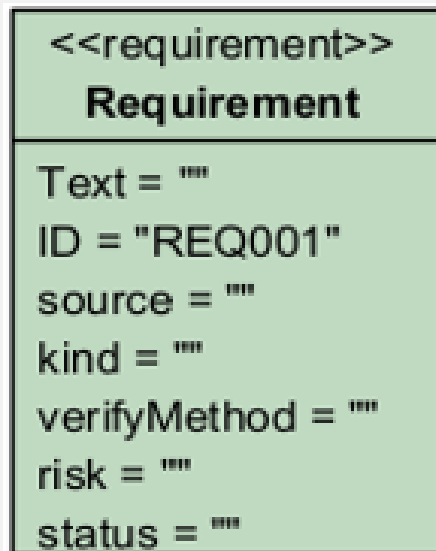


UML (Unified Modeling Language)

Requirements Diagram- Διάγραμμα απαιτήσεων

Requirement

Μια απαίτηση(requirement) καθορίζει μια ικανότητα ή προϋπόθεση που πρέπει (ή θα έπρεπε) να ικανοποιηθεί. Μια απαίτηση μπορεί να καθορίζει μια λειτουργία που πρέπει να εκτελεί ένα σύστημα ή μια συνθήκη απόδοσης που πρέπει να επιτύχει ένα σύστημα



Requirements

Τα χωρίζουμε κυρίως σε **Functional** και **Non-Functional** requirements

- **Functional**: Είναι αυτές οι απαιτήσεις που ο τελικός χρήστης ζητά ως βασικές λειτουργίες στο σύστημα και πρέπει απαραίτητως να υπάρχουν. Αυτές οι λειτουργίες πρέπει να ενσωματωθούν στο σύστημα ως μέρος της σύμβασης. Εκφράζονται σε όρους των εισόδων που δίνονται στο σύστημα, των λειτουργιών που εκτελεί και των αναμενόμενων αποτελεσμάτων. Κατά βάση, αυτές είναι οι απαιτήσεις του χρήστη που είναι άμεσα ορατές στο τελικό προϊόν, αντίθετα με τις μη λειτουργικές απαιτήσεις.

- **Non-Functional**: Αυτοί είναι ουσιαστικά οι περιορισμοί ποιότητας που το σύστημα πρέπει να ικανοποιήσει σύμφωνα με τη σύμβαση του έργου. Η προτεραιότητα ή η έκταση με την οποία αυτοί οι παράγοντες εφαρμόζονται ποικίλλει από ένα έργο σε ένα άλλο. Αυτοί οι περιορισμοί αποκαλούνται επίσης **non-behavioral requirements**.

Υποκατηγορίες non-functional requirements: User Interface, Reliability, Reusability, Portability, Security, Performance, Maintenance, Standards



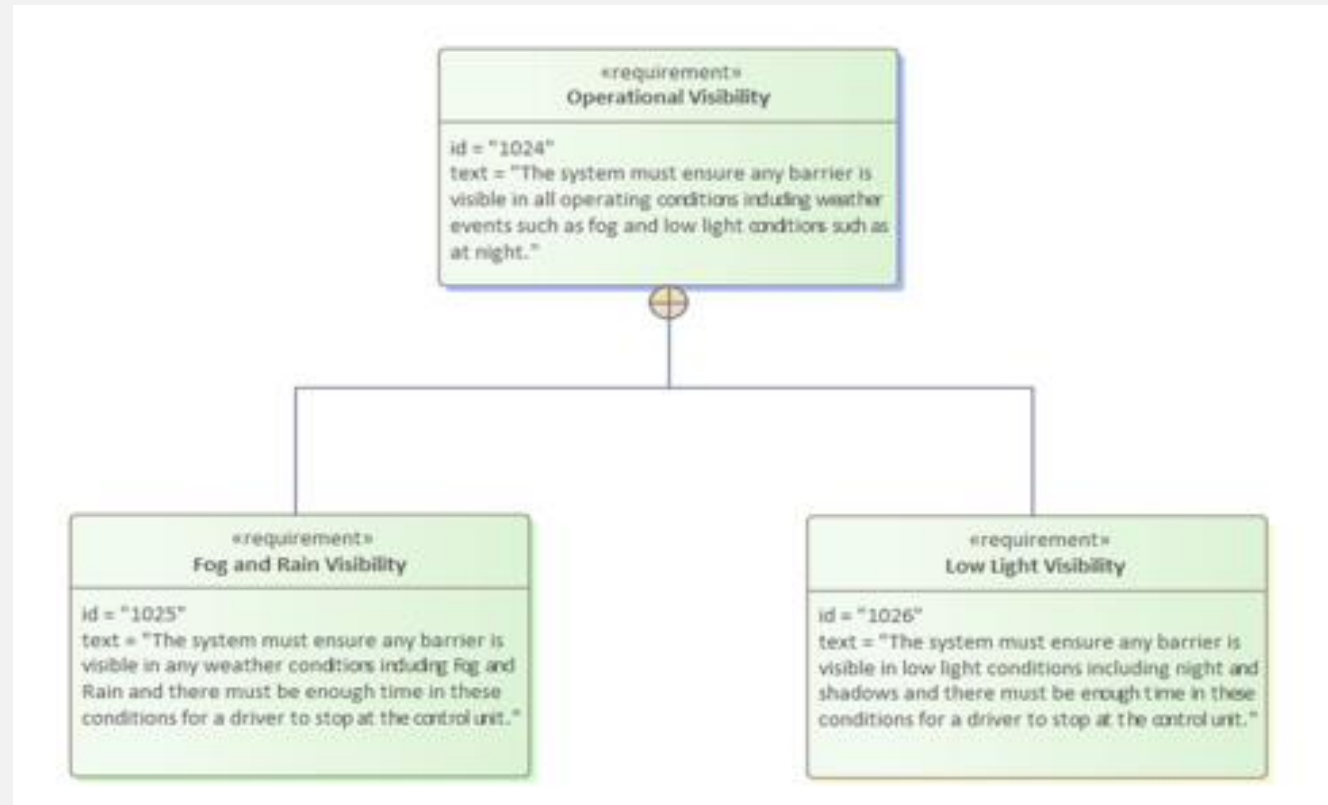
Non-Functional Requirements

Παραδείγματα:

1. Security: The system must be secure from unauthorized access.
2. Performance: The system must be able to handle the required number of users without any degradation in performance.
3. Scalability: The system must be able to scale up or down as needed.
4. Availability: The system must be available when needed.
5. Maintenance: The system must be easy to maintain and update.
6. Portability: The system must be able to run on different platforms with minimal changes.
7. Reliability: The system must be reliable and meet the requirements of the user.
8. Usability: The system must be easy to use and understand.
9. Compatibility: The system must be compatible with other systems.
10. Compliance: The system must comply with all applicable laws and regulations.

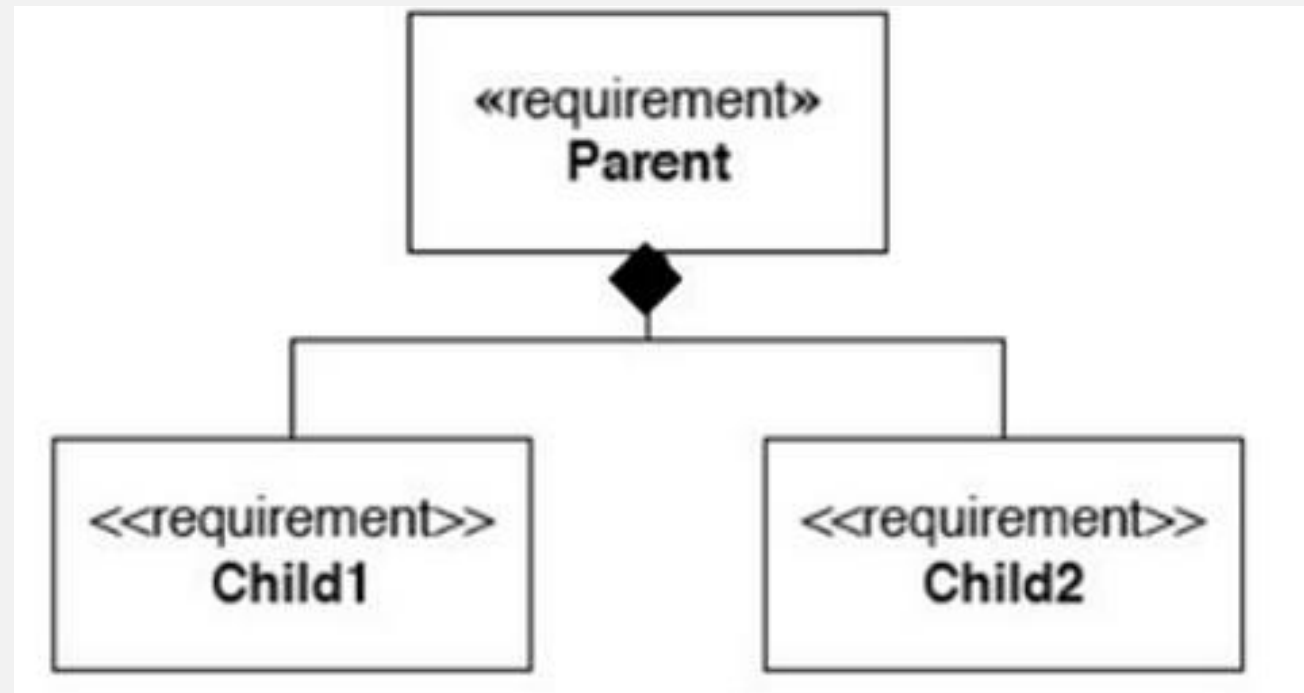
Requirement Diagram - Types

Η σχέση **Containment** μας επιτρέπει διαχωρισμός των σύνθετων απαιτήσεων σε απλούστερες απαιτήσεις με την μορφή μιας ιεραρχικής δομής.



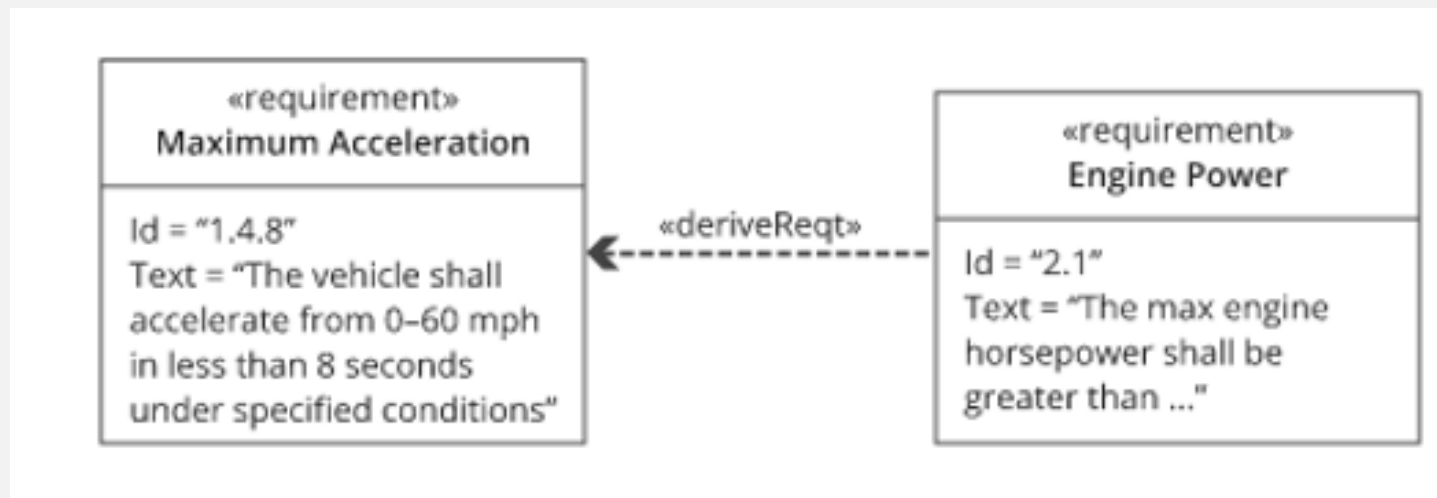
Requirement Diagram - Types

Η σχέση **Composition** μας επιτρέπει διαχωρισμό απαιτήσεων στα επιμέρους απαιτήσεις με την μορφή μιας ιεραρχικής δομής και την ένδειξη εξάρτησης παιδιού από τον πατέρα.



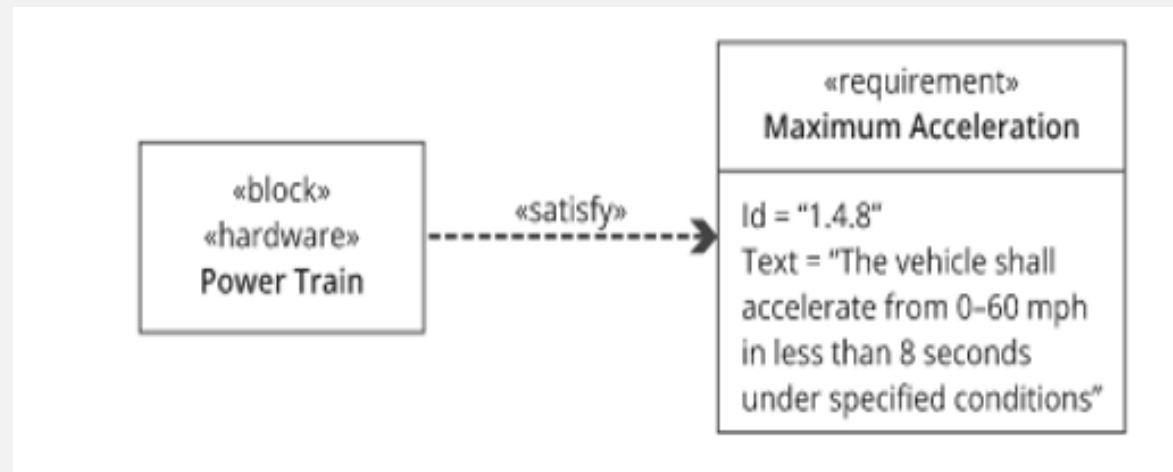
Requirement Diagram - Types

Η σχέση **Derive** αντιπροσωπεύει μια σχέση μεταξύ απαιτήσεων στο ίδιο επίπεδο της ιεραρχίας αλλά σε διαφορετικά επίπεδα αφαίρεσης. Για παράδειγμα, οι απαιτήσεις υλικού ή λογισμικού που καθορίζονται αρχικά από την ομάδα μηχανικής συστημάτων μπορεί να αναλυθούν από την ομάδα υλικού ή λογισμικού για να εξαχθούν πιο λεπτομερείς απαιτήσεις που αντικατοπτρίζουν πρόσθετους παράγοντες ή περιορισμούς υλοποίησης.



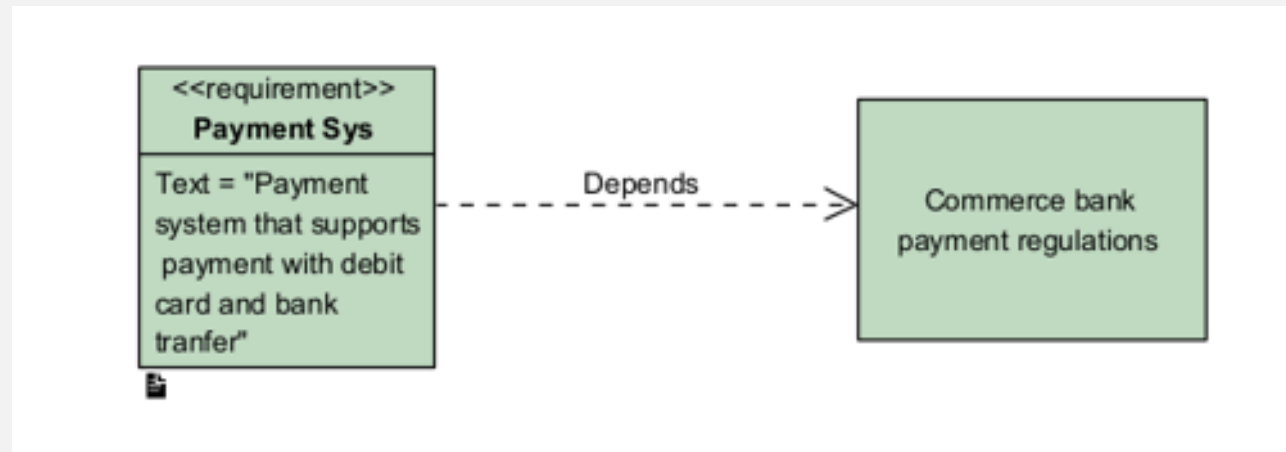
Requirement Diagram - Types

Η σχέση **Satisfy** περιγράφει πώς μια ιδέα σχεδιασμού ή μοντέλου υλοποίησης ικανοποιεί μία ή περισσότερες απαιτήσεις. Ένα σημαντικό σημείο, ωστόσο, είναι ότι ένας ισχυρισμός δεν αποτελεί απόδειξη. Η σχέση ικανοποιεί είναι απλώς ένας μηχανισμός για την κατανομή μιας απαίτησης σε μια δομή.



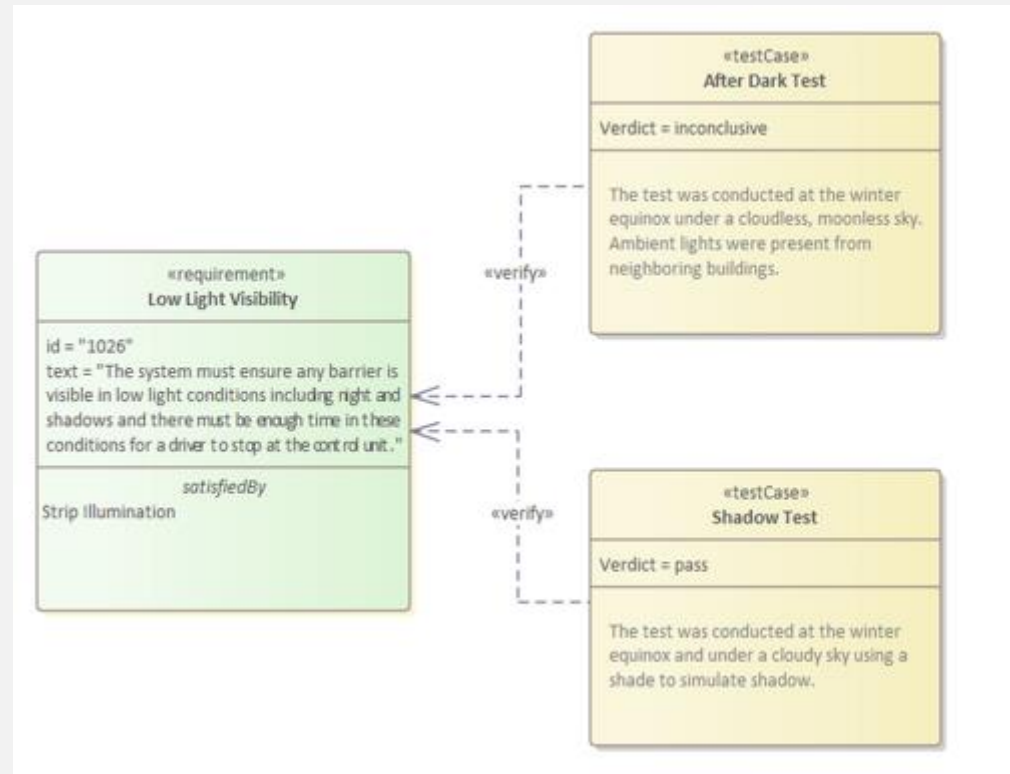
Requirement Diagram - Types

Η σχέση Dependency χρησιμοποιείται για να δείξει ότι μια Απαίτηση βασίζεται σε ένα άλλο στοιχείο (Model, απαίτηση).



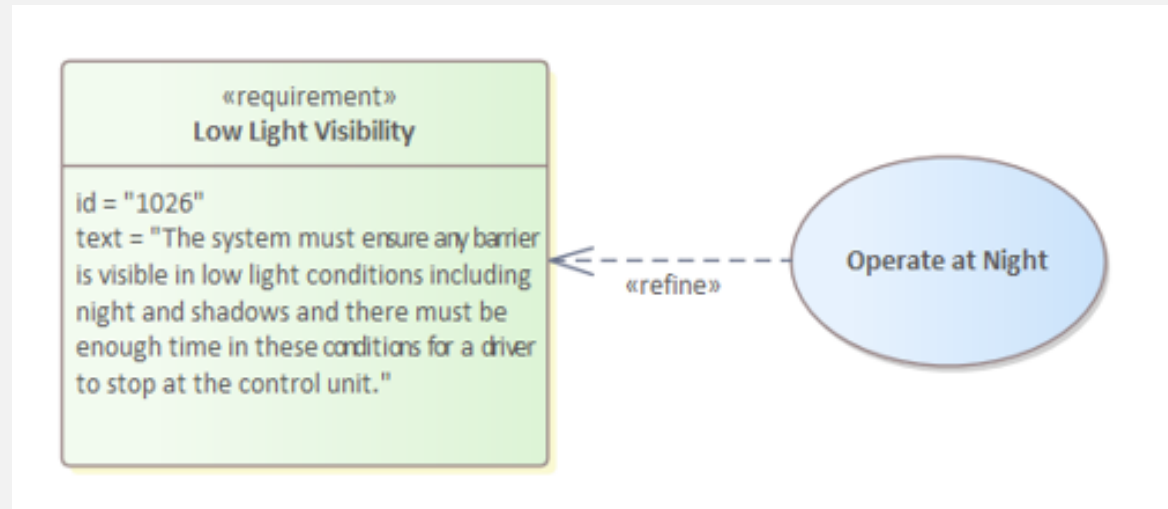
Requirement Diagram - Types

Η σχέση verify ορίζει πώς ένα test case ή άλλο μοντέλου επαληθεύει μια απαίτηση.



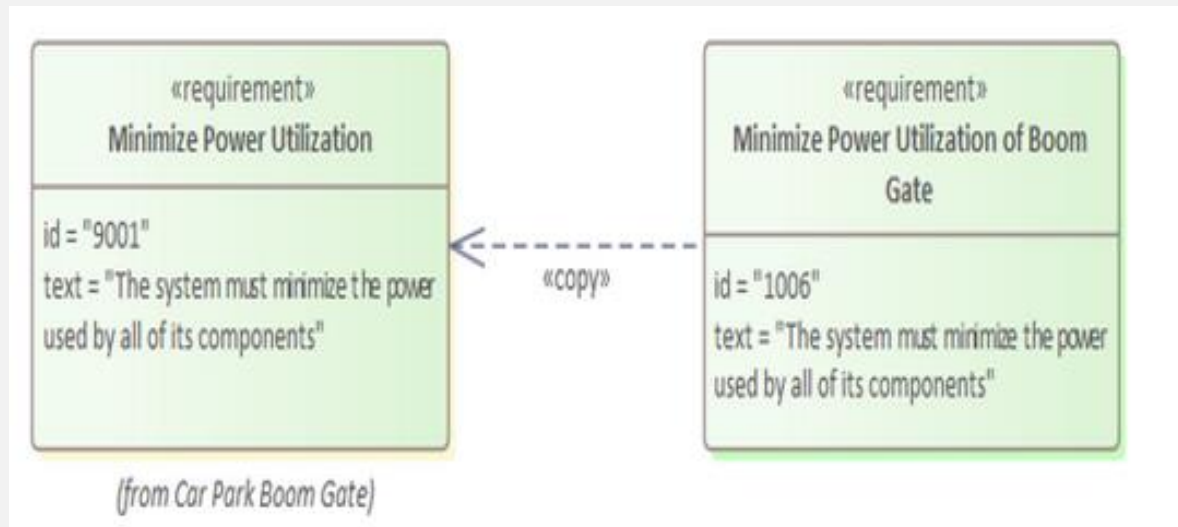
Requirement Diagram - Types

Η σχέση **Refine** είναι μια σχέση μεταξύ μιας απαίτησης και ενός άλλου στοιχείου model που προσθέτει βελτίωση ή πρόσθετες πληροφορίες που βοηθούν στην αποσαφήνιση της απαίτησης, ώστε το νόημά της να είναι πιο εμφανές.



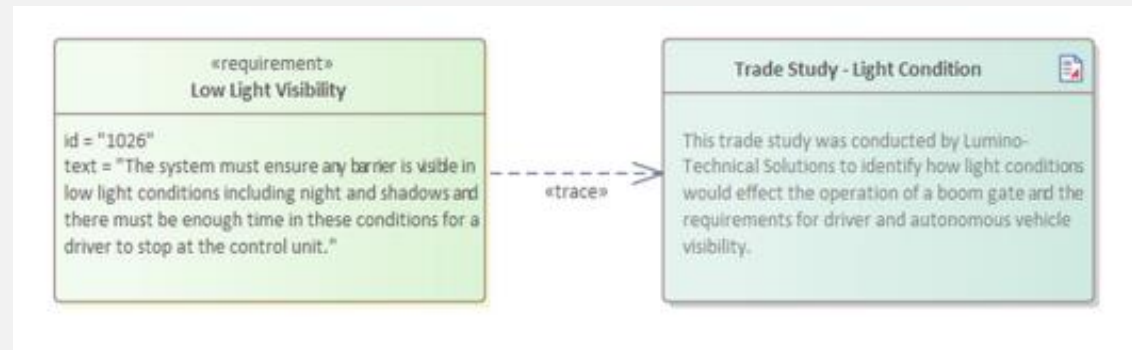
Requirement Diagram - Types

Η σχέση **Copy** γίνεται όταν υπάρχει πραγματική ανάγκη για επαναχρησιμοποίηση απαίτησης.



Requirement Diagram - Types

Η σχέση **Trace** παρέχει μια σχέση γενικού σκοπού μεταξύ μιας απαίτησης και οποιουδήποτε άλλου model.
Η σχέση μπορεί να είναι χρήσιμη για τη συσχέτιση απαιτήσεων με την μορφή documentation.



UML (Unified Modeling Language)

Use Cases – Περιπτώσεις Χρήσης



Use Cases – Περιπτώσεις Χρήσης

Ένα διάγραμμα χρήσης (use case) UML είναι η κύρια μορφή απαιτήσεων για ένα νέο λογισμικό πρόγραμμα που βρίσκεται υπό ανάπτυξη. Οι χρήσεις περιγράφουν την αναμενόμενη συμπεριφορά (what), και όχι τον ακριβή τρόπο με τον οποίο συμβαίνει αυτό (how). Αφού προσδιοριστούν οι χρήσεις, μπορούν να αναπαρασταθούν τόσο με κείμενο όσο και με οπτικό τρόπο (δηλαδή με διάγραμμα χρήσης). Έννοια-κλειδί του διαγράμματος χρήσης είναι ότι μας βοηθά να σχεδιάσουμε ένα σύστημα από την οπτική γωνία του τελικού χρήστη. Πρόκειται για μια αποτελεσματική τεχνική για την επικοινωνία της συμπεριφοράς του συστήματος με τους όρους του χρήστη, προσδιορίζοντας όλη την εξωτερικά ορατή συμπεριφορά του συστήματος. Ένα διάγραμμα χρήσης είναι συνήθως απλό. Δεν εμφανίζει τις λεπτομέρειες των χρήσεων:

- Απλώς περιλαμβάνει μια περίληψη των σχέσεων μεταξύ των χρήσεων, των χρηστών (actors) και του συστήματος.
- Δεν εμφανίζει τη σειρά με την οποία εκτελούνται τα βήματα για την επίτευξη των στόχων κάθε χρήσης.

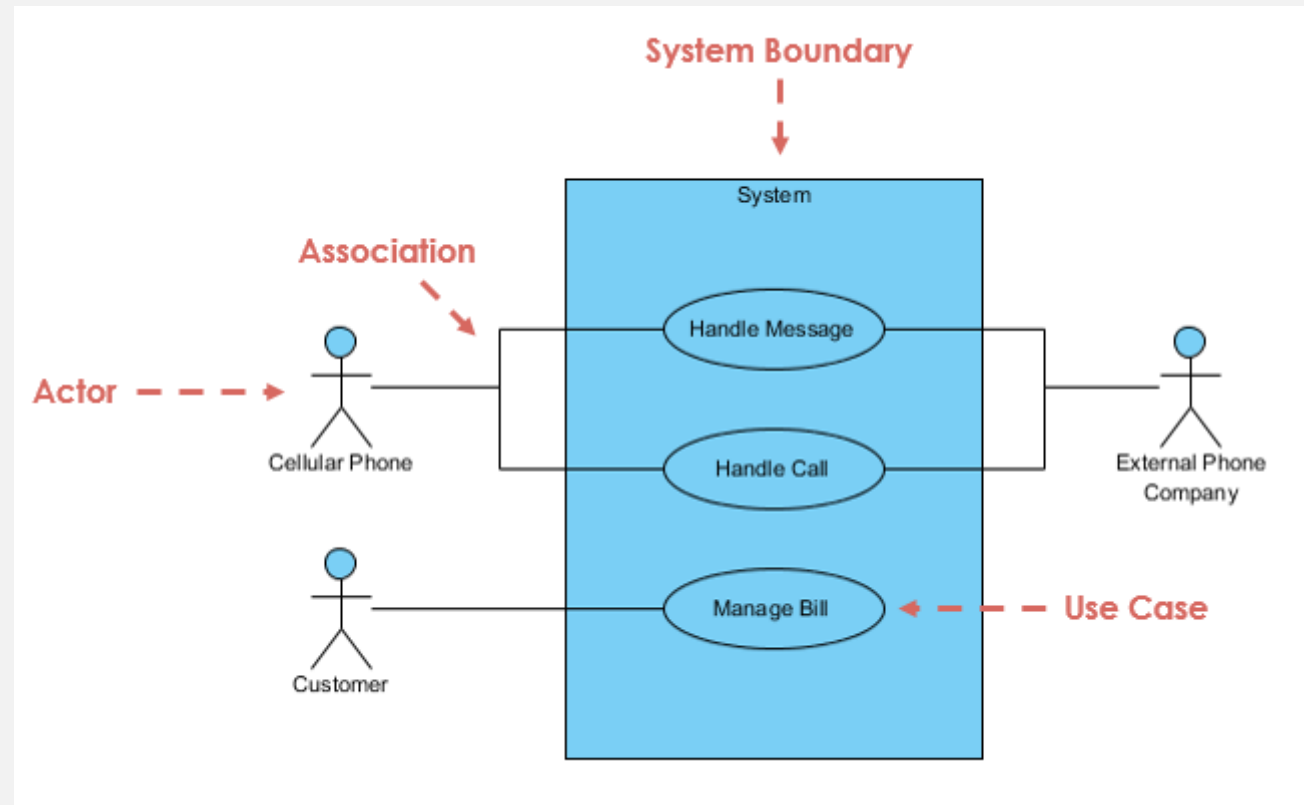
Use Cases – Περιπτώσεις Χρήσης

Τα διαγράμματα χρήσης (use case diagrams) αναπτύσσονται συνήθως στο αρχικό στάδιο της ανάπτυξης, και οι άνθρωποι συχνά εφαρμόζουν το μοντελοποίησης των χρήσεων (use case modeling) για τους ακόλουθους σκοπούς:

- Καθορισμός του πλαισίου ενός συστήματος
- Καταγραφή των απαιτήσεων ενός συστήματος
- Επικύρωση της αρχιτεκτονικής ενός συστήματος
- Καθοδήγηση της υλοποίησης και δημιουργία δοκιμαστικών περιπτώσεων
- Αναπτύσσονται από αναλυτές σε συνεργασία με ειδικούς του τομέα

Use Cases – Περιπτώσεις Χρήσης

Παράδειγμα



Use Cases – Κύριες έννοιες

Actor



- Κάποιος αλληλεπιδρά με τη χρήση περίπτωσης (συστηματική λειτουργία).
- Ονομασμένο από ουσιαστικό.
- Ο ρόλος του ηθοποιού στην επιχείρηση.
- Παρόμοιο με την έννοια του χρήστη, αλλά ένας χρήστης μπορεί να διαδραματίζει διάφορους ρόλους.
- Για παράδειγμα,
 - ένας καθηγητής μπορεί να είναι εκπαιδευτής αλλά και ερευνητής,
 - διαδραματίζοντας δύο ρόλους με δύο συστήματα.
- Ο “Actor” ενεργοποιεί χρήση περίπτωσης (περιπτώσεων).
- Ο “Actor” έχει ευθύνη απέναντι στο σύστημα (εισαγωγές), και ο ηθοποιός έχει προσδοκίες από το σύστημα (εξόδους).

Use Cases – Κύριες έννοιες

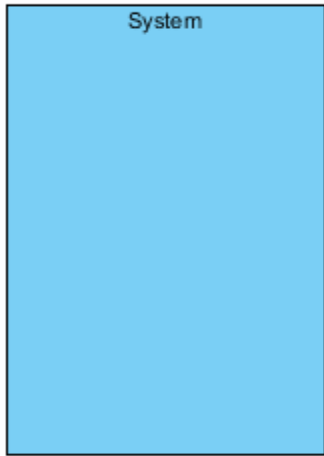
Use Case



- Συστηματική λειτουργία (διαδικασία - αυτοματοποιημένη ή χειροκίνητη)
- Ονομασμένη από ρήμα + ουσιαστικό (ή φράση ουσιαστικού).
- Π.χ. Κάνε κάτι.
- Κάθε “Actor” πρέπει να συνδέεται με μια περίπτωση χρήσης, ενώ κάποιες περιπτώσεις χρήσης ενδέχεται να μην συνδέονται με ηθοποιούς.

Use Cases – Κύριες έννοιες

Boundary System



- Ο όρος του συστήματος μπορεί να είναι πιθανώς το σύνολο του συστήματος, όπως καθορίζεται στο έγγραφο απαιτήσεων.
- Για μεγάλα και πολύπλοκα συστήματα, κάθε μονάδα μπορεί να αποτελεί τον όριο του συστήματος.
- Για παράδειγμα, για ένα σύστημα ERP για μια οργάνωση, κάθε ένα από τα modules, όπως προσωπικό, μισθοδοσία, λογιστικό, κλπ.,
 - μπορεί να αποτελέσει το όριο του συστήματος για τις περιπτώσεις χρήσης που είναι συγκεκριμένες για κάθε μία από αυτές τις επιχειρηματικές λειτουργίες.
- Το σύνολο του συστήματος μπορεί να εκτείνεται σε όλα αυτά τα modules, αποτελώντας το συνολικό όριο του συστήματος.

Use Cases – Κύριες έννοιες

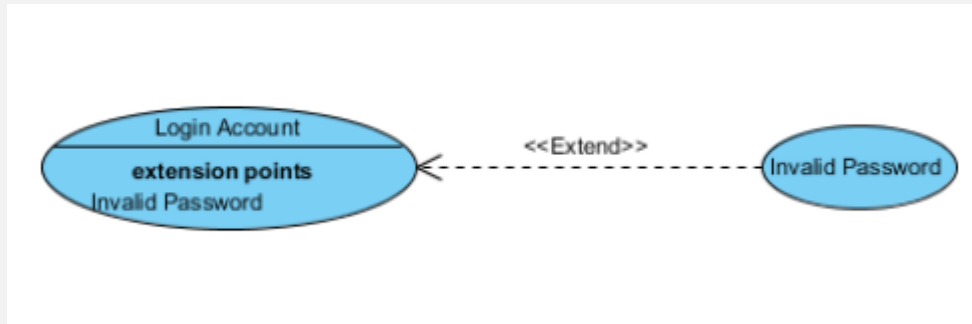
Communication Link



- Η συμμετοχή ενός “Actor” σε μια περίπτωση χρήσης εμφανίζεται συνδέοντας έναν ηθοποιό με μια περίπτωση χρήσης με έναν συμπαγή σύνδεσμο.
- Οι ηθοποιοί μπορεί να συνδεθούν με τις περιπτώσεις χρήσης με συσχετίσεις, υποδεικνύοντας ότι ο ηθοποιός και η περίπτωση χρήσης επικοινωνούν μεταξύ τους χρησιμοποιώντας μηνύματα.

Use Cases – Κύριες σχέσεις

Extends



- Υποδηλώνει ότι μια περίπτωση χρήσης "Άκυρος Κωδικός" μπορεί να περιλαμβάνει (υπό την προϋπόθεση που καθορίζεται στην επέκταση) τη συμπεριφορά που καθορίζεται από την βασική περίπτωση χρήσης "Είσοδος στον Λογαριασμό". Απεικονίζεται με ένα κατευθυνόμενο βέλος με παύλες.
- Η άκρη της αιχμής του βέλους δείχνει προς τη βασική περίπτωση χρήσης, και η περίπτωση χρήσης του παιδιού συνδέεται στη βάση του βέλους.
- Ο όρος "<<extends>>" ταυτοποιεί τη σχέση ως σχέση επέκτασης.

Use Cases – Κύριες σχέσεις

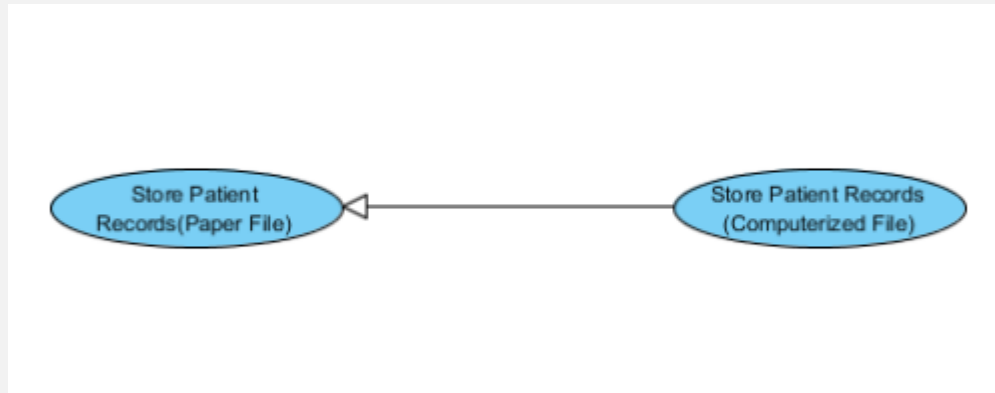
Include



- Όταν μια περίπτωση χρήσης απεικονίζεται ως χρησιμοποιώντας τη λειτουργικότητα μιας άλλης περίπτωσης χρήσης, η σχέση μεταξύ των περιπτώσεων χρήσης ονομάζεται σχέση “include” ή “uses”.
- Μια περίπτωση χρήσης περιλαμβάνει τη λειτουργικότητα που περιγράφεται σε μια άλλη περίπτωση χρήσης ως μέρος της ροής της επιχειρηματικής διαδικασίας της.
- Μια σχέση “uses” από τη βασική περίπτωση χρήσης προς την περίπτωση χρήσης του παιδιού υποδηλώνει ότι μια περίπτωση της βασικής περίπτωσης χρήσης θα περιλαμβάνει τη συμπεριφορά που καθορίζεται στην περίπτωση χρήσης του παιδιού.
- Η στερεότυπος "<<include>>" ταυτοποιεί τη σχέση ως σχέση “include”.

Use Cases – Κύριες σχέσεις

Generalization



- Η σχέση γενίκευσης (generalization relationship) είναι μια σχέση γονέα-παιδιού μεταξύ περιπτώσεων χρήσης.
- Η περίπτωση χρήσης του παιδιού είναι μια ενίσχυση της περίπτωσης χρήσης του γονέα.
- Η γενίκευση απεικονίζεται ως ένα κατευθυνόμενο βέλος με την άκρη σε μορφή τριγώνου ως αιχμή.
- Η περίπτωση χρήσης του παιδιού συνδέεται στη βάση του βέλους, και η άκρη του βέλους συνδέεται με την περίπτωση χρήσης του γονέα.

Ασκήσεις

A transportation company accepts customers for parcel shipments. For customers, it keeps their address, phone number, and a customer code assigned by the company itself. Furthermore, it categorizes customers into two groups: individuals and businesses. For individuals, it also keeps the customer's full name and ID number. For businesses, it records their company name, VAT number, and tax status. Each customer can make multiple shipments, while a shipment can belong to only one customer. For shipments, the company maintains a shipment ID, the recipient's name and address, and the sender's information. Additionally, it tracks the weight, cost, and priority of the shipments. Shipments are sent collectively using trucks. For shipping, information about the number of shipments, the departure date, the truck number for handling the shipment, and the driver's details are recorded.

Ασκήσεις

Μια εταιρία ταχυμεταφορών δέχεται πελάτες για την αποστολή δεμάτων. Για τους πελάτες της κρατάει την διεύθυνση, το τηλέφωνο και ένα κωδικό πελάτη που τον καταχωρεί η ίδια η εταιρία. Επιπλέον, χωρίζει τους πελάτες σε δύο κατηγορίες, ιδιώτες και επιχειρήσεις. Για τους ιδιώτες κρατάει επιπρόσθετα το ονοματεπώνυμο του πελάτη και τον αριθμό ταυτότητάς του. Για τις επιχειρήσεις κρατάει την επωνυμία, το ΑΦΜ και την φορολογική τους κατάσταση. Κάθε πελάτης μπορεί να κάνει πολλές ταχυμεταφορές, ενώ μια ταχυμεταφορά μπορεί να ανήκει μόνο σε έναν πελάτη.

Για τις ταχυμεταφορές η εταιρία κρατάει ένα αναγνωριστικό κωδικό ταχυμεταφοράς, το όνομα και την διεύθυνση του παραλήπτη καθώς και τον αποστολέα. Επιπλέον κρατάει για τις ταχυμεταφορές το βάρος, το κόστος και την προτεραιότητα τους. Οι ταχυμεταφορές αποστέλλονται μαζικά με φορτηγά. Για την αποστολή κρατείται η πληροφορία του πλήθους των ταχυμεταφορών, η ημερομηνία αναχώρησης, καθώς και ο αριθμός του φορτηγού που θα διεκπεραιώσει την αποστολή όπως και ο οδηγός του.

Χρήσιμα Links

- **Visual Paradigm Textual Analysis**

- <https://www.visual-paradigm.com/tutorials/how-to-use-textual-analysis.jsp>
- https://www.visual-paradigm.com/support/documents/vpuserguide/94/1288_textualanaly.html

- **Visual Paradigm Requirements Diagram**

- https://www.visual-paradigm.com/support/documents/vpuserguide/94/158/6516_drawingrequi.html
- https://www.visual-paradigm.com/support/documents/vpuserguide/94/158_requirementd.html
- <https://www.geeksforgeeks.org/functional-vs-non-functional-requirements/>
- <https://visuresolutions.com/requirements-management-traceability-guide/non-functional-requirements/>

- **Visual Paradigm Use Case Diagram**

- <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-use-case-diagram/>

Thank you

