

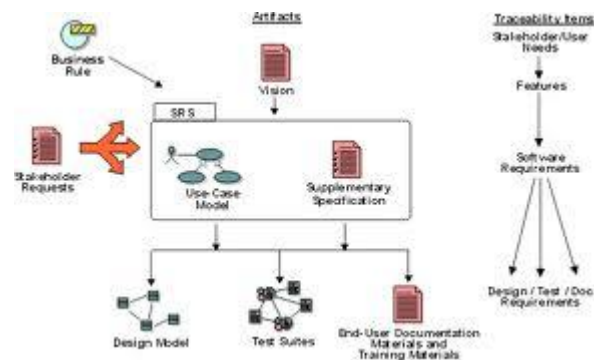


## ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Μελέτη Ιχνηλασιμότητας στην οριζόντια ανάπτυξη της Συμπεριφοράς  
–Μια νέα προσέγγιση

Οριζόντια ανάπτυξη βάσει συμβάντων

Μετρικές συμβάντων



Των φοιτητών

Κουρουλέας Γεώργιος , Πετρίδης Χρήστος

Αρ. Μητρώου: 052862 , 052850

Επιβλέπων καθηγητής

Δεληγιάννης Ιγνάτιος

## ΠΕΡΙΛΗΨΗ

Η ιχνηλασιμότητα είναι χρήσιμος τρόπος για τον εντοπισμό εννοιών και ιχνών καθ' όλη τη διάρκεια ζωής του λογισμικού. Μπορεί να ορισθεί ως το σύνολο των σχέσεων που υπάρχουν μεταξύ των πληροφοριών σε ένα λογισμικό. Μια σχέση για παράδειγμα θα μπορούσε να είναι αυτή που υπάρχει ανάμεσα σε ένα κομμάτι του πηγαίου κώδικα και σε μια δοκιμή (test case) του λογισμικού. Πιο συγκεκριμένα έχουμε 2 μορφές ιχνηλασιμότητας, την οριζόντια και την κάθετη. Η κάθετη ιχνηλασιμότητα εξασφαλίζει ότι οι απαιτήσεις είναι ανιχνεύσιμες στα συστατικά του συστήματος που υλοποιήθηκε και το αντίστροφο, ενώ η οριζόντια ιχνηλασιμότητα εξασφαλίζει ότι οι απαιτήσεις είναι ανιχνεύσιμες από στις περιπτώσεις δοκιμής. Εξασφαλίζει ότι τηρούνται όλες οι απαιτήσεις που πράγματι έχουν δοκιμαστεί και διευκολύνει την προσαρμογή των περιπτώσεων δοκιμών σε μεταβολές που γίνονται στα έγγραφα απαιτήσεων και αντιστρόφως. Σε αυτήν την πτυχιακή θα διερευνήσουμε την ιχνηλασιμότητα και κυρίως την οριζόντια, τις μετρικές που μπορούν να εφαρμοστούν πάνω στην οριζόντια ιχνηλασιμότητα, θα σχεδιάσουμε μια Περίπτωση Χρήσης αλλά και θα εφαρμόσουμε ένα πείραμα με τη βοήθεια σπουδαστών που μέσα από αυτό θα βγάλουμε συμπεράσματα για το αν η οριζόντια ιχνηλασιμότητα είναι το ίδιο εύχρηστη με την κάθετη.

## ABSTRACT

Traceability is a useful way in order to trace various concepts and traces during the lifetime of the software. It can be defined as the set of relationships that exist between the informations of a software. A relationship, for example could be the very one that exist between a part of the source code and a test case of the software. We have 2 kinds of traceability : horizontal, and vertical traceability. Vertical traceability ensures that the requirements are traceable throughout the components of the software system that was implemented and via versa, horizontal traceability ensures that the requiremenets are traceable throughout the test cases. It ensures that all requirements that have been tested are met and makes easier the versatility of the test cases in the changes of test cases in the requirements and via versa. In this paper, we are going to look deeper into traceability and mainly on the horizontal kind, the metrics that can be used onto horizontal traceability, we are going to build a test case and last but not least we are going to look into an experiment that we run with the help of some students who are a part of a Software Engineering class. Through this experiment, we are going to draw conclusions about if horizontal traceability is as useful and easy to adopt as the vertical one ( that is more widely used).

## **ΕΥΧΑΡΙΣΤΙΕΣ**

Ευχαριστούμε τον κ. Δεληγιάννη Ιγνάτιο για την πολύτιμη και σημαντική βοήθεια που μας παρείχε στην διεκπεραίωση της πτυχιακής, καθώς και τον κύριο Τσάνταλη για τον parser ( `myXmiReader` ) που μας παρείχε αλλά και για την βοήθεια πάνω στη μελέτη του πειράματος.

## ΠΕΡΙΕΧΟΜΕΝΑ

ΠΕΡΙΛΗΨΗ.....	2
ABSTRACT.....	3
ΕΥΧΑΡΙΣΤΙΕΣ.....	4
ΠΕΡΙΕΧΟΜΕΝΑ.....	5
ΠΡΟΛΟΓΟΣ – ΕΙΣΑΓΩΓΗ.....	8
ΚΕΦΑΛΑΙΟ 1.....	10
ΙΧΝΗΛΑΣΙΜΟΤΗΤΑ.....	10
1.1. Ορισμοί.....	10
Ιχνηλασιμότητα (Traceability).....	21
1.2. Οι διάφορες μορφές της ιχνηλασιμότητας.....	23
1.2.1. Οριζόντια και κάθετη ιχνηλασιμότητα.....	23
1.3. Κάθετη Ιχνηλασιμότητα.....	24
1.4. Οριζόντια Ιχνηλασιμότητα.....	28
1.5. Τεχνικές Ιχνηλασιμότητας (Traceability Techniques).....	34
1.6. Διάγραμμα Ακολουθίας Συστήματος SSD.....	50
1.7. Διάγραμμα Ακολουθίας (Sequence Diagram – SD).....	52
1.8. Διάγραμμα Κλάσεων (Class Diagram).....	53
ΚΕΦΑΛΑΙΟ 2.....	56
Περίπτωση Χρήσης.....	56
Εννοιολογικό Μοντέλο.....	58
Διάγραμμα Ακολουθίας.....	59

Διάγραμμα Κλάσεων.....	60
Κώδικας:.....	61
ΚΕΦΑΛΑΙΟ 3 .....	70
Μετρικές .....	70
Εισαγωγή.....	70
3.1. Βασικές έννοιες που σχετίζονται με τις μετρικές.....	71
3.2. Τι είναι οι Μετρικές και λίγα λόγια για αυτές.....	72
3.3. Τύποι Μετρικών.....	73
3.3.1. Μετρικές προϊόντων (Product Metrics).....	73
3.3.2. Διαδικαστικές Μετρικές (Process Metrics) .....	74
3.3.3. Μετρικές Απαιτήσεων( Requirements Metrics ) .....	75
3.4. Γιατί είναι απαραίτητες οι Μετρικές στην σύγχρονη ανάπτυξη λογισμικού .....	79
3.5. Περαιτέρω Ανάλυση των Μετρικών.....	80
3.6. Βασικές Υπάρχουσες Μετρικές.....	82
3.7.Μετρικές που μπορούν να χρησιμοποιηθούν στην οριζόντια / κάθετη ιχνηλασιμότητα. 84	
Metrics for Model .....	92
Metrics for Class.....	93
Metrics for Message .....	93
ΕΠΙΛΟΓΟΣ .....	94
ΚΕΦΑΛΑΙΟ 4 .....	96
ΠΕΙΡΑΜΑ.....	96
Εισαγωγή :.....	96
ΟΜΑΔΑ Α' .....	97

ΔΙΑΓΡΑΜΜΑΤΑ ΑΚΟΛΟΥΘΙΑΣ Α΄ ΟΜΑΔΑΣ.....	98
ΟΜΑΔΑ Β΄ .....	105
ΔΙΑΓΡΑΜΜΑΤΑ ΑΚΟΛΟΥΘΙΑΣ Β΄ ΟΜΑΔΑΣ.....	106
Παραγόμενα Στατιστικά Στοιχεία : .....	113
Συμπεράσματα Πειράματος : .....	114
Επίλογος - Συμπεράσματα και πιθανή μελλοντική δουλειά .....	115
ΒΙΒΛΙΟΓΡΑΦΙΑ.....	116
ΠΑΡΑΡΤΗΜΑ Α΄ .....	122
ΧΜΙ ΝΟ.2.....	122
ΠΑΡΑΡΤΗΜΑ Β΄ .....	161
Κώδικας Φοιτητή ΝΟ.2 .....	161
ΠΑΡΑΡΤΗΜΑ Γ΄.....	163
Κώδικας Πειράματος .....	163

## ΠΡΟΛΟΓΟΣ – ΕΙΣΑΓΩΓΗ

Για χρόνια τώρα, όταν ακούγαμε για ιχνηλασιμότητα, το μυαλό μας πήγαινε στην κάθετη ιχνηλασιμότητα θεωρώντας ότι είναι η κυρίαρχη μορφή της ιχνηλασιμότητας, ίσως και γιατί είναι και η πιο βασική μορφή της ιχνηλασιμότητας. Λίγοι έχουν ασχοληθεί με την οριζόντια μορφή της, καθώς και διάφορες επεκτάσεις της και μετρικές. Εμείς αυτό θα επιχειρήσουμε σε αυτήν την πτυχιακή. Ας ορίσουμε λοιπόν συνοπτικά τις δύο έννοιες, ξεκινώντας αρχικά από έναν γενικό ορισμό της ιχνηλασιμότητας.

Η ιχνηλασιμότητα γενικά μπορεί να ορισθεί ως το σύνολο των σχέσεων που υπάρχουν μεταξύ των στοιχείων πληροφοριών σε ένα έργο λογισμικού.

Η ιχνηλασιμότητα σε κάθε φάση της διαδικασίας ανάπτυξης λογισμικού αναφέρεται ως κάθετη, διότι περιέχει πληροφορίες που σχετίζονται με διάφορα στοιχεία της ίδιας φάσης

Η Οριζόντια ιχνηλασιμότητα ορίζεται ως οι πληροφορίες ιχνηλασιμότητας που δείχνουν τα διαδοχικά στάδια του προϊόντος ή τον προκάτοχό τους στην ανάπτυξη του κύκλου ζωής του λογισμικού.

Η πτυχιακή μας είναι οργανωμένη ως εξής:

Στο Κεφάλαιο 1, μιλάμε για την ιχνηλασιμότητα γενικά αλλά και πιο ειδικά για τα 2 είδη της, της κάθετη αλλά και την οριζόντια ( η οποία μας ενδιαφέρει και παραπάνω, άρα και αναπτύσσουμε περισσότερο ).

Στο κεφάλαιο 2, αναπτύσσουμε μια Περίπτωση Χρήσης μαζί με τα απαραίτητα Διαγράμματα Κλάσεων, Διαγράμματα Ακολουθίας, αλλά και τον πηγαίο κώδικα.

Στο κεφάλαιο 3, αναπτύσσουμε τις μετρικές, γενικά αλλά και ειδικότερα τις πιο γνωστές από αυτές, ενώ ερευνάμε και ποιες από αυτές μπορούν να εφαρμοστούν στην οριζόντια ιχνηλασιμότητα, σχεδιάζοντας και συγκεκριμένα παραδείγματα για αυτές. Επίσης, δίνουμε και κάποιες μετρικές οι οποίες θα μας απασχολήσουν πιθανόν μελλοντικά.

Τέλος, στο κεφάλαιο 4, υπάρχει το Πείραμά μας με τους φοιτητές από το μάθημα «Μεθοδολογίες Προγραμματισμού II» οι οποίοι είναι χωρισμένοι σε 2 ομάδες και δουλεύουν αντίστοιχα με την οριζόντια ή κάθετη ιχνηλασιμότητα καθώς και τα Διαγράμματα Ακολουθίας που έχουν σχεδιάσει οι διάφοροι φοιτητές, καθώς και τον πηγαίο κώδικα του βέλτιστου



γραπτού ( Παράρτημα ). Τελικά βγάζουμε διάφορα συμπεράσματα σχετικά με τα 2 είδη της ιχνηλασιμότητας .

# ΚΕΦΑΛΑΙΟ 1

## ΙΧΝΗΛΑΣΙΜΟΤΗΤΑ

### 1.1. Ορισμοί

Η ιχνηλασιμότητα μπορεί να ορισθεί ως το σύνολο των σχέσεων που υπάρχουν μεταξύ των στοιχείων πληροφοριών σε ένα έργο λογισμικού. Μια σχέση για παράδειγμα θα μπορούσε να είναι αυτή που υπάρχει ανάμεσα σε ένα κομμάτι του πηγαίου κώδικα και σε μια δοκιμή (test case) του λογισμικού.

Στο λεξικό Webster η ιχνηλασιμότητα ορίζεται ως εξής: Παρακολούθηση από τα ίχνη, τη διαδρομή, ή το μονοπάτι. Η λεπτομερής παρακολούθηση ή η βήμα-βήμα παρακολούθηση της διαδρομής της κυκλοφορίας του ίχνους μιας εργασίας . Παρακολούθηση βήμα προς βήμα της προέλευσης του ίχνους από την αρχή πηγαίνοντας προς τα πίσω κατά τη διάρκεια του ελέγχου. Ανακάλυψη των σημείων, των αποδεικτικών στοιχείων ή από κάποια απομεινάρια. Η ύπαρξη συνδέσμου μεταξύ δύο κόμβων χρησιμοποιείται για την διαμόρφωση μιας σχέσης υπερμέσων που παρέχει ένα ευέλικτο σύνολο τρόπων με τους οποίους η ιχνηλασιμότητα μπορεί να συλληφθεί.

Ένας άλλος ορισμός, πιο κοντά στην ανάπτυξη λογισμικού προέρχεται από το πρότυπο IEEE Γλωσσάρι του λογισμικού Ορολογίας [IEEEGST]: Ο βαθμός στον οποίο μια σχέση μπορεί να δημιουργηθεί μεταξύ δύο ή περισσότερων από τα προϊόντα της αναπτυξιακής διαδικασίας, ιδίως τα προϊόντα που έχουν σχέσεις προκατόχου-διάδοχου (predecessor-successor) ή master-slave μεταξύ τους , για παράδειγμα, ο βαθμός σχέσης των απαιτήσεων και του σχεδιασμού ενός συγκεκριμένου τμήματος λογισμικού. Ο βαθμός στον οποίο κάθε στοιχείο σε ένα προϊόν ανάπτυξης λογισμικού τεκμηριώνει το λόγο του για την ύπαρξή του: για παράδειγμα, ο βαθμός στον οποίο κάθε στοιχείο σε ένα γράφημα φυσαλίδα αναφέρει την απαίτηση η οποία τον ικανοποιεί.

#### **Επτά διαφορετικοί ορισμοί της οριζόντιας ιχνηλασιμότητας:**

Το άρθρο MIL-STD 498 [MILSTD498] διευκρινίζει ότι οι πληροφορίες ιχνηλασιμότητας θα πρέπει να συγκεντρωθούν, αλλά λέει πολύ λίγα για το πώς πρέπει να γίνει αυτό ή τις πληροφορίες που απαιτούνται για είναι η ιχνηλασιμότητα ικανοποιητική.

Κάθε περιγραφή στοιχείου δεδομένων (DID - Data Item Description, έγγραφο που πρέπει να γραφτούν), περιέχει ένα τμήμα ιχνηλασιμότητας που καθορίζει ποια στοιχεία πληροφορίας θα

πρέπει να είναι σε θέση να ανιχνευτούν, αλλά το ίδιο το έγγραφο δεν έχει σαφή ορισμό της ιχνηλασιμότητας. Η IEEE / EIA (ISO / IEC) 12207 [IEEE12207] διευκρινίζει ότι οι πληροφορίες ιχνηλασιμότητας πρέπει να συγκεντρώνονται καθ' όλη τη διαδικασία, αλλά λέει πολύ λίγα για το από πού αποτελούνται αυτές οι πληροφορίες στην πραγματικότητα. Επιπλέον, οι ορισμοί που χρησιμοποιούνται φαίνονται να ποικίλουν ανάλογα με το περιεχόμενο των πληροφοριών ιχνηλασιμότητας. Μπορεί να ειπωθεί, ωστόσο, ότι η όλη διαδικασία πρέπει να είναι εναρμονισμένη στον τελικό στόχο.

Ο Matthias Jarke ορίζει τα εξής [Jarke 1994]: Η ιχνηλασιμότητα απαιτήσεων, άρα, ορίζεται ως η ικανότητα να περιγραφεί και να ακολουθηθεί η ζωή μιας απαίτησης, με κατεύθυνση τόσο προς τα εμπρός και προς τα πίσω, κατά προτίμηση μέσω ολόκληρου του συνόλου του κύκλου ζωής του συστήματος.

Διακρίνονται τέσσερα είδη συνδέσεων ιχνηλασιμότητας συνήθως όσον αφορά τις σχέσεις με τις απαιτήσεις:

Μπροστά από τις απαιτήσεις. Η ευθύνη για το επίτευγμα των απαιτήσεων πρέπει να αποδίδεται στα στοιχεία του συστήματος, έτσι ώστε οι απαιτήσεις να είναι υπεύθυνα προσδιορισμένες και ο αντίκτυπος της αλλαγής των απαιτήσεων να μπορεί να αξιολογηθεί.

Πίσω προς τις απαιτήσεις. Η συμμόρφωση του συστήματος με τις απαιτήσεις πρέπει να επαληθευτεί, καθώς και το λεγόμενο gold plating(σχέδια για τα οποία δεν υπάρχουν απαιτήσεις) πρέπει να αποφευχθεί.

Μπροστά προς τις απαιτήσεις. Οι αλλαγές στις ανάγκες των κατάλληλων τμημάτων, καθώς και σε τεχνικές υποθέσεις, μπορεί να απαιτήσουν μια ριζική επανεκτίμηση της σχετικότητας των απαιτήσεων.

Πίσω από τις απαιτήσεις. Η συνεισφορά δημιουργεί υποθάλπτουσες απαιτήσεις (The contribution structures underlying requirements) οι οποίες είναι ζωτικής σημασίας για την επικύρωση των απαιτήσεων, ιδίως σε ρυθμίσεις πολιτικής.

Επιπλέον, ο Jarke μιλά για την *προ-ιχνηλασιμότητα* και την *μετά-ιχνηλασιμότητα*, με τη διαφορά των δύο να έγκειται στο "χρονοδιάγραμμα" της δημιουργίας των πληροφοριών ιχνηλασιμότητας. Η προ-ιχνηλασιμότητα πραγματοποιείται "στη διαδικασία", ενώ η μετά-ιχνηλασιμότητα είναι η δημιουργία πινάκων ιχνηλασιμότητας στο τέλος της φάσης του προγράμματος.

Στο [Schach 1996], ο Schach μιλά για την ιχνηλασιμότητα στο πλαίσιο των δοκιμών προδιαγραφών ως εξής: "Πρέπει να είναι εφικτός ο εντοπισμός κάθε πρότασης στο έγγραφο προδιαγραφής πίσω σε μια πρόταση που έκανε η ομάδα του πελάτη κατά τη φάση των απαιτήσεων." Για να βοηθηθεί η διαδικασία αυτή, οι απαιτήσεις και τα έγγραφα

προδιαγραφής θα πρέπει να διασταυρώνονται και να κατηγοριοποιούνται. Στο πλαίσιο των δοκιμών του σχεδιασμού, ο Schach ορίζει την ιχνηλασιμότητα, ως την περίπτωση κατά την οποία «κάθε μέρος του σχεδίου μπορεί να συνδέεται με μια δήλωση στο έγγραφο προδιαγραφής». Η προσέγγιση που ακολουθούμε είναι εκείνη κατά την οποία η ιχνηλασιμότητα υφίσταται μεταξύ περισσότερο από μόνο στις απαιτήσεις και τις προδιαγραφές ή τις απαιτήσεις και τον κώδικα, αλλά είναι σχετική όταν κάθε είδους πληροφοριών μπορούν να συνδεθούν μεταξύ τους.

Οι Deutsch και Willis ορίζουν την ιχνηλασιμότητα, [Deutsch 1988] ως εξής: "Το λογισμικό είναι ανιχνεύσιμο όταν είναι εύκολο να βρεθεί όλος ο κώδικας που υλοποιεί μια συγκεκριμένη απαίτηση και, αντιθέτως, όταν είναι εύκολο να βρείτε όλες τις απαιτήσεις που τίθενται σε εφαρμογή από ένα συγκεκριμένο τμήμα του κώδικα. Η ιχνηλασιμότητα είναι σημαντική για τον έλεγχο ορθότητας και κατά τη συντήρηση." Πρόκειται σαφώς για μια άποψη ιχνηλασιμότητας απαιτήσεων, που καθορίζεται σε ένα πλαίσιο παραγόντων ποιότητας λογισμικού.

Στο DoD 2167A, Clause 4.2.8 [Gillies 1992] η ιχνηλασιμότητα ορίζεται ως εξής: "Ιχνηλασιμότητα των απαιτήσεων στο σχέδιο : Θα αναπτυχθούν πίνακες ιχνηλασιμότητας για να δειχθεί η κατανομή των απαιτήσεων από τις προδιαγραφές του συστήματος για τη διαχείριση του λογισμικού των Υπολογιστών, το ανώτερο επίπεδο τμημάτων λογισμικού, το κατώτερο επίπεδο τμημάτων λογισμικού για τις μονάδες από το επίπεδο μονάδας πίσω στο σύστημα προδιαγραφών. Οι πίνακες ιχνηλασιμότητας θα πρέπει να καταγράφονται στις προδιαγραφές απαιτήσεων λογισμικού, έγγραφο του ανώτερου επιπέδου σχεδιασμού του λογισμικού και στο έγγραφο του αναλυτικού σχεδίου του λογισμικού.

Από τον [Sommerville 1995]: "Η Ιχνηλασιμότητα μεταξύ του μοντέλου απαιτήσεων για ένα πρόγραμμα και για το τελικό προϊόν είναι ένα μέσο για την εξασφάλιση της πληρότητας. Επιπλέον, η ιχνηλασιμότητα των τμημάτων της δομής του προγράμματος σε σχέση με τις μοντέλο απαιτήσεων του προγράμματος προωθεί την απλότητα. Ως κύριο χαρακτηριστικό της αξιοπιστίας, η ιχνηλασιμότητα αναφέρεται επίσης στην επάρκεια και την ακρίβεια των υποθέσεων ελέγχου που χρησιμοποιήθηκε για να προσδιοριστεί πόσο καλά ανταποκρίνεται το πρόγραμμα στις απαιτήσεις που ορίστηκαν για αυτό. Με αυτή την έννοια, η ιχνηλασιμότητα είναι μια ποιοτική τεχνική. Γενικά, μπορούμε να σκεφτούμε την ιχνηλασιμότητα από την άποψη μιας αλυσίδας από το μοντέλο απαιτήσεων, μέσω της διαδικασίας ανάπτυξης, έως την εγκατάσταση και την τελική ολοκλήρωση του λογισμικού."

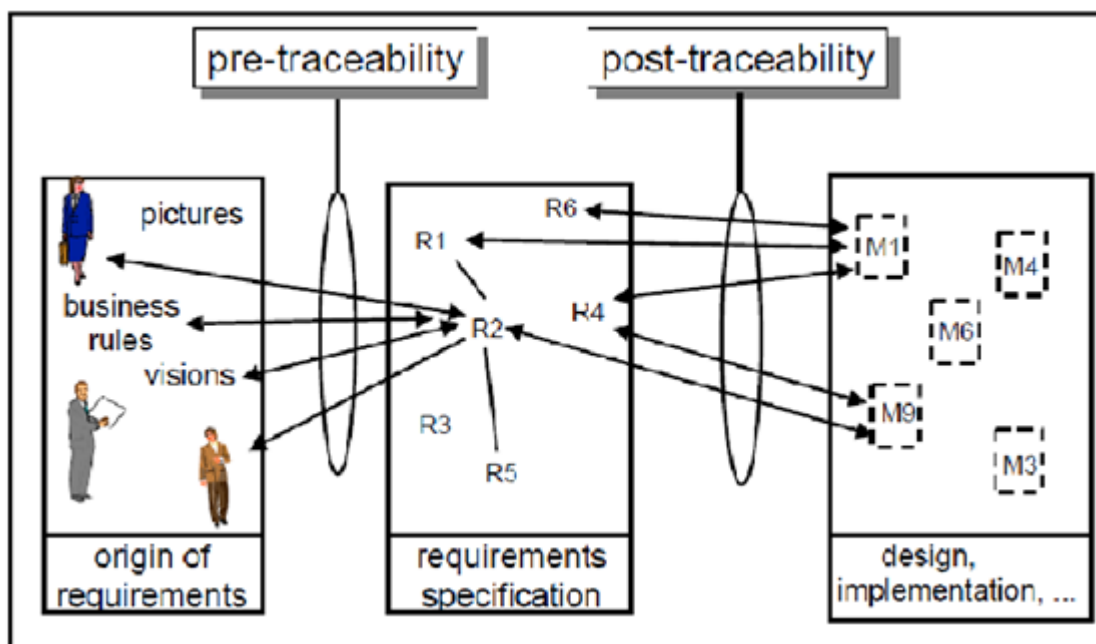
Ο πιο πλήρης ορισμός της ιχνηλασιμότητας απαιτήσεων (Requirements Traceability) δίνεται από Gotel και Finkelstein:

"η ικανότητα να περιγραφεί και να ακολουθηθεί η ζωή μιας απαίτησης, με κατεύθυνση τόσο

προς τα εμπρός όσο και προς τα πίσω (δηλ. από την αρχική του κατάσταση, μέσω της ανάπτυξής του και τις προδιαγραφές, με την επακόλουθη εγκατάσταση του και χρήση, καθώς και μέσω των περιόδων της συνεχούς βελτίωσης και επανάληψη σε οποιοδήποτε από αυτές τις φάσεις). "[Gotel και Finkelstein. 1993]

Όπως αναφέρεται στις αρχές της δεκαετίας ο καθορισμός των απαιτήσεων ιχνηλασιμότητας από τον ΙΕΕΕ Οδηγό των Προδιαγραφών Απαιτήσεων Λογισμικού [IEEE-830, 1984] και τον ορισμό που έδωσε ο [Johnson και οι λοιποί. 1991], η διαφοροποίηση μπορεί να γίνει στη διάκριση μεταξύ των δύο ειδών απαιτήσεων ιχνηλασιμότητας, δηλαδή στις απαιτήσεις προ και μετά της ιχνηλασιμότητας (Σχήμα 1.1.). « Ένα SRS (προδιαγραφή απαιτήσεων λογισμικού- software requirements specification) είναι ανιχνεύσιμο εάν η προέλευση των απαιτήσεών του είναι σαφής και αν διευκολύνει την αναφορά της κάθε απαίτησης στη μελλοντική ανάπτυξη και στην καταγραφή της καλυτέρευσης.»[IEEE-830, 1984].

"... κάθε απαίτηση που πρέπει να επισημανθεί προς την αρχική πηγή της σε άλλα έγγραφα και στο τμήμα λογισμικού που ικανοποιεί την απαίτηση." [Johnson και οι άλλοι. 1991]



Σχήμα 1.1. Απαιτήσεις πριν και μετά την ιχνηλασιμότητα.

Οι απαιτήσεις μετά την ιχνηλασιμότητα υποστηρίζουν κυρίως το μηχανικό των απαιτήσεων και τον μηχανικό του συστήματος για την παρακολούθηση των απαιτήσεων και τη διασφάλιση ότι τηρούνται όλες οι απαιτήσεις και κυλούν σωστά σε όλα τα επίπεδα, τις προδιαγραφές και στο

σχεδιασμό του συστήματος χωρίς να χάνεται και να προστίθεται καμία απαίτηση (cf. [Alford 1980, Flynn and Dorfman 1990, Alford 1990]).

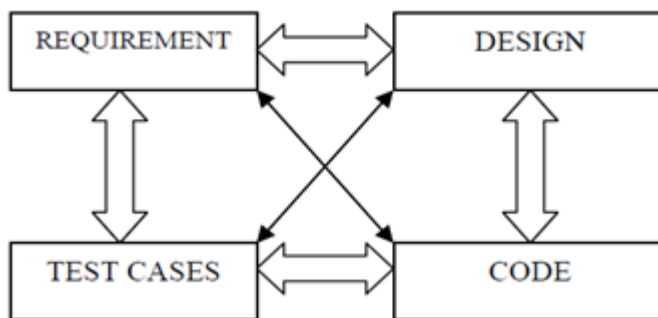
Οι απαιτήσεις πριν από την ιχνηλασιμότητα είναι προαπαιτούμενες για τη διαχείριση της εξέλιξης του συστήματος σε επίπεδο προδιαγραφών. Εξασφαλίζουν ότι η λογική και οι στόχοι πίσω από κάθε απαίτηση και τη σχέση της με σημαντικές πληροφορίες, π.χ., τους οργανωτικούς στόχους, τις επιχειρηματικές διαδικασίες, καταγράφονται. Ως εκ τούτου, του παρέχει την ταυτοποίηση του προσδιορισμού των απαιτήσεων που επηρεάζονται από ένα αίτημα αλλαγής, π.χ. αλλαγή της οργανωτικής πολιτικής.

Κατά συνέπεια, οι απαιτήσεις πριν της ιχνηλασιμότητας είναι εξίσου σημαντικές με τις απαιτήσεις μετά αυτής ειδικά για συστήματα τα οποία είναι ενσωματωμένα σε ένα συνεχώς μεταβαλλόμενο περιβάλλον (βλ. [Gotel και Finkelstein 1993, Ramesh και Edwards 1993, Pohl 1996b]).

Είναι αναπόφευκτο για ένα λογισμικό να υποβάλλεται σε κάποια αλλαγή στη διάρκεια ζωής του. Με κάποιες απαιτήσεις αλλαγών έρχεται η ανάγκη να υπολογιστεί το *scope* (μέγεθος και πολυπλοκότητα) των προτεινόμενων αλλαγών και του σχεδίου για την υλοποίησή τους. Το κύριο πρόβλημα για έναν συντηρητή είναι ότι φαινομενικά οι μικρές αλλαγές μπορούν να «κυματίσουν» σε όλο το σύστημα και να έχουν έναν ουσιαστικό αντίκτυπο αλλού μέσα στο σύστημα. Ο συντηρητής ολοκληρώνει γενικά την αλλαγή με την ανάλυση των υπαρχουσών εξαρτήσεων ή των σχέσεων μεταξύ των τμημάτων λογισμικού που συνθέτουν το σύστημα λογισμικού. Η Ανάλυση αντίκτυπου αλλαγής λογισμικού (*Software change impact analysis*), ή ανάλυση αντίκτυπου για χάρη συντομίας, προσφέρει ιδιαίτερη δύναμη στην κατανόηση και την εφαρμογή της αλλαγής στο σύστημα επειδή παρέχει μια λεπτομερή εξέταση των συνεπειών των αλλαγών στο λογισμικό. Η ανάλυση αντίκτυπου παρέχει τη διαφάνεια στα πιθανά αποτελέσματα των προτεινόμενων αλλαγών προτού να εφαρμοστούν οι πραγματικές αλλαγές. Η δυνατότητα να προσδιοριστεί η πιθανή επίδραση του αντίκτυπου της αλλαγής θα βοηθήσει πολύ τον συντηρητή ή το διαχειριστή για να καθορίσει τις κατάλληλες ενέργειες που θα παρθούν όσον αφορά την απόφαση για την αλλαγή, τα σχέδια προγράμματος, τις εκτιμήσεις δαπανών και των πόρων. Για να εφαρμοστεί ανάλυση αντίκτυπου σε μια ευρύτερη προοπτική είναι αρκετά δύσκολο δεδομένου ότι περιλαμβάνει την ιχνηλασιμότητα μέσα και κατά τα διαφορετικά πρότυπα στον κύκλο της ζωής λογισμικού, όπως από το πρότυπο σχεδίου στο πρότυπο κώδικα. Ο Ramesh ορίζει την ιχνηλασιμότητα ως τη δυνατότητα να επισημανθούν τα εξαρτώμενα στοιχεία μέσα σε ένα μοντέλο και τη δυνατότητα να επισημανθούν τα αντίστοιχα στοιχεία σε άλλα μοντέλα. Τέτοιο είδος ιχνηλασιμότητας καλείται ιχνηλασιμότητα απαιτήσεων. Σύμφωνα με αυτό, οι Turner και Munro υποθέτουν ότι μια ιχνηλασιμότητα συστημάτων υπονοεί ότι όλα τα πρότυπα του λογισμικού ενημερώνονται με συνέπεια.

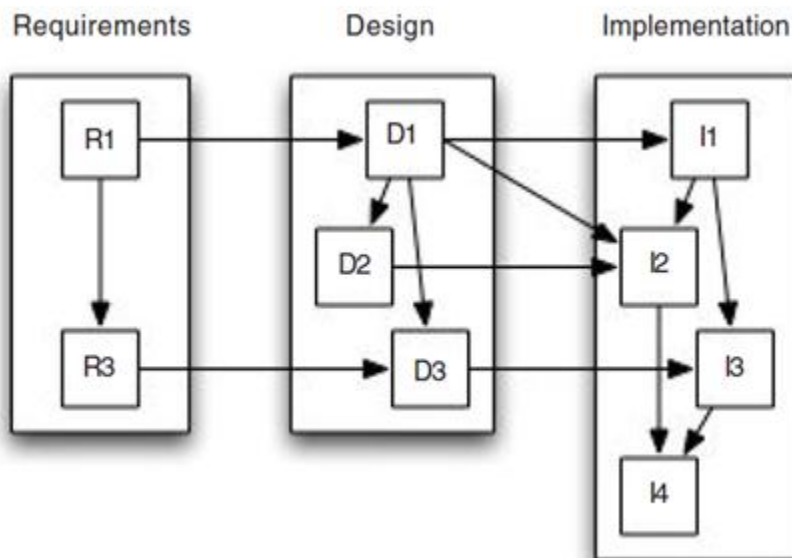
Η έρευνα για την ιχνηλασιμότητα απαιτήσεων έχει εξερευνηθεί ευρέως τις τελευταίες δύο δεκαετίες, από τότε δηλαδή που αυτήν υποστηρίζει πολλές εφαρμογές όπως την τεκμηρίωση εκ νέου (redocumentation), την οπτικοποίηση, την επαναχρησιμοποίηση κλπ. Η ιχνηλασιμότητα είναι θεμελιώδης στη ανάπτυξη λογισμικού και τη συντήρηση ενός μεγάλου συστήματος. Έχει τη δυνατότητα να επισημάνει στοιχεία από τις υψηλού επιπέδου αφαιρέσεις στις χαμηλού επιπέδου αφαιρέσεις π.χ. από μια απαίτηση στον κώδικα εφαρμογής του. Το γεγονός για αυτό το πρότυπο ιχνηλασιμότητας είναι ότι εάν οι συστατικές σχέσεις είναι πάρα πολύ πολύπλοκες, πρέπει να αποσυντεθούν για να κατανοηθούν οι σύνθετες σχέσεις. Από την άλλη, εάν είναι πάρα πολύ κοκκώδεις, είναι δύσκολο να αναδημιουργηθούν στα αναγνωρισμένα, εύκολα κατανοητά προϊόντα εργασίας λογισμικού. Θα επιθυμούσαμε να ερευνήσουμε μια ιχνηλασιμότητα απαιτήσεων για την ανάλυση αντίκτυπου αλλαγής από την οποία πρέπει να είμαστε σε θέση να συλλάβουμε τις επιδράσεις μιας προτεινόμενης αλλαγής. Όταν λέμε μια προτεινόμενη αλλαγή εννοούμε ένα τμήμα στόχων που πρέπει να τροποποιηθεί ως αποτέλεσμα του αιτήματος αλλαγής. Το αίτημα αλλαγής αρχίζει από τον πελάτη ή το εσωτερικό προσωπικό ανάπτυξης λόγω της ανάγκης να γίνει μια αλλαγή στο σύστημα λογισμικού. Πρέπει να μεταφραστεί σε μερικά πιο κατανοητά στοιχεία προτού να μπορέσει να εφαρμοστεί μια ανάλυση αντίκτυπου αλλαγής.

Το σχήμα 1.2. απεικονίζει τις σχέσεις μεταξύ των χειροποίητων αντικειμένων. Τα παχιά βέλη αντιπροσωπεύουν τις άμεσες σχέσεις ενώ τα λεπτά βέλη αντιπροσωπεύουν τις έμμεσες σχέσεις. Και οι άμεσες και έμμεσες σχέσεις μπορούν να προέλθουν από τη στατική ή δυναμική ανάλυση των σχέσεων τμημάτων. Οι άμεσες σχέσεις εφαρμόζουν τις πραγματικές τιμές δύο συστατικών, ενώ οι έμμεσες σχέσεις εφαρμόζουν τις ενδιάμεσες τιμές της σχέσης π.χ. χρησιμοποιώντας μια μεταβατική περάτωση. Οι στατικές σχέσεις είναι ίχνη λογισμικού μεταξύ των τμημάτων ως αποτέλεσμα μιας μελέτης της στατικής ανάλυσης στο πηγαίο κώδικα και άλλων σχετικών προτύπων. Η δυναμική ανάλυση από την άλλη, εξάγεται από την εκτέλεση του λογισμικού για να βρει τα ίχνη όπως η εκτέλεση των περιπτώσεων δοκιμής για να βρει τους επηρεασμένους κώδικες. Ταξινομούμε το μοντέλο μας σε δύο κατηγορίες: την κάθετη και την οριζόντια ιχνηλασιμότητα. Η κάθετη ιχνηλασιμότητα αναφέρεται στην ένωση των εξαρτώμενων στοιχείων μέσα σε ένα μοντέλο και η οριζόντια ανιχνευσιμότητα αναφέρεται στην ένωση των αντίστοιχων στοιχείων μεταξύ των διαφορετικών μοντέλων.



Σχήμα 1.2. Μετά-μοντέλο του συστήματος ιχνηλασιμότητας

### Βασικοί ορισμοί για την εξέλιξη τεχνικών υποστήριξης (Basic Definitions for Evolution Support Techniques)



Σχήμα 1.3. Σχέσεις οριζόντιας και κάθετης ιχνηλασιμότητας

Σε αυτή την ενότητα, παρέχουμε τους ορισμούς για τις ακόλουθες έννοιες της εξέλιξης: έλεγχος εκδόσεων, τη διαχείριση ιχνηλασιμότητας, αλλαγή διάδοσης, αλλαγή ανάλυση των



επιπτώσεων, την αναδιάρθρωση του λογισμικού, τη διαχείριση ασυνέπειας και μετ'επιστροφής μηχανική.

Η ιχνηλασιμότητα ορίζεται ως ο «βαθμός στον οποίο μια σχέση μπορεί να δημιουργηθεί μεταξύ δύο ή περισσότερων από τα προϊόντα της αναπτυξιακής διαδικασίας, ιδίως τα προϊόντα που έχουν προκάτοχο-διάδοχο (predecessor-successor) ή κυρίαρχη-εξαρτώμενη (master-subordinate) σχέση μεταξύ τους. Για παράδειγμα, ο βαθμός στον οποίο οι απαιτήσεις και ο σχεδιασμός ενός συγκεκριμένου χειροποίητου αντικειμένου λογισμικού» (IEEE, 1999).

Όπως φαίνεται στο Σχήμα 1.3., δύο τύποι ιχνηλασιμότητα, δηλαδή οριζόντια και κάθετη ιχνηλασιμότητα, μπορούν να προσδιοριστούν (IEEE, 1999). Η κάθετη ιχνηλασιμότητα εκφράζει σχέσεις μεταξύ των αντικειμένων του λογισμικού στην ίδια φάση του κύκλου ζωής του λογισμικού. Η οριζόντια ιχνηλασιμότητα εκφράζει τις σχέσεις μεταξύ των αντικειμένων λογισμικού που διαμένουν σε διάφορες φάσεις του κύκλου ζωής του λογισμικού, για παράδειγμα, τις εξαρτήσεις μεταξύ των προδιαγραφών απαιτήσεων και ενός μοντέλου σχεδιασμού. Υπογραμμίζοντας, η ιχνηλασιμότητα μειώνει το κόστος συντήρησης και βελτιώνει την ποιότητα των πτυχών της διαδικασίας ανάπτυξης λογισμικού. Η ανάλυση ιχνηλασιμότητας περιλαμβάνει την εξέταση των σχέσεων εξάρτησης μεταξύ των αντικειμένων λογισμικού του ίδιου είδους ή μεταξύ αντικειμένων λογισμικού σε διαφορετικές φάσεις του κύκλου ζωής του λογισμικού, για παράδειγμα, η εξάρτηση μεταξύ μιας απαίτησης προδιαγραφών και ενός αντίστοιχου μοντέλο σχεδιασμού. Εάν μια οντότητα λογισμικό τροποποιηθεί, είναι πιθανό η αλλαγή αυτή να διαδοθεί σε άλλες οντότητες, που χρειάζονται τροποποίηση επίσης. Η αλλαγή διάδοσης συμβαίνει όταν γίνει μια αλλαγή σε ένα κομμάτι λογισμικού του συστήματος, το οποίο απαιτεί άλλα μέρη του συστήματος που εξαρτώνται από αυτό (Rajlich, 1997). Οι αλλαγές εξαρτώνται από τα μέρη του συστήματος που μπορεί με τη σειρά τους να απαιτούν αλλαγές σε άλλα μέρη του συστήματος. Με τον τρόπο αυτό, μία μεταβολή σε ένα κομμάτι του συστήματος μπορεί να οδηγήσει σε διάδοση των αλλαγών που πρέπει να γίνουν σε ολόκληρο το σύστημα λογισμικού. Ανάλυση επιπτώσεων ορίζεται ως «η διαδικασία της αναγνώρισης των πιθανών επιπτώσεων (παρενέργειες) των αλλαγών και η εκτίμηση του τι πρέπει να τροποποιηθεί ώστε να επιτευχθεί μια αλλαγή» (Bohner and Arnold, 1996). Η ανάλυση επιπτώσεων προσπαθεί να αξιολογήσει το αντίκτυπο των αλλαγών στο υπόλοιπο σύστημα, όταν συμβούν αλλαγές σε ορισμένες οντότητες λογισμικού, ποιά μέρη του συστήματος θα θιγούν και πώς θα επηρεαστούν. Αυτή η γνώση είναι ιδιαίτερα χρήσιμη γιατί προβλέπει το κόστος και την πολυπλοκότητα των αλλαγών, επίσης βοηθάει στην απόφαση εάν θα εφαρμοστούν αυτές οι αλλαγές σε μια νέα έκδοση. Μια έκδοση χειροποίητου αντικειμένου λογισμικού αποτελεί ένα στιγμιότυπο στην ιστορία της εξέλιξης αυτού του χειροποίητου αντικειμένου λογισμικού. Εάν οι αλλαγές που γίνονται στο χειροποίητο αντικείμενο λογισμικού είναι ταυτόχρονες, τότε μπορεί να υπάρχουν κατά την ίδια χρονική στιγμή

διαφορετικές εκδοχές του ίδιου χειροποίητου αντικειμένου λογισμικού. Η έκδοση ελέγχου διαχειρίζεται τις ταυτόχρονες αλλαγές που έγιναν από πολλούς χρήστες σε μια συλλογή από χειροποίητα αντικείμενα λογισμικού, χωρίς να υπάρχει σύγχυση εκδόσεων (Ohst, Welle and Kelter, 2003). Ο έλεγχος των εκδόσεων είναι ουσιαστικά η ανάπτυξη του λογισμικού από μεγάλες ομάδες, όπου διαφορετικοί άνθρωποι μπορούν να κάνουν παράλληλα τροποποιήσεις στα διαγράμματα.

Αν χρησιμοποιήσουμε μια κατάσταση που βασίζεται (state-based) στις εκδόσεις, χρειαζόμαστε εργαλεία για να συγκρίνουν τις διαφορές μεταξύ των διαδοχικών εκδόσεων του ίδιου χειροποίητου αντικειμένου λογισμικού. Αν χρησιμοποιήσουμε την αλλαγή που βασίζεται (change-based) στις εκδόσεις, χρειαζόμαστε εργαλεία που αφορούν τις διαφορετικές εκδόσεις με βάση την ακολουθία των αλλαγών που χρησιμοποιήθηκαν για την απόκτηση της νέας έκδοσης από μια παλαιότερη (Conradi & Westfechtel, 1998). Αναδιάρθρωση λογισμικού είναι «η μετατροπή από μια μορφή αναπαράστασης σε άλλη, στο ίδιο σχετικό επίπεδο αφαίρεσης, διατηρώντας παράλληλα το θέμα του συστήματος στην εξωτερική συμπεριφορά (τη λειτουργικότητα και τη σημασιολογία)». Ενώ η αναδιάρθρωση δημιουργεί νέες εκδόσεις που εφαρμόζουν ή προτείνουν αλλαγή στο θέμα του συστήματος, δεν συνεπάγεται κατά κανόνα τροποποιήσεις λόγω των νέων απαιτήσεων. Ωστόσο, μπορεί να οδηγήσει σε καλύτερη παρατήρηση του θέματος του συστήματος και να προτείνονται αλλαγές που θα βελτίωναν τις πτυχές του συστήματος. Η αναδιάρθρωση χρησιμοποιείται συχνά ως μια μορφή προληπτικής συντήρησης με σκοπό να βελτιώσει την φυσική κατάσταση του θέματος του συστήματος σε σχέση με κάποια προτιμώμενο πρότυπο» (Chikofsky and Cross, 1990). Σε ένα αντικειμενοστρεφή πλαίσιο ο όρος Αναδόμηση (Refactoring) χρησιμοποιείται αντί της αναδιάρθρωσης (Ordyke 1992, Fowler 1999). Ως εκ τούτου, θα χρησιμοποιήσουμε τον όρο μοντέλο Refactoring για την αναδιάρθρωση του αντικειμενοστρεφούς μοντέλου σχεδιασμού. Τα μοντέλα σχεδιασμού περιγράφουν το σύστημα από διαφορετικές οπτικές γωνίες και σε διαφορετικά επίπεδα αφαίρεσης και τις ευαισθησίες του. Τα μοντέλα αυτά μπορούν επίσης να εκφραστούν χρησιμοποιώντας διαφορετικά σύμβολα και διαφορετικούς προγραμματιστές λογισμικού που μπορούν να συμμετέχουν στη διαδικασία διαμόρφωσης λογισμικού. Όλα αυτά τα ζητήματα αναμένεται να οδηγήσουν σε ασυνέπεια μεταξύ των μοντέλων. Μια ασυνέπεια ανεπίσημα περιγράφεται ως «μια κατάσταση κατά την οποία δύο ή περισσότερα επικαλυπτόμενα στοιχεία διαφορετικών μοντέλων λογισμικού κάνουν δηλώσεις σχετικά με τις πτυχές του συστήματος που περιγράφουν, οι οποίες δεν είναι από κοινού ικανοποιήσιμες» (Spanoudakis and Zisman, 2001). Οι επικαλύψεις ορίζονται ως "οι σχέσεις μεταξύ των ερμηνειών που αποδίδονται σε μοντέλα λογισμικού από ειδικούς παράγοντες" (Spanoudakis and Zisman). Η ασυνέπεια διαχείρισης έχει οριστεί από τον Finkelstein, τον Spanoudakis, και τον Till (1996) ως «η διαδικασία με την οποία δημιουργείτε ασυνέπεια μεταξύ των μοντέλων

λογισμικού και γίνεται με τρόπο που να υποστηρίζει τους στόχους των εν λόγω φορέων." Ο Nuseibeh, ο Easterbrook, και ο Russo (2000) προτείνουν γενικά πλαίσια που περιγράφουν τις δραστηριότητες των εν λόγω διαδικασιών. Και οι δύο προσεγγίσεις συμφωνούν ότι η διαδικασία διαχείρισης των αντιφάσεων περιλαμβάνει δραστηριότητες για την ανίχνευση, τη διάγνωση και το χειρισμό τους. Οι δραστηριότητες αυτές παρατείνονται κατά τον Σπανουδάκη και τον Zisman: η ανίχνευση των επικαλύψεων, η ανίχνευση των ανακολουθιών, η διάγνωση των αντιφάσεων, ο χειρισμός των αντιφάσεων, ο εντοπισμός και ο καθορισμός των προδιαγραφών και της εφαρμογής μιας πολιτικής διαχείρισης ασυνέπεια. Επιπλέον, ο Σπανουδάκης και ο Zisman παρουσίασαν μια έρευνα των τεχνικών και μεθόδων για τη στήριξη της διαχείρισης των αντιφάσεων. Η συν-εξέλιξη (Co-evolution) αναφέρεται στο σύνολο των εργαλείων και των τεχνικών που κρατάνε τις οντότητες του λογισμικού σε διαφορετικά επίπεδα του κύκλου ζωής λογισμικού συγχρονίζοντάς τα όταν το λογισμικό εξελίσσεται. Αυτό περιλαμβάνει τεχνικές όπως η παραγωγή κώδικα και αντίστροφη μηχανική (reverse engineering). Μια ειδική έννοια της συν-εξέλιξης είναι η μετ'επιστροφής μηχανική (*round-trip engineering*). Μετ'επιστροφής μηχανική είναι η «απρόσκοπτη ολοκλήρωση μεταξύ του σχεδιασμού και του πηγαίου κώδικα, μεταξύ των μοντέλων και την εφαρμογή. Με την *μηχανική μετ'επιστροφής* ο προγραμματιστής δημιουργεί κώδικα από ένα σχέδιο, οι αλλαγές του κώδικα γίνονται σε ένα ξεχωριστό περιβάλλον ανάπτυξης και αναδημιουργείτε το προσαρμοσμένο σχήμα σχεδιασμού πίσω από τον πηγαίο κώδικα» (Demeyer, Ducasse and Tichelaar, 1999).

Η ιχνηλασιμότητα είναι μια σχέση μεταξύ των οντοτήτων: ένα σύνολο πηγαίων οντοτήτων, και ένα σύνολο οντοτήτων-στόχων. Οι διαφορετικοί τύποι σχέσεων μπορούν να καθοριστούν με διαφορετικούς τύπους οντοτήτων. Οι Ramesh & Jarke πρότειναν ένα πρότυπο-μοντέλο αναφοράς, βασισμένο σε εμπειρικές παρατηρήσεις, περιλαμβάνοντας τα σημαντικότερα είδη συνδέσεων ιχνηλασιμότητας (traceability links) για τους στόχους ανάπτυξης. Παραδείγματος χάριν, τα σενάρια περιγράφουν τις απαιτήσεις, οι απαιτήσεις συστημάτων προέρχονται από τις απαιτήσεις χρηστών, τα components του συστήματος ικανοποιούν τις απαιτήσεις, και οι διαδικασίες επαλήθευσης ελέγχουν ότι οι απαιτήσεις ικανοποιούνται. Ο Letelier [4] παρουσιάζει ένα μεταμοντέλο αναφοράς (reference metamodel) για τις απαιτήσεις ιχνηλασιμότητας που επεκτείνει το UML metamodel, ενσωματώνοντας αντικείμενα μη-UML όπως τα στοιχεία κειμένων και οι προδιαγραφές δοκιμής.

**Η πορεία προς το Querying της πληροφορίας της ιχνηλασιμότητας στο πλαίσιο της εξέλιξης λογισμικού (Towards Querying of Traceability Information in the Context of Software Evolution)**

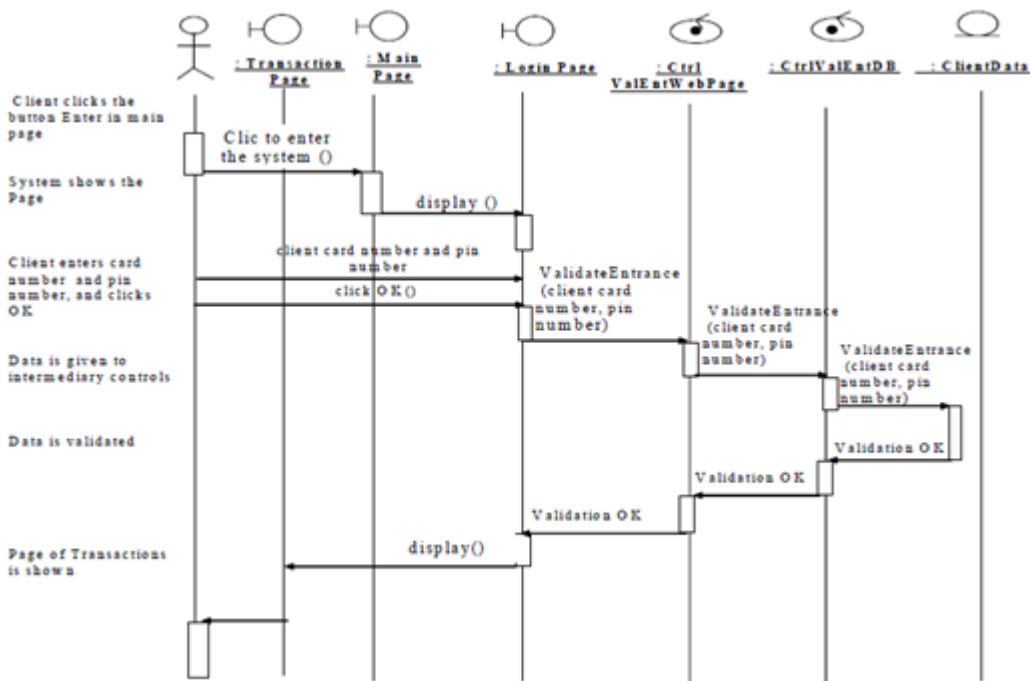
Η ιχνηλασιμότητα διαδραματίζει έναν σημαντικό ρόλο μέσα στον τομέα της ανάπτυξης λογισμικού και της εξέλιξης του. Μεταξύ των άλλων, οι τομείς της εφαρμογής είναι η επαναχρησιμοποίηση, η διαχείριση αλλαγής, και η συντήρηση. Πολύ συνδεδεμένες με τη διαχείριση αλλαγής και τη συντήρηση είναι η ανάλυση αντίκτυπου (impact analysis), που εξετάζει την επίδραση της αλλαγής ενός συγκεκριμένου τμήματος πληροφορίας σε άλλες απαιτήσεις, κομμάτια πληροφορίας/κώδικα, κ.λπ. Παραδείγματος χάριν, μια αλλαγμένη απαίτηση απαιτεί να ανακαλυφθούν τα κομμάτια του κώδικα που την απαρτίζουν προκειμένου να επιτραπεί η πραγματοποίηση της αλλαγής. Αντιθέτως, ένα τροποποιημένο κομμάτι κώδικα πρέπει να επισημανθεί πίσω στις απαιτήσεις που υλοποιεί, για να εξασφαλιστεί κατάλληλη εφαρμογή των σχετικών, αλλά αμετάβλητων απαιτήσεων. Δίπλα σε αυτήν την οριζόντια ιχνηλασιμότητα, αντιπροσωπεύοντας τις αλληλεξαρτήσεις των διαφορετικών τύπων των κομματιών της πληροφορίας, όπως τις απαιτήσεις, τις περιπτώσεις δοκιμής, και τον κώδικα, η ιχνηλασιμότητα στην εξέλιξη λογισμικού εξετάζει επίσης την κάθετη ιχνηλασιμότητα, που παρουσιάζει τις αλληλεξαρτήσεις μεταξύ των κομματιών της πληροφορίας του ίδιου είδους. Η κάθετη ιχνηλασιμότητα που προέρχεται από τον πηγαίο κώδικα περιλαμβάνει π.χ. την πρόσβαση εξωτερικών προγραμμάτων και βάσεων δεδομένων, τα οποία υπάγονται στις δραστηριότητες προγραμμάτων κατανόησης σε πολυ-γλωσσικά συστήματα.

Η ιχνηλασιμότητα λογισμικού - αυτή είναι η δυνατότητα να σχετίζονται τα κομμάτια πληροφορίας που δημιουργούνται κατά τη διάρκεια της ανάπτυξης ενός συστήματος λογισμικού (π.χ., απαιτήσεις, αντικείμενα σχεδίου και κώδικα) το ένα με το άλλο - έχει αναγνωριστεί ως σημαντική δυνατότητα στη διαδικασία ανάπτυξης λογισμικού και της συντήρησης, και ως σοβαρός παράγοντας για την ποιότητα του τελικού προϊόντος. Οι πληροφορίες ιχνηλασιμότητας μπορούν να χρησιμοποιηθούν για να υποστηρίξουν την ανάλυση των αλλαγών που ζητούνται στη διαδικασία ανάπτυξης συστημάτων, στη συντήρηση, στην εξέλιξη και στο documentation των συστημάτων λογισμικού, στην επαναχρησιμοποίηση των συστημάτων λογισμικού και των συστατικών τους, και στην δοκιμή. Η ανάπτυξη των συστημάτων λογισμικού μπορεί να είναι σύνθετη, και αυτή η πολυπλοκότητα είναι ακόμα μεγαλύτερη για τα κατανεμημένα συστήματα συνεπώς, η διαχείριση της ιχνηλασιμότητας λογισμικού για τα κατανεμημένα συστήματα είναι κρίσιμη.

Ένα σημαντικό μέρος της ιχνηλασιμότητας είναι ότι η τελική εφαρμογή είναι σύμφωνη με το σχέδιο και την ανάλυση. Καθώς το σχέδιο καλυτερεύει (getting refined) σε μια συγκεκριμένη εφαρμογή, είναι σημαντικό οι έννοιες να έχουν μια σαφή αντιστοίχιση στα υλοποιημένα αντικείμενα της εφαρμογής - ακόμα κι αν η χαρτογράφηση δεν είναι ένα προς ένα. Στην προσέγγισή μας, εξειδικευμένα αντικείμενα ελέγχου στην ανάλυση - που συνδέονται με τις περιπτώσεις χρήσης και καλούνται Remote Communication Control Objects είναι οι

αφαιρέσεις των συστατικών που είναι υπεύθυνα για τη μακρινή επικοινωνία στην εφαρμογή. Στο μεταξύ καθορίζουμε τις κλάσεις, που διευκρινίζονται από τις UML διεπαφές τους(UML interfaces).

## Ιχνηλασιμότητα (Traceability)



Σχήμα 1.4. Διάγραμμα ακολουθίας για την περίπτωση χρήσης Σύνδεση μέσω διαδικτύου.



Σχήμα 1.5. Διάγραμμα ελέγχου κλάσεων, CtrlValEntWebPag και CtrlValEntBD.

## Remote Interfaces (IDL) and Local Interfaces (UML)

Έχουμε δύο κλάσεις ελέγχου(control classes) σε αυτό το σχέδιο. Τις εξής: CtrlValEntWebPag και CtrlValEntBD. Το σχέδιο 1.4. δείχνει τις διεπαφές όπως έχουν σχεδιαστεί από το SD διάγραμμα. Παρατηρήστε ότι η διεπαφή CtrlValEntWebPag περιέχει την διαδικασία validationOK()και η διεπαφή CtrlValEntBD περιέχει τη διαδικασία validateEntrance(), διαδικασίες που αναγνωρίστηκαν στο sequence diagram.

```
//IDL
// Module: Control of data verification when entering the system via Internet
Module CtrlValEntWebPage {
// Operations on the Web page side
interface CtrlValEntWebPage {
void validationOK();
void validationError();
};
// Operations DB side
interface CtrlValEntBD {
void validateEntrance(
in long client_card,
in long pin_number);
};
}
```

## 1.2. Οι διάφορες μορφές της ιχνηλασιμότητας

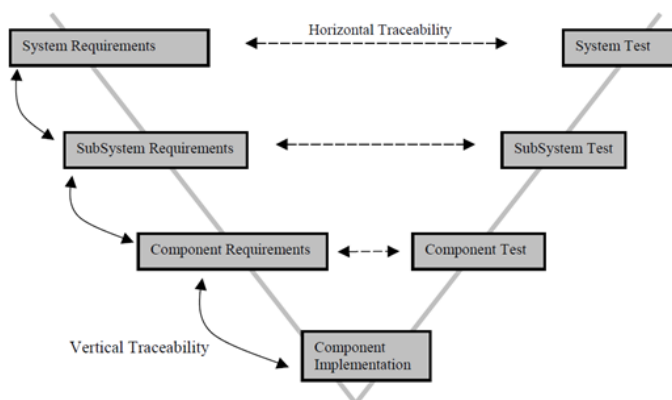
Η ιχνηλασιμότητα μπορεί, δεδομένης της πλούσιας σειράς ορισμών στη βιβλιογραφία, να χρησιμοποιηθεί για πολλά διαφορετικά πράγματα. Οι δύο κύριοι τρόποι ιχνηλασιμότητας είναι η οριζόντια και η κάθετη ιχνηλασιμότητα. Και οι δύο μορφές έχουν πλεονεκτήματα, με βάση το ποιες πληροφορίες πρέπει να συλληφθούν.

### 1.2.1. Οριζόντια και κάθετη ιχνηλασιμότητα

Ειδικά στην μετα-ιχνηλασιμότητα (φάση ανάπτυξης), υπάρχουν δύο κύριοι τύποι ιχνηλασιμότητας: η κάθετη και η οριζόντια ιχνηλασιμότητα (βλέπε το V-μοντέλο στο Figure 2).

Η κάθετη ιχνηλασιμότητα εξασφαλίζει ότι οι απαιτήσεις είναι ανιχνεύσιμες στα συστατικά του συστήματος που υλοποιήθηκε και το αντίστροφο. Με άλλα λόγια, εξασφαλίζει ότι γνωρίζουμε ποια συστατικά λογισμικού της πραγματικής έκδοσης του λογισμικού ικανοποιούν ποια απαίτηση. Η κατακόρυφη ιχνηλασιμότητα διευκολύνει την προσαρμογή των εγγράφων απαιτήσεων στις αλλαγές που έγιναν κατά την εφαρμογή του συστήματος ή τη χρήση του. Βοηθά να κρατήσει τις τεκμηριωμένες απαιτήσεις συνεπείς με την εφαρμογή του συστήματος.

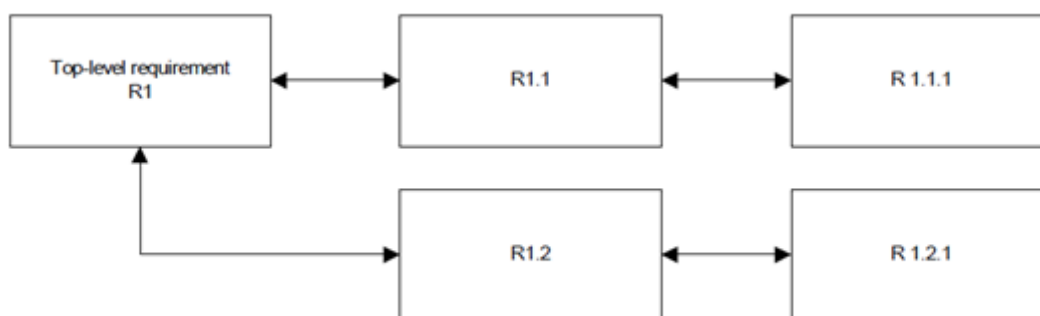
Η οριζόντια ιχνηλασιμότητα εξασφαλίζει ότι οι απαιτήσεις είναι ανιχνεύσιμες από στις περιπτώσεις δοκιμής. Εξασφαλίζει ότι τηρούνται όλες οι απαιτήσεις που πράγματι έχουν δοκιμαστεί και διευκολύνει την προσαρμογή των περιπτώσεων δοκιμών σε μεταβολές που γίνονται στα έγγραφα απαιτήσεων και αντιστρόφως. Με την ιχνηλασιμότητα και τις περιπτώσεις δοκιμής να έχουν εφαρμοστεί, η κατάσταση των απαιτήσεων μπορεί να οριστεί ανάλογα με τα αποτελέσματα των δοκιμών των σχετικών περιπτώσεων δοκιμής.



Σχήμα 1.6. Οριζόντια και Κάθετη Ιχνηλασιμότητα στο V-model.

### 1.3. Κάθετη Ιχνηλασιμότητα

Η ιχνηλασιμότητα σε κάθε φάση της διαδικασίας ανάπτυξης λογισμικού αναφέρεται ως κάθετη, διότι περιέχει πληροφορίες που σχετίζονται με διάφορα στοιχεία της ίδιας φάσης. Ένα παράδειγμα είναι να έχουμε απαιτήσεις που οδηγούν σε επιμέρους ανάγκες. Οι πληροφορίες ιχνηλασιμότητας μπορούν στη συνέχεια να χρησιμοποιηθούν για να συνδεθούν όλες αυτές μαζί, έτσι ώστε αν μία από τις βασικές απαιτήσεις αλλάξει, η επίδραση των λοιπών απαιτήσεων που απορρέουν από αυτές να μπορεί να προσδιοριστεί. Το Σχήμα 1.7. δείχνει πώς μπορεί να απεικονιστεί η σχέση μεταξύ υψηλότερου και χαμηλότερου επιπέδου απαιτήσεων.



Σχήμα 1.7. Η ροή προς τα χαμηλότερα επίπεδα των απαιτήσεων.

Η κατακόρυφη ιχνηλασιμότητα αναφέρεται στην ένωση των εξαρτώμενων στοιχείων μέσα σε ένα μοντέλο. Για την ανάλυση του πηγαίου κώδικα, τα αυτοματοποιημένα εργαλεία ανίχνευσης, την καταγραφή πληροφοριών εξάρτησης, την αξιολόγηση των δεδομένων, τον έλεγχο και εξαρτήσεων των συστατικών. Μια υποστηρικτική μέθοδος, που ονομάζεται τεμαχισμός προγράμματος είναι μια βασική τεχνική που εφαρμόστηκε για πρώτη φορά από τον Weiser (1984), που εκμεταλλεύεται τα δεδομένα ροής και τις εξαρτήσεις ελέγχου ροής για να συλλάβει "φέτες" των προγραμμάτων. Τεχνικές για την ανάλυση της εξάρτησης χρησιμοποιούνται ευρέως στην μεταγλωττιστές και στα εργαλεία ανάλυσης πηγαίου κώδικα. Η μοντελοποίηση ροής δεδομένων, ο έλεγχος ροής καθώς και οι σχέσεις και εξαρτήσεις των συστατικών είναι καρποφόρες μέθοδοι για να καθοριστούν οι επιπτώσεις της αλλαγής του λογισμικού μέσα σε ένα σύνολο αντικειμένων πηγαίου κώδικα. Μολονότι η ανάλυση της εξάρτησης επικεντρώνεται αυστηρά στις επιπτώσεις πάνω στην πληροφορία οι οποίες συλλαμβάνονται από τον πηγαίο κώδικα, είναι η πιο ώριμη τεχνική ανάλυσης των επιπτώσεων.



## Κάθετοι Σύνδεσμοι

Θεωρούμε ότι ένα μοντέλο κάθετης ιχνηλασιμότητας είναι ένα μοντέλο εσωτερικών ( στο μοντέλο ) χειροποίητων αντικειμένων από τα οποία ένα χειροποίητο αντικείμενο προσφέρει συνδέσεις με άλλα συστατικά στο ίδιο επίπεδο αντικειμένων. Στην θεωρία, θεωρούμε τα εξής ως πλατφόρμες κάθετης ιχνηλασιμότητας.

- a) Επίπεδο απαιτήσεων ( Requirement level ).
- b) Επίπεδο δοκιμής περίπτωσης ( Test case level ).
- c) Επίπεδο σχεδίασης ( Design level ).
- d) Επίπεδο κώδικα (Code level ).

Το επίπεδο απαιτήσεων αναφέρεται εδώ σε λειτουργικές καθαρά απαιτήσεις. Ενώ το επίπεδο δοκιμής περίπτωσης αναφέρεται σε περιγραφές δοκιμών που περιγράφονται ως πιθανές καταστάσεις που χρειάζεται να δοκιμαστούν ικανοποιηθεί μια απαίτηση. Σε μερικά συστήματα μπορεί να υπάρχουν μερικές απαιτήσεις ή μερικές δοκιμές περιπτώσεων που αποσυντίθενται σε υπό-συστατικά. Όμως, για να εναρμονιστούμε με το μοντέλο μας το καθένα από αυτά αναγνωρίζεται μοναδιαία. Για να απεικονιστεί αυτό το φαινόμενο, σκεφτείτε το εξής παράδειγμα.

Req#: 5

Code : SRS\_REQ-02-05

Περιγραφή: Ο οδηγός πατάει το κουμπί «Ενεργοποίηση» για να ενεργοποιήσει την λειτουργία AutoCruise.

Οι περιπτώσεις χρήσης που εμπλέκονται είναι οι εξής :

- 1) Test case #:1  
Code: TCASE-12-01  
Περιγραφή : Τρέξε με τη λειτουργία Auto Cruise με ταχύτητα > 80 km/hr.
- i) Test case#: 1.1  
Code : TCASE-12-01-01  
Περιγραφή: Τρέξε τη λειτουργία Auto Cruise ενώ δεν έχεις πέμπτη ταχύτητα.
- ii) Test case#: 1.2

Code : TCASE-12-01-02

Περιγραφή: Τρέξε τη λειτουργία Auto Cruise ενώ έχεις πέμπτη ταχύτητα.

2) Test case#: 2

Code : TCASE-12-02

Περιγραφή: Εμφάνισε το LED με ένα μήνυμα προειδοποίησης “Κίνδυνος!” ενώ είσαι σε λειτουργία auto cruise αν η ταχύτητα  $\geq 150$  km/h.

Μπορούμε να πούμε ότι η Απαίτηση *Req#5* απαιτεί 3 περιπτώσεις χρήσης αντί για 2 καθώς χρειαζόμαστε να χωρίσουμε την ομάδα της περίπτωσης χρήσης *test case#1* σε δύο ξεχωριστές περιπτώσεις χρήσης, τις *test case#1.1* και *test case#1.2*.

Στο σχέδιο και στον κώδικα, μπορεί να υπάρχουν κάποιες ασάφειες του τύπου ποια χειροποίητα αντικείμενα θα πρέπει να παρουσιαστούν καθώς και τα δύο μπορεί να αποτελούνται από μερικά συστατικά αλληλεπικαλυπτόμενα π.χ. θα έπρεπε οι κλάσεις να ταξινομηθούν στο σχέδιο ή στον κώδικα? Για εμάς, αυτό είναι απλά ένα ζήτημα προσέγγισης που θα επιλέξουμε απλά.

Το επίπεδο κώδικα μπορεί να ταξινομηθεί σε υψηλού επιπέδου σχεδιαστικές αφαιρέσεις (π.χ. μοντέλα συνεργασίας), αλλά και σε χαμηλού επιπέδου σχεδιαστικές αφαιρέσεις (e.g. διαγράμματα κλάσεων) ή και με ένα συνδυασμό και των δύο. Στην υλοποίησή μας, δίνουμε λιγότερη προσοχή στις υψηλού επιπέδου αφαιρέσεις για να παράγουμε την ιχνηλασιμότητα αφού αυτό απαιτεί περισσότερη έρευνα και θα περιέπλεκε σημαντικά την εργασία μας. Εφαρμόζουμε τις αφαιρέσεις χαμηλότερου επιπέδου που περιέχουν τα πακέτα λογισμικού και τις αλληλεπιδράσεις των κλάσεων. Ο κώδικας πρέπει να περιέχει όλες τις μεθόδους, τα περιεχόμενά τους και τις αλληλεπιδράσεις τους.

### **Κάθετες μονάδες (Vertical Units)**

Η αλγοριθμική έννοια του λογισμικού, όπως τυποποιείται από τον Dijkstra, έχει την επιθυμητή ιδιότητα ότι μπορεί να συντεθεί κατά τρόπο φυσικό από τις έννοιες των συστατικών σε μια συμβατική αρχιτεκτονική. Για να δούμε τι σημαίνει αυτό, σε όρους εκτέλεσης στα λειτουργικά πρότυπα, σκεφτείτε *state sequences* που υλοποιούν έναν επιθυμητό μετασχηματισμό.

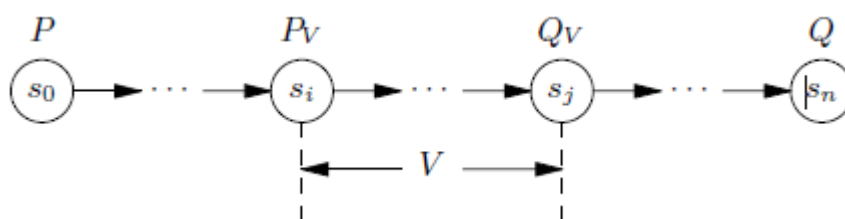
Ανεξάρτητα από τις σχεδιαστικές αρχές που εφαρμόζονται, μια συμβατική αρχιτεκτονική επιβάλλει έναν «κάθετο» τεμαχισμό σε αυτές τις ακολουθίες, έτσι ώστε κάθε μονάδα να είναι αρμόδια για ορισμένες υπο-ακολουθίες καταστάσεων. Αυτό απεικονίζεται στο σχήμα 1.9., όπου η ικανοποίηση του ζευγαριού «προϋπόθεση-τελική κατάσταση» (P, Q) για ολόκληρη την ακολουθία στηρίζεται στην υπόθεση ότι ένα *subsequence* V, που παράγεται από μια αρχιτεκτονική μονάδα, ικανοποιεί το ζεύγος «precondition-postcondition» (PV, QV).

Γενικότερα, μια αρχιτεκτονική που αποτελείται από συμβατικές μονάδες, επιβάλλει μια εμφωλευμένη δομή τέτοιων κάθετων «κομματιών» επάνω σε κάθε αλληλουχία κατάστασης.

Στην παραγωγή αυτών των ακολουθιών, οι δύο βασικές διαδικασίες μεταξύ των αρχιτεκτονικών μονάδων μπορούν να χαρακτηριστούν ως σειριακή σύνθεση και πρόκληση. Το πρώτο συνδέει τις ακολουθίες κατάστασης που παράγονται από τις συστατικές μονάδες, ενώ το τελευταίο ενσωματώνεται στις πιο μακροχρόνιες ακολουθίες μερικών ακολουθιών κατάστασης που παράγονται σαν μια συστατική μονάδα. Και στις δύο περιπτώσεις, οι προκύπτουσες ακολουθίες κατάστασης έχουν υπο-ακολουθίες για τις οποίες τα εκάστοτε συστατικά είναι αρμόδια. Στις τρέχουσες προσεγγίσεις τεχνολογίας λογισμικού, αυτή η άποψη έχει υιοθετηθεί ως η βάση για τον σχεδιασμό συμπεριφοράς των αντικειμενοστραφών συστημάτων, μεταθέτοντας την εστίαση στις διαδικασίες που πρόκειται να κληθούν, και στα σχετικά ζεύγη «precondition-postcondition». Αυτήν η αρχιτεκτονική διάσταση καλείται κάθετη.



Σχήμα 1.8. Εκτέλεση ακολουθίας καταστάσεων.

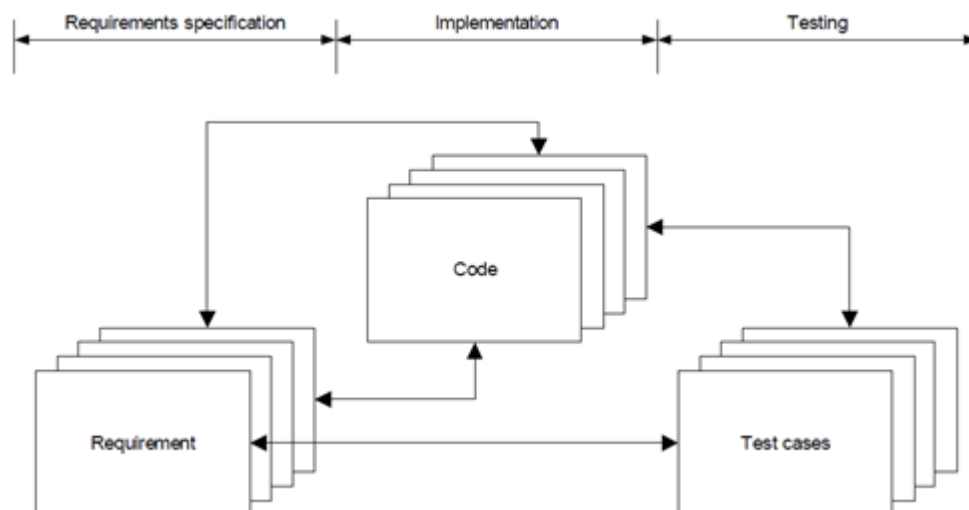


Σχήμα 1.9. Κάθετο απόσπασμα  $V$  σε μια εκτέλεση.

## 1.4. Οριζόντια Ιχνηλασιμότητα

Η Οριζόντια ιχνηλασιμότητα ορίζεται ως οι πληροφορίες ιχνηλασιμότητας που δείχνουν τα διαδοχικά στάδια του προϊόντος ή τον προκάτοχό τους στην ανάπτυξη του κύκλου ζωής του λογισμικού.

Οι πληροφορίες που αποκτήθηκαν μέσω της οριζόντιας ιχνηλασιμότητας μπορούν να χρησιμοποιηθούν για να πάρετε μια γενική επισκόπηση της επίδρασης των προηγούμενων στοιχείων – πληροφοριών . Το Σχήμα 1.10. παρουσιάζει την οριζόντια ιχνηλασιμότητα μεταξύ μιας σειράς εγγράφων (απλή περίπτωση).



Σχήμα 1.10. Διασταύρωση των σχέσεων μεταξύ των φάσεων .

Διαφορετικές παραλλαγές αυτής της μορφής υπάρχουν, οι περισσότερες εκ των οποίων ποικίλλουν όσον αφορά τον όγκο των πληροφοριών ιχνηλασιμότητας που συγκεντρώθηκαν και τις συγκεκριμένες φάσεις της διαδικασίας ανάπτυξης από τις οποίες μπορούν να ανιχνευθούν. Οι παραλλαγές κυμαίνονται από την πιο απλή μορφή, από τον εντοπισμό των απαιτήσεων στην εφαρμογή, στα μοντέλα που εντοπίζουν πληροφορία από κάθε μια φάση στην επόμενη(και μερικές φορές ακόμη και στην προηγούμενη).

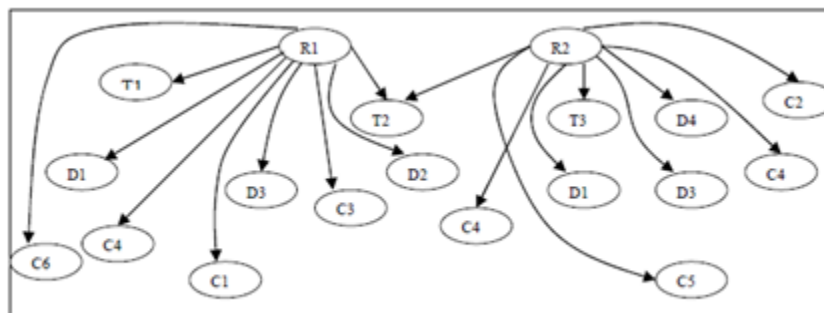
Η Οριζόντια ιχνηλασιμότητα χρησιμοποιείται συχνά για να διαπιστωθεί αν το αποτέλεσμα της πρώτης φάσης είναι αυτό που θα έπρεπε να είναι, δεδομένων των εισροών στη φάση.

## Οριζόντια Ιχνηλασιμότητα

Η Οριζόντια ανάλυση αναφέρεται στην ένωση των αντίστοιχων αντικειμένων μεταξύ των προϊόντων . Σε γενικές γραμμές, οι τεχνικές της ιχνηλασιμότητας θα μπορούσαν να ταξινομηθούν σε τέσσερις μεγάλες τεχνικές, τις εξής : Τις ρητές συνδέσεις ( explicit links ), εννοούμενες συνδέσεις( implicit links), τον εντοπισμό ονόματος και τον εντοπισμό της έννοιας

## Οριζόντιοι Σύνδεσμοι

Οι οριζόντιες συνδέσεις θεωρούνται ως μέρος του μοντέλου της ιχνηλασιμότητας που περιλαμβάνει τα χειροποίητα αντικείμενα μεταξύ όλων των επιπέδων κατά τέτοιο τρόπο ώστε κάθε τεχνούργημα σε ένα επίπεδο, να παρέχει συνδέσεις με άλλα αντικείμενα από διαφορετικά επίπεδα. Το παρακάτω σχήμα δείχνει μια πιθανή έννοια της ιχνηλασιμότητας από την άποψη των απαιτήσεων. Για παράδειγμα, η R1 είναι μια απαίτηση που έχει άμεσες επιπτώσεις στις περιπτώσεις δοκιμής T1 και T2. Η R1 έχει επίσης άμεσες επιπτώσεις στα σχέδια D1, D2, D3 και στα συστατικά του κώδικα C1, C3, C4. Η ίδια αρχή ισχύει και για το R2. Τα R1 και R2 θα μπορούσαν να έχουν αντίκτυπο στα ίδια αντικείμενα, για παράδειγμα τα D3, T2 και C4. Έτσι, σε γενικές γραμμές ο αντίκτυπος του συστήματος μπορεί να ερμηνευθεί ως εξής.



Σχήμα 1.11. Ιχνηλασιμότητα από την οπτική γωνία των απαιτήσεων.

$S = (G, E)$  ..... (H-i)

$G = GR ? GD ? GC ? GT$  ..... (H-ii)

$E = ER ? ED ? EC ? ET$  ..... (H-iii)

Όπου,

S – Συνολική επίδραση στο σύστημα

G – Χειροποίητα αντικείμενα

GR – χειροποίητα αντικείμενα απαιτήσεων

GD – χειροποίητα αντικείμενα σχεδίου

GC – χειροποίητα αντικείμενα κώδικα

GT – χειροποίητα αντικείμενα δοκιμής κώδικα

E - Σχέσεις

ER – σχέσεις από την οπτική γωνία ενός χειροποίητου αντικειμένου απαίτησης

ED – σχέσεις από την οπτική γωνία ενός χειροποίητου αντικειμένου σχεδίου

ET – σχέσεις από την οπτική γωνία ενός χειροποίητου αντικειμένου δοκιμής κώδικα

EC – σχέσεις από την οπτική γωνία ενός χειροποίητου αντικειμένου κώδικα

? – αναπαριστά το σύμβολο/ έννοια OR ( ή ).

Οι παραπάνω προδιαγραφές μπορούν να ερμηνευθούν ως η συνολική επίδραση (S) η οποία αποτελείται από χειροποίητα αντικείμενα λογισμικού και τις σχέσεις τους. Για ένα χειροποίητο αντικείμενο τύπου απαιτήσεων (GR), σχεδίου (GD), κώδικα (GC) ή δοκιμής κώδικα (GT), υπάρχει δυνατότητα να παρακολουθήσει τις σχέσεις της ιχνηλασιμότητας του με άλλα χειροποίητα αντικείμενα στο σύστημα.

Η οριζόντια ιχνηλασιμότητα μπορεί να αναχθεί στις ακόλουθες προοπτικές.

1) Η ιχνηλασιμότητα απαιτήσεων

$$ER \subseteq GR \times SGR \dots\dots\dots (H-iv)$$

$$SGR = GD \cup GC \cup GT \dots\dots\dots (H-v)$$

$$GR = \{R1, R2, \dots, Rn\} \dots\dots\dots (H-vi)$$

Όπου,

$\subseteq$  - συμβολίζει ένα υποσύνολο

$\times$  - συμβολίζει την ύπαρξη μιας σύνδεσης

SGR – χειροποίητα αντικείμενα των διαφόρων επιπέδων

{R1, R2, ... .., Rn} – το σύνολο των αναφορών απαίτησης μπορεί να ερμηνευθεί ως ER, το οποίο είναι ένα υποσύνολο των δεσμών μεταξύ των απαιτήσεων (GR), καθώς και άλλων χειροποίητων αντικειμένων των διαφόρων επιπέδων (SGR) που μπορεί να είναι ένα σχέδιο, κώδικας ή δοκιμές περιπτώσεων. Το GR είναι ένα σύνολο απαιτήσεων.

2) Η ιχνηλασιμότητα σχεδίου

$$ED \subseteq GD \times SGD \dots\dots\dots (H-vii)$$

$$SGD = GR \cup GD \cup GT \cup GC \dots\dots\dots (H-viii)$$

$$GD = \{D1, D2, \dots, Dn\} \dots\dots\dots (H-ix)$$

Το ED ορίζεται ως μια σχέση ιχνηλασιμότητας από τη σκοπιά ενός χειροποίητου αντικειμένου σχεδίου.

Το ED είναι ένα υποσύνολο των συνδέσεων μεταξύ του σχεδίου (GD) και άλλων χειροποίητων αντικειμένων διαφόρων επιπέδων (SGD) που μπορεί να απαιτήσεις, κώδικα ή δοκιμές περιπτώσεων. Το GD είναι ένα σύνολο αναφορών σχεδίου.

### 3) Ιχνηλασιμότητα δοκιμής περίπτωσης

$$ET \subseteq GT \times SGT \dots\dots\dots (H-x)$$

$$SGT = GR \cup GD \cup GC \dots\dots\dots (H-xi)$$

$$GT = \{T1, T2, \dots, Tn\} \dots\dots\dots (H-xii)$$

Το ET ορίζεται ως μια σχέση ιχνηλασιμότητας από την άποψη μιας δοκιμής περίπτωσης. Το ET είναι ένα υποσύνολο των συνδέσεων μεταξύ των δοκιμών περιπτώσεων (GT) και άλλων αντικειμένων των διαφόρων επιπέδων (SGT) που μπορεί να είναι απαιτήσεις, σχέδιο ή κώδικας. Το GT είναι ένα σύνολο δοκιμών περίπτωσης.

### 4) Η ιχνηλασιμότητα κώδικα

$$EC \subseteq GC \times SGC \dots\dots\dots (H-xiii)$$

$$SGC = GR \cup GD \cup GT \cup GC \dots\dots\dots (H-xiv)$$

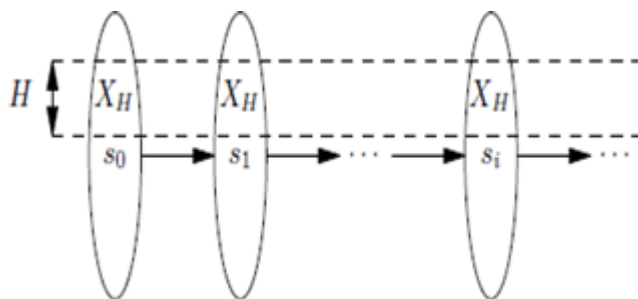
$$GC = \{C1, C2, \dots, Ct\} \dots\dots\dots (H-xv)$$

Το EC ορίζεται ως μια σχέση ιχνηλασιμότητας από τη σκοπιά ενός χειροποίητου αντικειμένου κώδικα. Το EC είναι ένα υποσύνολο των συνδέσεων μεταξύ των χειροποίητων αντικειμένων του κώδικα (GC) και άλλων αντικειμένων των διαφόρων επιπέδων (SGC), τα οποία μπορεί να είναι απαιτήσεις, σχέδια, ή δοκιμές περιπτώσεων. Το GC είναι ένα σύνολο αναφορών κώδικα. Ο κώδικας μπορεί να αναλυθεί περαιτέρω σε πιο λεπτομερές δομές των χειροποίητων αυτών αντικειμένων.

## Οριζόντιες μονάδες (Horizontal Units)

Η έννοια ενός συστήματος μπορεί επίσης να διαμορφωθεί από το πώς οι τιμές των μεταβλητών της, που δείχνονται από το σύνολο X, συμπεριφέρονται σε μη τερματιζόμενες ακολουθίες κατάστασης. Προκειμένου να υπάρξει αυτονομία, αυτό είναι φυσικό για μια τέτοια έννοια. Οι έννοιες των συστατικών πρέπει να είναι της ίδιας μορφής. Με άλλα λόγια, κάθε συστατικό πρέπει επίσης να παραγάγει μη τερματιζόμενες ακολουθίες κατάστασης, αλλά το σχετικό σύνολο μεταβλητών μπορεί να είναι ένα υποσύνολο του X. Μια αρχιτεκτονική των reactive μονάδων επομένως επιβάλλει έναν «οριζόντιο» τεμαχισμό των ακολουθιών κατάστασης, έτσι ώστε η κάθε μονάδα να είναι αρμόδια για κάποιο υποσύνολο XH των μεταβλητών σε όλες τις Si καταστάσεις, όπως απεικονίζεται στο σχήμα 1.12. Στην παραγωγή των ακολουθιών κατάστασης, μόνο μια βασική λειτουργία απαιτείται. Η Superposition χρησιμοποιεί ακολουθίες κατάστασης που παράγονται από μια οριζόντια «φέτα» και τις ενσωματώνει σε ακολουθίες που περιλαμβάνουν ένα μεγαλύτερο σύνολο μεταβλητών. Οι

ακολουθίες κατάστασης της προκύπτουσας κάθετης αρχιτεκτονικής έχουν προβολές για τις οποίες τα οριζόντια συστατικά είναι αρμόδια. Οι ιδιότητες των οριζόντιων φετών υπογραμμίζουν έπειτα τη συνεργασία μεταξύ των διαφορετικών κάθετων μονάδων, και τις σχέσεις μεταξύ των εσωτερικών τους καταστάσεων. Οι δύο διαστάσεις της αρχιτεκτονικής είναι υπό κάποια έννοια συμπληρωματικές η μία με την άλλη. Αφ' ενός, από την άποψη της κάθετης αρχιτεκτονικής, οι συμπεριφορές που παράγονται από τις οριζόντιες μονάδες αντιπροσωπεύουν θεμελιώδεις ανησυχίες. Από την οριζόντια άποψη, αφ' ετέρου, οι κάθετες μονάδες προκύπτουν αυξητικά.



Σχήμα 1.12. Οριζόντιο απόσπασμα H σε μια εκτέλεση.

### H Οριζόντια Αρχιτεκτονική Αφαιρέσεων (Architecting Horizontal Abstractions)

Για να επεξηγηθεί η φύση των οριζόντιων μονάδων, θεωρήστε ένα απλό παράδειγμα διαμόρφωσης ενός εξιδανικευμένου γραφείου των γιατρών, όπου οι άρρωστοι ασθενείς θεραπεύονται από τους γιατρούς. Οι κατακόρυφες μονάδες σε ένα τέτοιο πρότυπο θα περιελάμβαναν τους ασθενείς, τους γιατρούς και τους ρεσεψιονίστ. Οι οριζόντιες μονάδες, αφ' ετέρου, θα μοντελοποιούσαν τη συνεργασία τους ως συγκεκριμένες προβολές του συνολικού συστήματος, και ολόκληρη η προδιαγραφή θα μπορούσε να χτιστεί επαυξητικά πάνω σε αυτές.

Η διεργασία μπορεί να αρχίζει με ένα τετριμμένο μοντέλο του οι άνθρωποι να αρρωσταίνουν, και οι άρρωστοι ασθενείς τελικά να γίνονται καλά. Τα «κομμάτια ασθένειας» των ασθενών είναι οι μόνες μεταβλητές που απαιτούνται σε αυτήν την οριζόντια μονάδα. Έπειτα, αυτή η μονάδα μπορεί να ενσωματωθεί σε ένα μεγαλύτερο πρότυπο όπου ένας ασθενής γίνεται καλά μόνο όταν θεραπεύεται από έναν γιατρό. Αυτό το μοντέλο που επεκτάθηκε έχει γεγονότα όπως ένας γιατρός να αρχίζει να εξετάζει έναν ασθενή, και η συμμετοχή ενός γιατρού προστίθεται επίσης στα γεγονότα όπου ένας ασθενής γίνεται καλά. Τέλος, ένα περαιτέρω superposition βήμα μπορεί να προσθέσει την πτυχή ότι επίσης οι ρεσεψιονίστ απαιτούνται



στο μοντέλο, για να οργανώσουν τους ασθενείς να συναντήσουν τους γιατρούς, και να σιγουρευτούν ότι πληρώνουν τους λογαριασμούς τους. Αυτή η πτυχή είναι αληθινά θεμελιώδης υπό την έννοια ότι έχει επιπτώσεις σε όλες τις κάθετες μονάδες, δηλ., τους ασθενείς, τους γιατρούς και τους ρεσεψιονίστ.

Κάθε μονάδα σε αυτό το είδος μιας οριζόντιας αρχιτεκτονικής είναι μια αφαίρεση έννοιας του συνολικού συστήματος. Η πρώτη οριζόντια μονάδα σε αυτό το παράδειγμα είναι μια αφαίρεση όπου όλοι άλλες ιδιότητες συμπεριφοράς έχουν αφαιρεθεί μακριά εκτός από αυτές που αφορούν τα «κομμάτια ασθένειας» των ασθενών. Σε όρους Temporal Logic of Actions, το συνολικό σύστημα υπονοεί πάντα (τη έννοια) της κάθε οριζόντιας μονάδας σε αυτό. Όσον αφορά στις μεταβλητές, κάθε συστατικό στην οριζόντια δομή εστιάζει σε μερικές μεταβλητές που θα γίνουν «μυστικά» που ενκαψουλώνονται στα κάθετα συστατικά μέσα σε μια ενδεχόμενη εφαρμογή. Τέτοια μυστικά θεωρούνται περισσότερο σημαντικά από τις διεπαφές που συνδέονται με αυτές τις μεταβλητές κατά τις πρώτες φάσεις του σχεδίου. Αυτό δίνει μια επίσημη βάση για ένα reactive σύστημα - δηλ., για την έκφραση της προοριζόμενης έννοιάς του - επαυξητικά από την άποψη των λειτουργικών αφαιρέσεων που μπορούν να είναι τυπικά αιτιολογημένες. Δεδομένου ότι είναι μη ρεαλιστικό να διατυπωθούν οποιεσδήποτε σύνθετες προδιαγραφές σε ένα κομμάτι, αυτό είναι ένα σημαντικό πλεονέκτημα για τη χρησιμοποίηση των οριζόντιων αρχιτεκτονικών στην διαδικασία προδιαγραφής. Ένα κλασσικό παράδειγμα με χρήση οριζόντιων φετών είναι ο χωρισμός της ανίχνευσης ακρίβειας και λήξης σε έναν κατανεμημένο υπολογισμό. Αυτή είναι επίσης η πιο πρόωρη γνωστή χρήση του superposition στη βιβλιογραφία - η σχέση της είναι στενή με τον αντικειμενοστραφή προγραμματισμό. Για τη σύγκριση, εξετάστε πώς η διαδικασία του σταδιακού καθαρισμού (stepwise refinement) προχωρά με τις κάθετες αρχιτεκτονικές μονάδες. Από την άποψη των εκτελέσεων, κάθε λειτουργική αφαίρεση παράγει ακολουθίες καταστάσεων που οδηγούν από μια αρχική σε μια τελική κατάσταση, έτσι ώστε να ικανοποιούνται τα «precondition-postcondition» ζεύγη. Στο πιο υψηλό επίπεδο αφαίρεσης μπορεί να υπάρξει μόνο μια αλλαγή κατάστασης, και κάθε βήμα καθαρισμού αντικαθιστά μερικές αλλαγές κατάστασης από ακολουθίες καταστάσεων που παράγονται από περισσότερο καλές (more refined) αρχιτεκτονικές μονάδες. Αυτό οδηγεί σε ένα σχέδιο όπου η πρόωρη εστίαση είναι στις διεπαφές και τη χρήση τους, ενώ τα «μυστικά» μέσα στα κάθετα συστατικά μπορούν να διατεθούν μόνο προς τέλος του σχεδίου, όταν το επίπεδο της εκτελέσιμης διεπαφής έχει επιτευχθεί. Λόγω των ανωτέρω, οι αφαιρέσεις που μια κάθετη αρχιτεκτονική παρέχει δεν είναι αφαιρέσεις της έννοιας: η πλήρης έννοια υποτίθεται ότι ήταν διαθέσιμη ήδη στο πιο υψηλό επίπεδο, και αυτό παραμένει το ίδιο σε όλη τη σχεδιαστική διαδικασία. Αντ' αυτού, σε κάθε επίπεδο καθαρισμού (refinement), μια κάθετη αρχιτεκτονική δίνει μια αφαίρεση της δομής ενός υλοποιημένου λειτουργικού μοντέλου.

## 1.5. Τεχνικές Ιχνηλασιμότητας (Traceability Techniques)

Η ιχνηλασιμότητα προσφέρει μια πλατφόρμα για την ανάλυση επιπτώσεων ( impact analysis ). Μπορούμε να χωρίσουμε την ιχνηλασιμότητα σε 3 κατηγορίες.

### 1. Ιχνηλασιμότητα μέσω *explicit links*

Οι ρητές συνδέσεις ( explicit links ) παρέχουν τεχνικά μέσα της ιχνηλασιμότητας π.χ. αυτής που συνδέεται με τις βασικές inter-class σχέσεις σε ένα UML διάγραμμα κλάσης

### 2. Ιχνηλασιμότητα μέσω *name tracing*

Η επισήμανση ονόματος ( *name tracing* ) υποθέτει μια συνεπή στρατηγική ονομασίας και χρησιμοποιείται κατά το «χτίσιμο» των προτύπων. Παρουσιάζεται με την έρευνα των στοιχείων με ονόματα παρόμοια με αυτά στο αρχικό πρότυπο.

### 3. Ιχνηλασιμότητα μέσω *domain knowledge and concept location*.

Τα *Domain knowledge and concept location* κανονικά εκτελούνται από έμπειρους παραγωγούς λογισμικού που ανιχνεύουν «ίχνη» χρησιμοποιώντας την γνώση για το πώς σχετίζονται διαφορετικά αντικείμενα.

Εφαρμόζουμε:

1) και 3) στην προσέγγισή μας για την ιχνηλασιμότητα. Παίρνουμε τις ρητές συνδέσεις από τα αντικείμενα που έχουμε φτιάξει εμείς ( χειροποίητα αντικείμενα ) συμπεριλαμβανομένου των πινάκων μετασχηματισμού. Χρησιμοποιούμε το *concept location* για να δημιουργήσουμε σχέσεις μεταξύ των απαιτήσεων και των Περιπτώσεων χρήσης με τον υλοποιημένο κώδικα.

Αυτήν η διαδικασία απαιτεί έναν συντηρητή λογισμικού ο οποίος θα καταλάβει το *domain knowledge* του συστήματος που θέλει να μετατρέψει. Με αυτήν την πρόωρη γνώση μιας απαίτησης, ένας συντηρητής θα μπορεί να το αποσυνθέσει σε πιο «ρητά» αντικείμενα με όρους κλάσεων, μεθόδων ή μεταβλητών. Αυτά τα ρητά αντικείμενα αντιπροσωπεύουν μια απαίτηση ή μια έννοια που είναι τελείως ανιχνεύσιμη στον κώδικα[13]. Με την βοήθεια μιας περίπτωσης χρήσης που θα έχουμε η προσέγγισή μας μέσω του *codeMentor* θα πρέπει να είναι ικανή να υποστηρίξει μια ανίχνευση συντήρησης και να βρει τις μετα-επιπτώσεις των ορισμένων αντικειμένων στις μεθόδους και κλάσεις που επηρεάστηκαν.

Το *Name tracing* είναι μια ακόμη τεχνική για να υλοποιήσουμε την ιχνηλασιμότητα. Μπορεί να χρησιμοποιηθεί για να καθοριστούν τα αντικείμενα που επικοινωνούν με τα αντικείμενα που επικοινωνούν σε ένα άλλο, για παράδειγμα να καθορίσει τα occurrences ενός αντικειμένου με παρόμοιο όνομα, όπως ακριβώς παρουσιάστηκε σε μια απαίτηση σε αυτά στο παρακάτω σχήμα( σ.σ. 6 ), στον υλοποιημένο κώδικα σε μια προσπάθεια να φτιαχτούν συνδέσεις μεταξύ απαιτήσεων και κώδικα. Όμως, αυτή η στρατηγική δεν είναι πρακτικά σωστή στην δική μας μελέτη. Ο λόγος είναι ότι το *name tracing* δε μπορεί να χρησιμοποιηθεί για να ανιχνευτούν δομικές σχέσεις στις εξαρτήσεις του προγράμματος.

### **Απόλυτη Προσέγγιση Ιχνηλασιμότητας (Total Traceability Approach)**

Η εικόνα 4 απεικονίζει την υλοποίηση της προσέγγισης της απόλυτης ιχνηλασιμότητας. Οι οριζόντιες σχέσεις μπορούν να συμβούν can occur at the στα διαγώνια όρια όπως αυτό δείχνεται από τα παχιά βέλη, για παράδειγμα στο requirements-test (RxT), στο test case-code (TxM), και στα άλλα καθώς κινούμαστε προς τα μπροστά. Οι κάθετες σχέσεις μπορούν να συμβούν σε επίπεδο κώδικα (MxM - method interactions) και σε επίπεδο σχεδίασης (CxC - class interactions, PxP - package interactions) αντίστοιχα.

Αφού είχαμε τα RxT και το TxM από τα υποθετικά ίχνη νωρίτερα, μπορούμε να υπολογίσουμε το RxM χρησιμοποιώντας μια μεταβατική περάτωση,

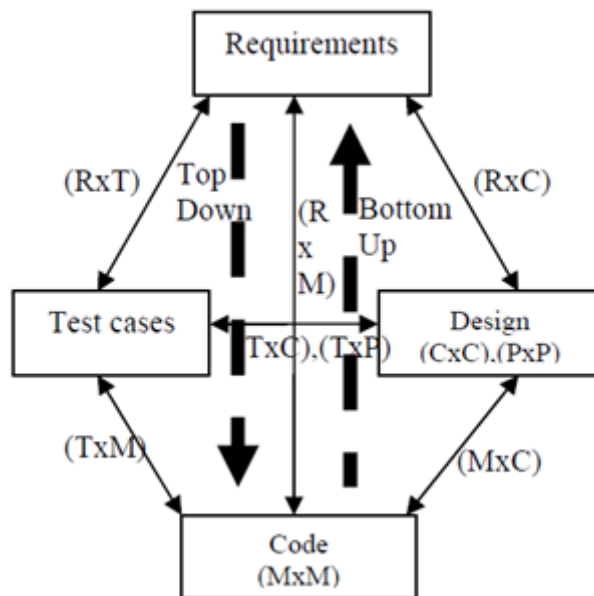
$$(RxT) \text{ και } (TxM) == (RxM)$$

Όπως αν το R επιδρά στο T και αν το T επιδρά στο M, μετά με αν R επιδρά στο M.

Οι αλληλεπιδράσεις των μεθόδων μπορούν απλά να μετασχηματισθούν σε αλληλεπιδράσεις κλάσεων και αλληλεπιδράσεις πακέτων με τη χρήση ενός μηχανισμού αντιστοίχισης βασισμένου στο γεγονός ότι ένα πακέτο κατασκευάζεται από μια ή περισσότερες κλάσεις και μια κλάση κατασκευάζεται από μια ή περισσότερες μεθόδους. Οι παχιές διακεκομμένες γραμμές απεικονίζουν την συνολική ιχνηλασιμότητα total traceability που χρειάζεται για να υλοποιήσουμε την εξεύρεση ιχνών είτε από πάνω προς τα κάτω, είτε από κάτω προς τα πάνω. Με την ανίχνευση από πάνω προς τα κάτω, εννοούμε ότι μπορούμε να εντοπίσουμε την ιχνηλασιμότητα από τα χειροποίητα αντικείμενα υψηλότερης ιεραρχίας προς τα αντικείμενα χαμηλότερης ιεραρχίας, για παράδειγμα από μια περίπτωση χρήσης μπορούμε να αναγνωρίσουμε τον σχετιζόμενο υλοποιημένο κώδικα.

Η ανίχνευση από κάτω προς τα πάνω, μας επιτρέπει να αναγνωρίσουμε τα επηρεασμένα από τις αλλαγές χειροποίητα αντικείμενα από ένα χαμηλότερο σε ένα υψηλότερο επίπεδο τους, π.χ. από μια μέθοδο μπορούμε να βρούμε τις περιπτώσεις χρήσης που επηρεάστηκαν από αυτήν και τις απαιτήσεις που επίσης επηρεάστηκαν. Ενόσω το σύστημα συνεχίζει να ψάχνει

για πιθανώς επηρεασμένα αντικείμενα ( με την γενική έννοια του όρου ) με την ιχνηλασιμότητα από πάνω προς τα κάτω ή με την ιχνηλασιμότητα από πάνω προς τα κάτω, μαζεύει και αθροίζει το μέγεθος των μετρικών των επηρεασμένων χειροποίητων αντικειμένων. Τα μεγέθη των μετρικών μετρώνται με όρους της LOC (lines of code) ή και VG (value of program complexity). Χρειάζεται να μετρήσουμε όλες τις μεθόδους και τις κλάσεις με τις LOC και VG με τη βοήθεια ενός code parser, π.χ. χρησιμοποιώντας το *McCabe tool*.



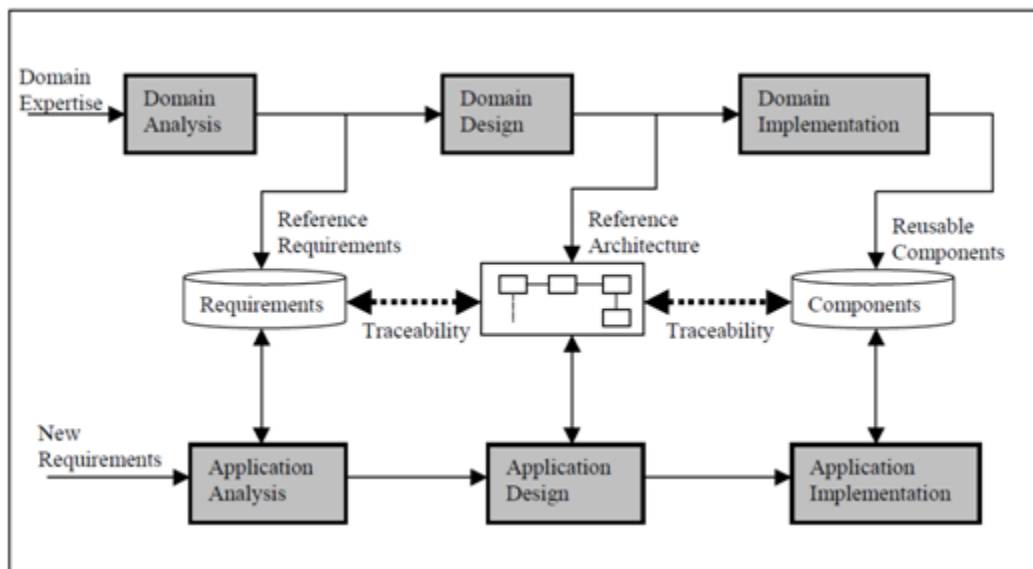
Σχήμα 1.13. Χειροποίητα αντικείμενα του συστήματος και οι δεσμοί τους.

Η ιχνηλασιμότητα στην Διαδικασία Ανάπτυξης Οικογενείας Συστήματος (System Family Development Process )

Στο σχήμα 1.14. απεικονίζεται η Διαδικασία Ανάπτυξης Οικογενείας Συστήματος. Στην πραγματικότητα, η αριστερή πλευρά του V-μοντέλου διαχωρίζεται σε επαναχρησιμοποιήσιμα μέρη, το Domain Engineering, και το συγκεκριμένο μέρος, το Application Engineering.

Με την ιχνηλασιμότητα μπορούν να συνδυαστούν νέες απαιτήσεις με τις υπάρχουσες απαιτήσεις αναφοράς που ορίζονται στην περιοχή βοηθειών ( domain asset ) και η σχέση της με την αρχιτεκτονική αναφοράς και τα επαναχρησιμοποιήσιμα συστατικά μπορεί να γίνει ρητή. Όταν οι νέες απαιτήσεις ταιριάζουν με τις απαιτήσεις αναφοράς, τα αντίστοιχα συστατικά της αρχιτεκτονικής αναφοράς μπορούν να επαναχρησιμοποιηθούν. Όταν οι νέες απαιτήσεις δεν ταιριάζουν με τις απαιτήσεις αναφοράς πρέπει να ληφθούν νέες αποφάσεις σχεδιασμού. Για να υποστηριχθούν οι μελλοντικές εξελίξεις αυτές τις σημαντικές αποφάσεις για το σχεδιασμό θα πρέπει να καταγράφονται και να είναι ανιχνεύσιμες.

Σε γενικές γραμμές, με τη δομή ιχνηλασιμότητας καταφέρνουμε το εξής : οποιεσδήποτε τροποποιήσεις στις απαιτήσεις, το σχέδιο και την υλοποίηση, τόσο στο domain asset όσο και στην εφαρμογή, μπορούν να επισημανθούν τα εμπρός και προς τα πίσω προκειμένου να αναλυθούν οι επιπτώσεις και να ελέγξουμε την συνέπεια.



Σχήμα 1.14. Σύστημα αναπηξιακής διαδικασίας οικογένειας “six pack”.

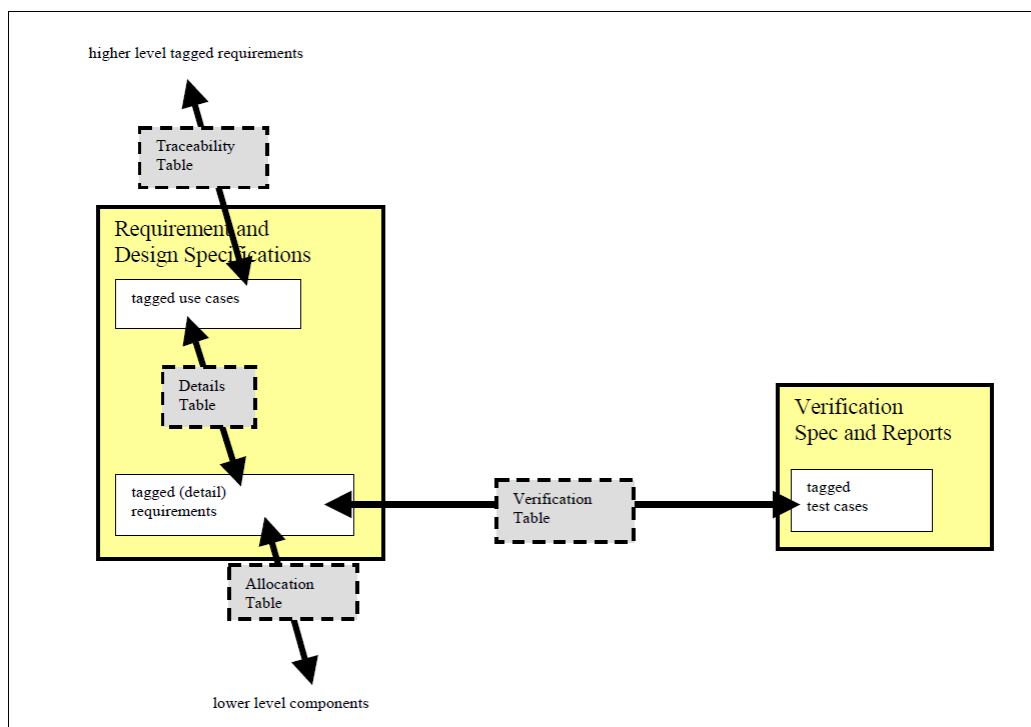
### Κάθετη / Οριζόντια Ιχνηλασιμότητα (Vertical/Horizontal Traceability)

Οι απαιτήσεις ιχνηλασιμότητας εφαρμόζεται με την κάθετη στο σχεδιασμό και την οριζόντια στην επαλήθευση. Η κάθετη ιχνηλασιμότητα ακολουθεί το είδος της εφαρμογής, τις σχέσεις μεταξύ των απαιτήσεων σε επίπεδο συστήματος, τις απαιτήσεις σε επίπεδο μονάδας, τις σχέσεις μεταξύ των απαιτήσεων σε επίπεδο μονάδας και στο επίπεδο ενότητας απαιτήσεις.

Ενδιάμεσα ένα είδος λεπτομερών σχέσεων μπορεί να είναι χρήσιμο για τον περιορισμό του αριθμού των ιχνών από τις λεπτομερείς απαιτήσεις σε υψηλότερου επιπέδου απαιτήσεις, όπου αυτό το επίπεδο λεπτομερώς δεν υπάρχει.

Η οριζόντια ιχνηλασιμότητα ακολουθεί τον τύπο επαλήθευσης της σχέσης ανάμεσα στις απαιτήσεις και τις περιπτώσεις δοκιμής.

Το μοντέλο ιχνηλασιμότητας βασίζεται στον προσδιορισμό των απαιτήσεων με ετικέτες απαίτησης στο (υπάρχον) έγγραφο προδιαγραφών και στους πίνακες με το μοντέλο των σχέσεων, όπως φαίνεται στο σχήμα 1.15.



Σχήμα 1.15. Μοντέλο υλοποίησης ιχνηλασιμότητας

Αυτό το μοντέλο ιχνηλασιμότητα επαναλαμβάνεται σε κάθε επίπεδο (συστήματος, μονάδας και ενότητα-επίπεδο), στο πλαίσιο κάθε έργου και υποέργου μέχρι το επίπεδο των λεπτομερειών στο οποίο αποφασίστηκε να εφαρμοστεί η κάθετη ιχνηλασιμότητα.

### Χρήσης (Usage)

Η απαίτηση αυτή μπορεί να αποδοθεί οριζόντια ή κάθετα και προς τα εμπρός όσο και προς τα πίσω.

Οριζόντια εμπρός ίχνος (Horizontal forward trace)

Ο οριζόντιος εμπρός εντοπισμός αποσκοπεί στην επαλήθευση των απαιτήσεων τόσο για τον προγραμματισμό του έργου όσο και για την παροχή αποδεικτικών στοιχείων για την ασφάλεια, π.χ. προς την FDA. Δείχνει εύκολα για κάθε απαίτηση πόσες δοκιμές περιπτώσεων πραγματοποιούνται και έχουν επιτύχει ή όχι και πόσες δοκιμές περιπτώσεων δεν έχουν πραγματοποιηθεί ακόμη.

- Επιλέξτε μια απαίτηση (π.χ.) την Ενότητα Απαίτησης προδιαγραφής της ενότητας 2B.

- Βρείτε την ετικέτα της απαίτησης για την επαλήθευση του πίνακα της ενότητας 2.
- Κρατήστε τις ετικέτες όλων των αποτελεσμάτων των δοκιμών και βρείτε τα αποτελέσματα των δοκιμών στην ενότητα Έλεγχος Έκθεσης (MVR2).
- Ελέγξτε τη δοκιμή κατάστασης όλων των περιπτώσεων δοκιμών για την απαίτηση. Όταν όλες οι περιπτώσεις ελέγχου έχουν περάσει, ικανοποιείται η απαίτηση.

#### Οριζόντιο πίσω ίχνος (Horizontal backwards trace)

Οριζόντια πίσω ανίχνευση επιτρέπει σε κάποιον να ελέγξει το κείμενο απαιτήσεων κατά την εκτέλεση της δοκιμής και να ελέγξει την πληρότητα της προδιαγραφής 2.

- Επιλέξτε μια δοκιμή στην ενότητα Έλεγχος προδιαγραφών της ενότητας 2B.
- Για κάθε σχετική απαίτηση στον πίνακα επαλήθευσης κάνετε τα εξής:
- Βρείτε την ενότητα Προδιαγραφές στο εφαρμοστέο έγγραφο στη MasterList του εγγράφου χρησιμοποιώντας το DocID.
- Αναζητήστε την ετικέτα με το δείκτη του εγγράφου που βρέθηκε, για να βρείτε το ίδιο το κείμενο της απαίτησης.

#### Κάθετο εμπρός ίχνος (Vertical forward trace)

Η κάθετη προς τα εμπρός ανίχνευση επιτρέπει τον έλεγχο, εάν η απαίτηση καθορίζεται στο χαμηλότερο επίπεδο των εγγράφων και τον προσδιορισμό του καθεστώτος της απαίτησης για το σχεδιασμό του έργου W.R.T. προσπάθειας αναπτυξιακής.

- Επιλέξτε μια μονάδα απαίτηση π.χ. σε μια URS.
- Ελέγξτε την Κατανομή του πίνακα της μονάδας για να μάθετε ποιες ενότητες υλοποίησαν τη μονάδα-απαίτηση(unit-requirement) και τη κάθε ευθύνη.
- Για κάθε εμπλεκόμενη ενότητα βρείτε την ενότητα απαιτήσεων της ιχνηλασιμότητας πίνακα που θα πρέπει να πληρούν την απαίτηση ενότητας
- εάν δεν βρίσκονται τα ίχνη της ενότητας-απαιτήσεις, αποτυγχάνει και είναι αναγκαία κάποια κατάλληλη δράση (η οποία είναι ένα έργο-ευθύνη(project-responsibility))
- αν οι ενότητες απαιτήσεων (module-requirements) που βρέθηκαν μπορούν να επαληθεύσουν ότι οι απαιτήσεις που συναντώνται καλύπτουν την κατανεμημένη ευθύνη.

Για κάθε ενότητα απαιτείτε:

- Βρείτε το εφαρμοστέο έγγραφο ενότητας και τις Προδιαγραφές στη MasterList των εγγράφων που χρησιμοποιούν το DocID.
- Αναζητήστε την ετικέτα με το δείκτη του εγγράφου που βρέθηκε, για να βρείτε το ίδιο το κείμενο της απαίτησης.
- επαναλάβετε τα παραπάνω βήματα σε ένα χαμηλότερο επίπεδο εάν χρειάζεται, π.χ. στο αρχείο έργου του υποέργου.

### **Κάθετο πίσω ίχνος (Vertical backwards trace)**

Η κάθετη προς τα πίσω ανίχνευση επιτρέπει σε κάποιον να ελέγξει το κείμενο απαιτήσεων κατά το σχεδιασμό, στο χαμηλότερο επίπεδο απαιτήσεων και να ελέγξει την πληρότητα των προδιαγραφών. Όταν ένα υψηλότερο επίπεδο απαίτησης λείπει, τότε μπορεί η απαίτηση του χαμηλότερου επιπέδου να μην είναι απαραίτητη. Όταν μια απαίτηση υψηλότερου επιπέδου θα πρέπει να προστεθεί λόγω συγκεκριμένου σχεδιασμού ή λόγω τεχνογνωσία τότε προστίθεται σε αυτό το επίπεδο, όπως αναφέρεται στην παρατήρηση σχετικά με τον κατάλογο του πίνακα Ιχνηλασιμότητας.

- Επιλέξτε χαμηλότερου επιπέδου απαίτηση MRS2A
- Βρείτε όλες τις εμφανίσεις της απαίτησης στη 3η στήλη του Πίνακα Ιχνηλασιμότητας
- Για κάθε περιστατικό ελέγξτε το υψηλότερο επίπεδο απαίτησης στην πρώτη στήλη και με τη χορηγηθέντα ευθύνη στη δεύτερη στήλη
- Βρείτε το εφαρμοστέο έγγραφο στην ενότητα Προδιαγραφές στη MasterList των εγγράφων χρησιμοποιώντας το DocID.
- Αναζητήστε την ετικέτα με το δείκτη του εγγράφου που βρέθηκε, για να βρείτε το ίδιο το κείμενο της απαίτησης.
- Αν χρειαστεί τα παραπάνω βήματα μπορεί να επαναληφθούν για την απαίτηση αυτή, για την ανίχνευση ενός επιπέδου περαιτέρω.

### **Απαιτήσεις ιχνηλασιμότητας**

Υπάρχει ανάγκη συσχέτισης ενός μοντέλου με ένα άλλο, το οποίο αποτελεί την ιχνηλασιμότητα π.χ. από το σχεδιασμό μέχρι τον κώδικα. Ramesh (2001) αφορά την ιχνηλασιμότητα και την ικανότητα να εντοπίζουν τα εξαρτημένα στοιχεία μέσα σε ένα μοντέλο και την ικανότητα να εντοπίσει τις αντίστοιχες θέσεις τους σε άλλα μοντέλα. Έρευνες σχετικά με την ιχνηλασιμότητα του λογισμικού έχουν γίνει ευρέως, και έχει διερευνηθεί τις τελευταίες δύο δεκαετίες ότι υποστηρίζει πολλές εφαρμογές, όπως η δοκιμές παλινδρόμησης (al Jang et., 2001), της αλλαγής των επιπτώσεων (Lee et al., 2001) και την οπτικοποίηση (Bohner και



Gracanin, 2003). Έχουν γίνει προσπάθειες για την απλούστευση της εξέλιξης του λογισμικού για το υψηλότερο επίπεδο του συστήματος λογισμικού που περιλαμβάνει τις επιχειρηματικές απαιτήσεις και τις προδιαγραφές. Για παράδειγμα, ο Wan Kadir (2005) αναπτύσσει ένα μετά-μοντέλο και ένα εργαλείο που ονομάζεται Brood (Business Rule-Driven Object-Oriented Design), το οποίο δείχνει ότι οι επαγγελματικοί κανόνες συνδέονται με το σχεδιασμό του λογισμικού. Η ιχνηλασιμότητα είναι θεμελιώδους σημασίας για την ανάπτυξη λογισμικού και τη συντήρηση των μεγάλων συστημάτων. Η οποία δείχνει την ικανότητα εντοπισμού από τις υψηλότερου επιπέδου περιλήψεις για τα χαμηλότερου επιπέδου αποσπάσματα, π.χ. από την απαίτηση με τον κωδικό της εφαρμογής ή αντιστρόφως. Ο προσδιορισμός του προαναφερθέντος είδους ιχνηλασιμότητας καλείται απαίτηση ιχνηλασιμότητας (Ramesh, 2001). Σύμφωνα με αυτό, ο Burd και ο Munro (1999) υποθέτουν ότι οι απαιτήσεις ιχνηλασιμότητας συνεπάγονται ότι όλα τα μοντέλα του συστήματος ενημερώνονται με συνέπεια, διαφορετικά είναι αδύνατο να εφαρμοστεί η ιχνηλασιμότητα λογισμικού του συστήματος. Πολλά πρότυπα τεχνολογίας λογισμικού που διέπουν την ανάπτυξη των συστημάτων (π.χ. IEEE / EIA 12207) υπογραμμίζουν την ανάγκη των απαιτήσεων ιχνηλασιμότητας που πρέπει να περιληφθούν στη διαδικασία ανάπτυξης λογισμικού και να αντιμετωπιστούν ως ένας παράγοντας ποιότητας προς την επίτευξη βελτίωσης της διαδικασίας. Η διαδικασία βελτίωσης μέσω του CMM (Capability Maturity Model) είναι ένας οδηγός επικοινωνίας ανάπτυξης για την εξέλιξη προς μια μηχανική λογισμικού τελειότητας για την οργανωτική βελτίωση (Saemu και Promproon, 2004). Για την καλύτερη διαχείριση μιας ανάλυσης επιπτώσεων αλλαγής σε μια ευρύτερη προοπτική, θα πρέπει να συσχετιστεί με την προσέγγιση ιχνηλασιμότητας που απαιτεί ένα πλούσιο σύνολο διαφόρων σχέσεων εντός και μεταξύ των διαφορετικών επιπέδου μοντέλων λογισμικού. Το Έργο του Sneed (Sneed, 2001) συσχετίζει την προσέγγιση της ιχνηλασιμότητας με μια αποθήκη δεδομένων κατασκευάζοντας εργασίες συντήρησης που συνδέονται με τον κώδικα για τις δοκιμές και την έννοια των μοντέλων. Η έννοια του μοντέλου του Sneed φαίνεται να είναι πάρα πολύ γενικευμένη, δηλαδή περιλαμβάνει τις απαιτήσεις, τους κανόνες των επιχειρήσεων, τις αναφορές, τις περιπτώσεις χρήσης, τις λειτουργίες των υπηρεσιών και των αντικείμενα των δεδομένων αλλά δεν περιγράφεται κάποιου είδους κριτική. Ο Bianchi και λοιποί (2000) εισάγουν ένα μοντέλο ιχνηλασιμότητας για να υποστηρίξει την ανάλυση των επιπτώσεων σε ένα Αντικειμενοστραφές περιβάλλον. Ωστόσο, η εργασία τους ασχολείται περισσότερο με χειρωνακτική διαδικασία η οποία είναι χρονοβόρα και επιρρεπής σε λάθη. Οι Lindvall και Sandahl (1998) παρουσιάζουν μια προσέγγιση ιχνηλασιμότητας που βασίζεται στη γνώση και στην ικανότητα αυτής να μας βοηθήσει να συλλέξουμε και να αναλύσουμε τις μετρήσεις στην αλλαγή του λογισμικού που σχετίζεται με την ανάλυση των επιπτώσεων για τις εκτιμήσεις των πόρων. Αναφέρονται ορισμένα αιτήματα αλλαγών στον επηρεασμένο κώδικα που από την άποψη των κλάσεων, αλλά δεν εμπλέκονται καθόλου απαιτήσεις και περιπτώσεις δοκιμής. Έρευνα σχετικά με τις

απαιτήσεις ιχνηλασιμότητας αποκαλύπτει ότι αν και τα παραπάνω εφαρμόζουν τις απαιτήσεις ιχνηλασιμότητας για την ανάλυση των επιπτώσεων, η έμφαση και οι προσεγγίσεις τους δεν επιτρέπουν μια άμεση σύνδεση μεταξύ των χειροποίητων αντικειμένων. Θα μπορούσε να προταθεί μια νέα προσέγγιση ιχνηλασιμότητας απαιτήσεων για την υποστήριξη άμεσων συνδέσεων μεταξύ των απαιτήσεων σε υψηλό επίπεδο και άλλων αντικείμενων σε χαμηλό επίπεδο ή αντίστροφα για να παρατηρηθούν πιθανές επιπτώσεις που θα μπορούσε να έχει η αλλαγή.

### **Αποκόμιση Πληροφοριών Ιχνηλασιμότητας**

Ο Letelier παρουσιάζει ένα πλαίσιο για τον προσδιορισμό των συνδέσεων της ιχνηλασιμότητας μεταξύ των απαιτήσεων υψηλού επιπέδου και των UML μοντέλων.

Παρά το γεγονός ότι προορίζονται κυρίως για τη βοήθεια των συμμετεχόντων στοιχείων στον εντοπισμό υψηλού επιπέδου (σε μορφή κειμένου) απαιτήσεων για διάφορα UML μοντέλα κατά την αρχική φάση ανάπτυξης, το πλαίσιο επιτρέπει επίσης συνδέσεις ιχνηλασιμότητας μεταξύ των στοιχείων των UML μοντέλων. Ο καθορισμός τέτοιου είδους συνδέσεων ιχνηλασιμότητας είναι όμως η ευθύνη των στοιχείων που συμμετέχουν. Οι συγγραφείς κάνουν χρήση ενός στοιχείου μεταμοντέλου UML ( UML metamodel element ), που ονομάζεται αφαίρεση ( abstraction ), για να συνδέσουν βελτιωμένα στοιχεία και στοιχεία που πρόκειται να βελτιωθούν ( refined and refinement elements ), επιτρέποντας έτσι τη δημιουργία μιας (ενδεχομένως επίσημης) χαρτογράφησης. Οι βελτιώσεις, ωστόσο, πρέπει να καθορίζονται από το σχεδιαστή, χρησιμοποιώντας ένα συγκεκριμένο στερεότυπο <<refine>> για να αντιπροσωπεύσει τις αναφορές της αφαίρεσης.

Ένας άλλος τρόπος για τη δημιουργία δεσμών ιχνηλασιμότητας αυτόματα είναι να βασίζεται αυτό σε ένα αυτοματοποιημένο μοντέλο μετασχηματισμών. Μόλις κάποιος ξέρει πώς μετασχηματίζεται ένα μοντέλο, η μετατροπή μπορεί να μελετηθεί και μπορούν να καθοριστούν οι συνδέσεις ιχνηλασιμότητας μεταξύ των στοιχείων των δύο εκδόσεων των μοντέλων. Οι εν λόγω μεταβολές αποτελούν τον πυρήνα της MDA και έχουν κάποιους αυτοματοποιημένους μετασχηματισμούς που ήδη έχουν περιγραφεί, για παράδειγμα για τα σχεδιαστικά πρότυπα. Η χρήση αυτών των μετασχηματισμών, για τη δημιουργία ιχνηλασιμότητας δεν έχει λάβει πολλή προσοχή γενικά με κάποιες εξαιρέσεις.

### **Πραγματοποίηση ανάλυσης αντίκτυπου**

Οι περισσότερες τεχνικές ανάλυσης των επιπτώσεων βασίζονται σε μια γραφική απεικόνιση των χειροποίητων αντικειμένων λογισμικού (κόμβοι) και τις εξαρτήσεις αυτών (άκρων). Ένα γράφημα μπορεί (για παράδειγμα) να αντιπροσωπεύει τις κλάσεις, τις ιδιότητες και τις

λειτουργίες (κόμβοι) και τους ορισμούς των χαρακτηριστικών / λειτουργιών στους κλάδους καθώς και τις κλήσεις λειτουργιών (άκρα), ή απλά τις εξαρτήσεις των κλάσεων. Αυτές οι εξαρτήσεις μπορούν να ανακτηθούν από τον κώδικα ή από τα μοντέλα σχεδίου. Σε αντίθεση με την απλή διάσχιση του γράφου εξάρτησης για τον εντοπισμό άμεσων και έμμεσων επιπτώσεων, η εργασία στο paper των Li L. και Offutt A. J., "Algorithmic analysis of the impact of changes to object-oriented software," *Proc. IEEE International Conference on Software Maintenance*, προτείνει μια λεπτομερή ανάλυση των αλλαγών, οργανωμένη σε μια έγγραφο που ταξινομεί την αλλαγή, για να μελετηθεί πώς ακριβώς διαδίδονται οι αλλαγές (αν το κάνουν).

Οι κανόνες δίνουν επακριβώς τους όρους υπό τους οποίους μόνο διαδίδονται οι αλλαγές, για παράδειγμα, υπό ποιες προϋποθέσεις θα λάβουν έμμεσες επιπτώσεις.

Σημειώστε ότι όλες αυτές οι στρατηγικές έχουν οριστεί για την οριζόντια ανάλυση των επιπτώσεων. Η κάθετη ανάλυση του αντικτύπου προτείνεται στα εξής papers:

Bohner S. A., "Software Change Impacts - An Evolving Perspective," *Proc. IEEE ICSM*, pp. 263-272, 2002. Mens T. and D'Hondt T., "Automating Support for Software Evolution in UML," *Automated Software Engineering (Springer)*, vol. 7 (1), pp. 39-59, 2000. Αν και αυτή η ορολογία δεν χρησιμοποιείται από αυτούς τους συγγραφείς. Δεν περιγράφεται ακριβής κάθετη προσέγγιση για την ανάλυση των επιπτώσεων όμως. Από τα όσα γνωρίζουμε, δεν υπάρχει τέτοια εργασία που να έχει αναφερθεί μέχρι σήμερα. Μια αξιοσημείωτη εξαίρεση είναι το εξής paper:

von Knethen A. and Grund M., "QuaTrace: A Tool Environment for (Semi-)Automatic Impact Analysis Based on Traces," *Proc. ICSM*, pp. 246-255, 2003, όπου η διατήρηση ιχνηλασιμότητας περιορίζεται σε ιχνηλασιμότητα μεταξύ υψηλού επιπέδου περιγραφής κειμένου απαίτησης και των περιπτώσεων χρήσης (και των περιγραφών περίπτωσης χρήσης).

Στο ίδιο πνεύμα, ένα πλαίσιο για τον συγχρονισμό και την εξέλιξη των μοντέλων διαφόρων ειδών (δηλαδή, διαφόρων metamodels) προτείνεται στο Ivkovic I. και Kontogiannis K., "Tracing evolution changes of software artifacts through model synchronization," *Proc. IEEE ICSM*, pp. 252-261, 2004, τα μοντέλα που θα συγχρονιστούν αντιμετωπίζονται ως γραφήματα και συγκεκριμένα σύμφωνα με το πλαίσιο. Οι εξαρτήσεις μεταξύ των προδιαγραφών του μοντέλου ( γραφήματα ) καθορίζονται σύμφωνα με το πλαίσιο (περιγραφές ιχνηλασιμότητας), και στη συνέχεια οι αλλαγές στα μοντέλα μπορούν να διαδοθούν. Οι συγγραφείς, ωστόσο, υποθέτουν ότι τα μοντέλα έχουν ήδη συγχρονιστεί, δηλαδή, θεωρούν ότι οι συνδέσεις ιχνηλασιμότητας είναι γνωστές και επομένως επικεντρώνονται στην εκπροσώπηση των δεσμών αυτών και τη χρήση προς για να διαδώσουν προς αλλαγές στα μοντέλα των οποίων τα μεταμοντέλα μπορεί να είναι και διαφορετικά.

## ΚΑΘΕΤΗ ΑΝΑΛΥΣΗ ΤΩΝ ΕΠΙΠΤΩΣΕΩΝ

Ο στόχος προς προσέγγισής προς είναι να εκτελεστεί κάθετη ανάλυση αντίκτυπου (vertical impact analysis VIA) των μοντέλων προς UML 2.0 με τον όσο πιο δυνατό αυτοματοποιημένο και αποτελεσματικότερο τρόπο. Για να γίνει αυτό, προτείνουμε μια προσέγγιση που περιλαμβάνει την οριζόντια ανάλυση αντίκτυπου (horizontal impact analysis – HIA), καθώς και προς πληροφορίες ιχνηλασιμότητας.

Πρώτον, οι ατομικές αλλαγές, οι οποίες είναι τα στοιχειώδη βήματα με τα οποία το ένα μοντέλο εξελίσσεται σε ένα άλλο, εντοπίζονται αυτόματα.

Δεύτερον, προκύπτουν βελτιώσεις από προς προσδιοριζόμενες ατομικές αλλαγές, συλλαμβάνοντας κατά συνέπεια την πρόθεση του χρήστη για ένα υψηλότερο επίπεδο αφαίρεσης από προς ατομικές αλλαγές.

Τρίτον, οι συνδέσεις ιχνηλασιμότητας, προς ορίζεται ανάλογα με το μοντέλο στοιχεία προς UML εκδόσεις των μοντέλων αυτόματα (π.χ., από τα στοιχεία του μοντέλου ανάλυσης στα στοιχεία του μοντέλο σχεδίου), με βάση προς προσδιορισθέντες βελτιώσεις, για να παρακολουθηθεί η πρόθεση του χρήστη.

Τέλος, με δύο εκδόσεις μοντέλων UML, προς αντίστοιχες συνδέσεις ιχνηλασιμότητας, καθώς και μια μία HIA προσέγγιση έτοιμη , η VIA μπορεί να γίνει για να δοθεί απάντηση στο ερώτημα: ποια είναι η επίπτωση των αλλαγών στο «βελτιωμένο» μοντέλο σε σχέση με το αρχικό μοντέλο;

Σκοπός είναι να αυτοματοποιηθούν τα παραπάνω τέσσερα βήματα όσο το δυνατόν περισσότερο, χωρίς να απαιτείται από τον σχεδιαστή να υλοποιεί συνεχώς βελτιώσεις, και να επικεντρωθεί στα τρία πρώτα.

Αν και η είσοδος του χρήστη μπορεί να είναι αναγκαία για ορισμένες αποφάσεις, ο στόχος είναι η ελαχιστοποίηση προς εμφάνισης τέτοιων ερωτημάτων του χρήστη.

Είναι σημαντικό να σημειωθεί ότι, σύμφωνα με προς τέσσερις δραστηριότητες που αναφέραμε παραπάνω, μια αλλαγή σε ένα μοντέλο μπορεί να χρησιμοποιηθεί για να προσδιοριστεί είτε η βελτίωση ( συγκεκριμένα ) και, συνεπώς, οι συνδέσεις προς ιχνηλασιμότητας ή να εκτελεστεί μια HIA ή/και μια VIA (αφού έχουν αναγνωριστεί οι συνδέσεις ιχνηλασιμότητας ).

Το σημαντικό είναι, στη συνέχεια, να είμαστε σε θέση να διακρίνουμε προς δύο διαδικασίες ( HIA / VIA ) κατά τη διάρκεια προς διαδικασίας ανάπτυξης λογισμικού.

Ως απλούστευση προς υπόθεσης εργασίας, και δεδομένου ότι έχουμε την πρόθεση να αυτοματοποιηθούν οι VIA και HIA στο πλαίσιο προς εργαλείου υπόθεσης UML ( UML case tool ) προς το Rational Rose Architect, θα θεωρήσουμε το ακόλουθο σενάριο:

1. Ο σχεδιαστής που εργάζεται πάνω σε ένα σύνολο UML μοντέλων λέει στο εργαλείο ( case tool ) σχετικά με το πότε αρχίζει τη βελτίωση ( refinement ) αυτών των μοντέλων. Σε αυτό το σημείο εγκαίρως, το εργαλείο υποθέτει ότι μπορεί να αναφέρεται σε αυτά τα μοντέλα, προς τα μοντέλα στο πρώτο επίπεδο προς αφαίρεσης (προς πούμε, στην ανάλυση) και προς τροποποιήσεις για προς σκοπούς προς αναγνώρισης των τελειοποιήσεων και προς συνδέσεις τη ιχνηλασιμότητας.
2. Όταν η βελτιωμένη δραστηριότητα τελειώσει ( δηλαδή έχει βελτιστοποιηθεί ), ο σχεδιαστής ενημερώνει το εργαλείο υπόθεσης: το recording τελειώνει. Σε αυτό το σημείο εγκαίρως, τα σημερινά μοντέλα UML θεωρούνται υποδείγματα στο δεύτερο επίπεδο προς αφαίρεσης (προς πούμε, στον σχεδιασμό). Οι βελτιώσεις και οι συνδέσεις προς ιχνηλασιμότητας έχουν εντοπιστεί.
3. Κάθε τροποποίηση σε ένα μοντέλο σε ένα από τα δύο επίπεδα αφαίρεσης προκαλεί στη συνέχεια κάποιες HIA και VIA. Για παράδειγμα, μια τροποποίηση σε ένα μοντέλο στο πρώτο επίπεδο προς αφαίρεσης ενεργοποιεί μια HIA (για τον προσδιορισμό των συγκρουόμενων στοιχείων στο ίδιο επίπεδο αφαίρεσης) και, στη συνέχεια, μια VIA (για τον προσδιορισμό των συγκρουόμενων στοιχείων στο δεύτερο επίπεδο αφαίρεσης).

Εάν αυτή η απλουστευμένη παραδοχή ως προς τον τρόπο HIA και VIA ταίριαζε σε μια εξέλιξη του κύκλου ζωής του λογισμικού θα αποτελούσε αντικείμενο των μελλοντικών ερευνών.

### **Ενσωματωμένες συνδέσεις ιχνηλασιμότητας.**

Σε αυτήν την προσέγγιση, οι συνδέσεις ιχνηλασιμότητας ενσωματώνονται στα μοντέλα που αναφέρονται υπό τη μορφή νέων στοιχείων μοντέλων. Παραδείγματος χάριν, μια κοινή πρακτική για να αντιπροσωπευτούν οι συνδέσεις ιχνηλασιμότητας καλυτέρευσης (refinement traceability links) στα πρότυπα UML είναι να προστίθενται στερεοτυπημένες ενώσεις (π.χ. «refines») που συνδέουν τα συμμετέχοντα στοιχεία. Αυτή η προσέγγιση είναι δημοφιλής με τους σχεδιαστές για την user-friendly ιδιότητά της καθώς αντιπροσωπεύει τις συνδέσεις ιχνηλασιμότητας ως οπτικά στοιχεία μοντέλων που οι άνθρωποι μπορούν εύκολα να επιθεωρήσουν και στα οποία μπορούν εύκολα να πλοηγηθούν.

Από την άλλη, αυτή η προσέγγιση ισχύει εγγενώς μόνο για την αντιπροσώπευση συνδέσεις ιχνηλασιμότητας μέσα στο ίδιο το μοντέλο. Επιπλέον, η ενσωμάτωση αυξανόμενων ποσών συνδέσεων ιχνηλασιμότητας μέσα σε ένα πρότυπο προκαλεί τη ρύπανσή του. Πρέπει να έχουμε στο νου μας ότι η ιχνηλασιμότητα είναι μόνο μια από τις πολλές ανησυχίες κατά τη σχεδίαση-modelling, και η ενσωμάτωση πληροφοριών σχετικών με όλες τις ανησυχίες μας μέσα στο πρότυπο βαθμιαία το επιβαρύνει με στοιχεία δευτεροβάθμιας σπουδαιότητας, και

άρα το μολύνει. Ένα άλλο θέμα είναι η ομοιομορφία. Εάν οι συνδέσεις ιχνηλασιμότητας αντιπροσωπεύονται με ad-hoc τρόπους (π.χ. χρησιμοποιώντας διαφορετικούς τύπους ενώσεων και ad-hoc ονομάτων για τα στερεότυπα σε UML), γίνεται δύσκολο να τα διακρίνουμε από τα άλλα πρότυπα στοιχεία. Αυτό κάνει την αυτοματοποιημένη επεξεργασία του project δύσκολη. Για να συμβάλει στην ομοιομορφία, συζητάτε ένα UML profile για την κωδικοποίηση της ιχνηλασιμότητας στα πρότυπα UML που χρησιμοποιούν ένα σαφώς ορισμένο σύνολο καθορισμένων στερεοτύπων.

### **Εξωτερικές συνδέσεις ιχνηλασιμότητας**

Σε αυτήν την προσέγγιση, οι συνδέσεις ιχνηλασιμότητας αποθηκεύονται εξωτερικά στα μοντέλα στα οποία αναφέρονται (δηλαδή σε ένα χωριστό μοντέλο). Μια απαίτηση αυτής της προσέγγισης, είναι ότι κάθε στοιχείο των σχετικών μοντέλων πρέπει να έχει ένα μοναδικό και μόνιμο χαρακτηριστικό γνώρισμα προσδιορισμού – ID, έτσι ώστε οι εξωτερικές συνδέσεις με αυτό να μπορούν να επιλυθούν σαφώς.

Εάν οι συνδέσεις πρόκειται να αποθηκευτούν σε ένα εξωτερικό μοντέλο, ένα ουσιαστικό βήμα είναι να καθοριστεί το metamodel με το οποίο αυτό το πρότυπο ιχνηλασιμότητας θα αντιστοιχιστεί. Η απλούστερη προσέγγιση είναι να καθοριστεί ένα custom metamodel για κάθε τύπο ιχνηλασιμότητας που χρειάζεται να αποθηκευτεί.

Μια περιπλοκότερη προσέγγιση είναι να καθοριστεί ένα γενικό metamodel, Αυτό το metamodel, με τους κατάλληλους μηχανισμούς επεκτασιμότητας, πρέπει να είναι σε θέση να κωδικοποιήσει οποιοδήποτε τύπο συνδέσεων ιχνηλασιμότητας.

Τέλος, μια άλλη επιλογή είναι να καθοριστεί ένας πυρήνας metamodel ιχνηλασιμότητας που κωδικοποιεί ένα βασικό σύνολο κοινών χαρακτηριστικών γνωρισμάτων και έπειτα να καθοριστούν επεκτάσεις για τους διαφορετικούς τύπους συνδέσεων ιχνηλασιμότητας. Αν και αυτό μπορεί να εφαρμοστεί με τα γενικά weaving tools όπως το ModelWeaver, η σημασιολογία των επεκτάσεων metamodel δεν είναι ακόμα σαφώς ορισμένη. Περαιτέρω συζήτηση στα πλεονεκτήματα και τα μειονεκτήματα ενός γενικού metamodel εναντίον ενός μικρού πυρήνα που μπορεί να επεκταθεί για κάθε περιοχή ιχνηλασιμότητας μπορεί να μειωθεί σε μια γενική συζήτηση για τις γενικές γλώσσες διαμόρφωσης (όπως η UML), εναντίον των domain specific languages.

Ανεξάρτητα από την ιχνηλασιμότητα του metamodel, ένα πλεονέκτημα της αποθήκευσης συνδέσεων ιχνηλασιμότητας σε χωριστά μοντέλα είναι ότι διευκολύνει το loose coupling μεταξύ των μοντέλων και των συνδέσεων, κρατώντας και κατά συνέπεια τα μοντέλα καθαρά. Επιπλέον, οι συνδέσεις που είναι βασισμένες σε id οι οποίες κωδικοποιούνται σε ένα

metamodel με σαφώς ορισμένη σημασιολογία είναι ευκολότερες να διαχειριστούν και να αναλυθούν με αυτοματοποιημένα εργαλεία. Σε αντίθεση, οι συνδέσεις που είναι βασισμένες σε id , δεν είναι ιδιαίτερα χρήσιμες στους ανθρώπους που χρειάζεται να επιθεωρήσουν τα μοντέλα. Λαμβάνοντας υπόψη ότι ένας κύριος ρόλος των συνδέσεων ιχνηλασιμότητας είναι να βοηθηθούν τους σχεδιαστές-modellers στην λήψη αποφάσεων, αυτό θεωρείται ένα σημαντικό μειονέκτημα αυτής της προσέγγισης.

Επισκόπηση στην ανάλυση των επιπτώσεων και στις τεχνικές ιχνηλασιμότητας

Η Ανάλυση των επιπτώσεων ( impact analysis ) χρησιμοποιείται ευρέως σε πολλές εφαρμογές, όπως στις δοκιμές παλινδρόμησης, στην οπτικοποίηση , στον έλεγχο συνέπειας και στις επιπτώσεις της αλλαγής. Εν τω μεταξύ, η ιχνηλασιμότητα έχει τη δυνατότητα να διευρύνει το πεδίο της ανάλυσης των επιπτώσεων σε διαφορετικά επίπεδα στο σύστημα λογισμικού. Ο τεμαχισμός του προγράμματος ,η ροή δεδομένων και η ροή του ελέγχου αποτελούν τη βάση για να υποστηριχθεί η ανάλυση των επιπτώσεων της αλλαγής . Οι τεχνικές αυτές παρέχουν κάποια σημαντικά αισιόδοξα αποτελέσματα όταν εφαρμόζεται σε μικρό κώδικα ή ένα πρόγραμμα. Καθώς ένα έργο μεγαλώνει, ο κώδικας γίνεται μεγαλύτερος και πιο πολύπλοκος. Η ανάγκη της διαχείρισης της ανάλυση των επιπτώσεων σε ξεχωριστές καταστάσεις δεν είναι πλέον εφικτή . Οι μηχανικοί λογισμικού ή οι συντηρητές μπορεί να προτιμούν να διαχειρίζονται τα χειροποίητα αντικείμενα του λογισμικού με μεγαλύτερες επιπτώσεις. Για παράδειγμα, σε μεγάλα συστήματα λογισμικού, το να αναγνωρίζουμε της μεθόδους, κλάσεις ή τα πακέτα που επηρεάστηκαν είναι πιο αποτελεσματικό από το αντίστοιχο σε ατομικές καταστάσεις του καθενός συστατικού ( πχ κλάσης ή πακέτου ή αντικειμένου ) ξεχωριστά (Zhoa, 2002). Όταν αυτά τα αντικείμενα θα αναγνωρίζονται ρητά στη συνέχεια ο χρήστης μπορεί διαισθητικά να αναγνωρίσει τις ατομικές καταστάσεις που έχουν τέτοια προβλήματα, τα οποία χρειάζονται τροποποίηση. Η πρώτη εντύπωση είναι το πώς να συνδέσετε τον κώδικα με τα συστατικά του υψηλότερου επιπέδου, για παράδειγμα τις απαιτήσεις. Στο σημείο αυτό υπεισέρχεται μια έννοια γνώσης λογισμικού. Μελετήθηκαν προσεκτικά μερικές στρατηγικές δεδομένου ότι αυτό θα επηρέαζε την ακρίβεια και την πληρότητα των πληττόμενων χειροποίητων αντικειμένων. Ένα τυπικό σενάριο των δομών του προγράμματος θα ήταν καλύτερα να εξηγηθεί με στρατηγικές «από πάνω προς τα κάτω» ή από κάτω προς τα πάνω (Mayrhauser and Lang, 1999). Όντας λειτουργικές οι γνώσεις που έχουν δημιουργηθεί με την στρατηγική από πάνω προς τα κάτω υποστηρίζουν την ανάγκη για μια καλά δομημένη ιεραρχία των εννοιών. Αυτές οι δομές που υποστηρίζονται από τη ροή δεδομένων και τον έλεγχο των εξαρτήσεων ροών παρέχουν ένα καλό αντίκτυπο αναζήτησης της γνώσης λογισμικού. Κατά την εφαρμογή της ανάλυσης των επιπτώσεων της αλλαγής, θα θέλαμε να προσδιοριστούν οι παράπλευρες επιπτώσεις ενός χειροποίητου αντικειμένου λογισμικού. Εάν ένα χειροποίητο αντικείμενο λογισμικού (π.χ. η μεταβλητή x) είναι η αλλαγή, θα επηρεάσει τις

άλλες κλήσεις και τα μέρη του προγράμματος που εξαρτώνται από αυτήν, μέσω των σχέσεων του προγράμματος. Οι κλήσεις πχ μεθόδων υπάγονται στις εξαρτήσεις ελέγχου, ενώ οι άλλες τις σχέσεις του προγράμματος, όπως η κληρονομικότητα, η σύζευξη, κλπ. υπάγονται στις εξαρτήσεις των δεδομένων που πρέπει να συλληφθούν από τον κώδικα. Αν κάποιος επεκτείνει την έννοια της μεταβλητής  $x$  σε μια κατηγορία για να αντιπροσωπεύουν μια παγκόσμια έννοια του τεχνούργημα λογισμικό (πρόκειται να τροποποιηθεί), τότε εφαρμόζεται στο συστατικό των εξαρτήσεων, όπως αναφέρθηκε προηγουμένως. Ως εκ τούτου, η ιχνηλασιμότητα λογισμικό χρειάζεται να εξετάσει αυτά τα τρία είδη των ρητών συνδέσεων όταν ασχολείται με τις εξαρτήσεις ανάλυση των αλλαγών των επιπτώσεων, δηλαδή τα δεδομένα, τις εξαρτήσεις τον έλεγχο και εξαρτήσεις συνιστώσα. Ονομάζοντας τον εντοπισμό ή μοτίβο string είναι μια άλλη ιχνηλασιμότητα σημαίνει χρησιμοποιούνται ευρέως σε προγραμματιστικό περιβάλλον. Ωστόσο, ονομάζοντας τον εντοπισμό έχει σοβαρές ελλείψεις. Είναι συχνά αποτυγχάνει, ιδιαίτερα όταν οι έννοιες είναι έμμεσα διαθέσιμες στον κώδικα, αλλά δεν είναι ανιχνεύσιμες ή κατανοητή από οποιαδήποτε αναγνωριστικά πρόγραμμα. Ονομάζοντας τον εντοπισμό δεν έχει θέση στο σημερινό πλαίσιο της έρευνας, διότι αγνοεί τις εξαρτήσεις πρόγραμμα συντακτική του οποίου η πληροφορία αυτή είναι πολύ χρήσιμο να λαμβάνονται υπόψη τα αποτελέσματα της αλλαγής. Η ανησυχία τώρα είναι το πώς να δημιουργούν συνδέσμους μεταξύ απαιτήσεων και εφαρμογή του κώδικα; Στο νέο προτεινόμενο μοντέλο, η στατική ανάλυση μόνο του δεν επαρκεί για την υλοποίηση των οριζόντιων ιχνηλασιμότητας. Στατική ανάλυση μπορεί να είναι αποδεκτό στον κωδικό, αλλά να τη συνδέσει με υψηλό επίπεδο, π.χ. απαιτήσεις, χρειάζεται μια πιο συγκεκριμένη λύση της ανάλυσης της έννοιας. Μια αποδεκτή προσέγγιση χρησιμοποιεί μια δυναμική ανάλυση, γνωστή ως τεχνική αναγνώρισης (Wilde and Casey, 1996). Η τεχνική αυτή εφαρμόζεται με βάση την εκτέλεση των δοκιμών ελέγχου ή επίσης γνωστά ως σενάρια δοκιμής. Καθώς η απαίτηση χαρακτηρίζεται από μία ή περισσότερες δοκιμές ελέγχου, η υλοποίηση κάθε απαίτησης μπορεί να υλοποιηθεί από την εκτέλεση αυτών των περιπτώσεων δοκιμής και συλλαμβάνοντας στοιχεία λογισμικού που έχουν επηρεαστεί από τις αλλαγές ( και συγκεκριμένα τις κλάσεις, τις μεθόδους η/και τα πακέτα ). Το όφελος από τη τοποθεσία της γενικότερης έννοιας-concept ( concept location ) μέσω της δυναμικής ανάλυσης είναι ότι το αποτέλεσμα είναι πιο ακριβές σε σχέση με το αποτέλεσμα που προκύπτει από τη στατική ανάλυση. Ο λόγος είναι απλά επειδή περιέχει την εκτέλεση του προγράμματος που μπορεί να διαπεράσει όλες τις πιθανές διαδρομές που η στατική ανάλυση μπορεί να χάσει. Ο στόχος εδώ είναι να τηρούνται οι απαιτήσεις της ιχνηλασιμότητας που περιλαμβάνουν τις απαιτήσεις, το σχέδιο, τον κώδικα και των περιπτώσεις δοκιμής, ενώ άλλα συστατικά υψηλού επιπέδου αφήνονται για μελλοντική δουλειά.



## Λειτουργική σημασιολογία (Operational Semantics)

Μια βασική ανησυχία και μια ερώτηση που πρέπει να ρωτήσουμε είναι: γιατί απαιτούνται διαφορετικοί τύποι σχέσεων ιχνηλασιμότητας; Πιο συγκεκριμένα, πρέπει να ρωτήσουμε: γιατί δεν είναι μια σχέση ιχνηλασιμότητας ικανοποιητική; Η απάντηση βρίσκεται στην προοριζόμενη χρήση για τις σχέσεις που καθορίζονται

Η ιχνηλασιμότητα έχει τρεις κύριες χρήσεις:

1. Querying: π.χ. ανάλυση αντίκτυπου, ερωτήσεις κάλυψης
2. Ακολουθία συνδέσεων σε όλο το κύκλο ζωής του project. Παραδείγματος χάριν, εάν θέλουμε να καθορίσουμε την απαίτηση που μια ιδιαίτερη δοκιμή αποδοχής ελέγχει.
3. Ενημερώνοντας το σύστημα και το documentation του. Στην Model-Driven Development το documentation θα περιλάβει τα ακριβώς καθορισμένα μοντέλα-πρότυπα, π.χ., όπως γράφονται σε UML.

Ο ρόλος της σημασιολογίας της ιχνηλασιμότητας είναι να βοηθήσει με αυτά τα έργα:

1. Μια σημασιολογία για την ιχνηλασιμότητα θα βοηθήσει τους υπεύθυνους τα queries τους (δηλ., θα τις βοηθήσει να γράψουν τις ακριβέστερες ερωτήσεις-queries, γρηγορότερα)
2. Θα βοηθήσει τους υπεύθυνους για την ανάπτυξη να βρουν τη σωστή σύνδεση για να ακολουθήσουν όταν θέλουν να καθορίσουν τις σχέσεις μεταξύ των οντοτήτων (ή εάν θέλουν γρήγορα και ανεπίσημα να καθορίσουν ότι μια σχέση ιχνηλασιμότητας υπάρχει μεταξύ των αντικειμένων) και για να υποστηρίξουν την αυτοματοποίηση της διαχείρισης της συνέπειας (π.χ., για να βοηθήσει να εξασφαλίσει ότι οι διαφορετικές εκδόσεις των προτύπων UML δεν περιέχουν ασυμβίβαστες πληροφορίες. Παραδείγματος χάριν, δύο διαφορετικές εκδόσεις ενός διαγράμματος κλάσης μεταξύ των οποίων μια σχέση ιχνηλασιμότητας υπάρχει μπορεί να πρέπει να είναι συνεπείς σύμφωνα με ένα συγκεκριμένο σύνολο κανόνων).

Μια λειτουργική σημασιολογία βοηθά πρώτιστα με την τελευταία χρήση της ιχνηλασιμότητας.

Μια λειτουργική σημασιολογία για μια σύνδεση ιχνηλασιμότητας περιγράφει:

- ποιες πληροφορίες πρέπει να διατηρηθούν μεταξύ των σχετικών στοιχείων
- ποιες ενέργειες πρέπει να προκληθούν όταν γίνεται κάτι που επηρεάζει ένα ή περισσότερα από τα σχετικά στοιχεία
- ποιες ενέργειες πρέπει να προκληθούν όταν εμφανίζεται κάτι σχετικό – που επιδρά στη σχέση από μόνο του - προκειμένου να εξασφαλιστεί ότι η σχέση είναι ακόμα ενεργή σύμφωνα με τη σημασιολογία του.

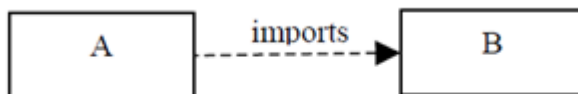
Με βάση αυτό, καθορίζουμε δύο τύπους σημασιολογίας για την ιχνηλασιμότητα:

1. Προληπτική σημασιολογία (Preventative semantics)
2. Αντιδραστική σημασιολογία (Reactive semantics)

Η προληπτική σημασιολογία περιγράφει τα πράγματα που δεν πρέπει να συμβούν.

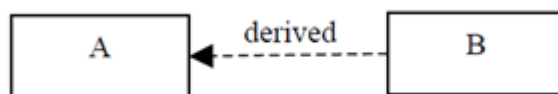
Παραδείγματος χάριν, εξετάστε την εξάρτηση στο σχήμα 1.16., εάν μια κλάση A εισάγει μια

κλάση B τότε η κλάση B δεν πρέπει να διαγραφεί προτού να αφαιρεθεί η εξάρτηση εισαγωγής(import).



Σχήμα 1.16. Παράδειγμα της σχέσης “imports”.

Η αντιδραστική σημασιολογία περιγράφει τι πρέπει να προκληθεί όταν συμβαίνει κάτι σε ένα ή περισσότερα από τα σχετικά στοιχεία ή στην ίδια τη σχέση. Παραδείγματος χάριν εάν η ιδιότητα B που προέρχεται από την ιδιότητα A, όποτε η τιμή της A αλλάζει, τότε πρέπει να υπολογιστεί ξανά η τιμή της ιδιότητας B.



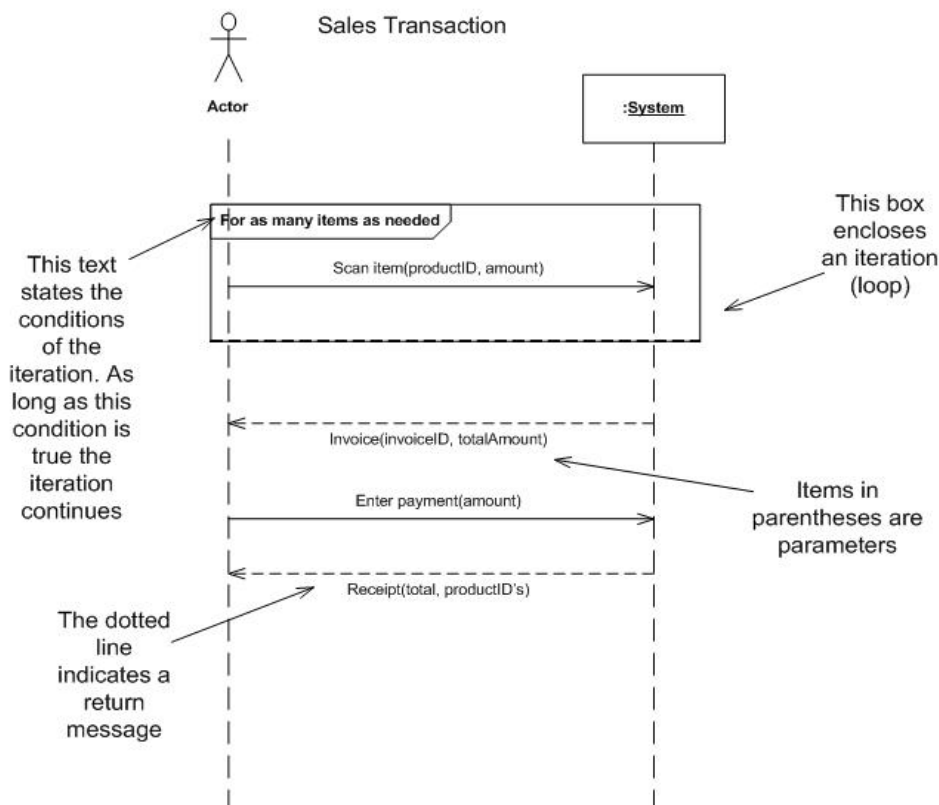
Σχήμα 1.17. Παράδειγμα της σχέσης “derived”.

## 1.6.Διάγραμμα Ακολουθίας Συστήματος SSD

Στη μηχανική λογισμικού, ένα Διάγραμμα Ακολουθίας Συστήματος ( **System Sequence Diagram – SSD** ) είναι ένα διάγραμμα ακολουθίας που δείχνει, για ένα συγκεκριμένο σενάριο μιας Περίπτωσης Χρήσης, τα γεγονότα που παράγουν οι εξωτερικοί actors, τη σειρά τους και πιθανώς τα γεγονότα που συμβαίνουν μέσα στο σύστημα. Τα διαγράμματα Ακολουθίας Συστήματος είναι οπτικές περιλήψεις από μεμονωμένες Περιπτώσεις Χρήστης. Όλα τα συστήματα μεταχειρίζονται σαν ένα μαύρο κουτί. Το διάγραμμα δίνει έμφαση σε γεγονότα που διασχίζουν το σύστημα, δηλαδή από τους actors έως τα συστήματα. Ένα SSD θα πρέπει να γίνει για το κύριο σενάριο της περίπτωσης χρήσης και για τα δευτερεύοντα σενάρια.

Ένα SSD θα πρέπει να διευκρινίζει και να δείχνει τα εξής:

- Εξωτερικοί actors
- Μηνύματα (μέθοδοι ) που προκαλούνται από αυτούς τους actors.
- Τιμές επιστροφής (αν υπάρχουν) που συσχετίζονται με τα προηγούμενα μηνύματα.
- Ενδείξεις για τυχόν επαναλήψεις/loops.
- Η παρακάτω εικόνα απεικονίζει τα παραπάνω.



Σχήμα 1.18. Διάγραμμα SSD

### Περισσότερα λόγια για τα Διαγράμματα Ακολουθίας Συστήματος

Οι επαγγελματίες, κατά την κατασκευή ενός λογισμικού, συχνά χρησιμοποιούν Διαγράμματα Ακολουθίας Συστήματος για να δείξουν πως γίνονται συγκεκριμένες ενέργειες μεταξύ χρηστών και συστήματος. Αυτές οι ενέργειες εμπεριέχουν επαναληπτικά, απλά ή και πιο σύνθετα καθήκοντα. Ο σκοπός είναι να απεικονιστεί η Περίπτωση χρήσης οπτικά. Για να κατασκευάσουμε λοιπόν ένα τέτοιο διάγραμμα, πρέπει να είμαστε εξοικειωμένοι με την UML. Αυτά τα μοντέλα δείχνουν την λογική πίσω από τους Actors (ανθρώπους που επηρεάζουν το σύστημα) και το Σύστημα καθώς εκτελείται μια ενέργεια. Η μελέτη ενός διαγράμματος ακολουθίας ξεκινά στην κορυφή με τους Actor/s ή το Σύστημα/τα (τα οποία βρίσκονται προφανώς στην κορυφή της σελίδας). Κάτω από τον κάθε Actor ή Σύστημα υπάρχουν μακριές παρατεταμένες γραμμές ζωής (lifelines); που συνεχίζουν από αυτά και προς τα κάτω. Οι ενέργειες πραγματοποιούνται με οριζόντιες γραμμές μεταξύ αυτών των γραμμών ζωής. Όταν μια ενέργεια συνδέεται με μια γραμμή ζωής δείχνει την διάδραση μεταξύ του Actor και του Συστήματος. Για παράδειγμα, ο actor μπορεί να κάνει μια αίτηση για να συνδεθεί, και αυτό θα παρουσιαζόταν με την `login(username, password)`. Αφού κάθε ενέργεια εκτελεστεί, η απάντηση ή η επόμενη ενέργεια τοποθετείται κάτω ακριβώς από την αμέσως προηγούμενη. Καθώς προχωράμε προς τα κάτω βλέπουμε λεπτομερώς πως εκτελούνται συγκεκριμένες ενέργειες στο σύστημα

## 1.7. Διάγραμμα Ακολουθίας (Sequence Diagram – SD)

### Σύνθεση και ιχνηλασιμότητα εκτελέσιμων μοντέλων που βασίζονται σε κάποιο σενάριο (Synthesis and Traceability of Scenario-based Executable Models)

Τα Message Sequence Charts (MSCs) ή Sequence Diagrams (Διαγράμματα ακολουθίας μηνυμάτων - Msc) ή διαγράμματα ακολουθίας χρησιμοποιούνται ευρέως από τους μηχανικούς απαιτήσεων στα πρώτα στάδια του reactive system design. Συμβατικά, τα Msc χρησιμοποιούνται στο έγγραφο απαιτήσεων συστήματος για να περιγράψουν τα σενάρια - πιθανοί τρόποι με τους οποίους οι διαδικασίες που αποτελούν ένα διανεμημένο αντιδραστικό σύστημα μπορούν να επικοινωνήσουν η μια με την άλλη καθώς επίσης και με το περιβάλλον. Λόγω της διαδεδομένης χρήσης τους στην εφαρμοσμένη μηχανική απαιτήσεων, τα Msc έχουν ενσωματωθεί στην UML. Στην UML, είναι γενικά γνωστά ως διαγράμματα ακολουθίας (Sequence Diagrams – SD). Οπτικά, ένα διάγραμμα ακολουθίας μηνυμάτων (Msc) αποτελείται από έναν αριθμό διεργασιών που επικοινωνούν η μια με την άλλη και που η καθεμία από αυτές παρουσιάζεται ως μια κάθετη γραμμή. Η επικοινωνία μηνυμάτων μεταξύ των διαδικασιών παρουσιάζεται ως οριζόντια ή με βέλη προς τα κάτω. Η επικοινωνία των μηνυμάτων μπορεί να είναι σύγχρονη («χειραψία» αποστολέα/δέκτη) ή ασύγχρονη (non-blocking send events). Σημασιολογικά, ένα Msc-SD δείχνει μια «ταμπελοποιημένη» μερική διάταξη των γεγονότων. Αυτή η μερική διάταξη είναι η μεταβατική περάτωση:

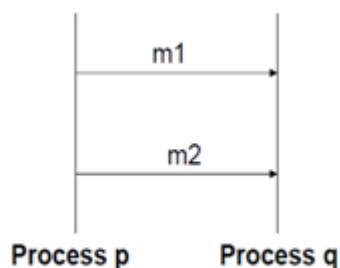
(α) της συνολικής διάταξης των γεγονότων σε κάθε διαδικασία και

(β) της διάταξης που επιβάλλεται από την επικοινωνία ενός μηνύματος - ένα μήνυμα παραλαμβάνεται αφότου στέλνεται.

Ένα SD παράδειγμα εμφανίζεται στο σχήμα 1.19.

Με βάση τη μετρική διάταξης που δείχνεται από αυτό το SD, δεν μπορούμε να συμπεράνουμε ότι οποιαδήποτε διάταξης μεταξύ του παραληφθέντος γεγονότος του μηνύματος m1 και του αποστέλλοντος γεγονότος του μηνύματος m2. Δεδομένου ότι ένα SD δείχνει ένα σενάριο στην εκτέλεση του συστήματος, δεν διαμορφώνει την πλήρη περιγραφή του. Αυτό το πρόβλημα μπορεί να λυθεί με τα MSC-graphs. Κάθε κόμβος ενός MSC-graph είναι ένα SD. Κατά συνέπεια, ένα MSC-graph αντιπροσωπεύει τα ίχνη εκτέλεσης που διαμορφώνονται με το ξετύλιγμα του MSC-graph από την αρχική κατάσταση και έπειτα την γραμμοποίηση των SD διαγραμμάτων που έχουν αποκτηθεί. Το πρότυπο των MSC-graphs είναι κατάλληλο για την περιγραφή του πλήρους συστήματος. Όταν επεκτείνεται με ιεραρχία διαμορφώνει τα δημοφιλή High-level Message Sequence Charts (HMSCs). Όμως, δεν μπορούμε απευθείας να κατασκευάσουμε τις εφαρμογές για τις συμμετέχουσες διαδικασίες με να προβάλουμε τις

συμπεριφορές τους μέσα στα Msc ενός HMSC, λόγω της παρουσίας “implied scenarios” («υπονοούμενων σεναρίων»)



Σχήμα 1.19. Διάγραμμα Ακολουθίας Μηνυμάτων (MSC)

## 1.8. Διάγραμμα Κλάσεων (Class Diagram)

### Διάγραμμα Κλάσεων :

#### Γενικά:

Το διάγραμμα κλάσης είναι ένα στατικό διάγραμμα. Αντιπροσωπεύει τη στατική άποψη μιας εφαρμογής. Το διάγραμμα κλάσης δε χρησιμοποιείται μόνο για απεικόνιση, περιγραφή και καταγραφή των διαφορετικών πτυχών ενός συστήματος αλλά επίσης για την κατασκευή εκτελέσιμου κώδικα της εφαρμογής του λογισμικού.

Το διάγραμμα κλάσης περιγράφει τις ιδιότητες και τις διαδικασίες-μεθόδους μιας κλάσης και επίσης τους περιορισμούς που τίθενται στο σύστημα constraints. Τα διαγράμματα κλάσης χρησιμοποιούνται ευρέως στη μοντελοποίηση Αντικειμενοστραφών συστημάτων επειδή είναι τα μόνα UML διαγράμματα που μπορούν να αντιστοιχιστούν κατευθείαν με αντικειμενοστραφής γλώσσες προγραμματισμού.

Το διάγραμμα κλάσης δείχνει μια συλλογή κλάσεων, διεπαφών, συσχετίσεων, συνδέσεων και περιορισμών. Είναι επίσης γνωστό και ως διάγραμμα δομής.

#### Σκοπός:

Ο σκοπός του διαγράμματος κλάσης είναι να μοντελοποιήσει τη στατική άποψη μιας εφαρμογής. Τα διαγράμματα κλάσης είναι τα μοναδικά διαγράμματα τα οποία μπορούν να αντιστοιχιστούν άμεσα με κάποια Αντικειμενοστραφή γλώσσα Προγραμματισμού και άρα, να χρησιμοποιηθούν την ώρα της κατασκευής του λογισμικού.

Τα UML διαγράμματα όπως τα Διαγράμματα Ενεργειών ή Ακολουθίας μπορούν μόνο να δώσουν την ροή της ακολουθίας της εφαρμογής αλλά το διάγραμμα κλάσης είναι λίγο διαφορετικό. Για αυτόν το λόγο, είναι το πιο δημοφιλές UML διάγραμμα στην προγραμματιστική κοινότητα.

Έτσι, ο σκοπός του διαγράμματος κλάσης είναι ο εξής, σύντομα:

- Ανάλυση και σχεδίαση της στατικής άποψης του συστήματος.
- Περιγραφή των ευθυνών του συστήματος.
- Αποτελεί τη βάση για Ψηφιδικά ( component ) και Παραταξιακά ( deployment ) διαγράμματα .
- Ευθεία και ανάστροφη μηχανική.

### **Που χρησιμοποιούνται τα Διαγράμματα Κλάσης;**

Η στατική άποψη του διαγράμματος κλάσης που αναφέραμε πιο πάνω περιγράφει το λεξικό του συστήματος. Το διάγραμμα κλάσης θεωρείται επίσης ως η βάση για άλλα διαγράμματα, όπως τα Ψηφιδικά ( component ) και Παραταξιακά ( deployment ) διαγράμματα . Το διάγραμμα κλάσης δε χρησιμοποιείται μόνο για απεικόνιση των διαφορετικών πτυχών ενός συστήματος , όπως προείπαμε , αλλά επίσης για την κατασκευή εκτελέσιμου κώδικα της εφαρμογής του λογισμικού για ευθεία και ανάστροφη μηχανική.

Γενικά τα UML διαγράμματα δεν αντιστοιχούν με κάποια Αντικειμενοστραφής γλώσσα προγραμματισμού αλλά τα διαγράμματα κλάσης αποτελούν μια εξαίρεση.

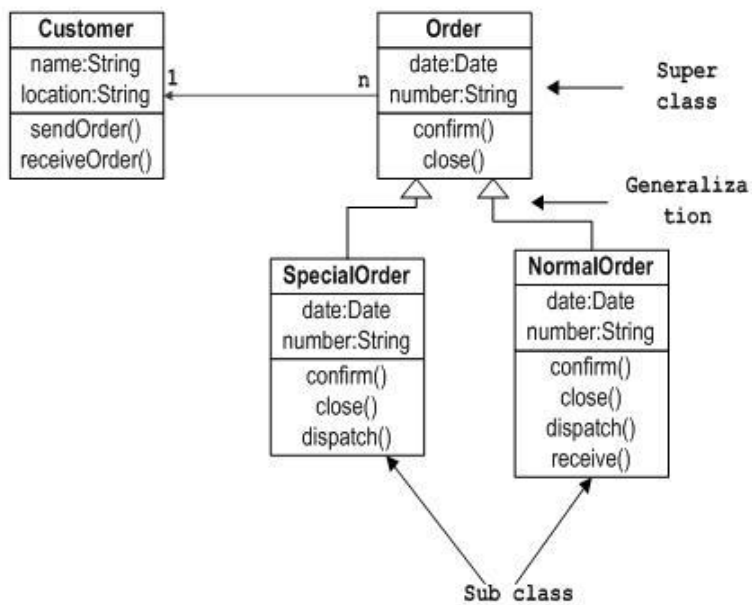
Το διάγραμμα κλάσης δείχνει ξεκάθαρα την αντιστοίχιση με αντικειμενοστραφής γλώσσες όπως η Java. Έτσι, από πρακτική εμπειρία το διάγραμμα κλάσης χρησιμοποιείται γενικότερα για κατασκευαστικούς σκοπούς.

Εν συντομία το Διάγραμμα Κλάσης έχει τις εξής χρήσεις :

- Περιγραφή της στατικής απόψεως ενός συστήματος.
- Να δείξει τη συνεργασία μεταξύ των στοιχείων της στατικής απόψεως.
- Περιγραφή των λειτουργιών που παρουσιάζονται από το σύστημα.
- Κατασκευή εφαρμογών λογισμικού με τη χρήση Αντικειμενοστραφών γλωσσών Προγραμματισμού.

## Παράδειγμα ενός Διαγράμματος Κλάσεως:

Sample Class Diagram



## ΚΕΦΑΛΑΙΟ 2

### Περίπτωση Χρήσης

#### Ορισμός :

Ως περίπτωση Χρήσης ( Π. Χ. ) ορίζεται ένα σύνολο σεναρίων που συνδέονται με ένα κοινό σκοπό (για τον χρήστη)

Συνήθως περιλαμβάνει:

- Ένα πρωταρχικό σενάριο (σαν μια ακολουθία από βήματα)
- Τις εναλλακτικές περιπτώσεις (σαν παραλλαγές της κύριας ακολουθίας βημάτων)

#### Στόχοι της Μοντελοποίησης Περιπτώσεων Χρήσης

- Να καθορίσει και να περιγράψει τις *λειτουργικές απαιτήσεις* του συστήματος .
- Να δώσει μια σαφή και συνεπή περιγραφή για το *τι θα πρέπει να κάνει* το σύστημα .
- Να παρέχει την κατάλληλη βάση για να γίνονται έλεγχοι για επαλήθευση του συστήματος .
- Να παρέχει την ικανότητα ν' *ανιχνεύονται (εντοπίζονται)* οι λειτουργικές απαιτήσεις μέσα στις κλάσεις και τις *λειτουργίες του συστήματος* .

#### Ανάπτυξη της Περίπτωσης Χρήσης “Process Sale”- Διεκπεραίωση Πώλησης

##### Περίπτωση Χρήσης ΠΧ1: Διεκπεραίωση Πώλησης(Process sale)

**Κύριος Χρήστης:** Ταμίας

...

##### Κύριο Σενάριο Επιτυχίας (ή Βασική Ροή):

1. Ο Πελάτης έρχεται σε ένα ταμείο POS για να αγοράσει προϊόντα.
2. Ο Ταμίας ξεκινά μια νέα πώληση.

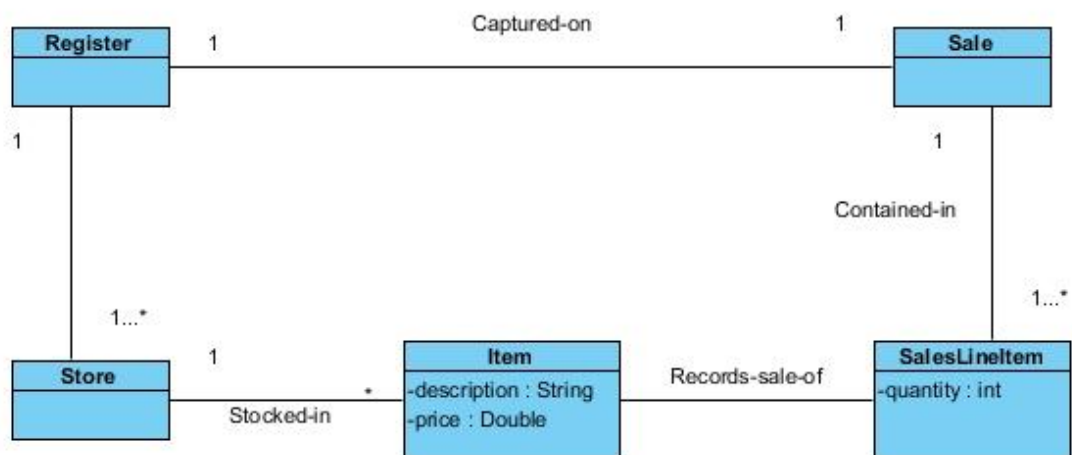


3. Ο Ταμίας εισάγει τον κωδικό του προϊόντος και την ποσότητα.
  4. Το σύστημα καταγράφει το προϊόν και παρουσιάζει την περιγραφή, την τιμή και το τρέχων σύνολο.
- Ο Ταμίας επαναλαμβάνει τα βήματα 3-4 μέχρι να δηλώσει την ολοκλήρωση της διαδικασίας.
5. Το σύστημα εμφανίζει το συνολικό κόστος, συμπεριλαμβανομένων και των φόρων.
  6. Ο Ταμίας ενημερώνει τον πελάτη για το κόστος, και ζητάει την πληρωμή.
  7. Ο Πελάτης πληρώνει και το Σύστημα διαχειρίζεται την πληρωμή.
  8. Το Σύστημα καταγράφει την ολοκληρωμένη πώληση και στέλνει πληροφορίες σχετικές με την πώληση και την πληρωμή στο εξωτερικό Λογιστικό Σύστημα (για λογιστική επεξεργασία και υπολογισμό προμηθειών), καθώς και στο Αποθεματικό σύστημα (για την ενημέρωση των αποθεμάτων).
  9. Το Σύστημα εκδίδει απόδειξη.
  10. Ο Πελάτης αναχωρεί με την απόδειξη και τα προϊόντα.

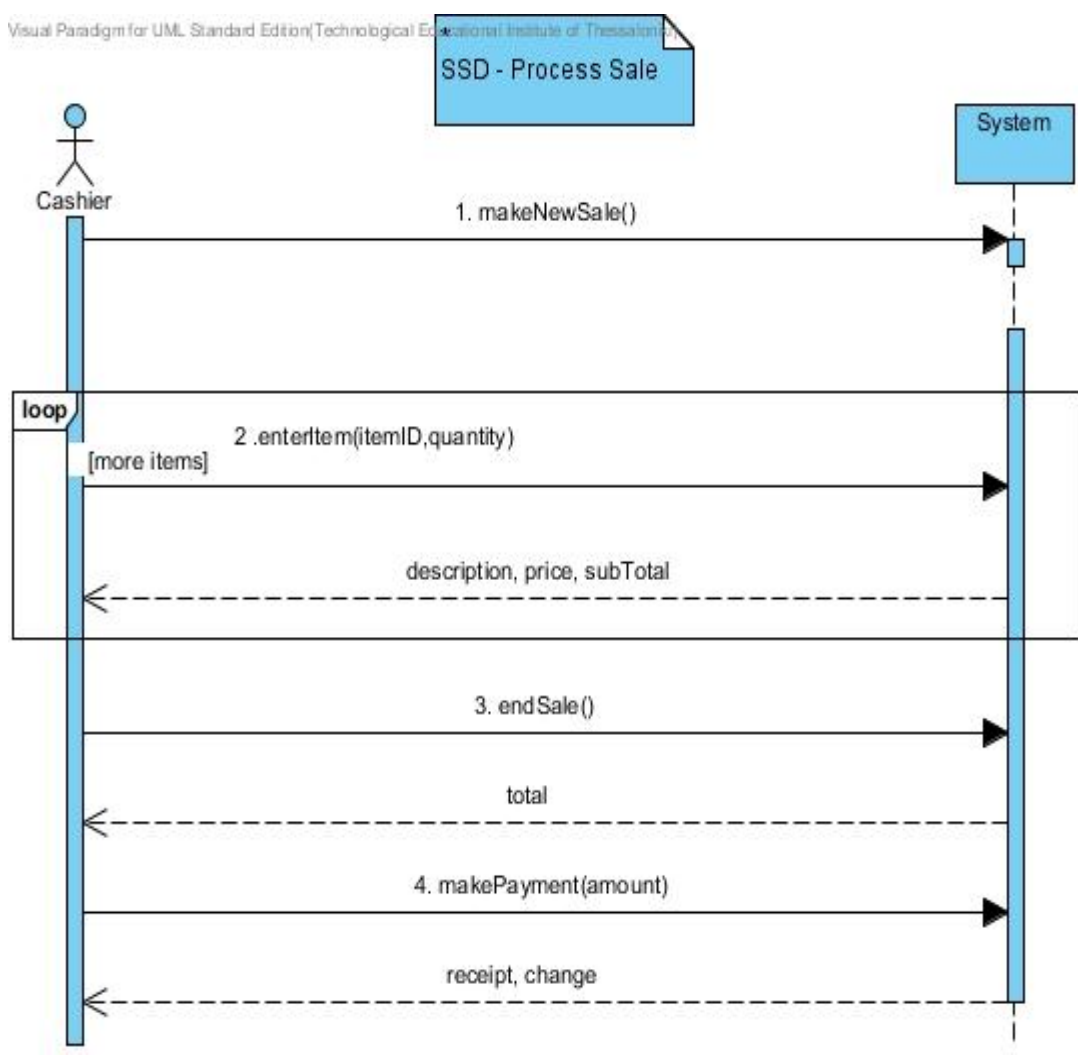
## Εννοιολογικό Μοντέλο

Visual Paradigm (© UML Standard Edition (Technical Educational Institute of Thessaloniki))

Εννοιολογικό Μοντέλο

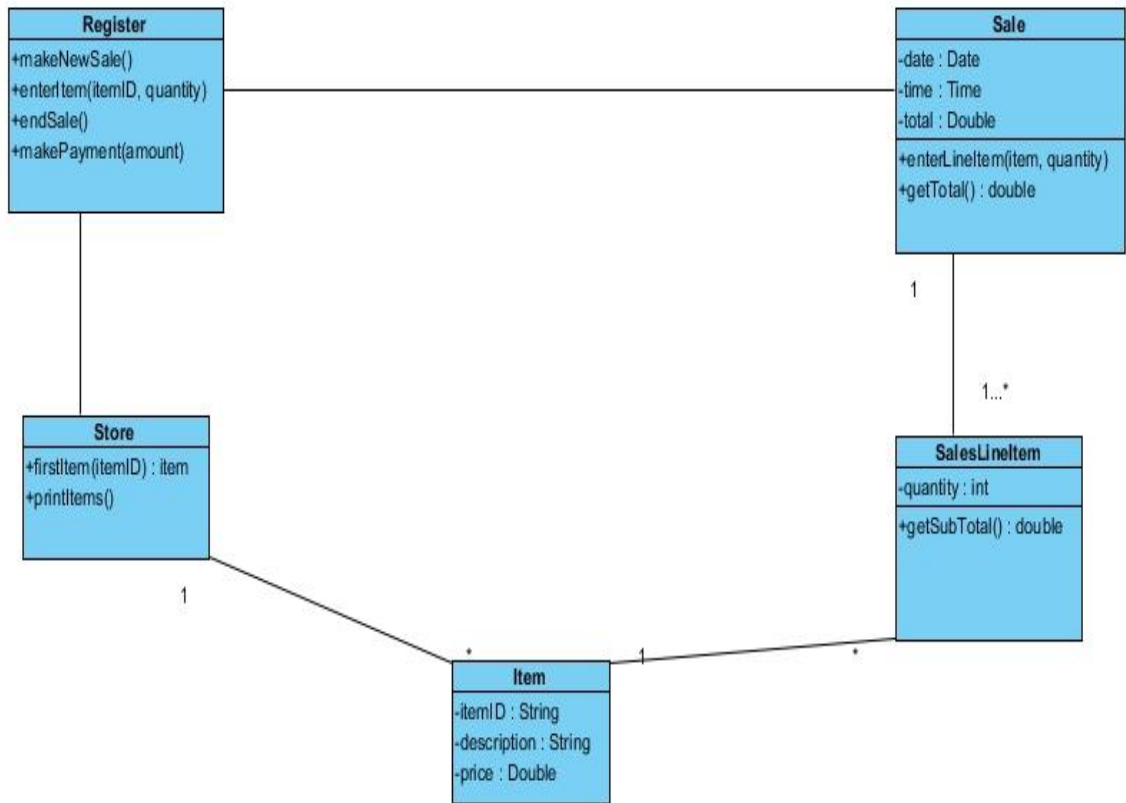


# Διάγραμμα Ακολουθίας



# Διάγραμμα Κλάσεων

UML Paradigm for UML Standard Edition (Technological Educational Institute of Thessaloniki)  
Class Diagram



**Κώδικας:****Class Main:**

```
public class Main {  
  
    public static void main(String[] args) {  
  
        Store store = new Store();  
        Register reg = new Register(store);  
        store.addRegister(reg);  
  
        // ΣΥΜΒΑΝ 1: makeNewSale()  
        reg.makeNewSale("19/3/2010", "10:24");  
        // ΣΥΜΒΑΝ 2: enterItem(itemID, quantity) - Επαναληπτική διαδικασία...  
        // εισάγουμε τον κωδικό του προϊόντος και την ποσότητα  
        reg.enterItem("x1", 4);  
        // εισάγουμε ένα ακόμη προϊόν με την ποσότητα  
        reg.enterItem("x2", 3);  
        // ΣΥΜΒΑΝ 3:  
        reg.endSale();  
        // ΣΥΜΒΑΝ 4: makePayment(amount)  
  
    }  
}
```

**Class Register :**

```
//import java.util.*;
```

```

public class Register {
    private Store store;
    private Sale currentSale;

    public Register(Store store) {
        this.store = store;
    }

    public void makeNewSale(String date, String time) {
        currentSale = new Sale(date, time);
        System.out.println("\n" + currentSale);
    }

    public void enterItem(String itemID, int quantity){
        // πρώτα βρίσκουμε το Προϊόν
        Item item = store.findItem(itemID);

        // μετά δημιουργούμε μία γραμμή πώλησης με το προϊόν, η οποία εμφανίζει
        // και το τρέχον σύνολο
        currentSale.enterLineItem(item, quantity);
    }

    public void endSale(){
        System.out.println("ΓΕΝΙΚΟ ΣΥΝΟΛΟ: " + currentSale.getTotal());
        System.out.println("-----");
    }
}

```

**Class Store :**

```
import java.util.*;
import java.util.ArrayList;
import java.util.GregorianCalendar;
import java.util.Iterator;

public class Store {
    private Register reg;
    private ArrayList<Item> prodCat;
    public Store() {
        prodCat = new ArrayList();
        // Δημιουργεί ένα αρχικό κατάστημα
        makeInitialStore();
    }
    public void addRegister(Register reg) {
        this.reg = reg;
    }
    /**
     * Δημιουργεί ένα αρχικό κατάστημα και πραγματοποιεί
     * κάποιες αρχικές συναλλαγές. */
    private void makeInitialStore(){
        // Δημιουργεί και καταχωρεί στο κατάστημα 5 προϊόντα
        prodCat.add(new Item("x1", "Σαμπουάν", 4.35));
        prodCat.add(new Item("x2", "Σαπούνι", 1.28));
        prodCat.add(new Item("x3", "Μαλακτικό Ρούχων", 8.72));
    }
}
```

```
//Εμφάνισε τα προϊόντα
System.out.println("Διαθέσιμα προϊόντα στο κατάστημα");
printItems();
}

public Item findItem(String itemId) {
    Iterator itr = prodCat.iterator();
    while (itr.hasNext()) {
        Item item = (Item) itr.next();
        if (item.getItemId().equals(itemId)) {
            return item;
        }
    }
    return null;
}

public void printItems() {
    Iterator itr = prodCat.iterator();
    while (itr.hasNext()) {
        Item item = (Item) itr.next();
        System.out.println(item);
    }
    System.out.println("\n");
}
}
```



**Class Item :**

```
public class Item {  
    private String itemID;  
    private String description;  
    private double price;  
  
    public Item(String itemID, String description, double price) {  
        this.itemID = itemID;  
        this.description = description;  
        this.price = price;  
    }  
  
    public void setDescription(String description) {  
        this.description = description;  
    }  
  
    public void setItemID(String itemID) {  
        this.itemID = itemID;  
    }  
  
    public void setPrice(double price) {  
        this.price = price;  
    }  
  
    public String getItemID() {  
        return itemID;  
    }  
  
    public double getPrice() {
```

```

        return price;
    }
    public String getDescription() {
        return description;
    }
    @Override
    public String toString() {
        return "" + itemID +
            ". " + description +
            " (" + price + ") Euro";
    }
}

```

### **Class SalesLineItem :**

```

public class SalesLineItem {
    private Item item;
    private int quantity;

    public SalesLineItem(Item item, int quantity) {
        this.item = item;
        this.quantity = quantity;
    }

    public double getSubTotal() {
        double subTotal = item.getPrice() * quantity;
    }
}

```

```
        return(subTotal);
    }

    public String getDescription() {
        return item.getDescription();
    }

    public double getPrice () {
        return item.getPrice();
    }

    public String toString() {
        return ("SalesLineItem: " + item.getItemID() + " " +
            item.getDescription() + " " +
            item.getPrice() + " * " +
            quantity + " " +
            "= Ποσό: " + getSubTotal());
    }
}
```

**Class Sale :**

```
import java.util.ArrayList;
import java.util.Iterator;
```

```
public class Sale {  
    private String date;  
    private String time;  
    private double total =0;  
    private ArrayList <SalesLineItem> saleLines;  
    //private Payment payment;  
  
    public Sale(String date, String time) {  
  
        this.date = date;  
        this.time = time;  
        this.saleLines = new ArrayList();  
    }  
  
    public void enterLineItem(Item item, int quantity) {  
        // 2.2.1: δημιουργείται μία γραμμή πώλησης  
        SalesLineItem sli = new SalesLineItem(item, quantity);  
        // 2.2.2: η γραμμή πώλησης προστίθεται στον πίνακα saleLines της Sale  
        saleLines.add(sli);  
        // 2.2.3: η γραμμή πώλησης ενημερώνεται με το τρέχον σύνολο  
        total+=sli.getSubTotal();  
        // 2.2.4: το σύστημα εμφανίζει την γραμμή πώλησης  
        showLineItem(sli);  
    }  
}
```

```

public void showLineItem(SalesLineItem sli) {
    // 2.2.4.1: το σύστημα εμφανίζει την περιγραφή, τιμή και τρέχον σύνολο
    System.out.print(sli.getDescription() + " " + sli.getPrice() + " " + sli.getSubTotal());
}

public double getTotal() {
    return total;
}

/*
public void makePayment(double amount){
    payment = new Payment(amount);
}

public void becomeComplete() {
    if(payment.getMoney()>=getTotal() && entries.size(>0)
        setComplete(true);
    else
        setComplete(false);
}
*/
public String toString() {
    return ("Sale: " + date + " " + time);
}
}

```

## ΚΕΦΑΛΑΙΟ 3

### Μετρικές

#### Εισαγωγή

Το αντικειμενοστραφές σχέδιο και η ανάπτυξη γίνονται πολύ δημοφιλή στο σύγχρονο περιβάλλον ανάπτυξης λογισμικού. Η αντικειμενοστραφής ανάπτυξη απαιτεί όχι μόνο μια διαφορετική προσέγγιση στο σχέδιο και την υλοποίηση, αλλά απαιτεί μια διαφορετική προσέγγιση στις μετρικές λογισμικού. Δεδομένου ότι η αντικειμενοστραφής τεχνολογία χρησιμοποιεί τα αντικείμενα και όχι τους αλγορίθμους ως τα «θεμέλιά» της, η προσέγγιση στις μετρικές λογισμικού για τα αντικειμενοστραφή προγράμματα πρέπει να είναι διαφορετική από ένα τυποποιημένο σύνολο μετρικών. Μερικές μετρικές, όπως οι γραμμές κώδικα και η cyclomatic complexity(CC), έχουν γίνει αποδεκτές ως "standard" για τα παραδοσιακά λειτουργικά/διαδικαστικά προγράμματα, αλλά για τα αντικειμενοστρεφή, υπάρχουν πολλές προτεινόμενες αντικειμενοστρεφείς μετρικές στη βιβλιογραφία. Η ερώτηση είναι, " Ποιες αντικειμενοστρεφείς μετρικές πρέπει ένα project να χρησιμοποιήσει, και μπορούν οποιεσδήποτε από τις παραδοσιακές μετρικές να είναι προσαρμοστούν και να λειτουργήσουν σωστά στο αντικειμενοστρεφές περιβάλλον; " Το αντικειμενοστρεφές σχέδιο και η ανάπτυξη είναι δημοφιλείς έννοιες στο σύγχρονο περιβάλλον ανάπτυξης λογισμικού. Λέγεται συχνά ότι είναι ο από μηχανής Θεός για την επίλυση των προβλημάτων του λογισμικού, ενώ στην πραγματικότητα δεν υπάρχει κανένας από μηχανής Θεός. Η αντικειμενοστρεφής ανάπτυξη έχει αποδείξει την αξία της για τα συστήματα που πρέπει να συντηρηθούν και να τροποποιηθούν. Η αντικειμενοστρεφής ανάπτυξη λογισμικού απαιτεί μια διαφορετική προσέγγιση από την παραδοσιακή λειτουργική ροή αποσύνθεσης και τις μεθόδους ανάπτυξης ροής δεδομένων(data flow development methods). Ενώ οι παραδοσιακές αυτές προσεγγίσεις αρχίζουν με την εξέταση ή την συμπεριφορά του συστήματος ή/και των στοιχείων χωριστά, η αντικειμενοστρεφής ανάλυση προσεγγίζει το πρόβλημα ψάχνοντας τις οντότητες συστημάτων που συνδυάζουν την συμπεριφορά αυτή του συστήματος και τα δεδομένα. Η αντικειμενοστρεφής ανάλυση και το σχέδιο στρέφονται στα αντικείμενα ως τους αρχικούς πράκτορες που συμμετέχουν σε έναν υπολογισμό. Κάθε κλάση δεδομένων και σχετικών διαδικασιών συλλέγονται σε μια μοναδική οντότητα συστημάτων.

### 3.1. Βασικές έννοιες που σχετίζονται με τις μετρικές.

**List of classes.** Ένας κατάλογος κλάσεων, για την περιγραφή ενός διαγράμματος κλάσης, είναι μια συνδεμένη λίστα κόμβων κλάσεων που αποτελούνται από το όνομα των κλάσεων, των ιδιοτήτων και των διαδικασιών κάθε κατηγορίας, και των σχέσεων κληρονομιάς.

**Collaboration sequence.** Μια ακολουθία συνεργασίας, που προέρχεται από ένα διάγραμμα συνεργασίας, είναι μια διαταγμένη συνδεμένη λίστα κόμβων αντικειμένου συνεργασίας που αποτελούνται από τα ονόματα των αντικειμένων, τα ονόματα των αντικειμένων-συνεργατών, τα μηνύματα μεταξύ των συνεργατών-αντικειμένων και των αντικειμένων, και την ακολουθία και την κατεύθυνση των μηνυμάτων.

**Activity graph.** Μια γραφική παράσταση δραστηριότητας, που αντιπροσωπεύει ένα διάγραμμα δραστηριότητας, είναι μια ένας κατευθυνόμενος γράφος κόμβων κατάστασης δραστηριότητας που αποτελείται από τα ονόματα των καταστάσεων, τους τύπους καταστάσεων, τις ενέργειες των καταστάσεων, τα ονόματα και τις κατευθύνσεις των μεταβάσεων, και των ενεργειών των μεταβάσεων.

**Attribute:** Το attribute καθορίζει τις δομικές ιδιότητες των κλάσεων, μοναδικές μέσα σε μια κλάση, συνήθως ένα ουσιαστικό.

**Class:** Ένα σύνολο αντικειμένων που μοιράζονται μια κοινή δομή και μια κοινή συμπεριφορά που φανερώνονται με ένα σύνολο μεθόδων το σύνολο χρησιμεύει ως ένα πρότυπο από το οποίο το αντικείμενο μπορεί να είναι instantiated (δημιουργημένος).

**Συνοχή:** Ο βαθμός με τον οποίο οι μέθοδοι μέσα σε μια κατηγορία συσχετίζονται η μία με την άλλη.

**Σύζευξη:** Το αντικείμενο X συνδέεται με το αντικείμενο Y εάν και μόνο εάν το X στέλνει ένα μήνυμα στο Y.

**Inheritance:** Μια σχέση μεταξύ κλάσεων, όπου ένα αντικείμενο σε μια κλάση αποκτά τα χαρακτηριστικά από μια ή περισσότερες άλλες κλάση.

**Instantiation:** Η διαδικασία της δημιουργίας μιας αναφοράς ενός αντικειμένου και την δέσμευση η προσθήκη συγκεκριμένων δεδομένων.

**Message:** Μια αίτηση που ένα αντικείμενο σε ένα άλλο αντικείμενο κάνει προκειμένου να εκτελέσει μια λειτουργία.

**Method:** Μια αίτηση που ένα αντικείμενο υποβάλλει σε ένα άλλο αντικείμενο προκειμένου να εκτελέσει μια λειτουργία.

**Object:** Μια αναφορά μιας κλάσης που μπορεί να σώσει μια κατάσταση(πληροφορία) και που προσφέρει έναν αριθμό από λειτουργίες για να εξετάσει ή να επηρεάσει αυτήν την κατάσταση.

**Operation:** Μια δράση που εκτελείται από ή σε ένα αντικείμενο, διαθέσιμο σε όλες τις αναφορές της κλάσης αυτής, δεν χρειάζεται να είναι μοναδική.

### **3.2. Τι είναι οι Μετρικές και λίγα λόγια για αυτές.**

Ο Mills προτείνει ότι οι ιδανικές μετρικές πρέπει να είναι απλές διευκολύνοντας την αξιολόγησή. Πρέπει να μετράνε αυτό που προορίζονται να μετράνε. Πρέπει επίσης να είναι αντικειμενικές, σταθερές και εύκολα αποκτήσιμες. «Μια μετρική λογισμικού είναι μια μέτρηση που προέρχεται από ένα προϊόν λογισμικού, μια διαδικασία, ή έναν πόρο. Ο σκοπός της είναι να παράσχει μια ποσοτική αξιολόγηση για το βαθμό στον οποίο το προϊόν, η διαδικασία, ή ο πόρος κατέχουν ορισμένες ιδιότητες.» Σύμφωνα με τον Costers οι μετρικές λογισμικού είναι απαραίτητες προκειμένου να διαχειριστεί αποτελεσματικά η ανάπτυξη λογισμικού. Αυτό μπορεί να γίνει με το ακριβές πρόγραμμα, με προϊόντα καλύτερης ποιότητας, καλή παραγωγικότητα και καλούς προϋπολογισμοί δαπανών. Οι στόχοι των μετρικών λογισμικού είναι να αναγνωρίσουν και να μετρήσουν τους ουσιαστικούς παράγοντες που επηρεάζουν την ανάπτυξη λογισμικού. Εντούτοις, οι οργανισμοί αντιμετωπίζουν ορισμένα εμπόδια για τις μετρικές όπως οι μη σωστά καθορισμένες μετρικές, τη διαστρέλωση από τον κύκλο ζωής λογισμικού, την ημιτέλεια των προτύπων, την αδυναμία της διάσπασης σε μεμονωμένα κομμάτια μέτρησης και την έλλειψη της επικύρωσης των μετρικών.

Οι μετρικές που μαζεύονται από τη φάση απαιτήσεων περιλαμβάνουν τη συγκρότηση λειτουργιών μεγέθους μετρικών, τις γραμμές κώδικα και την πολυπλοκότητα. Η ποιότητα των απαιτήσεων παραδείγματος χάριν, μπορεί να είναι μετράται ως αστάθεια - ο βαθμός στον οποίο οι απαιτήσεις αλλάζουν κατά τη διάρκεια της χρονικής περιόδου. Η ιχνηλασιμότητα μπορεί να είναι από τις απαιτήσεις στις απαιτήσεις και από τις απαιτήσεις στο σχέδιο ή στο έγγραφο ελέγχου ( test document ).

Οι αντικειμενοστρεφείς μετρικές είναι χρήσιμες για την κατανομή των πόρων στην ανάπτυξη λογισμικού. Αυτές οι μετρικές χρησιμοποιούνται ιδιαίτερα για να προσδιορίσουν οι κλάσεις που είναι επιρρεπείς στα ελαττώματα και προβλέπουν τις προσπάθειες συντήρησης, το πόσο επιρρεπής είναι κάτι στο λάθος, και το ποσοστό του λάθους.



### 3.3. Τύποι Μετρικών

Για να μετρηθεί το λογισμικό χρησιμοποιούνται διαφορετικοί τύποι μετρικών.

Οι μετρικές κατηγοριοποιούνται σε products, process και resource metrics.

#### 3.3.1. Μετρικές προϊόντων (Product Metrics)

Οι μετρικές προϊόντων μετράνε τα προϊόντα του λογισμικού σε οποιοδήποτε στάδιο της ανάπτυξης του. Μπορούν να εφαρμοστούν από τη φάση απαιτήσεων μέσω της εγκατάστασης. Αυτές οι μετρικές μετράνε το μέγεθος του προγράμματος. Μετρούν τον αριθμό σελίδων των εγγράφων και της πολυπλοκότητας του σχεδίου λογισμικού. Οι πρωτόγονες-primitive μετρικές ή η άμεση μέτρηση μιας ιδιότητας δεν περιλαμβάνει καμία άλλη οντότητα. Είναι ανεξάρτητες από άλλες οντότητες. Μπορεί να είναι LOC, διάρκεια των δοκιμών που υπολογίζονται σε αριθμό των ωρών, οι μήνες και ο αριθμός ατελειών που ανακαλύπτονται( πίνακας 4.2 ;;). Το μέγεθος του λογισμικού μπορεί να μετρηθεί από το μήκος, τη λειτουργία και την πολυπλοκότητά του. Τα LOC είναι κοινά κριτήρια που χρησιμοποιούνται για να μετρήσουν το μήκος του λογισμικού. Είναι το φυσικό μέγεθος των προγραμμάτων που χρησιμοποιούνται για να γράψουν τον κώδικα. Υπάρχουν διαφορετικές σημειώσεις για το γράψιμο του κώδικα. Ο κώδικας μπορεί να έχει μόνο τις εκτελέσιμες δηλώσεις, τις κενές γραμμές, τα σχόλια και τα headers. Τα βιομηχανικά πρότυπα για τη μέτρηση των LOC είναι να μετρηθεί ο μη-σχολιασμένος κώδικας. Τα προγράμματα έχουν προτάσεις- δηλώσεις, προτάσεις-header και εκτελέσιμες προτάσεις στις οποίες μόνο οι πρώην εκτελέσιμη προτάσεις μετριοούνται ως LOC και οι σχολιασμένες γραμμές αγνοούνται. Το Software Engineering Institute προτείνει να μετρηθεί ο κώδικας κρατώντας τον αυτοποιημένο από τις γεννήτριες κώδικα. Το Software Engineering Institute προτείνει επίσης να εξεταστούν οι διαφοροποιημένες φυσικές γραμμές κώδικα από τις λογικές γραμμές κώδικα και ο επαναχρησιμοποιημένος κώδικας από τον αντιγραμμένος και τον τροποποιημένος κώδικα[22].

#### **Object oriented metrics :**

Οι αντικειμενοστρεφείς μετρικές είναι χρήσιμες για την κατανομή των πόρων στην ανάπτυξη λογισμικού. Αυτές οι μετρικές χρησιμοποιούνται ιδιαίτερα για να προσδιορίσουν οι κλάσεις που είναι επιρρεπείς στα ελαττώματα και προβλέπουν τις προσπάθειες συντήρησης, το πόσο επιρρεπής είναι κάτι στο λάθος, και το ποσοστό του λάθους. Οι αντικειμενοστρεφείς μετρικές περιορίζονται κυρίως στις αλλαγές κατά τη διάρκεια των φάσεων σχεδίου και της εφαρμογής ανάπτυξης λογισμικού.

#### **Communication metrics :**

Η επικοινωνία μέσα σε μια ομάδα ορίζεται ως η εσωτερική επικοινωνία μέσα στην ίδια την ομάδα (intra-team communication) ενώ η επικοινωνία μεταξύ των μελών των διαφορετικών ομάδων είναι γνωστή ως inter-team communication. Παραδείγματα επικοινωνίας χειροποίητων αντικειμένων επικοινωνίας παραδείγματος χάριν, είναι το ηλεκτρονικό ταχυδρομείο, οι εβδομαδιαίες συνεδριάσεις, οι ανεπίσημες αλληλεπιδράσεις και είναι διαθέσιμες σε ολόκληρο το πρόγραμμα συλλαμβάνοντας πληροφορίες όπως ο κώδικας, οι εκάστοτε πολιτικές και η διαδικασία. Αυτή η μετρική επικοινωνίας δίνει μια καλή εικόνα της διαδικασίας ανάπτυξης λογισμικού. Αυτοί οι μετρικές μετρούν τα γεγονότα ενός συγκεκριμένου τύπου μέσα σε μια διαδικασία. Είναι διαθέσιμες στις αρχικές φάσεις ανάπτυξης λογισμικού και είναι ευκολότερο να τις συλλέξουμε. Δεδομένου ότι η επικοινωνία γίνεται τακτικά κατά τη διάρκεια της ανάπτυξης λογισμικού μπορεί να οδηγήσει σε φοβερές ατέλειες αργότερα.

#### **Design Metrics :**

Οι μετρικές σχεδίου σε αντίθεση με τις παραδοσιακές μετρικές δεν βασίζονται σε δομές σύνταξης ή δεδομένων. Αυτές οι μετρικές υπολογίζονται από τις απαιτήσεις ή τα έγγραφα σχεδίου προτού να τεθεί σε εφαρμογή το σύστημα. Αυτές οι μετρικές υπολογίζονται με βάση τη λειτουργία του κώδικα στην περιοχή προβλήματος. Αυτό υπολογίζει την πολυπλοκότητα του module αντί του αριθμού μονοπατιών μέσω ενός κομματιού του κώδικα [3]. Δεδομένου ότι αυτές οι μετρικές είναι ανεξάρτητες από τη δομή του κώδικα μπορούν να υπολογιστούν από τις προδιαγραφές του σχεδίου που επιτρέπουν μια πρώιμη αίσθηση της ποιότητας του λογισμικού. Οι μετρικές σχεδίου είναι χρήσιμες για την ποιότητα του λογισμικού.

### **3.3.2. Διαδικαστικές Μετρικές (Process Metrics)**

Οι μετρικές διαδικασίας μετράνε την διαδικασία της ανάπτυξης του λογισμικού. Περιλαμβάνουν τον τύπο της μεθοδολογίας που χρησιμοποιήθηκε, το επίπεδο εμπειρίας των ανθρωπίνων πόρων και το συνολικό χρόνο ανάπτυξης. Ο συνολικός χρόνος ανάπτυξης ενός προϊόντος λογισμικού είναι ένα αντικειμενικό κριτήριο και κατά συνέπεια μια αντικειμενική μετρική. Εάν δοθεί ως μια εργασία, αυτήν η μετρική πρέπει να έχει τα ίδια αποτελέσματα από όλους τους παρατηρητές. Οι υπολογισμένες μετρικές ή η έμμεσες μετρήσεις γίνονται για να γίνει ορατή η άμεση αλληλεπίδραση της μέτρησης. Αυτές οι μετρικές είναι ένας συνδυασμός μετρικών ( βλέπε τον πίνακα 5.1 )

Category	Formula
Programmer productivity	LOS Produced/Persons months of effort
Module defect density	no. of defects/module size
Defect detection efficiency	no. of defects detected/total no. of defects
Requirement stability	no. of initial requirements/total no. of requirements
Test effectiveness ration	no.of items covered/total no. of items
System spoilage	effort spent fixing faults/total project effort

*(Πίνακας 5.1: Παραδείγματα κοινών έμμεσων μετρήσεων που χρησιμοποιούνται στην Μηχανική Λογισμικού)*

### 3.3.3. Μετρικές Απαιτήσεων( Requirements Metrics )

Η εφαρμοσμένη μηχανική απαιτήσεων εξετάζει την απόκτηση, την ανάλυση, την επικοινωνία και την επικύρωση των απαιτήσεων. Όσο πιο νωρίς προσδιορίζονται τα λάθη, τόσο πιο εύκολο είναι να διορθωθούν σε σύγκριση με το να προσδιοριστούν πιο μετά. Είναι προφανές έτσι ότι το κόστος αναγνώρισης και διόρθωσης αυτών των λαθών στα αρχικά στάδια είναι χαμηλότερο από το κόστος αναγνώρισής τους τα στα προχωρημένα στάδια της ανάπτυξης λογισμικού. Πολλά από αυτά τα λάθη προκαλούνται λόγω των αλλαγών στις απαιτήσεις. Προκειμένου να εξαλειφτούν τα λάθη πρέπει να υπάρξει κάποια μέτρηση. Οι μετρικές που μπορούν να μαζευτούν από τις απαιτήσεις είναι το μέγεθος των απαιτήσεων, η ιχνηλασιμότητα των απαιτήσεων, η πληρότητα απαιτήσεων και η αστάθεια απαιτήσεων.

#### Use Case/Size metrics

Το μέγεθος είναι μια σημαντική και γενική μετρική που χρησιμοποιείται για να μετρήσει τις απαιτήσεις. Το Los είναι μια κοινή μετρική που χρησιμοποιείται για να μετρήσει το μέγεθος του λογισμικού. Οι γραμμές σχολίων και οι κενές γραμμές δεν λογίζονται ως κώδικας. Μια μη σχολιασμένη γραμμή θεωρείται ως κώδικας αλλά περιλαμβάνει επίσης τις εκτελέσιμες εντολές, τις εκτελέσιμες δηλώσεις και τις δηλώσεις δεδομένων. Οι περιπτώσεις χρήσης μπορούν επίσης να θεωρηθούν ως μετρήσεις ενός μεγέθους όταν χρησιμοποιούνται για να

περιγράψουν τις απαιτήσεις. Παραδείγματος χάριν ο αριθμός των περιπτώσεων χρήσης, ο αριθμός των λειτουργιών που καλύπτονται κ.λπ. Το μοντέλο Περίπτωσης Χρήσης χρησιμοποιείται για να εξασφαλίσει ότι χτίζεται το σωστό σύστημα. Οι περιπτώσεις χρήσης βρίσκονται σε υψηλό επίπεδο της αφαίρεσης. Εντούτοις, μπορούν να συλλάβουν μόνο τις λειτουργικές απαιτήσεις ενός συστήματος. Οι μη λειτουργικές απαιτήσεις δεν συλλαμβάνονται στις περιπτώσεις χρήσης. Ο Bullishness [1] λέει ότι οι μετρικές γιατί τις περιπτώσεις χρήσης μπορούν να παραχθούν με τον υπολογισμό του αριθμού των actors σύμφωνα με τα βάρη.

Ένα συσσωρευτικό ποσό, τα Unadjusted Use Case Points - UNC ( σημεία περίπτωσης χρήσης χωρίς διόρθωση ), παράγεται βασιζόμενο στα σημεία της περίπτωσης χρήσης( use case points – UP ). Τα UP είναι βάρη των σημείων λειτουργίας όπου η προσπάθεια υπολογίζεται από τον αριθμό ωρών ανά άτομο πολλαπλασιαζόμενα επί τον αριθμό UP. Μια ατομική δράση( atomic action ) είναι μια δράση η οποία δεν μπορεί να αποσυντεθεί περαιτέρω μέσα σε βαθύτερο επίπεδο. Μια ατομική δράση χρησιμοποιείται ως μονάδα της μέτρησης για τις περιπτώσεις χρήσης.

Ο Bellahsene [1] ταξινόμησε τις μετρικές περίπτωσης χρήσης σε 3 κατηγορίες.

Η μετρική μεγέθους περίπτωσης χρήσης περιλαμβάνει τον αριθμό ατομικών ενεργειών στην κύρια ροή, τον αριθμό ατομικών ενεργειών στην εναλλακτική ροή, την μακρύτερη πορεία μεταξύ των πρώτων και τελευταίων ατομικών ενεργειών της περίπτωσης χρήσης και του αριθμού εναλλακτικών ροών. Εδώ, οι εναλλακτικές ροές μετριοούνται από την έναρξη και το τέλος της περίπτωσης χρήσης. Το μέγεθος της περίπτωσης χρήσης μπορεί να καθοριστεί με τον υπολογισμό του αριθμού των actors που σταθμίζεται από την πολυπλοκότητά τους, τον αριθμό των περιπτώσεων χρήσης που σταθμίζονται από τον αριθμό συναλλαγών που περιέχουν και τον αριθμό των κλάσεων ανάλυσης που χρησιμοποιούνται για να εφαρμόσουν κάθε περίπτωση χρήσης. Οι μετρικές περίπτωσης χρήσης περιβάλλοντος είναι οι παράγοντες που έχουν επιρροή στο επίπεδο πολυπλοκότητας των περιπτώσεων χρήσης. Αυτοί είναι ανεξάρτητοι από το μέγεθος των περιπτώσεων χρήσης. Περιλαμβάνουν τον αριθμό των actors και τον συνολικό αριθμό των στόχων. Οι σύνθετες μετρικές περίπτωσης χρήσης προέρχονται από τις μετρικές μεγέθους και τις μετρικές περιβάλλοντος. Περιλαμβάνουν το συνολικό αριθμό των ατομικών ενεργειών στις εναλλακτικές ροές, τον συνολικό αριθμό των ατομικών ενεργειών σε όλες τις ροές και τον αριθμός των ατομικών ενεργειών ανά actor και ανά στόχο.

## Requirements Traceability Metrics

Η ιχνηλασιμότητα είναι η δυνατότητα να επισημανθούν οι απαιτήσεις σε μια προδιαγραφή στην προέλευσή τους από υψηλότερου επιπέδου σε χαμηλότερες απαιτήσεις επιπέδων σε ένα σύνολο συνδέσεων. Η ιχνηλασιμότητα παρέχει πληροφορίες που βοηθούν στον καθορισμό εάν όλες οι σχέσεις και οι εξαρτήσεις εξετάζονται. Αυτό κάνει επίσης τη διαδρομή των απαιτήσεων - ύπαρξη του χαμηλότερου επιπέδου απαιτήσεων χωρίς έγκυρη προέλευση από τα υψηλότερα επίπεδα - ορατή. Βοηθά στην παρεμπόδιση λανθασμένων ερμηνειών άλλων μετρικών.

Υπάρχουν 5 τύποι μετρικών ιχνηλασιμότητας.

1) Next level coverage metrics – COB ( Μετρικές κάλυψης επόμενου επιπέδου ). Αυτές οι μετρικές προσπαθούν να εντοπίσουν τα ίχνη ( traces ) ψάχνοντας στο επόμενο επίπεδο προς τα επάνω και στο επόμενο επίπεδο προς τα κάτω. Αυτό βρίσκει επίσης τον αριθμό απαιτήσεων και στις δύο κατευθύνσεις.

2) Full depth and height coverage – DECO ( Πλήρους βάθους και κάλυψη ύψους ). Αυτό είναι παρόμοιο με το COB αλλά ψάχνει τα ίχνη ( traces ) των απαιτήσεων μόνο στο υψηλότερο και στο χαμηλότερο επίπεδο αντί μόνο του άμεσου επόμενου επιπέδου και στις δύο κατευθύνσεις. Αυτό βολεύει για 3 ή λιγότερες προδιαγραφές επιπέδων.

3) Inconsistent traceability metrics ( Μετρικές ιχνηλασιμότητας δίχως συνέπεια ). Αυτές περιλαμβάνουν τον αριθμό των απαιτήσεων σε μια προδιαγραφή που έχει τουλάχιστον έναν μη συνεπή σύνδεσμο στις συνδέσεις και προς τα πάνω , αλλά και προς τα κάτω.

4) Undefined traceability metrics ( Μη-ορισμένες μετρικές ιχνηλασιμότητας ) Περιλαμβάνει τους αριθμούς των απαιτήσεων σε μια προδιαγραφή που δεν έχουν συνδέσμους ιχνηλασιμότητας ούτε προς τα πάνω ούτε προς τα κάτω. Όμως ο υψηλότερος σύνδεσμος δεν θα έχει σύνδεσμο ιχνηλασιμότητας προς τα πάνω αφού παράγεται από το σχεδιαστικό έγγραφο και ο χαμηλότερος σύνδεσμος δεν θα έχει αντίστοιχα σύνδεσμο προς τα κάτω.

4) Linkage statistic metric ( Μετρικές στατιστικών συνδέσεων )

Μετράνε το πόσο σύνθετο είναι το επίπεδο των δεδομένων ιχνηλασιμότητας μετρώντας τον αριθμό των χαμηλότερων ή υψηλότερων απαιτήσεων με τις οποίες κάθε απαίτηση σε μια προδιαγραφή εντοπίζεται.

## Requirements Completeness Metrics

Οι μετρικές πληρότητας απαιτήσεων χρησιμοποιούνται για να αξιολογήσουν εάν μια απαίτηση είναι σε λανθασμένο επίπεδο ιεραρχίας ή αν μια απαίτηση είναι υπερβολικά σύνθετη. Οι απαιτήσεις θεωρούνται ως πλήρεις εάν όλες οι σελίδες είναι αριθμημένες, όλοι οι αριθμοί και

οι πίνακες έχουν τους τίτλους τους και τους αριθμούς τους, όλες οι αναφορές είναι παρούσες, όλες τα τμήματα και οι υποενότητες είναι αριθμημένα και τίποτα να μην είναι μη πλήρες. Αυτές οι μετρικές ποσοτικοποιούν το επίπεδο της αποσύνθεσης της απαίτησης του υψηλότερου επιπέδου που βρίσκεται σε μια προδιαγραφή. Αυτές είναι μαρκαρισμένες με τη σήμανση 'to be completed' και τον αριθμό της χαμηλότερης προδιαγραφής επιπέδων.

Οι μετρικές πληρότητας απαιτήσεων είναι 3 τύπων.

1) Requirements Decomposition Metrics ( Μετρικές αποσύνθεσης απαιτήσεων ) Αυτές οι μετρικές χρησιμοποιούνται για να ξέρουμε εάν οι συγκεκριμένες απαιτήσεις έχουν αποσυντεθεί αρκετά για να χρησιμοποιηθούν στην επόμενη φάση. Μια σύνθετη λειτουργία θα έχει πολλά επίπεδα προδιαγραφής και μια απλή λειτουργία θα έχει λίγα επίπεδα.

2) Requirements Specification Development ( Ανάπτυξη προδιαγραφών Απαιτήσεων ). Αυτές οι μετρικές χρησιμοποιούνται για να παρέχουν τις πληροφορίες για την εργασία που ολοκληρώνεται και την εργασία που δεν έχει τελειώσει μέσα σε μια δεδομένη προδιαγραφή. Αυτές οι μετρικές παρέχουν μια λύση στην ερώτηση γιατί τα queries – ερωτήματα έχουν όλες τις απαιτήσεις στα πιο υψηλά επίπεδα ενώ έχουν τις προδιαγραφές σε χαμηλά επίπεδα.

3) Requirements Specification Metrics ( Μετρικές προδιαγραφών απαιτήσεων ) Αυτές οι μετρικές παρέχουν τις πληροφορίες σχετικά με την ποσότητα στοιχείων για τις οποίες απαιτείται διόρθωση ενός προβλήματος που έχουν. Παρέχουν επίσης τις πληροφορίες για τα εκκρεμή ζητήματα τα οποία θα επηρεάσουν τις απαιτήσεις υψηλότερων επιπέδων. Αυτές οι μετρικές χρησιμοποιούν ετικέτες όπως 'to be determined', 'to be με σκοπό την ολοκλήρωση του εγγράφου απαιτήσεων. Αυτό επισημαίνει επίσης τον αριθμό των απαιτήσεων υψηλού επιπέδου που διατίθενται στην προδιαγραφή για τις ελλιπείς απαιτήσεις.

### **Requirements Volatility Metrics**

Μετρικές αστάθειας απαιτήσεων. Ο βαθμός κατά τον οποίο οι απαιτήσεις αλλάζουν κατά τη διάρκεια ενός χρονικού διαστήματος καλείται αστάθεια των απαιτήσεων. Αυτή η μετρική χρησιμοποιείται για να αξιολογήσει τους λόγους για τους οποίους έγινε η αλλαγή των απαιτήσεων σε μια χρονική περίοδο. Από αυτήν την μετρική ελέγχεται επίσης ο βαθμός της αλλαγής των απαιτήσεων. Και οι δύο αυτοί παράγοντες ελέγχονται για να μας δείξουν εάν οι αλλαγές είναι σύμφωνες με τις τρέχουσες δραστηριότητες ανάπτυξης. Δείχνει τις αλλαγές όπως η προσθήκη, η διαγραφή και η τροποποίηση. Βοηθά στην επισήμανση των μελλοντικών απαιτήσεων, της αστάθειας σχεδίου και κώδικα. Η αστάθεια μπορεί να είναι υψηλή στην αρχική φάση ανάπτυξης λογισμικού. Πρέπει να μειωθεί όσο το λογισμικό αναπτύσσεται έτσι ώστε η περαιτέρω ανάπτυξη να μην επηρεαστεί. Αυτήν η μετρική είναι μια καλή ένδειξη

άλλων μετρικών παραδείγματος χάριν, εάν ένα συγκεκριμένο κομμάτι ενός λογισμικού εξετάζεται χωρίς τη σωστή( ή καθόλου ) της γνώσης της αστάθειας των απαιτήσεων τότε αυτή η δοκιμή θα ήταν τελείως λανθασμένη.

### **3.4. Γιατί είναι απαραίτητες οι Μετρικές στην σύγχρονη ανάπτυξη λογισμικού**

Πολλές εταιρείες εξακολουθούν να ρωτάνε πώς γίνεται να αξιολογηθεί και να προβλεφθεί το κόστος συντήρησης του λογισμικού τους. Μέτρα για το κόστος συντήρησης του λογισμικού μπορούν να ληφθούν είτε στο τέλος είτε στις αρχές στην διαδικασία ανάπτυξης. Τα πρόωρα μέτρα του κόστους συντήρησης του λογισμικού είναι ευεργετικά, επειδή μπορούν να βοηθήσουν στην την κατανομή των πόρων του έργου αποτελεσματικά, προβλέποντας την προσπάθεια των εργασιών συντήρησης και ελέγχου της διαδικασίας συντήρησης.

Έχει μετρηθεί ότι στη φάση συντήρησης του λογισμικού οι επαγγελματίες έχουν δαπανήσει τουλάχιστον το ήμισυ του χρόνου τους στην ανάλυση λογισμικού για να καταλάβουν.

Επιπλέον, το κόστος της συντήρησης λογισμικού δαπανά ένα μεγάλο ποσοστό του συνολικού κόστους ενός συστήματος λογισμικού.

Είναι πολύ σημαντικό για τις εταιρείες να αξιολογούν και να προβλέπουν το κόστος συντήρησης του λογισμικού τους. Μάλιστα τα τελευταία χρόνια έχει σχεδιαστεί και ένα εργαλείο που ονομάζεται WarMetrics για μετρήσεις σχεδιαστικών μετρικών UML WarMetrics. Το WarMetrics παρέχει έναν αυτοματοποιημένο τρόπο μέτρησης UML σχεδιαστικών μετρικών, και παρέχει τη δυνατότητα να δείτε τα αποτελέσματα σε διάφορα outputs.

Πολλές μελέτες έχουν δείξει ότι η έγκαιρες μετρικές( μετρικές που χρησιμοποιούνται νωρίς στον σχεδιασμό του συστήματος ) είναι πολύ πιο χρήσιμες.

Οι μετρικές μπορεί να είναι χρήσιμες με τους παρακάτω τρόπους.

1. Προβλέποντας την προσπάθεια συντήρησης και το κόστος των εργασιών συντήρησης το οποίο βοηθά στο να παρέχονται ακριβείς εκτιμήσεις που μπορούν να βοηθήσουν στην κατανομή σωστών πόρων για το έργο για τις εργασίες συντήρησης.
2. Συγκρίνοντας τα σχεδιαστικά έγγραφα, που μπορούν να βοηθήσουν κατά την επιλογή μεταξύ των διαφόρων βασιζόμενες στην σταθερότητα του σχεδίου.
3. Με τον εντοπισμό των επικίνδυνων συστατικών στοιχεία ενός συστήματος, δεδομένου ότι ορισμένες μελέτες δείχνουν ότι τα περισσότερα σφάλματα εμφανίζονται μόνο σε μερικά μόνο συστατικά του λογισμικού του συστήματος.
4. Με τη θέσπιση σχεδιαστικών και προγραμματιστικών κατευθυντήριων γραμμών για τα συστατικά του λογισμικού. Αυτό μπορεί να γίνει με καθορισμό αξιών οι οποίες είναι αποδεκτές ή μη αποδεκτές και με ανάληψη δράσης με τις μη αποδεκτές τιμές στα

συστατικά λογισμικού. Αυτό σημαίνει ότι πρέπει το κατώφλι των μετρικών προϊόντος λογισμικού να παρέχει έγκαιρες προειδοποιήσεις του συστήματος.

5. Κάνοντας πρόβλεψη του επιπέδου(level) του συστήματος όπου η συντηρησιμότητα όλων των συστατικών μπορεί να προβλεφθεί συναθροίζοντας την σταθερότητα όλων των μεμονωμένων στοιχείων. Αυτό μπορεί να χρησιμοποιηθεί για να προβλεφθεί η προσπάθεια που θα χρειαστεί για την ανάπτυξη του συνόλου του συστήματος λογισμικού.

Επαναχρησιμοποίηση κώδικα με κληρονομικότητα και σύνθεση.

Ο Sneed (1996) χαρακτήρισε τη σχέση μεταξύ object-oriented software Components σύμφωνα με αυτό το σχήμα.

Objects	Build	Classes
Objects	Have (are composed of)	Attributes
Objects	Inherit	Attributes
Objects	Have (are composed of)	Methods
Objects	Inherit	Methods
Objects	Send	Messages
Objects	Receive	Messages
Messages	Are	Data
Messages	Are	Relations

**Πίνακας 5.2: Συστατικά του Αντικειμενοστραφούς Λογισμικού σύμφωνα με τον Sneed(1996)**

### 3.5. Περαιτέρω Ανάλυση των Μετρικών

Σε ένα Αντικειμενοστραφές Μοντέλο, έχουμε έννοιες όπως :

object(αντικείμενο),

class(τάξεις),

attributes (γνωρίσματα),

inheritance (κληρονομικότητα),

method (μεθόδους),

message passing («πέραςμα» του μηνύματος).

Οι Chidamber and Kemerer: πρότειναν μια «σουίτα» από 6 Αντικειμενοστραφής σχεδιαστικές μετρικές, τις εξής:



- RFC
- DIT
- NOC
- CBO
- LCOM
- WMC.

Όσον αφορά τις κατηγορίες των μετρικών καθαυτών, χωρίζονται σε αυτές τις τρεις υποομάδες:

- Μετρικές συστήματος, για παράδειγμα τον αριθμό των αρχείων, τις κλάσεις, και τα δέντρα ( της κληρονομικότητας ).
- Μετρικές των δέντρων, για παράδειγμα ο αριθμός των παιδιών (NOC) ή κλάσεων.
- Μετρήσεις κλάσεων, για παράδειγμα τον αριθμό των μεθόδων.

Ο Peter Rosner αναφέρει ότι οι object-oriented μετρικές χρησιμοποιούνται κυρίως για να καταλάβουμε σε ποιο βαθμό οι έννοιες του object orientation πραγματοποιούνται σε ένα σύστημα όπως αξιολογήθηκε στην κλάση, τη μέθοδο, και τα επίπεδα του συστήματος. Ο Henderson-Sellers (1996) καλεί τις ιδιότητες που συμβάλλουν στη συνολική πολυπλοκότητα ως χαρακτηριστικό του προγραμματιστή.

Τα αποτελέσματα της ανάλυσης δείχναν ότι η «σουίτα» των μετρικών μπορεί να μειωθεί σε 5 σχετικές και ποσοτικές μετρήσεις αξιολόγησης του μεγέθους και της πολυπλοκότητας του Αντικειμενοστραφούς λογισμικού :

- Ο αριθμός των «weighted methods per class» (WMC) σαν ένας δείκτης για το μέγεθος του συστήματος. Το βάρος σε αυτήν την περίπτωση προκαλείται από την πολυπλοκότητα της αντίστοιχης μεθόδου.
- Το βάθος του δέντρου κληρονομιάς μετρά την πολυπλοκότητα του συστήματος.
- Το NOC είναι ένα μέτρο της επαναχρησιμοποίησης μιας κλάσης.
- Ο βαθμός σύζευξης μεταξύ των κλάσεων δείχνει το βαθμό στον οποίο οι κλάσεις είναι ανεξάρτητες. Αυτό αποτελεί σημαντικό δείκτη για την κατανόηση των κλάσεων και τη κατανομή τους σε υποκατηγορίες. Ισχυρή σύζευξη δείχνει αμέλεια του σχεδιασμού στο σύστημά μας.
- Η συμπεριφορά μιας τάξης(response behavior).

Ένα ενδιαφέρον αποτέλεσμα ήταν ότι κατά την ανάλυση και των δύο κατηγοριών (μετρικών δέντρων κληρονομιάς αλλά και μετρικών κλάσεων) δεν μπόρεσε να βρεθεί μια σχέση μεταξύ των πηγαίων γραμμών κώδικα και το βάθος των τάξεων. (Ωστόσο, υπάρχουν ορισμένες μετρικές που βασίζονται στις σχέσεις μεταξύ τους).

Ένας μεγάλος αριθμός αντικειμενοστρεφών μετρικών έχει προταθεί στη βιβλιογραφία. Χρησιμοποιούνται για να αξιολογήσουν τις διαφορετικές ιδιότητες λογισμικού. Εντούτοις, δεν είναι προφανές για έναν developer ή έναν διευθυντή προγράμματος να επιλέξουν τις μετρικές που είναι πιο χρήσιμες. Επιπλέον, αυτές οι μετρικές δεν είναι απολύτως ανεξάρτητες. Χρησιμοποιώντας διάφορα metrics στον ίδιο χρόνο είναι χρονοβόρο και μπορεί να παραγάγει ένα αρκετά μεγάλο σύνολο στοιχείων, το οποίο μπορεί να είναι δύσκολο να αναλυθεί και να ερμηνευτεί.

Οι μετρικές λογισμικού έχουν γίνει ένα στοιχείο-κλειδί σε διάφορες περιοχές της Μηχανικής Λογισμικού. [Pressman 05, Sommerville 04]. Χρησιμοποιούνται για να έχουν πρόσβαση σε διάφορες ιδιότητες που αφορούν το λογισμικό ή τη διαδικασία. Όταν εφαρμόζεται στο λογισμικό, επιτρέπουν τη συλλογή ποσοτικών στοιχείων σε διάφορες πτυχές σχετικές με τις ιδιότητές του. Αυτά τα στοιχεία αναλύονται, τελικά συγκρίνονται με παλαιότερα στοιχεία όταν αυτά διατίθενται, και χρησιμοποιούνται για να βελτιωθεί η ποιότητά του. Η αληθινή αξία των μετρικών λογισμικού προέρχεται από το τους ένωσή τους με τις σημαντικές εξωτερικές ιδιότητες(attributes) λογισμικού όπως η δυνατότητα δοκιμής, αξιοπιστία και συντηρησιμότητα [EL Emam 01]. Οι μετρικές χρησιμοποιούνται έπειτα για να προβλέψουν την ποιότητα λογισμικού [Fenton 96]. Ένας μεγάλος αριθμός αντικειμενοστρεφών μετρικών (OOM) έχει προταθεί στη βιβλιογραφία[henderson-sellers 96]. Χρησιμοποιούνται για να αξιολογήσουν τις διαφορετικές ιδιότητες λογισμικού όπως το μέγεθος, την πολυπλοκότητα, τη σύζευξη και τη συνοχή. Εντούτοις, υπάρχει μια μικρή κατανόηση των εμπειρικών υποθέσεων και της εφαρμογής για πολλές από αυτές τις μετρικές [Aggarwal 06]. Επιπλέον, δεν είναι προφανές για έναν υπεύθυνο για την ανάπτυξη να επιλέξει τις μετρικές που είναι πιο χρήσιμες για να τις χρησιμοποιήσουν οι υπάλληλοι. Επίσης, αυτοί οι μετρικές δεν είναι απολύτως ανεξάρτητες [Aggarwal 06]. Πολλές από αυτές τις μετρικές επικαλύπτουν πληροφορίες. Χρησιμοποιώντας διάφορες μετρικές συγχρόνως, ιδιαίτερα στην περίπτωση των σύνθετων και μεγάλης κλίμακας συστημάτων λογισμικού, είναι χρονοβόρο και μπορεί να παράγει ένα αρκετά μεγάλο σύνολο στοιχείων, το οποίο μπορεί να είναι δύσκολο να αναλυθεί και να ερμηνευτεί. Είναι κατόπιν δύσκολο να βγάλει κανείς συμπεράσματα από τις παρεχόμενες πληροφορίες. Πρέπει έπειτα να περιορίσουμε τον αριθμό των μετρικών που χρησιμοποιούνται σε ένα υποσύνολο τους που παρέχει χρήσιμη πληροφορία.

### **3.6. Βασικές Υπάρχουσες Μετρικές**

#### **1.Depth of Inheritance Tree (DIT)**

Ο αριθμός των κλάσεων γονέων. Δείχνει πόσες άλλες κλάσεις έχουν πιθανώς αντίκτυπο σε αυτήν την κλάση.

**2. Number Of Children (NOC)****3. Between Objects (CBO)****4. Response For a Class (RFC)****5. Lack of Cohesion in Methods (LCOM)**

Ο αριθμός των άσχετων «κομματιών» σε μια κλάση. Για παράδειγμα, μια κλάση μπορεί να έχει τρεις αναφορές-μεταβλητές (Ia, Ib, Ic) και πέντε μεθόδους (M1, M2, ..., M5) όπου μόνο οι M1, M2 χρησιμοποιούν την Ia, οι M3, M4 χρησιμοποιούν την Ib και μόνο η M5 χρησιμοποιεί την Ic. Αυτή η κλάση θα πάρει μια LCOM τιμή των δύο μονάδων (2).

**6. Weighted Methods Per Class (WMC)**

Είναι το άθροισμα της αξίας της πολυπλοκότητας για όλες τις μεθόδους σε μια κλάση.

**7. Message Passing Coupling (MPC)**

Ο αριθμός των αποσταλμένων δηλώσεων ως ορίζονται σε μια κλάση.

**8 Data Abstraction Coupling (DAC)**

Ο αριθμός των αφηρημένων δεδομένων που χρησιμοποιούνται σαν αναφορές-μεταβλητές από μια κλάση.

**9. Number Of local Methods (NOM)**

Ο αριθμός των τοπικών μεθόδων σε μια κλάση ( Οι κληρονομούμενες μέθοδοι δεν λογίζονται ως τοπικές )

**10. Lines of code (SIZE1)**

Γραμμές κώδικα.

**11. Number of Properties (SIZE2)**

Αριθμός των properties, για παράδειγμα attributes + ο αριθμός των τοπικών μεθόδων

**12. Method Invocation Coupling (MIC)**

Η MIC δείχνει το σχετικό αριθμό κλάσεων στον οποίο μια δεδομένη κλάση στέλνει μηνύματα.

### 3.7.Μετρικές που μπορούν να χρησιμοποιηθούν στην οριζόντια / κάθετη ιχνηλασιμότητα.

- **Coupling Between Objects (CBO)**

Ο αριθμός των μη κληρονομούμενων σχετιζόμενων ζευγών με άλλες κλάσεις, για παράδειγμα, μια κλήση μιας μεθόδου. Η CBO μετρά για μια κλάση τον αριθμό άλλων κλάσεων με τις οποίες συνδέεται [Chidamber 94]. Δύο κλάσεις συνδέονται όταν οι μέθοδοι που είναι δηλωμένες σε αυτήν την κλάση χρησιμοποιούν μεθόδους ή αναφορές μεταβλητών που είναι δηλωμένες στην άλλη κλάση. Στην ουσία η Coupling Between Objects (CBO) είναι μια *αρίθμηση* της ποσότητας των άλλων κλάσεων με τις οποίες μια κλάση συνδέεται. Μετριέται με τον υπολογισμό του αριθμού των ευδιάκριτων μη-κληρονομούμενων σχετικών ιεραρχιών από τις οποίες μια κλάση εξαρτάται. Η υπερβολική σύζευξη είναι καταστρεπτική στο σχεδιασμό και αποτρέπει την επαναχρησιμοποίηση(reuse). Όσο πιο ανεξάρτητη είναι μια κλάση, τόσο ευκολότερο είναι να επαναχρησιμοποιηθεί σε μια άλλη εφαρμογή. Όσο μεγαλύτερος είναι ο αριθμός των ζευγών, τόσο υψηλότερη είναι η ευαισθησία στις αλλαγές σε άλλα μέρη από το σχέδιο και επομένως η συντήρηση είναι δυσκολότερη. Η ισχυρή σύζευξη περιπλέκει ένα σύστημα δεδομένου ότι μια κλάση είναι δυσκολότερο να κατανοηθεί, να αλλάξει ή να διορθωθεί εάν συσχετίζεται με άλλες κατηγορίες. Η πολυπλοκότητα μπορεί να μειωθεί στο σχεδιασμό των συστημάτων με την όσο γίνεται μικρότερη σύζευξη μεταξύ των κατηγοριών. Αυτό βελτιώνει τη διαμόρφωση(modularity) και βοηθά την ενθυλάκωση. Τιμές CBO μπορούν να επιτευχθούν, με τον υπολογισμό των αριθμών των τύπων αναφοράς που χρησιμοποιούνται, όταν είναι διαθέσιμα μόνο τα διαγράμματα κλάσης. Καθώς το σχέδιο του συστήματος λογισμικού προχωράει, περισσότερα διαγράμματα που απεικονίζουν τις αλληλεπιδράσεις μεταξύ των κλάσεων, συμπεριφορές των κλάσεων και τις λεπτομερείς δομές των κλάσεων, όπως τα διαγράμματα συνεργασίας και τα διαγράμματα δραστηριότητας αναπτύσσονται. Τώρα, οι ακριβέστερες τιμές CBO, που λαμβάνουν υπόψη όχι μόνο τη σύζευξη που προκαλείται με τους στατικούς τύπους αναφοράς αλλά και τις αλληλεπιδράσεις και το πέρασμα-διαβίβαση των μηνυμάτων, μπορούν να μετρηθούν.

Σύζευξη μεταξύ αντικειμενοστραφών κλάσεων

*CBO = είναι ο αριθμός των κλάσεων όπου οι κλάσεις είναι συνδεδεμένες.*

Πολλαπλές προσβάσεις στην ίδια κλάση υπολογίζονται ως μία πρόσβαση. Μόνο καλούμενες μέθοδοι και μεταβλητές με αναφορά υπολογίζονται. Άλλοι τύποι αναφορών, όπως σταθερές, κλήσεις σε API's, χειρισμοί γεγονότων, χρήση τύπων καθορισμένων από τον

χρήστη και αναφορές, αγνοούνται. Εάν η μέθοδος είναι πολυμορφική (είτε λόγω υπέρβασης είτε λόγω υπερφόρτωσης), όλες οι κλάσεις που μπορούν να λάβουν κλήσεις συμπεριλαμβάνονται στην μέτρηση της σύζευξης.

Η υψηλή CBO είναι ανεπιθύμητη. Υπερβολική σύζευξη μεταξύ κλάσεων είναι επιζήμια για τον σχεδιασμό και αποτρέπει την επαναχρησιμοποίηση. Όσο πιο ανεξάρτητη είναι μια κλάση, τόσο πιο εύκολη είναι η επαναχρησιμοποίηση της σε κάποια άλλη εφαρμογή. Προκειμένου προωθηθεί η ενθουσία, η σύζευξη μεταξύ των αντικειμένων πρέπει να κρατηθεί στο ελάχιστο. Όσο μεγαλύτερος είναι ο αριθμός των ζευγαριών, τόσο υψηλότερη είναι η ευαισθησία στις αλλαγές σε άλλα σημεία του σχεδιασμού, και ως εκ τούτου η συντήρηση καθίσταται πιο δύσκολη. Η υψηλή σύζευξη έχει αποδειχθεί ότι προκαλεί επιρρέπεια σε σφάλματα. Αυτό που απαιτείται είναι αυστηρός και παρατεταμένος έλεγχος. Οι Sahraoui, Godin και Miceli σε ένα άρθρο τους είχαν ορίσει ως πολύ υψηλή τιμή την  $CBO > 14$ .

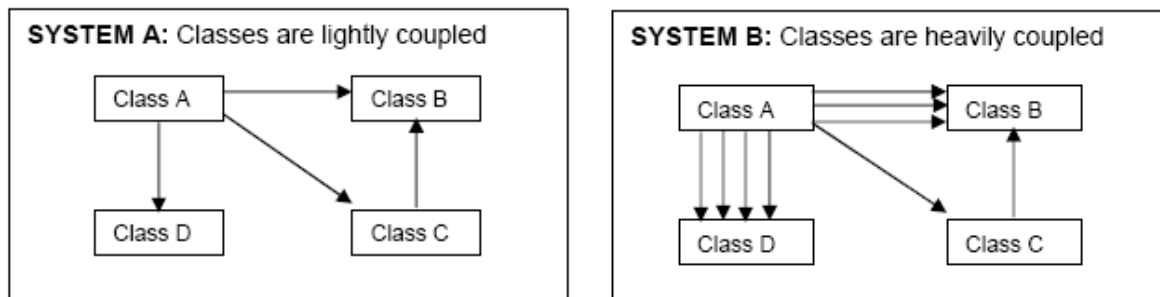
Λεπτομέρειες εφαρμογής.

Τον αρχικό ορισμό της CBO αν και δεν είναι απόλυτα σαφής οφείλουμε να τον αναφέρουμε. Ο ορισμός της CBO ασχολείται με τα ίχνη των μεταβλητών (instance variables) και όλες τις μεθόδους της κλάσης. Στο VB.NET, αυτό σημαίνει μη κοινές μεταβλητές, κατανεμημένες ή μη μέθοδοι. Αυτές οι κοινές μεταβλητές δεν λαμβάνονται υπόψη της μετρικής. Αντιθέτως, όλες οι καλούμενες μέθοδοι υπολογίζονται είτε είναι κατανεμημένες είτε όχι.

Εάν η κλήση είναι πολυμορφική είναι στην διεπαφή της μεθόδου στο .NET και δεν προσμετράται στην σύζευξη, ούτε της διεπαφής ούτε στις κλάσεις που υλοποιούν την διεπαφή. Εάν η κλήση είναι πολυμορφική είναι στο ορισμό της μεθόδου στη Visual Basic διεπαφή της κλάσης (βασική κλάση), υπολογίζεται στην διεπαφή της κλάσης αλλά όχι σε όποια κλάση υλοποιεί την διεπαφή. Αυτός είναι περιορισμός της υλοποίησης και όχι της CBO.

Στην υλοποίηση της CBO, όταν η παράγωγη κλάση καλεί τις βασικές της μεθόδους που κληρονομεί, συνδέεται με την κλάση γονέα στην οποία έχουν οριστεί οι μέθοδοι. Ο αρχικός ορισμός δεν ορίζει αν η κληρονομικότητα πρέπει να χειριστεί με κάποιο συγκεκριμένο τρόπο.

Όπως ορίσαμε και στην αρχή CBO είναι το πλήθος των συνδεδεμένων κλάσεων. Αυτός ο ορισμός είναι απλός αλλά δεν αποσαφηνίζει την πολυπλοκότητα της σύνδεσης μεταξύ των κλάσεων. Το ακόλουθο παράδειγμα μας διευκολύνει στην κατανόηση.

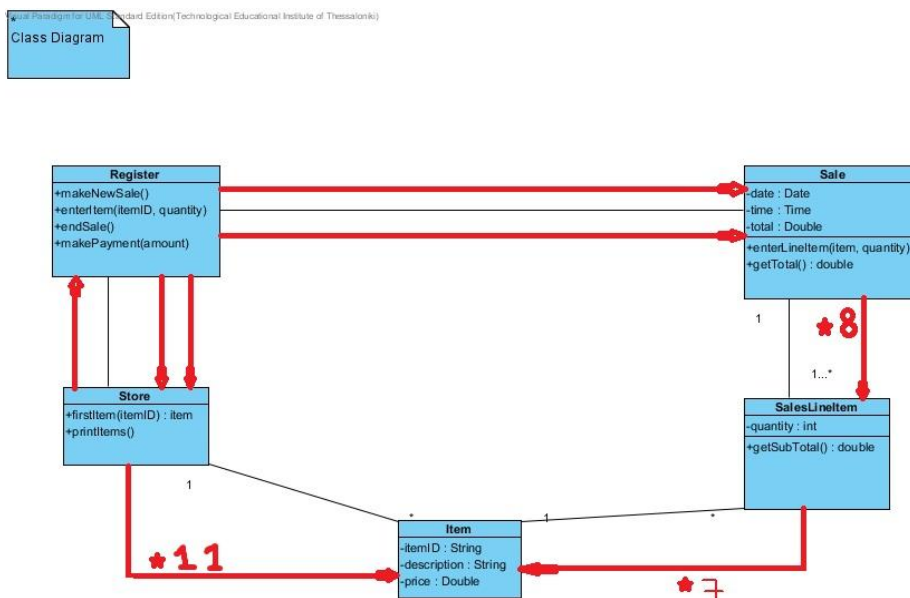


Εικόνα 5.3: Σύζευξη Κλάσεων

Τα δύο διαγράμματα παρουσιάζουν τον σχεδιασμό δύο διαφορετικών συστημάτων. Χρησιμοποιώντας την μετρική CBO μετράμε την κλάση A η οποία είναι 3. Ωστόσο, είναι ευδιάκριτο ότι η κλάση A του συστήματος B έχει μεγαλύτερη πολυπλοκότητα συγκριτικά με την κλάση A του συστήματος A.

Ο ορισμός της CBO εδώ θα πρέπει να επαναπροσδιοριστεί με πιο συγκεκριμένο τρόπο. Η CBO θα πρέπει να συνυπολογίζει τον αριθμό των μηνυμάτων που πέρασαν μεταξύ των κλάσεων. Στο παραπάνω παράδειγμα αυτός ο επαναπροσδιορισμός θα μετέτρεπε την τιμή της CBO από 3 σε 8 για το σύστημα B.

### Εφαρμογή της CBO πάνω στην Περίπτωση Χρήσης Process Sale.



## Response For a Class (RFC)

### Απόκριση κλάσης

Απόκριση κλάσης (Response For a Class). Η απόκριση μιας κλάσης είναι « το σύνολο των μεθόδων που μπορούν να εκτελεστούν ως απόκριση της κλάσης στην λήψη ενός μηνύματος από ένα αντικείμενο αυτής της κλάσης» (Chidamber και Kemerer, 1994). Η RFC είναι το πλήθος αυτών των μεθόδων ως απάντηση. Όσο αυξάνεται η RFC τόσο αυξάνεται και η καταβολή προσπαθειών για δοκιμές εξαιτίας της αύξησης της ακολουθίας των δοκιμών. Επίσης καθώς αυξάνεται η RFC είναι αναμενόμενο ότι αυξάνεται και η συνολική σχεδιαστική πολυπλοκότητα της κλάσης.

Η RFC μετράει τόσο την εξωτερική όσο και την εσωτερική επικοινωνία, αλλά κυρίως περιλαμβάνει μεθόδους που καλούνται εξωτερικά της κλάσης, επομένως αποτελεί και μέτρηση της δυναμικής επικοινωνίας μεταξύ της εν λόγω κλάσης και άλλων [ChK94, AIC98]. Η RFC είναι πιο ευαίσθητη μετρική σύζευξης από την CBO (που θα μελετηθεί αμέσως μετά) καθώς υπολογίζει μεθόδους αντί για κλάσεις.

Εάν εμπλακεί, ως απόκριση σε μηνύματα που λαμβάνει (message passing), μεγάλο πλήθος μεθόδων τότε η δοκιμές και η αποσφαλμάτωση της κλάσης καθίσταται πιο πολύπλοκη εφόσον απαιτεί μεγαλύτερη κατανόηση από την πλευρά του δοκιμαστή. Η τιμή της χειρότερης περίπτωσης για πιθανές αποκρίσεις θα βοηθήσουν στην κατάλληλη κατανομή της διάρκειας των δοκιμών.

**RFC = M + R (Πρώτο βήμα μέτρησης)**

**RFC' = M + R' (Πλήρης μέτρηση)**

M = ο αριθμός των μεθόδων της κλάσης

R = ο αριθμός των απομακρυσμένων μεθόδων που καλούνται άμεσα από την κλάση

R' = ο αριθμός των απομακρυσμένων μεθόδων που καλούνται αναδρομικά μέσω της συνολικής ιεραρχίας της κλάσης.

Μία δοθείσα μέθοδος υπολογίζεται μόνο μία φορά ως R (και R') ακόμα και εάν εκτελείται από διάφορες μεθόδους M.

Για παράδειγμα, η RFC για την κλάση SalesLineItem RFC = 5+6 = 11, δεδομένου ότι έχει 5 τοπικές μεθόδους (SalesLineItem(Item item, int quantity), getSubTotal(), getDescription(), getPrice(), toString()) και καλεί εξωτερικές κλάσεις (τις getPrice(), getDescription(), getPrice(), getItemID(), getDescription(), getPrice()) όλες από την Item κλάση).

Μια μελέτη 30 C++ έργων έδειξε ότι η αύξηση της RFC αυξάνει και την πυκνότητα των σφαλμάτων καθώς παράλληλα μειώνει την ποιότητα. Η μελέτη προτείνει την «βέλτιστη» χρήση της RFC, χωρίς όμως να λέει ποια είναι η βέλτιστη. Φαίνεται ασφαλές να υποθέσουμε ότι μία υψηλή RFC είναι επιζήμια σε Visual Basic προγράμματα ωστόσο (Misra & Bhavsar, 2003).

Η RFC είναι ο αρχικός ορισμός της μέτρησης. Μετράει μόνο το πρώτο επίπεδο των κλήσεων εκτός της κλάσης. Η RFC' «μετράει το σύνολο των αποκρίσεων, συμπεριλαμβανομένων και των μεθόδων που καλούνται εξωτερικά, αναδρομικά, μέχρις ότου να μην υπάρχουν νέες μέθοδοι. Αν η καλούμενη μέθοδος είναι πολυμορφική, όλες οι πιθανές απομακρυσμένες μέθοδοι που εκτελούνται περιλαμβάνονται στις R και R'.

Η χρήση της RFC', θα πρέπει να προτιμάται έναντι της RFC. Η RFC που ορίστηκε πρώτη, δεν είναι ικανή να υπολογίσει όλη την ιεραρχία της κλάσης με χειροκίνητο τρόπο. Με αυτοματοποιημένη ανάλυση του κώδικα από κάποιο εργαλείο, λαμβάνοντας τις τιμές της RFC' δεν αντιμετωπίζουμε καμία δυσκολία, έχοντας μια πιο εμπειριστατωμένη μέτρηση του εκτελέσιμου κώδικα.

### Παράδειγμα 1 RFC

```
public class A {
    private B aB;

    public void methodA1() {
        return aB.methodB1();
    }

    public void methodA2(C aC) {
        return aC.methodC1();
    }
}

RS = { methodA1, methodA2, methodB1, methodC1}
```

### Παράδειγμα 2 RFC



```

public class A {
    private B aB;

    public void methodA1() {
        return aB.methodB1();
    }

    public void methodA2() {
        return aB.methodB1();
    }
}

RS = { methodA1, methodA2, methodB1}
RS = { methodA1, methodA2, methodB1, methodB1}

```

### **Εφαρμογή της RFC πάνω στην Περίπτωση Χρήσης Process Sale.**

Έχουμε τις εξής 5 κλάσεις:

Sale, Store, Register, Item, SalesLineItem ( εκτός της main που δε προσμετράται ).

Υπολογίζουμε τις τιμές της Response For Class (RFC) metrics , αντίστοιχα για την κάθε κλάση.

Έχουμε :

$RFC(Sale) = M+R = 5 + 5(\text{ salesLineItem ,salesLineItem, salesLineItem, salesLineItem, salesLineItem } ) = 10 .$

$RFC(Store) = M + R = 5 + 1(Item)=6 .$

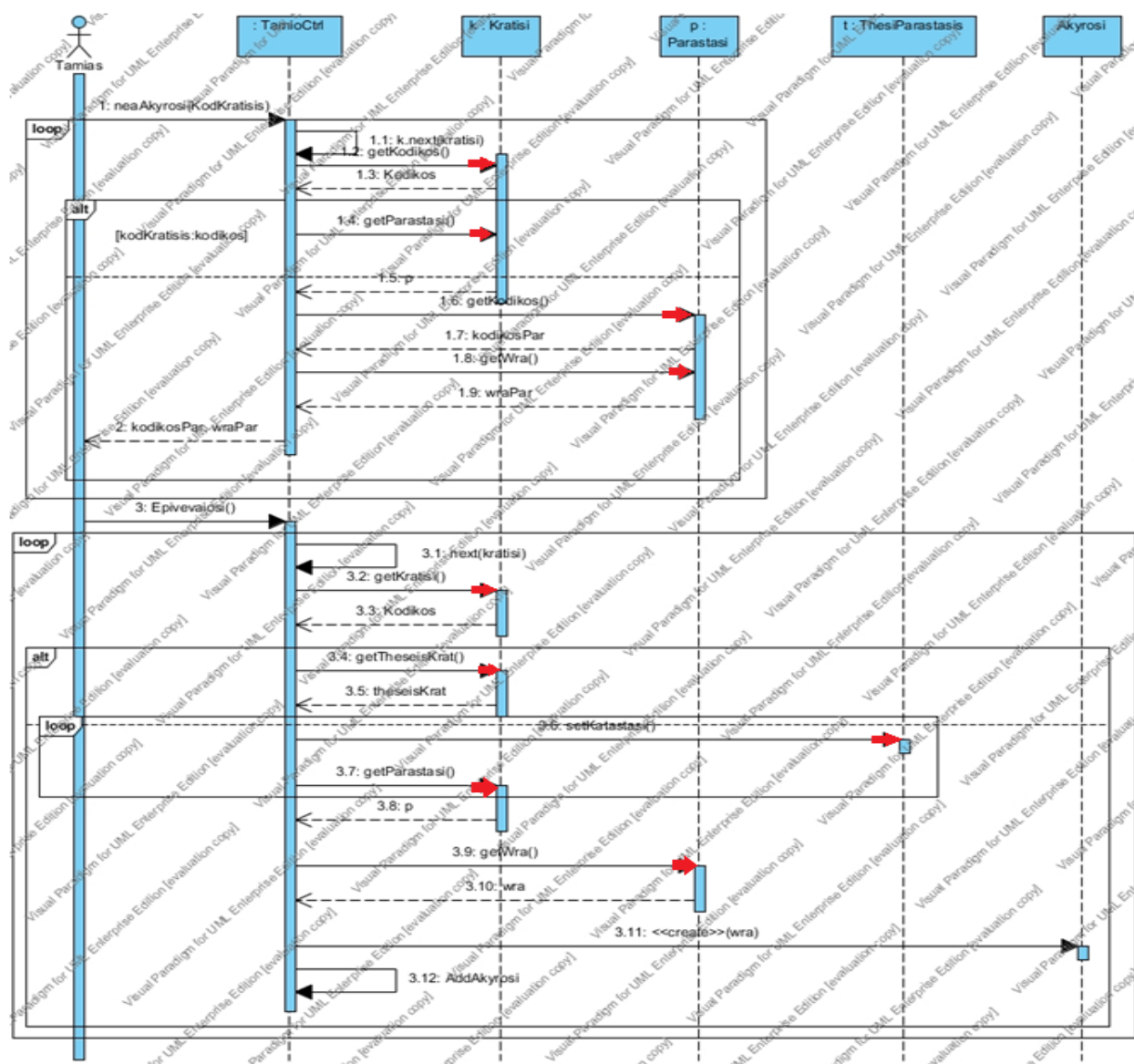
$RFC (Item) = 8 + 0 = 8 .$

$RFC (Register) = M + R = 4 + 3(Store, Sale, Sale) = 7 .$

$RFC ( SalesLineItem ) = M + R = 5+ 6(Item, item, Item, Item, Item, Item) = 9 .$

Κάθε γραμμή στο διάγραμμα αναφέρεται σε μήνυμα από μια κλάση σε μια άλλη.





RFC ( Φοιτητή νούμερο 2 for Class TameioCtrl ) = M (ο αριθμός των μεθόδων της κλάσης)+ R (ο αριθμός των απομακρυσμένων μεθόδων που καλούνται άμεσα από την κλάση) = **2 ( neaAkyrosi(int kwdKratisis), Epibebaiosi() ) + 7 ( getKwdikos(), getKwdikos(), ,getWra(),getParastasi(int kwdikos), getWra(), getTheseisKrat(), setKatastasi() ) = 9**

Όπως βλέπουμε ο αριθμός των απομακρυσμένων μεθόδων που καλούνται άμεσα από την κλάση ( κόκκινα βελάκια στο Διάγραμμα Ακολουθίας ) είναι ακριβώς 9, όσο και στον κώδικα του φοιτητή( βλέπε Παράρτημα Β' – Κώδικας Φοιτητή ). Άρα υπάρχει αντιστοιχία κώδικα-Διαγράμματος και μπορούμε με ευκολία να βγάλουμε σωστά συμπεράσματα. ( Υπενθυμίζουμε ότι η μέτρηση γίνεται στο πρότυπο-βέλτιστο γραπτό του φοιτητή νο. 2, για την κλάση

TameioCtrl μόνο και για την οποία μόνο έχει γραφεί κώδικα από τον συγκεκριμένο φοιτητή, όπως ήτανε άλλωστε και το ζητούμενο.

## Metrics for Model

Οι μετρικές του μοντέλου χρησιμοποιούνται για τον υπολογισμό του μεγέθους ή της ποσότητας της πληροφορίας που υπάρχει μέσα σε ένα μοντέλο. Ένα μοντέλο είναι ένας καθρέφτης ενός συστήματος που θα αναπτυχθεί σε έναν μετέπειτα κύκλο ζωής του λογισμικού. Έτσι, αυτές οι μετρικές είναι χρήσιμες αφού οι υπεύθυνοι για την ανάπτυξη λογισμικού μπορούν να προβλέψουν τα χαρακτηριστικά των συστημάτων που δεν έχουν υλοποιηθεί ακόμη.

**Number of the classes in a model (NCM)** : Μια κλάση σε ένα μοντέλο είναι μια αναφορά της μετακλάσης “class”. Αυτή η μετρική μετράει τον αριθμό των κλάσεων σε ένα μοντέλο. Αυτή η μετρική είναι συγκρίσιμη με την παραδοσιακή LOC (lines of code) ή μια πιο εξειδικευμένη cyclomatic complexity (MVG) του McCabe που υπολογίζει το μέγεθος του συστήματος. Έτσι, στον Αντικειμενοστραφή Προγραμματισμό αυτή η μετρική μπορεί να χρησιμοποιηθεί για την σύγκριση των μεγεθών των συστημάτων.

**Number of the objects in a model (NOM)**: Κατά έναν παρόμοιο τρόπο που μια κλάση είναι μια αναφορά της μετακλάσης “Class”, ένα αντικείμενο μπορεί να είναι μια αναφορά μιας κλάσης. Μπορούμε να μετρήσουμε αυτά τα αντικείμενα.

**Number of the messages in a model (NMM)**: Ένα μήνυμα είναι μια αναφορά της μετακλάσης “Message”. Μηνύματα ανταλλάσσονται μεταξύ αντικειμένων δημιουργώντας διάφορες διαδράσεις. Αυτά τα μηνύματα μπορούμε να τα μετρήσουμε.

**Number of the associations in a model (NASM)**: Μια ένωση ( association ),είναι μια σύνδεση ή ένας σύνδεσμος μεταξύ των κλάσεων. Αυτή η μετρική είναι χρήσιμη για να υπολογίζει το βαθμό των σχέσεων μεταξύ των κλάσεων.

## Metrics for Class

Οι μετρικές των κλάσεων μας αφορούν με διάφορα στοιχεία μιας κλάσης όπως είναι τα χαρακτηριστικά, σχέση και αναφορές αντικειμένων.

**Number of the attributes in a class - unweighted (NATC1):** Αυτή η μετρική μετράει τον αριθμό των χαρακτηριστικών σε μια κλάση. Τα public, private και protected χαρακτηριστικά αντιμετωπίζονται όμοια ( δεν έχουμε βάρη )

**Number of the operations in a class - unweighted (NOPC1):** Αυτή είναι μια μετρική που μετράει τον αριθμό των λειτουργιών σε μια κλάση.

**Number of the associations linked to a class (NASC):** Ο αριθμός των ενώσεων συμπεριλαμβανομένου και των συνθέσεων μετρούνται. Αυτή η μετρική είναι χρήσιμη για τον υπολογισμό των στατικών σχέσεων μεταξύ των κλάσεων.

**Coupling between classes (CBC):** Also known as message delivery channels, Αυτή η μετρική μετρά τον αριθμό των ενώσεων σε μια κλάση και τα χαρακτηριστικά που είναι παράμετροι ενός τύπου μιας άλλης κλάσης. Μηνύματα μπορούν μόνο να σταλούν όταν ένα αντικείμενο μιας κλάσης κρατά μια αναφορά σε ένα άλλο αντικείμενο μιας άλλης κλάσης.

## Metrics for Message

Στην UML, τα μηνύματα που στέλνονται από μια οποιαδήποτε πηγή σε έναν οποιοδήποτε προορισμό αποτελούν διαδράσεις.

**Number of the directly dispatched messages of a message (NDM):** Σύμφωνα με τη σημασιολογία της UML, ένα μήνυμα μπορεί να ενεργοποιήσει άλλα μηνύματα. Αυτή η μετρική μετράει αυτή τη συνέχεια των περαιτέρω μηνυμάτων.

## ΕΠΙΛΟΓΟΣ

Οι μετρικές που πάρθηκαν από το Αντικειμενοστραφές μοντέλο και ενσωματώθηκαν στην ανάπτυξη συστημάτων δεν έχουν επικυρωθεί ως προάγγελοι της συντήρησης λογισμικού. Αυτό είναι φυσικά ένα μειονέκτημα και παραμένει το υψηλότερο καθήκον προτεραιότητας των μελλοντικών εργασιών. Η υπόθεση στη μελέτη σπουδών είναι ότι υπάρχει σημαντική σχέση μεταξύ των μετρικών στην αρχιτεκτονική λογισμικού και τη συντηρησιμότητα. Ο πιο απλός τρόπος να γίνει αυτό είναι να βρούμε ένα σύνολο έργων και να συλλέξουμε τα δεδομένα που εξάγονται από τις μετρικές από την αρχιτεκτονική του λογισμικού και την τροποποίηση των δεδομένων κατά τη διάρκεια της περιόδου συντήρησης. Τότε θα μπορούσαμε να κάνουμε στατιστική ανάλυση σχετικά με τα δεδομένα. Υπάρχει ένα πρόβλημα όμως με αυτή την προσέγγιση. Η τρέχουσα κατάσταση στην ανάπτυξη λογισμικού δεν μας επιτρέπει να συλλέξουμε και τα δυο ειδών δεδομένα από το ίδιο έργο, αλλά μόνο από ένα είδος. Έτσι θα βρεθούμε σε μία από τις εξής δύο καταστάσεις:

Πρώτον, έχουμε ένα σχέδιο που θα χρησιμοποιήσουμε για την τροποποίηση των δεδομένων κατά τη διάρκεια της περιόδου συντήρησης, αλλά δεδομένου ότι τα στοιχεία συντήρησης είναι διαθέσιμα η συγκεκριμένη μελέτη έγινε κάποια χρόνια πριν. Εκείνη την ώρα είναι περισσότερο από πιθανό ότι οι σχεδιαστές δεν γνώριζαν την έννοια της αρχιτεκτονικής λογισμικού και κατά συνέπεια δεν βασίστηκαν πάνω της για την ανάπτυξη του λογισμικού. Ακόμα και αν έχουμε ανακτήσει την αρχιτεκτονική, η σχέση μεταξύ της αρχιτεκτονικής του λογισμικού και της συντήρησης του λογισμικού δεν υπήρχε κατά την ανάπτυξη του έργου.. Ως εκ τούτου, η σχέση που αποτελεί το θεμέλιο για την πρόβλεψη δεν ήταν παρούσα και έτσι η περίπτωση δεν μπορεί να βασίσει την υπόθεση.

Δεύτερον, έχουμε θεμελιώσει σχέσεις με μια εταιρεία ανάπτυξης λογισμικού που έχει υιοθετήσει τις έννοιες της αρχιτεκτονικής λογισμικού στις μεθόδους ανάπτυξης. Καταφέραμε να τους πείσουμε για να μας δώσουν πλήρη πρόσβαση στην αρχιτεκτονική κατά τη διάρκεια του χρόνου ζωής του έργου. Τώρα το πρόβλημα είναι ότι δεν έχουμε πρόσβαση στα τροποποιημένα δεδομένα αφού η φάση της συντήρησης δεν θα παράγει όγκο στοιχείων που χρειαζόμαστε για αρκετό καιρό ακόμα. Για να παράγουμε αποτελέσματα στο εγγύς μέλλον που επικυρώνουν ή και όχι τις προτεινόμενες μετρικές αρχιτεκτονικής του λογισμικού πρέπει να βρούμε άλλα μέσα από παραδοσιακή μελέτη περίπτωσης.

Οι εφαρμογές Web έχουν εξελιχθεί σε πολύπλοκες εφαρμογές που έχουν υψηλό κόστος συντήρησης. Το υψηλό κόστος οφείλεται στα εγγενή χαρακτηριστικά των διαδικτυακών εφαρμογών, στην ταχεία εξέλιξη του Διαδικτύου, καθώς και στις πιεστικές αγορές οι οποίες επιβάλλουν μικρούς κύκλους ανάπτυξης και συχνές τροποποιήσεις. Προκειμένου να ελεγχθεί το κόστος συντήρησης στις δικτυακές εφαρμογές, έχουμε ορίσει διάφορες μεθόδους

μέτρησης, σχεδιασμού για δικτυακές εφαρμογές και έχουν διεξαχθεί πολλές εμπειρικές μελέτες που δείχνουν την χρησιμότητα των μετρικών.

## ΚΕΦΑΛΑΙΟ 4

### ΠΕΙΡΑΜΑ

#### Εισαγωγή :

Το πείραμα διεξάχθηκε σε μια ομάδα φοιτητών του κύριου Δεληγιάνη στο μάθημα Μεθοδολογίες Προγραμματισμού II . Ο αριθμός των φοιτητών ήταν 24, χωρισμένοι σε δύο ομάδες ( Α' και Β' ) των 12 ατόμων η καθεμία. Η Α' ομάδα προσπάθησε να διεκπεραιώσει την άσκηση με μια μεθοδολογία κάθετης ιχνηλασιμότητας , ενώ η Β' αντίστοιχα με μια μεθοδολογία οριζόντιας ιχνηλασιμότητας . Ο σκοπός του πειράματος ήταν να εξετάσουμε αν οι φοιτητές μπορούν με την ίδια άνεση να εργαστούν πάνω σε μια Περίπτωση Χρήσης με τη βοήθεια των Διαγραμμάτων Ακολουθίας και Κλάσης μέσα από την εκάστοτε αυτή μεθοδολογία, δηλαδή κάθετης ή οριζόντιας ιχνηλασιμότητας. Στις δύο ομάδες δόθηκε ο κώδικας του παραρτήματος Γ' και του ζητήθηκε να συμπληρωθεί κατάλληλα.

Δουλέψαμε ως εξής: Εισάγαμε τα διαγράμματα όλων των φοιτητών στο εργαλείο Visual Paradigm (VP) , τα κάναμε εξαγωγή σε μορφή XMI ( έκδοσης v1.2 – βλέπε παράρτημα Α') τα οποία μελετήθηκαν με τη βοήθεια του εργαλείου MyXMIReader . Ο MyXMIReader είναι ένα εργαλείο που περιέχει κώδικα ο οποίος εκτυπώνει για ένα διάγραμμα ακολουθίας τα μηνύματα (messages) , τους χρήστες(actors) και τις γραμμές ζωής ( lifelines ) που υπάρχουν στο διάγραμμα. Το MyXMIReader είναι ένα έτοιμο Eclipse project που του εισάχθηκε ένα αρχείο vrr για να βλέπουμε οπτικά στο Visual Paradigm τα διαγράμματα .

Όπως φαίνεται στον κώδικα (παράρτημα Α') για κάθε model element μπορούμε να διαβάσουμε τις ιδιότητές του. Για παράδειγμα, για ένα μήνυμα(message) μπορούμε να πάρουμε το όνομά του (name), τον αποστολέα (messagesender) και τον παραλήπτη του (messagereceiver).



## ΟΜΑΔΑ Α΄

### ΠΧ 1: Ακύρωση Κράτησης

Ο Ταμίας θέλει να ακυρώσει μια κράτηση.

1. Ο Πελάτης έρχεται στο ταμείο και ζητά ακύρωση μιας κράτησης που είχε κάνει.
2. **Ο Ταμίας ζητά νέα ακύρωση εισάγοντας τον κωδικό της κράτησης.(ΣΥΜΒΑΝ)**
3. Το Σύστημα αναζητεί την κράτηση.
4. Το Σύστημα εμφανίζει την κράτηση με τις θέσεις που είχαν δεσμευτεί για την παράσταση ( κωδικός, ώρα ).
5. **Ο Ταμίας πατάει το πλήκτρο ‘Επιβεβαίωση’ . (ΣΥΜΒΑΝ)**
6. Το σύστημα αποδεσμεύει (βάζει ‘ ‘ )τις θέσεις της παράστασης.
7. Το σύστημα ενημερώνει την κράτηση ως ‘διαγραμμένη’ ( τοποθετώντας το αντικείμενο της ακύρωσης σε πεδίο-αναφορά της κράτησης ).

Για το συμβάν του βήματος 2:

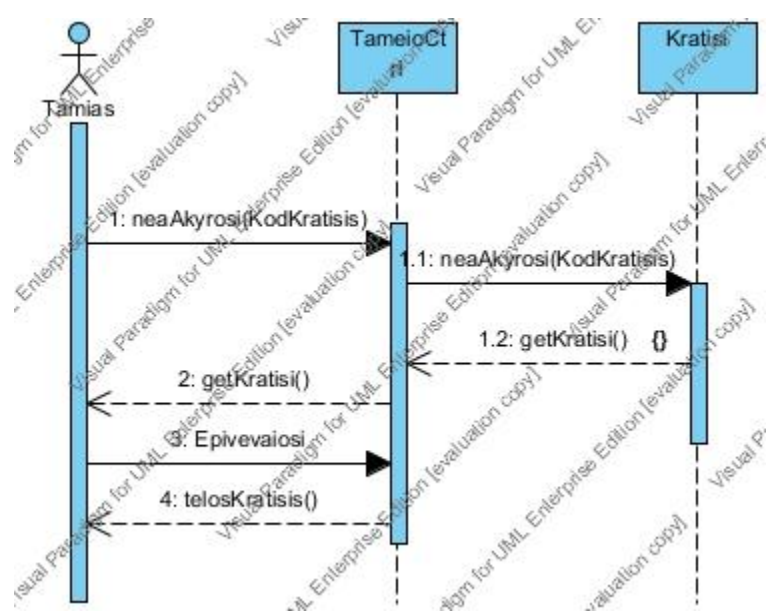
- A. Να σχεδιάσετε το μέρος του διαγράμματος Ακολουθίας που αναλογεί.
- B. Να σχεδιάσετε το μέρος του διαγράμματος Κλάσεων που αναλογεί.
- Γ. Να κωδικοποιήσετε σε Java τη σχεδίαση που αναλογεί.

Για το συμβάν του βήματος 5:

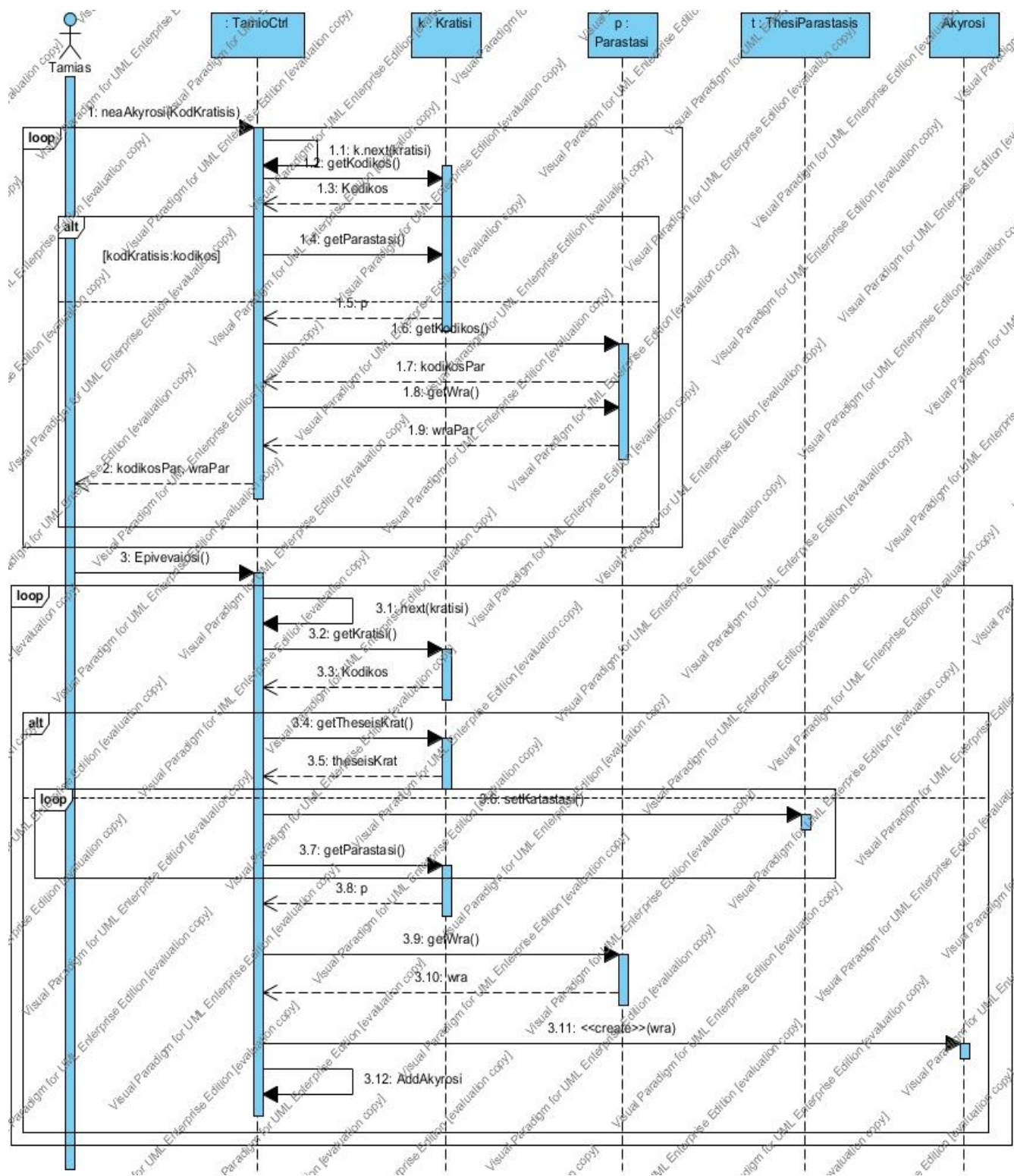
- A. Να σχεδιάσετε το μέρος του διαγράμματος Ακολουθίας που αναλογεί.
- B. Να σχεδιάσετε το μέρος του διαγράμματος Κλάσεων που αναλογεί.
- Γ. Να κωδικοποιήσετε σε Java τη σχεδίαση που αναλογεί.

# ΔΙΑΓΡΑΜΜΑΤΑ ΑΚΟΛΟΥΘΙΑΣ Α΄ ΟΜΑΔΑΣ

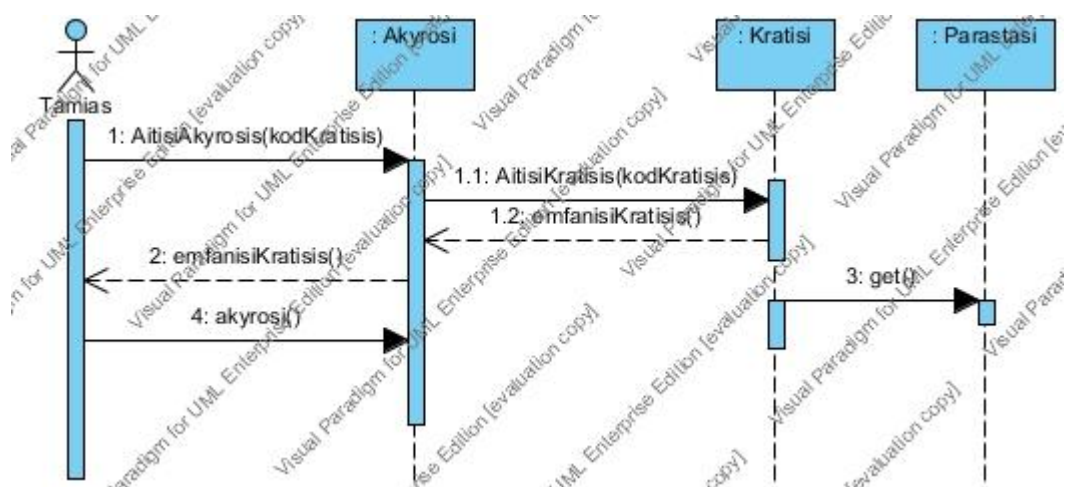
Sequence Diagram 1:



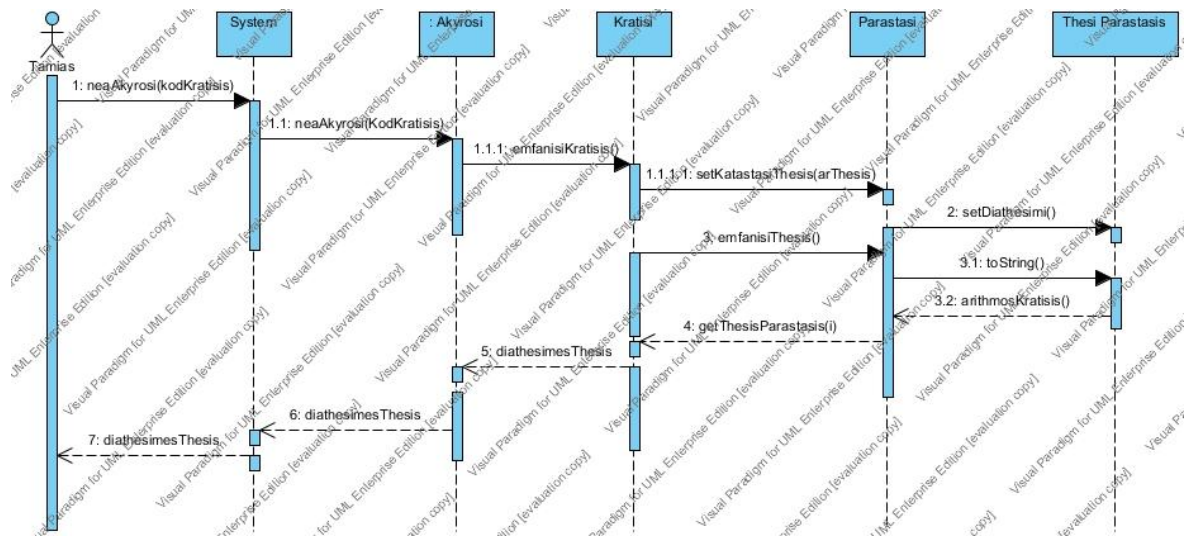
Sequence Diagram 2 :



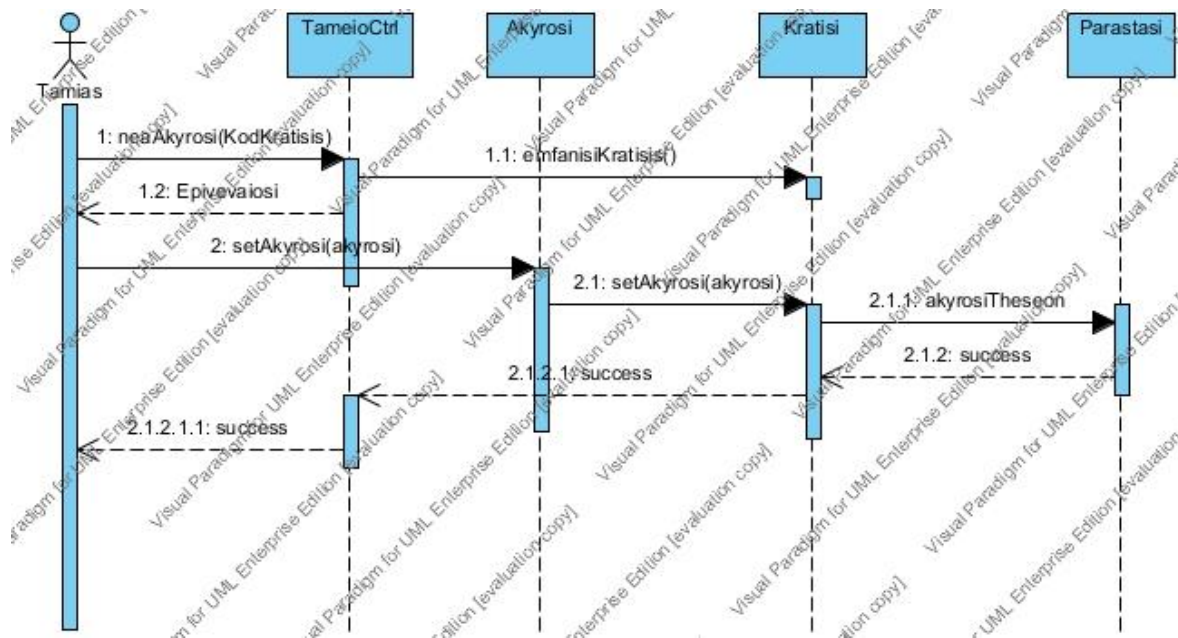
Sequence Diagram 5:



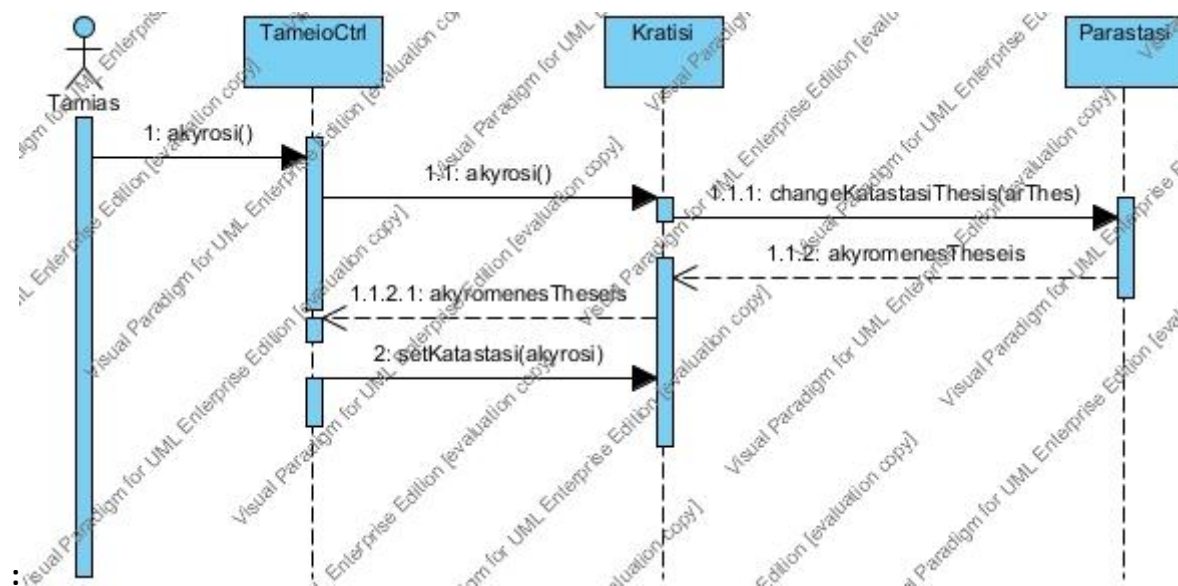
Sequence Diagram 13 :



Sequence Diagram 14 :

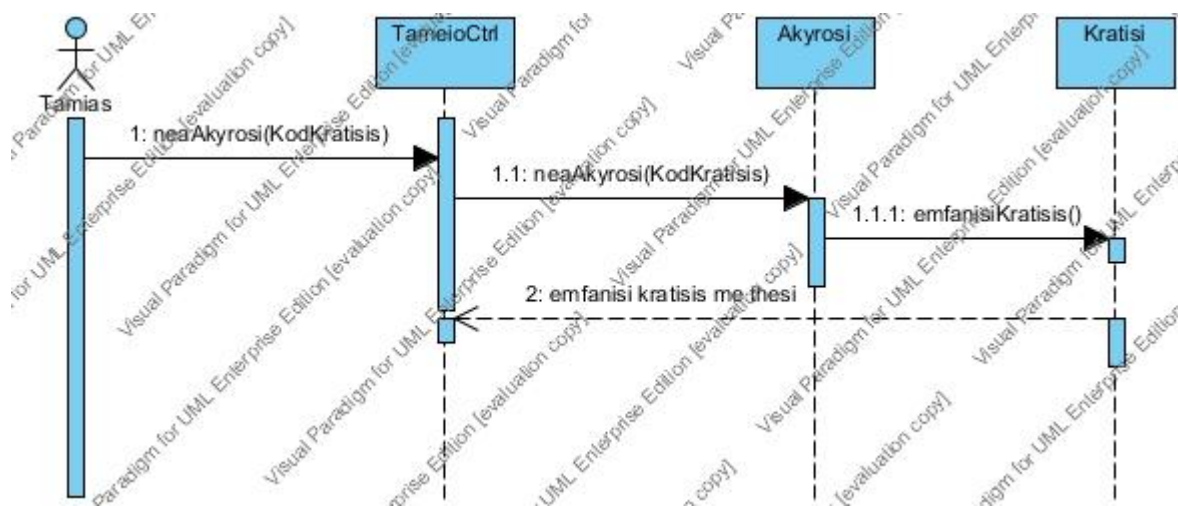


Sequence Diagram 16 A ( για το Βήμα 5 )

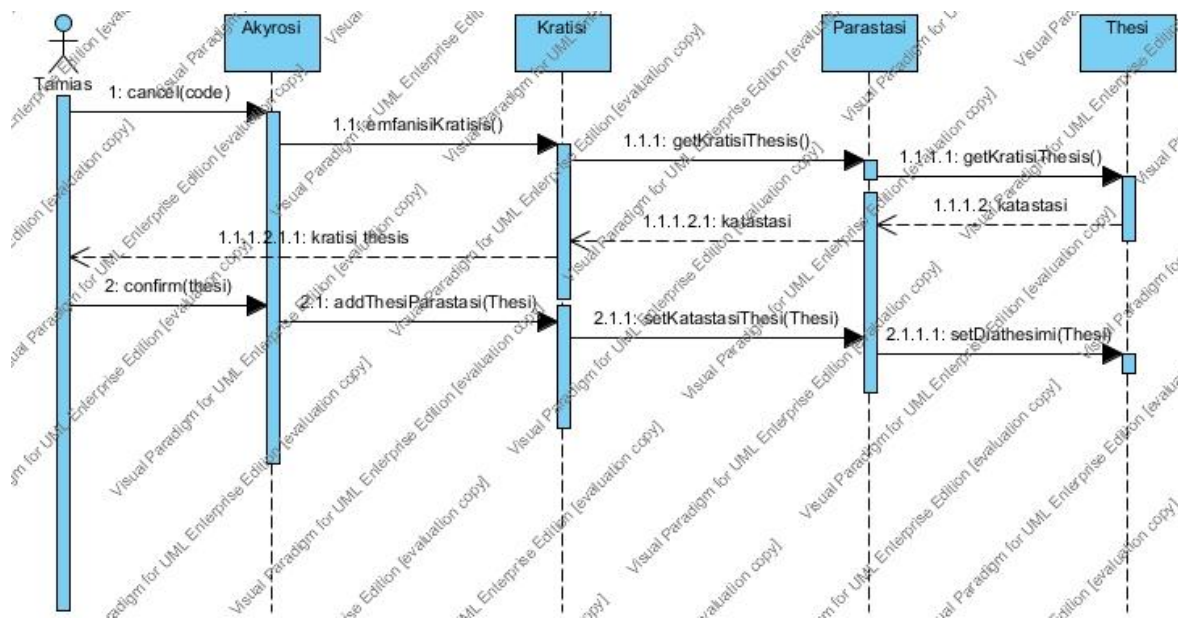




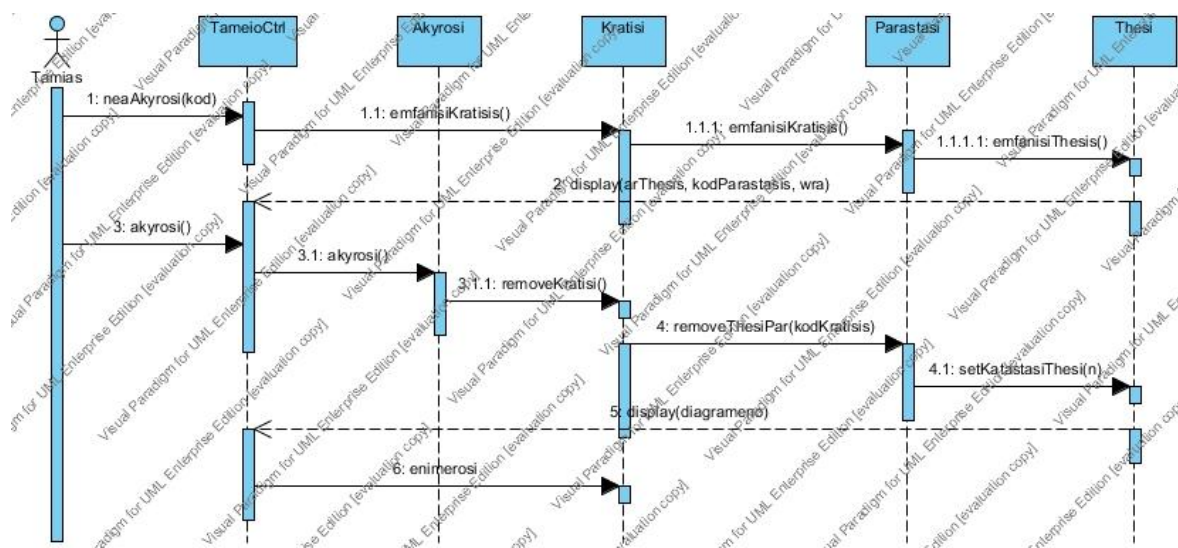
Sequence Diagram 16 B ( για το Βήμα 2 ) :



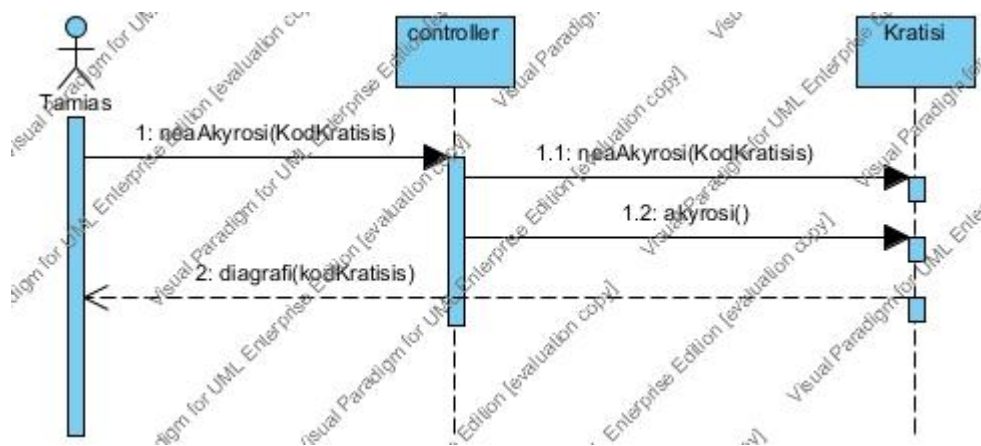
Sequence Diagram 17 :



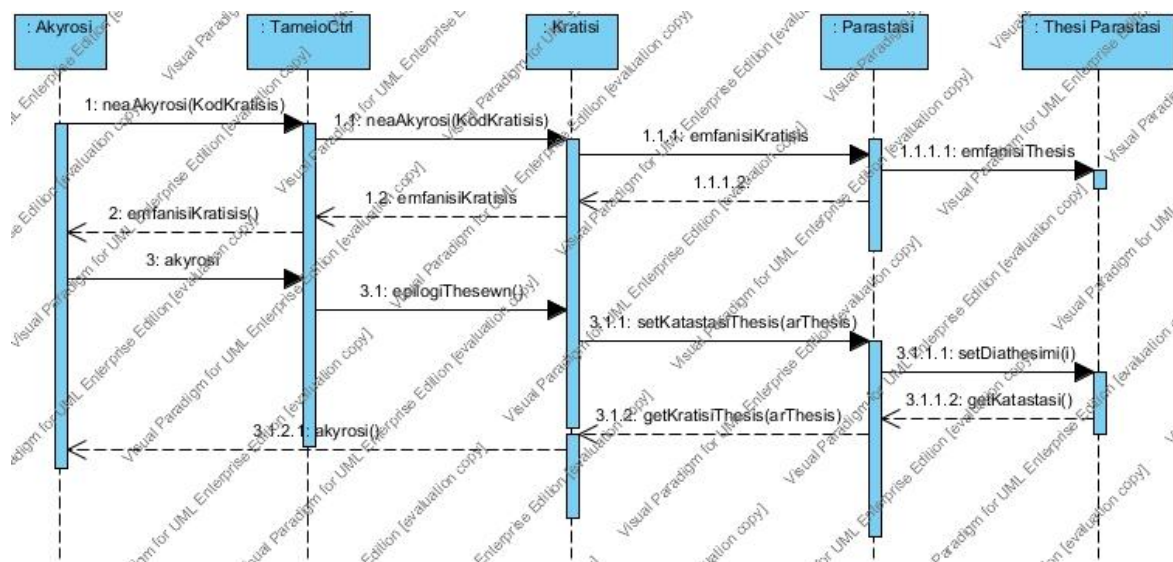
Sequence Diagram 19 :



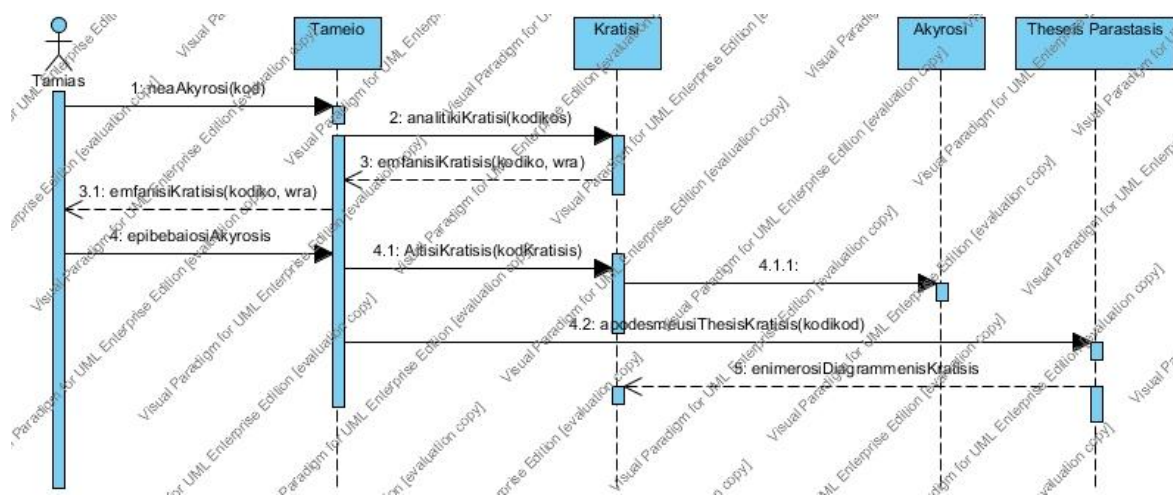
Sequence Diagram 22 :



Sequence Diagram 25 :

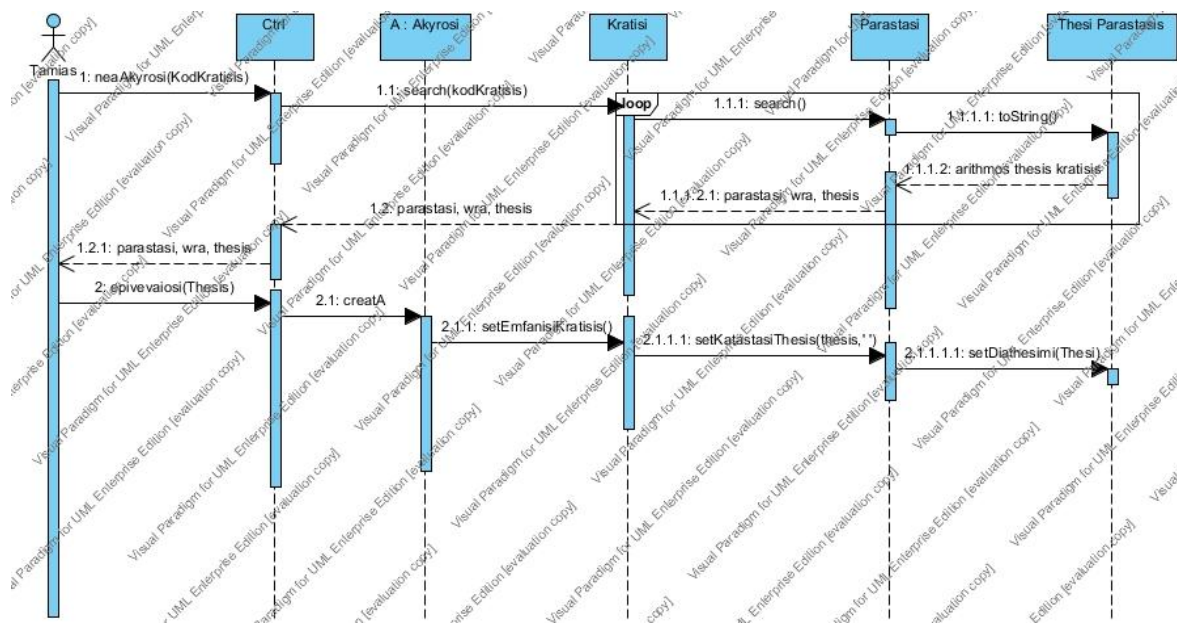


Sequence Diagram 28 :





Sequence Diagram 33 :



## ΟΜΑΔΑ Β'

### ΠΧ 1: Ακύρωση Κράτησης

Ο Ταμίας θέλει να ακυρώσει μια κράτηση.

1. Ο Πελάτης έρχεται στο ταμείο και ζητά ακύρωση μιας κράτησης που είχε κάνει.
2. **Ο Ταμίας ζητά νέα ακύρωση εισάγοντας τον κωδικό της κράτησης.(ΣΥΜΒΑΝ)**
3. Το Σύστημα αναζητεί την κράτηση.
4. Το Σύστημα εμφανίζει την κράτηση με τις θέσεις που είχαν δεσμευτεί για την παράσταση ( κωδικός, ώρα ).
5. **Ο Ταμίας πατάει το πλήκτρο 'Επιβεβαίωση'. (ΣΥΜΒΑΝ)**
6. Το σύστημα αποδεσμεύει (βάζει ' ') τις θέσεις της παράστασης.
7. Το σύστημα ενημερώνει την κράτηση ως 'διαγραμμένη' ( τοποθετώντας το αντικείμενο της ακύρωσης σε πεδίο-αναφορά της κράτησης ).

### A. Να σχεδιάσετε:

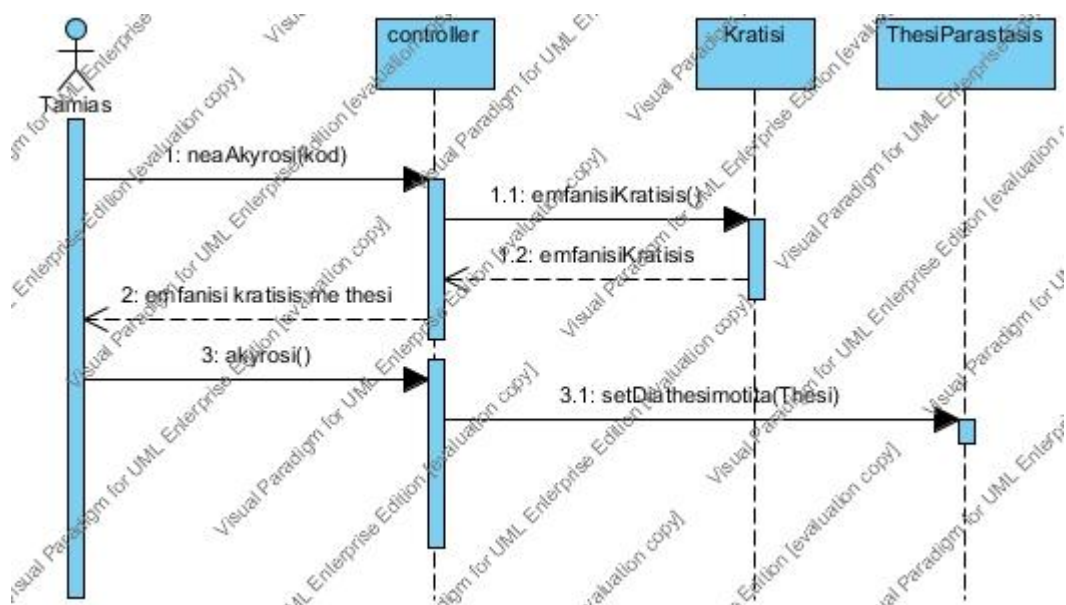
1) Το Διάγραμμα Ακολουθίας

2) Το διάγραμμα Κλάσεων

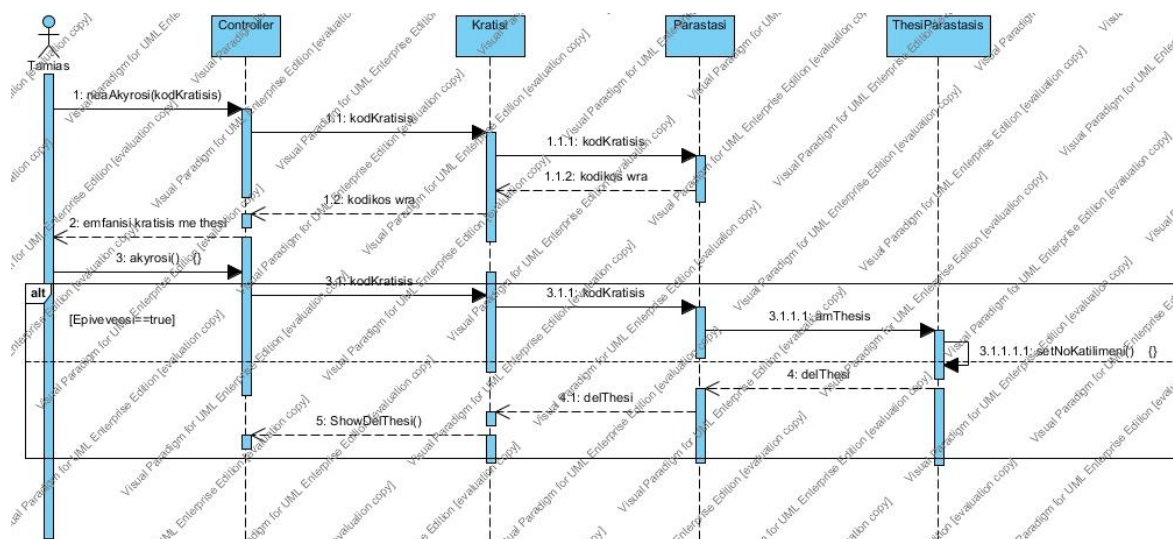
B. Να κωδικοποιήσετε σε Java την σχεδιάσή σας.

## ΔΙΑΓΡΑΜΜΑΤΑ ΑΚΟΛΟΥΘΙΑΣ Β' ΟΜΑΔΑΣ

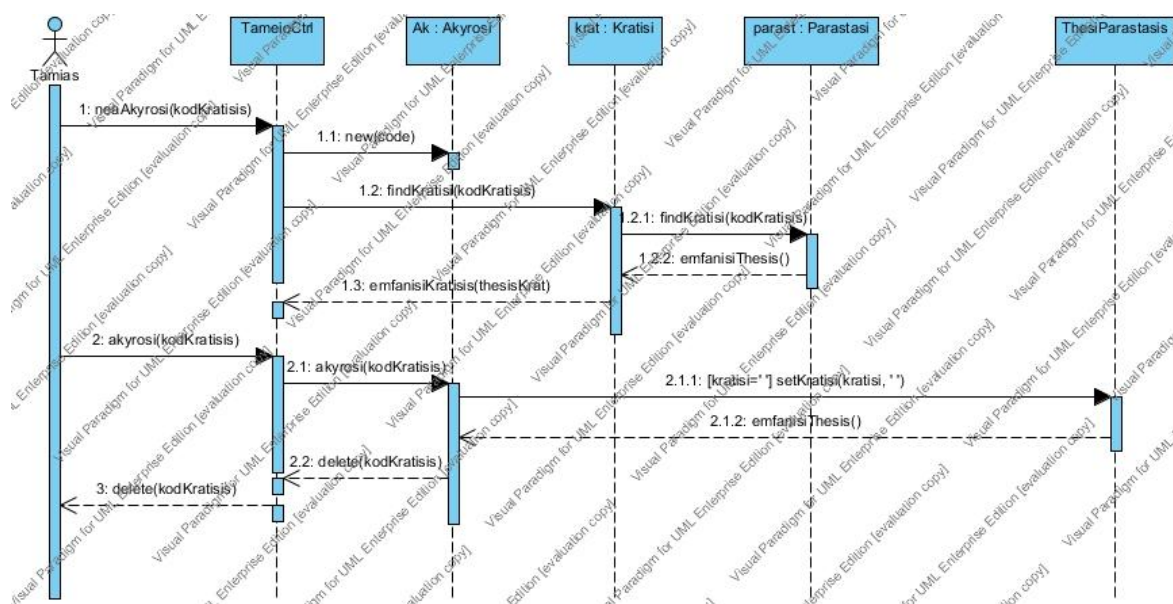
Sequence Diagram 6 :



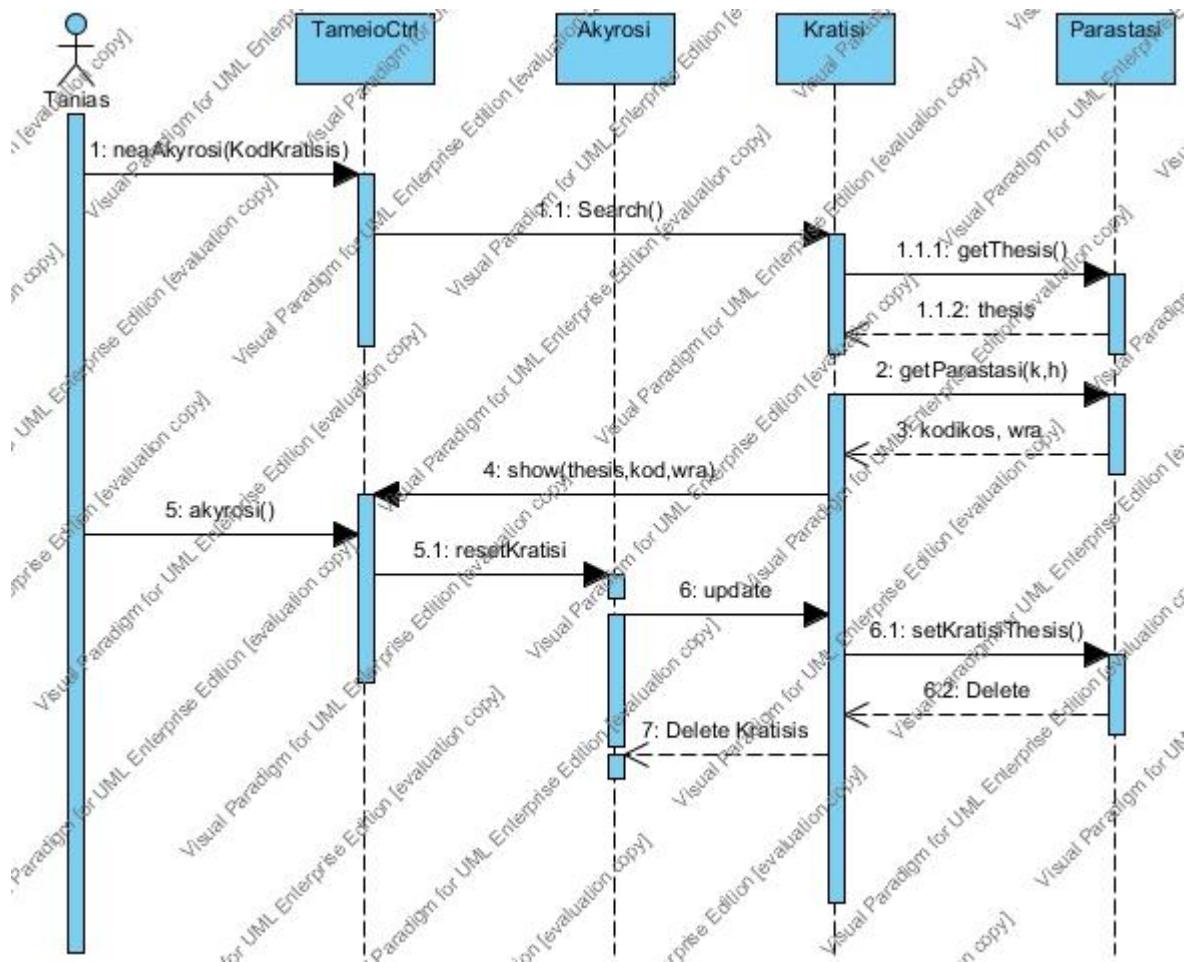
Sequence Diagram 8 :



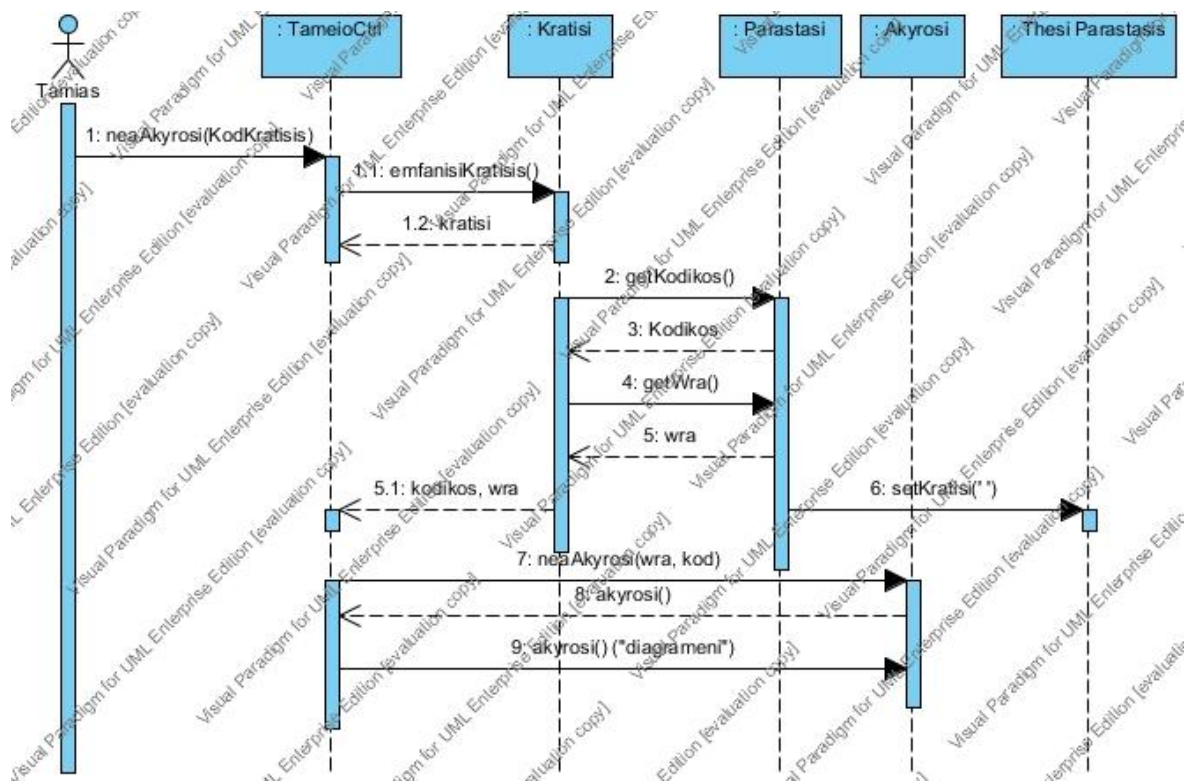
Sequence Diagram 10 :



Sequence Diagram 11 :

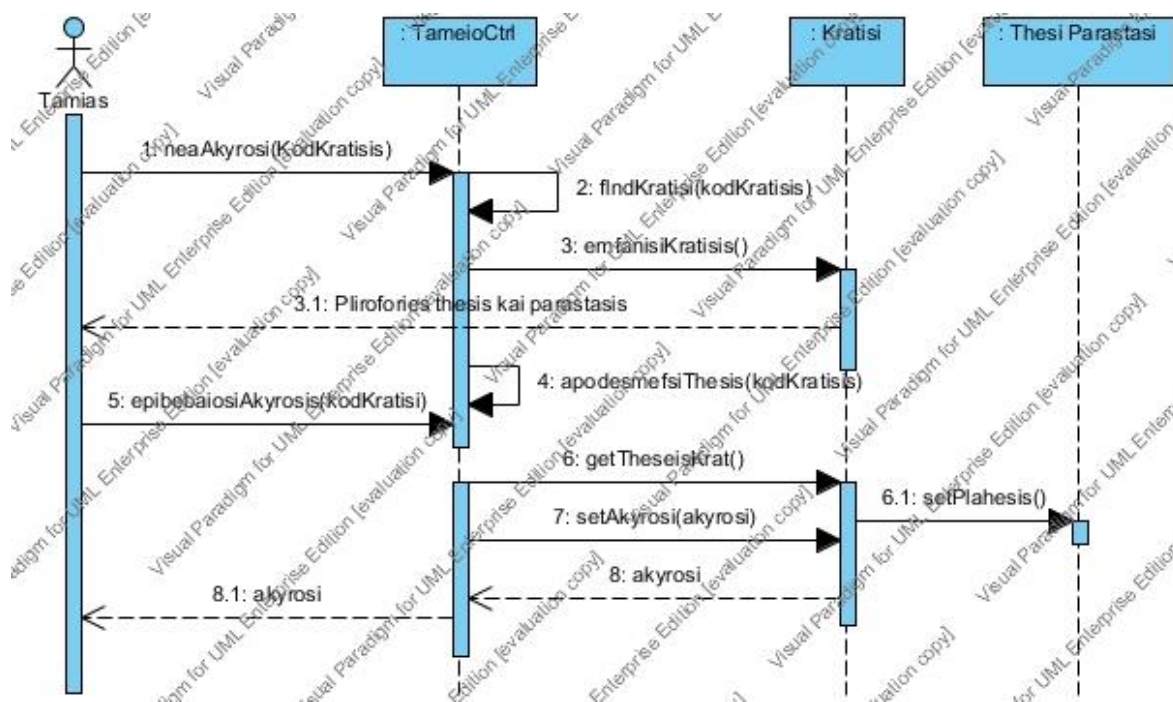


Sequence Diagram 15 :

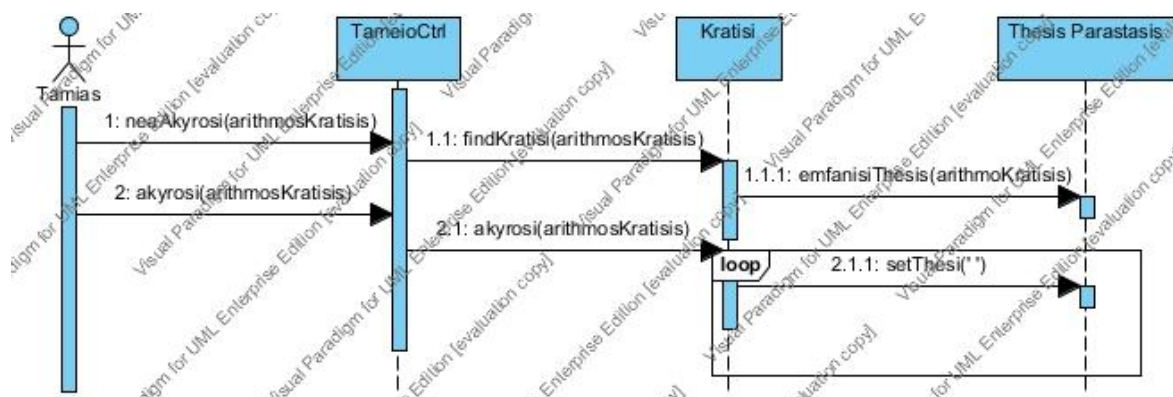




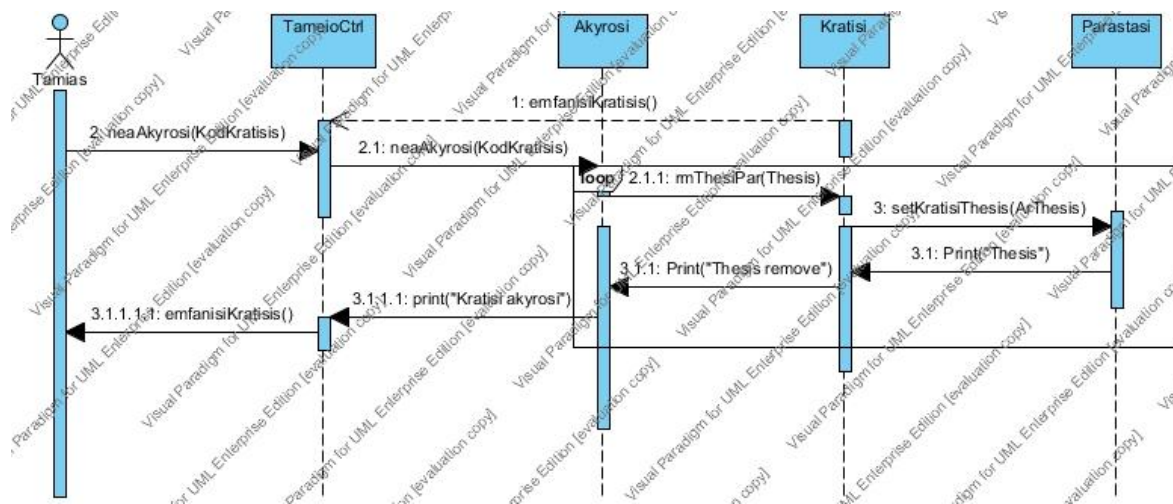
Sequence Diagram 18 :



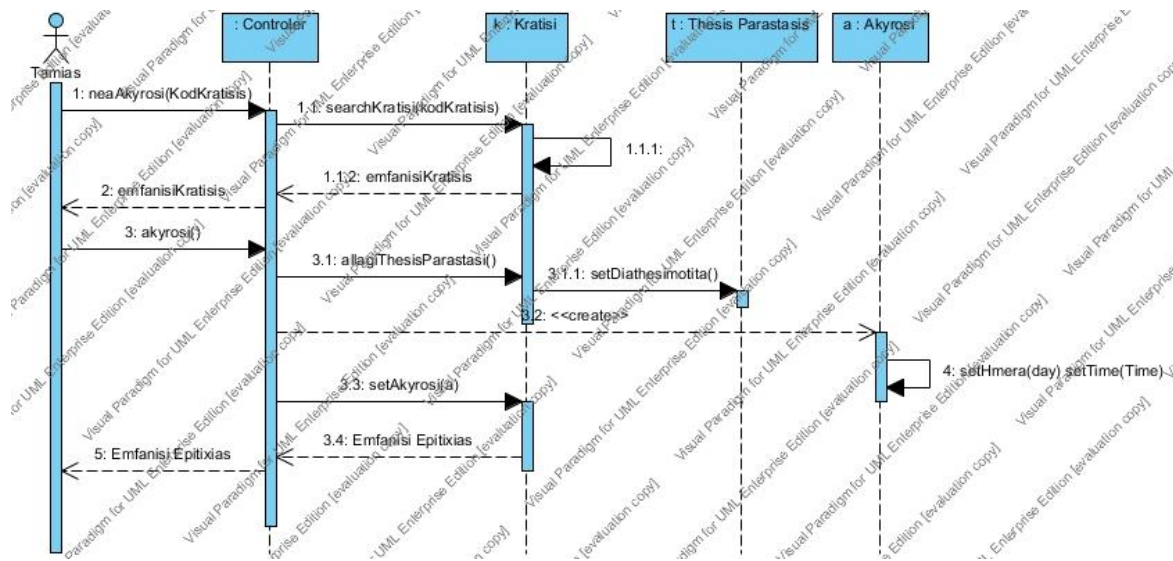
Sequence Diagram 26 :



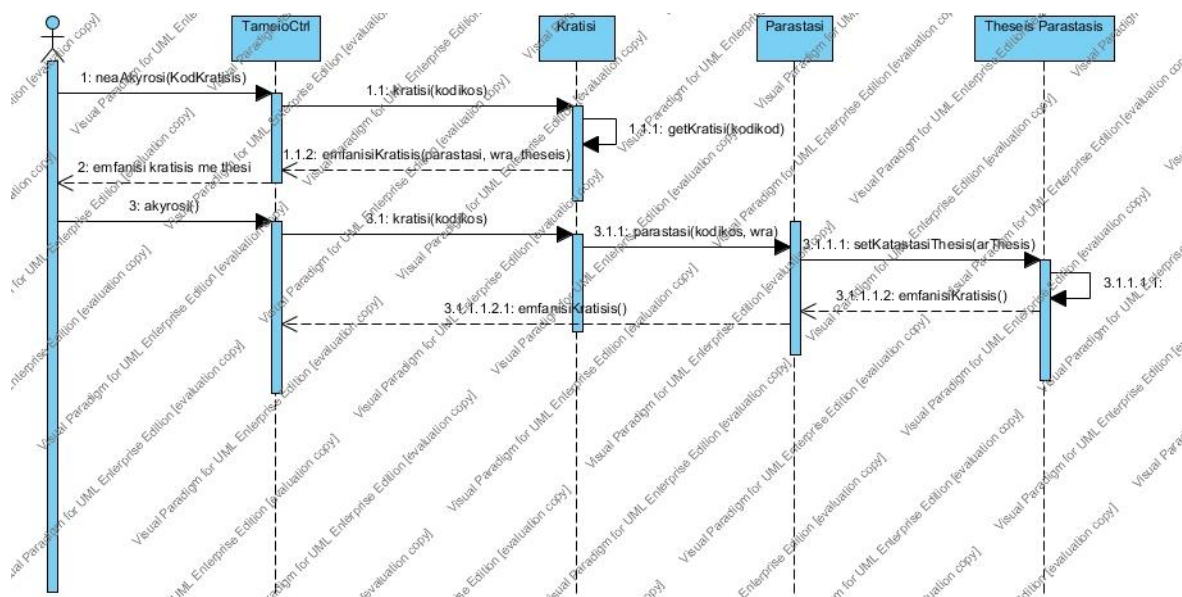
Sequence Diagram 27 :



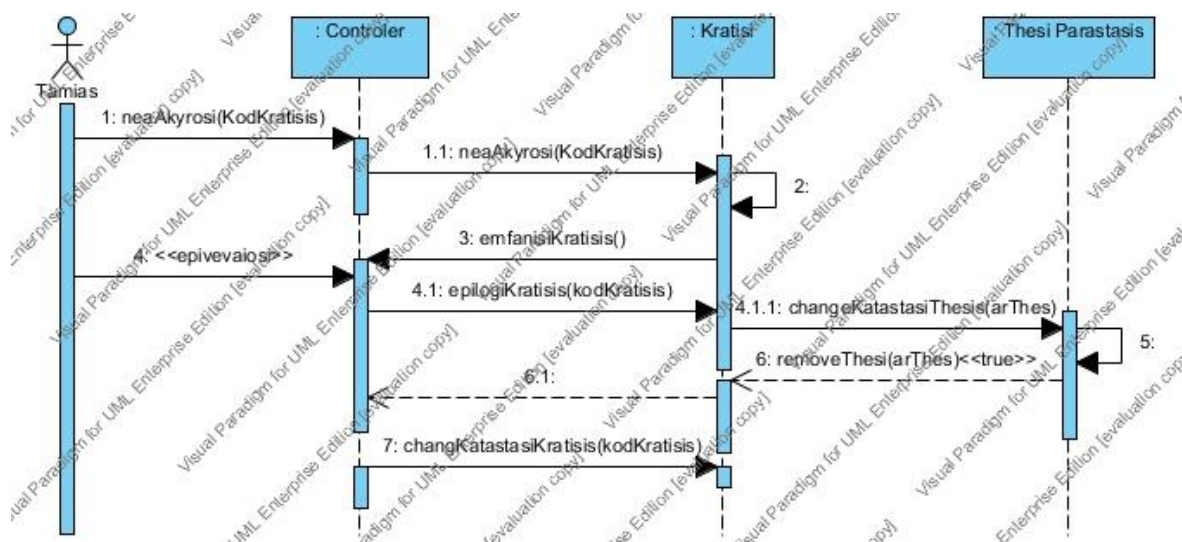
Sequence Diagram 30 :



Sequence Diagram 31 :

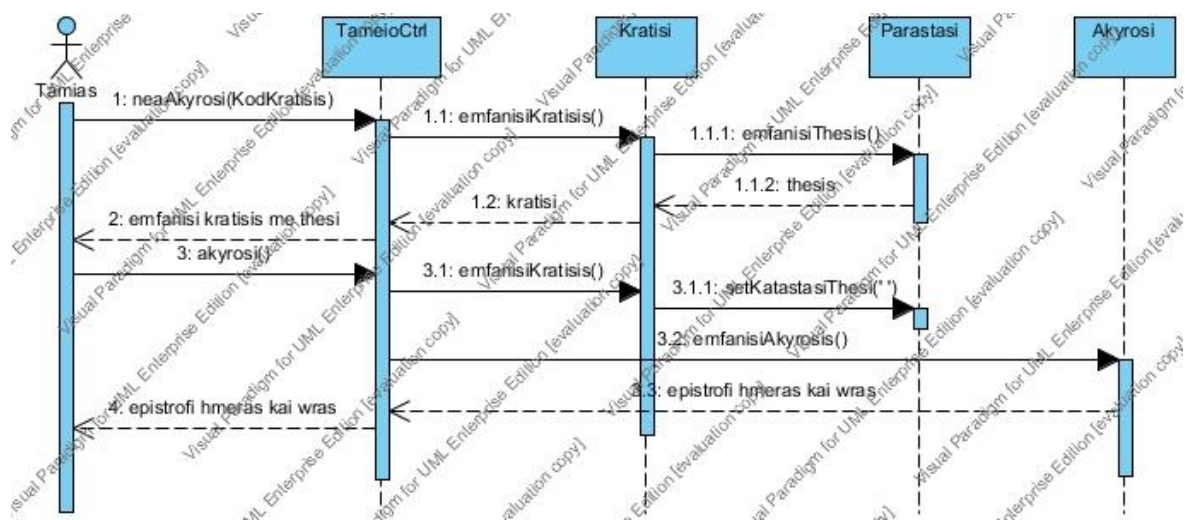


Sequence Diagram 32 :





Sequence Diagram 34 :



### Παραγόμενα Στατιστικά Στοιχεία :

	A	B	C	D	E	F	G	H
1	Κωδικός	Ομάδα	Δ. Ακολουθίας	Δ. Κλάσεων	Κώδικα	Messages	Actors	Lifelines
2	1	A	ΝΑΙ	ΝΑΙ	ΝΑΙ	6	1	2
3	2	A	ΝΑΙ	ΝΑΙ	ΝΑΙ	24	1	5
4	5	A	ΝΑΙ	ΝΑΙ	ΝΑΙ	6	1	3
5	13	A	ΝΑΙ	ΌΧΙ	ΝΑΙ	12	1	5
6	14	A	ΝΑΙ	ΝΑΙ	ΝΑΙ	9	1	4
7	16	A	ΝΑΙ	ΝΑΙ	ΝΑΙ	6	1	3
8	17	A	ΝΑΙ	ΝΑΙ	ΝΑΙ	11	1	4
9	19	A	ΝΑΙ	ΝΑΙ	ΝΑΙ	12	1	5
10	22	A	ΝΑΙ	ΝΑΙ	ΝΑΙ	4	1	2
11	25	A	ΝΑΙ	ΝΑΙ	ΝΑΙ	16	0	5
12	28	A	ΝΑΙ	ΝΑΙ	ΝΑΙ	9	1	4
13	33	A	ΝΑΙ	ΝΑΙ	ΝΑΙ	13	1	5
14	6	B	ΝΑΙ	ΝΑΙ	ΝΑΙ	6	1	3
15	8	B	ΝΑΙ	ΝΑΙ	ΝΑΙ	14	1	4
16	10	B	ΝΑΙ	ΝΑΙ	ΝΑΙ	12	1	5
17	11	B	ΝΑΙ	ΝΑΙ	ΝΑΙ	13	1	4
18	15	B	ΝΑΙ	ΝΑΙ	ΝΑΙ	12	1	5
19	18	B	ΝΑΙ	ΝΑΙ	ΝΑΙ	11	1	3
20	26	B	ΝΑΙ	ΝΑΙ	ΝΑΙ	6	1	3
21	27	B	ΝΑΙ	ΝΑΙ	ΝΑΙ	9	1	4
22	30	B	ΝΑΙ	ΝΑΙ	ΝΑΙ	13	1	4
23	31	B	ΝΑΙ	ΝΑΙ	ΝΑΙ	12	1	4
24	32	B	ΝΑΙ	ΝΑΙ	ΝΑΙ	11	1	3
25	34	B	ΝΑΙ	ΝΑΙ	ΝΑΙ	12	1	4

<b>Various Statistics Comparing A to B</b>	
Average Messages of Group A	10,66667
Average Messages of Group B	10,91667
Average Lifelines of Group A	3,916667
Average Lifelines of Group B	3,833333

### **Συμπεράσματα Πειράματος :**

Σύμφωνα με τον σκοπό του πειράματος τα αποτελέσματά που πήραμε ήταν τα εξής :

Η ομάδα Α' που δούλεψε με την μεθοδολογία της κάθετης ιχνηλασιμότητας είχε ένα Μέσο Όρο 10.67 απεσταλμένων μηνυμάτων, ενώ η Β' που δούλεψε με την μεθοδολογία της οριζόντιας ιχνηλασιμότητας είχε αντίστοιχα ένα Μέσο Όρο 10.92 απεσταλμένων μηνυμάτων. Άρα, βασιζόμενοι στους δύο Μέσους Όρους απεσταλμένων μηνυμάτων που είναι σχεδόν ίσοι, μπορούμε να συμπεράνουμε ότι οι φοιτητές κινήθηκαν με την ίδια ευκολία και στις δύο μεθοδολογίες και δεν παρουσίασαν σημαντικά προβλήματα στην μεθοδολογία της οριζόντιας ιχνηλασιμότητας. Για να είμαστε ακριβείς, επειδή δεν είναι ακριβώς ίσοι και ο Μέσος Όρος της Β' ομάδας είναι ελαφρώς μεγαλύτερος, βλέπουμε ότι όχι μόνο δεν αντιμετώπισαν πρόβλημα οι φοιτητές με την μεθοδολογία της οριζόντιας, αλλά τα πήγαν και καλύτερα από τους φοιτητές της Β' ομάδας.

Όσον αφορά τις γραμμές ζωής, η ομάδα Α' της κάθετης ιχνηλασιμότητας είχε ένα Μέσο Όρο 3.92 γραμμών ζωής, την ίδια στιγμή που η ομάδα Β' είχε έναν αντίστοιχο Μέσο Όρο 3.8 γραμμών ζωής. Άρα συμπεραίνουμε ότι τα διαγράμματα ακολουθίας των φοιτητών της Α' ομάδας που δούλεψαν με την μεθοδολογία της κάθετης ιχνηλασιμότητας είναι πληρέστερα κατά ένα μικρό βαθμό από τα αντίστοιχα της Β' ομάδας.

## **Επίλογος - Συμπεράσματα και πιθανή μελλοντική δουλειά .**

Μέσα από την πτυχιακή αυτή βγάλαμε τα εξής συμπεράσματα:

Όσον αφορά τους δύο τύπους της ιχνηλασιμότητας και με βάση το πείραμά μας, οι φοιτητές που δούλεψαν με την μεθοδολογία της οριζόντιας ιχνηλασιμότητας τα πήγαν ελαφρώς καλύτερα από τους φοιτητές που δούλεψαν με την μεθοδολογία της κάθετης ιχνηλασιμότητας, όσον αφορά τον αριθμό των απεσταλμένων μηνυμάτων μέσα σε ένα Διάγραμμα Ακολουθίας, αλλά τα πήγαν ελαφρώς χειρότερα στην δημιουργία γραμμών ζωής των Διαγραμμάτων Ακολουθίας, όσον αφορά όχι μόνο την ποσότητα, αλλά και την ποιότητα.

Όσον αφορά τις μετρικές που μπορούν να εφαρμοστούν , διαπιστώσαμε ότι οι μετρικές CBO και RFC μπορούν να εφαρμοστούν με επιτυχία για να μετρήσουν έννοιες στα πλαίσια της οριζόντιας ιχνηλασιμότητας(και δώσαμε συγκεκριμένα παραδείγματα πάνω σε Διαγράμματα Κλάσεων αλλά και Ακολουθίας).

Προτείνουμε και άλλες πιθανές μετρικές που θα μπορούσε μελλοντικά να γίνει έρευνα σε αυτές για να διαπιστωθεί αν μπορούν και αυτές με την σειρά τους να είναι χρήσιμες στην οριζόντια ιχνηλασιμότητα, αλλά και να ανακαλυφθούν ακόμη παραπάνω μετρικές για το συγκεκριμένο τομέα.

## BIBΛΙΟΓΡΑΦΙΑ

[1]: *Integrating Software Traceability for Change Impact Analysis*

Suhaimi Ibrahim<sup>1</sup>, Norbik Bashah Idris<sup>1</sup>, Malcolm Munro<sup>2</sup>, and Aziz Deraman<sup>3</sup>

1) Centre for Advanced Software Engineering, University of Technology Malaysia, Malaysia

2) Department of Computer Science, University of Durham, United Kingdom

3) Faculty of Technology and Information System, University Kebangsaan Malaysia, Malaysia

[2]: Syllabus REQB<sup>®</sup>

*Certified Professional for Requirements Engineering, Foundation Level, Version 1.2*

[3]: *Automated Traceability Analysis for UML Model Refinements*

Lionel C. Briand, Yvan Labiche, Tao Yue

Carleton University, Department SCE, Software Quality Engineering Laboratory

[4]: *Use case to source code traceability: The developer navigation view point*

Inah Omoronyia, Guttorm Sindre y Dept. of Computer and Information Science

Norwegian University of Science and Technology, Trondheim, Norway

Marc Roper, John Ferguson, Murray Wood z

Dept. of Computer and Information Sciences, University of Strathclyde, Glasgow, Scotland

[4]: *A DOCUMENT-BASED SOFTWARE TRACEABILITY TO SUPPORT CHANGE IMPACT ANALYSIS OF OBJECT-ORIENTED SOFTWARE*

SUHAIMI BIN IBRAHIM, UNIVERSITI TEKNOLOGI MALAYSIA.

[5]: *A Framework for Managing Consistency of Evolving UML Models*

Tom Mens, Université de Mons-Hainaut, Belgium

Ragnhild Van Der Straeten, Vrije Universiteit Brussel, Belgium

Jocelyn Simmonds, Universidad de Chile, Chile

[6]: *A SOFTWARE TRACEABILITY VALIDATION FOR CHANGE IMPACT ANALYSIS OF OBJECT ORIENTED SOFTWARE*

Suhaimi Ibrahim, Norbik Bashah Idris, Centre For Advanced Software Engineering,  
Universiti Teknologi Malaysia, Kuala Lumpur, Malaysia

Malcolm Munro, Department of Computer Science, University of Durham, United Kingdom

Aziz Deraman, Fac. of Technology & Infor. System, Universiti Kebangsaan Malaysia, Selangor,  
Malaysia

[7]: *A REQUIREMENTS TRACEABILITY TO SUPPORT CHANGE IMPACT ANALYSIS*

Suhaimi Ibrahim, Norbik Bashah Idris, Centre For Advanced Software Engineering,  
Universiti Teknologi Malaysia, Kuala Lumpur, Malaysia

Malcolm Munro, Department of Computer Science, University of Durham, United Kingdom

Aziz Deraman, Fac. of Technology & Infor. System, Universiti Kebangsaan Malaysia, Selangor,  
Malaysia

[8]: *<<UML>> 2003, Modeling Languages and Applications: Workshop on "Consistency Problems  
in UML-based Software Development II"*

October 20, 2003, San Francisco, USA

Editors: Ludwik Kuzniarz, Zbigniew Huzar, Gianna Reggio, Jean Louis Sourrouille, Mirosław Staron

[9]: *<<UML>> 2004, Modeling Languages and Applications: Third Workshop on Consistency  
Problems in UML-based Software Development: Understanding and Usage of Dependency  
Relationships*

Lisbon, Portugal, October 11, 2004

Editors: Zbigniew Huzar, Ludwik Kuzniarz, Gianna Reggio, Jean Louis Sourrouille

[10]: *Using Traceability for Reverse Instance Transformations with SiTra*

Seyyed Shah, Kyriakos Anastasakis, Behzad Bordbar

School of Computer Science

University of Birmingham, Edgbaston

[11]: *FOAL 2004 Proceedings Foundations of Aspect-Oriented Languages Workshop at AOSD  
2004*

Curtis Clifton, Ralf Lammel, and Gary T. Leavens (editors)

March 2004

[12]:*IMPLEMENTING A DOCUMENT-BASED REQUIREMENTS*

*TRACEABILITY: A CASE STUDY*

Suhaimi Ibrahim, Norbik Bashah Idris, Centre For Advanced Software Engineering,  
Universiti Teknologi Malaysia, Kuala Lumpur, Malaysia.

Malcolm Munro, Department of Computer Science, University of Durham, United Kingdom

Aziz Deraman, Fac. of Technology & Infor. System, Universiti Kebangsaan Malaysia, Selangor,  
Malaysia

[13]:*Towards a Generic Solution for Traceability in MDD*

Ståle Walderhaug, Ulrik Johansen, Erlend Stav, Jan Aagedal, SINTEF ICT, Norway

{stale.waldehaug | ulrik.johansen | erlend.stav | jan.aagedal}@sintef.no}

[14]:*Model traceability, 2006*

N. Aizenbud-Reshef, B. T. Nolan, J. Rubin, Y. Shaham-Gafni.

[15]:*On-Demand Merging of Traceability Links with Models*

Dimitrios S. Kolovos, Richard F. Paige and Fiona A.C. Polack

Department of Computer Science, The University of York, UK, York, YO10 5DD

{dkolovos, paige, fiona}@cs.york.ac.uk

[16]:*Reading Techniques for OO Design Inspections,*

Guilherme H. Travassos, Forrest Shull, Jeffrey Carver, Victor R. Basili

[17]:*Operational Semantics for Traceability*

Netta Aizenbud-Reshef<sup>1</sup>, Richard F. Paige<sup>2</sup>, Julia Rubin<sup>1</sup>, Yael Shaham-Gafni<sup>1</sup> and Dimitrios S.  
Kolovos<sup>2</sup>

<sup>1</sup>IBM Haifa Research Labs, Israel

<sup>2</sup> Department of Computer Science, University of York, UK

[18]:*Requirements Modelling and Traceability.*

Paul Luttikhuisen (ed.)

Partner: Philips, Siemens, Alcatel, Nokia, University of Essen, Fraunhofer IESE

Date: June 01, 2001

Number: Philips-WP3-0106-01

Level: ESAPS (Consortium-wide)

Topic: Task 3.1: Requirements Modelling and Traceability

[19]:Towards Querying of Traceability Information in the Context of Software Evolution

Hannes Schwarz,Juurgen Ebert,Volker Riediger,AndreasWinter

[20]:*Synthesis and Traceability of Scenario-based Executable Models*

Ankit Goel Abhik Roychoudhury,Department of Computer Science, National University of Singapore.

[21]:*Traceability Management through Use Cases when Developing Distributed Object Applications*

Nelly Bencomo<sup>1</sup>,Alfredo Matteo<sup>2</sup>

<sup>1</sup> Computing Department, Lancaster University,

Bailrigg, Lancaster, LA1 4YR, UK

<sup>2</sup> Laboratorio TOOLS, Escuela de Computación, Facultad de Ciencias, UCV.

[22]:*USING HYPERLINKS FOR TRACEABILITY IN SOFTWARE ENGINEERING*

By GEIR A. BJUNE.

MOS IN COMPUTER SCIENCE WASHINGTON STATE UNIVERSITY,School of Electrical Engineering and Computer Science,December 2000

[23]:*Supporting Modular Transformation Units with Precise Transformation Traceability Metadata*

Bert Vanhooff<sup>1</sup> and Yolande Berbers<sup>1</sup>

Katholieke Universiteit Leuven, Celestijnenlaan 200A, 3001 Leuven, Belgium

[23]*WapMetrics: a tool for computing UML Design Metrics forWeb Applications*

Emad Ghosheh,University of Westminster,Department of Information and Software Systems

Sue Black,University of Westminster Head of Department of Information and Software Systems,London HA1 3TP, UK

[24]:Title: Applying and Interpreting Object Oriented Metrics,Presenter: Dr. Linda H. Rosenberg

[25]:*Empirical Analysis of Object-Oriented Design Metrics: Towards a New Metric Using Control Flow Paths and Probabilities*

Mourad Badri, Linda Badri and Fadel Touré,

Software Engineering Research Laboratory, Department of Mathematics and Computer Science, University of Quebec at Trois-Rivieres, Québec, Canada.

[26]:*Measuring OO Design Metrics from UML*

Mei-Huei Tang and Mei-Hwa Chen

Computer Science Department,SUNY at Albany,Albany,USA

[27]:*Object Oriented Metrics Which predict Maintainability*

Wei Li and Sallie Henry,Blacksburg,Virginia,1993

[28]:*Towards Maintainability Metrics on Software Architecture: An Adaptation of Object-Oriented Metrics.*

PerOlof Bengtsson,University of Karlskrona/Ronneby,Department of Computer Science and Business Administration, Sweden

[29]:*Detecting Defects in Object Oriented Designs: Using Reading Techniques to Increase Software Quality.*

Guilherme H. Travassos<sup>1</sup>,o Forrest Shull Michael Fredericks Victor R. Basiliw

[30]:*Object-Oriented Reading Techniques for Inspection of UML Models – An Industrial Experiment*

Reidar Conradi<sup>1</sup>, Parastoo Mohagheghi<sup>2</sup>, Tayyaba Arif<sup>1</sup>, Lars Christian Hegde<sup>1</sup>, Geir Arne Bunde<sup>3</sup>, and Anders Pedersen<sup>3</sup>

<sup>1</sup> Department of Computer and Information Science, NTNU, NO-7491 Trondheim, Norway

[31]: T. Corbi. *Program understanding: challenge for the 1990s*. IBM Systems Journal, 28(2):294–306, 1989

[32]: N. Wilde and R. Huitt. Maintenance support for objectoriented programs. IEEE Transactions on Software Engineering,18(12):1038–1044, 1992

[33]:L. Briand and J. Wurst. Modeling development effort in object-oriented systems using design properties. IEEE Transactions on Software Engineering, 27(11):963–986,2001



- [34]:D. Card, K. El-Emam, and B. Scalzo. Measurement of object-oriented software development projects. Software Productivity Consortium NFP, 2001.
- [35]:F. Fioravanti and P. Nesi. Estimation and prediction metrics for adaptive maintenance effort of object-oriented systems[IEEE Transactions on Software Engineering, 27(12):1062–1084, 2001
- [36]:N. Fenton and N. Ohlsson. Quantitative analysis of faults and failures in a complex software system. IEEE Transactions on Software Engineering, 26(8):797–814, 2000
- [37]:K.-H. Moller and D. Paulish. An empirical investigation of software fault distribution. In Proceedings of the 4th International Software Metrics Symposium, pages 82–90. IEEE Computer Society Press, 1993
- [38]:K. EL-Emam. A methodology for validating software product metrics. Technical Report NRC 44142, National Research Council Canada, 2000.
- [39]:K. EL-Emam. A methodology for validating software product metrics. Technical Report NRC 44142, National Research Council Canada, 2000.
- [40]: Metrics for Requirements Engineering ,  
Mohammed Javeed Ali, June 15, 2006, Master’s Thesis,
- [41] [http://rup.hops-fp6.org/process/workflow/requirem/co\\_trace.htm](http://rup.hops-fp6.org/process/workflow/requirem/co_trace.htm)
- [42] Source : <http://www.tutorialspoint.com>

## ΠΑΡΑΡΤΗΜΑ Α'

### XMI NO.2

```

<XMI xmlns:UML="org.omg/UML/1.4" xmi.version
="1.2">
<XMI.header>
<XMI.documentation>
<XMI.exporter exportedFromDifferentName="fal
se">Visual Paradigm for UML</XMI.exporter>
<XMI.exporterVersion>7.0.0</XMI.exporterVers
ion>
</XMI.documentation>
<XMI.metamodel xmi.name="UML" xmi.version="1
.4"/>
</XMI.header>
<XMI.content>
<UML:Model name="peirama" xmi.id="c16oIVSGAq
ACBQDr">
<UML:Namespace.ownedElement>
<XMI.extension xmi.extender="Visual Paradigm
for UML">
<vpumlChildModels>
<vpumlModel id="H16oIVSGAqACBQFI" modelType=
"UseCasePropConfig">
<properties>
<property name="name" type="string" value=""
/>
<property name="modelType" type="string" val
ue="UseCasePropConfig"/>
<property htmlValue="" name="documentation"
type="htmlString" value=""/>
<property name="pmAuthor" type="string" valu
e="Chris"/>
<property name="pmCreateDateTime" type="stri
ng" value="1282116254456"/>
<property name="pmLastModified" type="string
" value="1282116883348"/>
</properties>
<vpumlChildModelRefs>
<modelRef value="QN6oIVSGAqACBQFS"/>
<modelRef value="QN6oIVSGAqACBQFb"/>
<modelRef value="wN6oIVSGAqACBQFk"/>
</vpumlChildModelRefs>
<vpumlChildModels>
<vpumlModel id="QN6oIVSGAqACBQFS" modelType=
"UseCaseRank">
<properties>
<property name="name" type="string" value="L
ow"/>
<property name="modelType" type="string" val
ue="UseCaseRank"/>
<property htmlValue="" name="documentation"
type="htmlString" value=""/>
<property name="pmAuthor" type="string" valu
e="Chris"/>
<property name="pmCreateDateTime" type="stri
ng" value="1282116254466"/>
<property name="pmLastModified" type="string
" value="1282116883351"/>
</properties>
</vpumlModel>

```

```

<vpumlModel id="QN6oIVSGAqACBQFb" modelType=
"UseCaseRank">
<properties>
<property name="name" type="string" value="M
edium"/>
<property name="modelType" type="string" val
ue="UseCaseRank"/>
<property htmlValue="" name="documentation"
type="htmlString" value=""/>
<property name="pmAuthor" type="string" valu
e="Chris"/>
<property name="pmCreateDateTime" type="stri
ng" value="1282116254466"/>
<property name="pmLastModified" type="string
" value="1282116883352"/>
</properties>
</vpumlModel>
<vpumlModel id="wN6oIVSGAqACBQFk" modelType=
"UseCaseRank">
<properties>
<property name="name" type="string" value="H
igh"/>
<property name="modelType" type="string" val
ue="UseCaseRank"/>
<property htmlValue="" name="documentation"
type="htmlString" value=""/>
<property name="pmAuthor" type="string" valu
e="Chris"/>
<property name="pmCreateDateTime" type="stri
ng" value="1282116254467"/>
<property name="pmLastModified" type="string
" value="1282116883352"/>
</properties>
</vpumlModel>
</vpumlChildModels>
</vpumlModel>
<vpumlModel id="qN6oIVSGAqACBQFu" modelType=
"WSDLContainer">
<properties>
<property name="name" type="string" value="W
SDLContainer"/>
<property name="modelType" type="string" val
ue="WSDLContainer"/>
</properties>
<vpumlChildModelRefs>
<modelRef value="5N6oIVSGAqACBQFw"/>
</vpumlChildModelRefs>
<vpumlChildModels>
<vpumlModel id="5N6oIVSGAqACBQFw" modelType=
"XSDSchema">
<properties>
<property name="name" type="string" value="h
ttp://www.w3.org/2001/XMLSchema"/>
<property name="modelType" type="string" val
ue="XSDSchema"/>
<property name="import" type="boolean" value
="false"/>
<property htmlValue="" name="documentation"
type="htmlString" value=""/>

```

```

<property name="pmAuthor" type="string" value="Chris"/>
<property name="pmCreateDateTime" type="string" value="1282116254503"/>
<property name="pmLastModified" type="string" value="1282116883352"/>
<property name="attributeFormDefault" type="int" value="0"/>
<property name="elementFormDefault" type="int" value="0"/>
<property name="targetNamespace" type="string" value="http://www.w3.org/2001/XMLSchema"/>
</properties>
<vpumlChildModelRefs>
<modelRef value="0t6oIVSGAqACBQF6"/>
<modelRef value="0t6oIVSGAqACBQGD"/>
<modelRef value="0t6oIVSGAqACBQGM"/>
<modelRef value="Mt6oIVSGAqACBQGV"/>
<modelRef value="Mt6oIVSGAqACBQGe"/>
<modelRef value="Mt6oIVSGAqACBQGn"/>
<modelRef value="Mt6oIVSGAqACBQGW"/>
<modelRef value="st6oIVSGAqACBQGS"/>
<modelRef value="st6oIVSGAqACBQHC"/>
<modelRef value="st6oIVSGAqACBQHL"/>
<modelRef value="st6oIVSGAqACBQHU"/>
<modelRef value="st6oIVSGAqACBQHd"/>
<modelRef value="st6oIVSGAqACBQHm"/>
<modelRef value="st6oIVSGAqACBQHv"/>
<modelRef value="st6oIVSGAqACBQH4"/>
<modelRef value="ct6oIVSGAqACBQIB"/>
<modelRef value="8t6oIVSGAqACBQIK"/>
<modelRef value="8t6oIVSGAqACBQIT"/>
<modelRef value="8t6oIVSGAqACBQIc"/>
<modelRef value="Ct6oIVSGAqACBQI1"/>
<modelRef value="Ct6oIVSGAqACBQIu"/>
<modelRef value="Ct6oIVSGAqACBQI3"/>
<modelRef value="it6oIVSGAqACBQJA"/>
<modelRef value="it6oIVSGAqACBQJJ"/>
<modelRef value="it6oIVSGAqACBQJS"/>
<modelRef value="St6oIVSGAqACBQJb"/>
<modelRef value="St6oIVSGAqACBQJk"/>
<modelRef value="St6oIVSGAqACBQJt"/>
<modelRef value="yt6oIVSGAqACBQJ2"/>
<modelRef value="yt6oIVSGAqACBQJ"/>
<modelRef value="yt6oIVSGAqACBQKI"/>
<modelRef value="yt6oIVSGAqACBQKR"/>
<modelRef value="Kt6oIVSGAqACBQKa"/>
<modelRef value="Kt6oIVSGAqACBQKj"/>
<modelRef value="Kt6oIVSGAqACBQKs"/>
<modelRef value="Kt6oIVSGAqACBQK1"/>
<modelRef value="qt6oIVSGAqACBQK."/>
<modelRef value="qt6oIVSGAqACBQLH"/>
<modelRef value="qt6oIVSGAqACBQLQ"/>
<modelRef value="at6oIVSGAqACBQLZ"/>
<modelRef value="at6oIVSGAqACBQLi"/>
<modelRef value="at6oIVSGAqACBQLr"/>
<modelRef value="6t6oIVSGAqACBQL0"/>
<modelRef value="6t6oIVSGAqACBQL9"/>
</vpumlChildModelRefs>
<vpumlChildModels>
<vpumlModel id="0t6oIVSGAqACBQF6" modelType="XSDSimpleType">
<properties>
<property name="name" type="string" value="ENTITIES"/>
<property name="modelType" type="string" value="XSDSimpleType"/>
</properties>
<vpumlModel id="0t6oIVSGAqACBQGD" modelType="XSDSimpleType">
<properties>
<property name="name" type="string" value="ENTITY"/>
<property name="modelType" type="string" value="XSDSimpleType"/>
<property name="import" type="boolean" value="false"/>
<property htmlValue="" name="documentation" type="htmlString" value=""/>
<property name="pmAuthor" type="string" value="Chris"/>
<property name="pmCreateDateTime" type="string" value="1282116254539"/>
<property name="pmLastModified" type="string" value="1282116883352"/>
</properties>
</vpumlModel>
<vpumlModel id="0t6oIVSGAqACBQGM" modelType="XSDSimpleType">
<properties>
<property name="name" type="string" value="ID"/>
<property name="modelType" type="string" value="XSDSimpleType"/>
<property name="import" type="boolean" value="false"/>
<property htmlValue="" name="documentation" type="htmlString" value=""/>
<property name="pmAuthor" type="string" value="Chris"/>
<property name="pmCreateDateTime" type="string" value="1282116254539"/>
<property name="pmLastModified" type="string" value="1282116883353"/>
</properties>
</vpumlModel>
<vpumlModel id="Mt6oIVSGAqACBQGV" modelType="XSDSimpleType">
<properties>
<property name="name" type="string" value="IDREF"/>
<property name="modelType" type="string" value="XSDSimpleType"/>
<property name="import" type="boolean" value="false"/>
<property htmlValue="" name="documentation" type="htmlString" value=""/>
<property name="pmAuthor" type="string" value="Chris"/>
<property name="pmCreateDateTime" type="string" value="1282116254540"/>
<property name="pmLastModified" type="string" value="1282116883353"/>
</properties>
</vpumlModel>

```













```

<vpumlModel id="at6oIVSGAqACBQLi" modelType=
"XSDSimpleType">
<properties>
<property name="name" type="string" value="u
nsignedByte"/>
<property name="modelType" type="string" val
ue="XSDSimpleType"/>
<property name="import" type="boolean" value
="false"/>
<property htmlValue="" name="documentation"
type="htmlString" value=""/>
<property name="pmAuthor" type="string" valu
e="Chris"/>
<property name="pmCreateDateTime" type="stri
ng" value="1282116254550"/>
<property name="pmLastModified" type="string
" value="1282116883354"/>
</properties>
</vpumlModel>
<vpumlModel id="at6oIVSGAqACBQLr" modelType=
"XSDSimpleType">
<properties>
<property name="name" type="string" value="u
nsignedInt"/>
<property name="modelType" type="string" val
ue="XSDSimpleType"/>
<property name="import" type="boolean" value
="false"/>
<property htmlValue="" name="documentation"
type="htmlString" value=""/>
<property name="pmAuthor" type="string" valu
e="Chris"/>
<property name="pmCreateDateTime" type="stri
ng" value="1282116254550"/>
<property name="pmLastModified" type="string
" value="1282116883354"/>
</properties>
</vpumlModel>
<vpumlModel id="6t6oIVSGAqACBQL0" modelType=
"XSDSimpleType">
<properties>
<property name="name" type="string" value="u
nsignedLong"/>
<property name="modelType" type="string" val
ue="XSDSimpleType"/>
<property name="import" type="boolean" value
="false"/>
<property htmlValue="" name="documentation"
type="htmlString" value=""/>
<property name="pmAuthor" type="string" valu
e="Chris"/>
<property name="pmCreateDateTime" type="stri
ng" value="1282116254551"/>
<property name="pmLastModified" type="string
" value="1282116883354"/>
</properties>
</vpumlModel>
<vpumlModel id="6t6oIVSGAqACBQL9" modelType=
"XSDSimpleType">
<properties>
<property name="name" type="string" value="u
nsignedShort"/>
<property name="modelType" type="string" val
ue="XSDSimpleType"/>
<property name="import" type="boolean" value
="false"/>
<property htmlValue="" name="documentation"
type="htmlString" value=""/>
<property name="pmAuthor" type="string" valu
e="Chris"/>
<property name="pmCreateDateTime" type="stri
ng" value="1282116254551"/>
<property name="pmLastModified" type="string
" value="1282116883355"/>
</properties>
</vpumlModel>
</vpumlChildModels>
</vpumlModel>
</vpumlChildModels>
</vpumlModel>
<vpumlModel id="dt6oIVSGAqACBQMH" modelType=
"PMStatusContainer">
<properties>
<property name="name" type="string" value=""
/>
<property name="modelType" type="string" val
ue="PMStatusContainer"/>
</properties>
<vpumlChildModelRefs>
<modelRef value="3t6oIVSGAqACBQMJ"/>
<modelRef value="Pt6oIVSGAqACBQMK"/>
<modelRef value="Pt6oIVSGAqACBQML"/>
<modelRef value="vt6oIVSGAqACBQMM"/>
<modelRef value="ft6oIVSGAqACBQMN"/>
<modelRef value="ft6oIVSGAqACBQMO"/>
<modelRef value="_t6oIVSGAqACBQMP"/>
</vpumlChildModelRefs>
<vpumlChildModels>
<vpumlModel id="3t6oIVSGAqACBQMJ" modelType=
"PMStatus">
<properties>
<property name="name" type="string" value="P
roposed"/>
<property name="modelType" type="string" val
ue="PMStatus"/>
</properties>
</vpumlModel>
<vpumlModel id="Pt6oIVSGAqACBQMK" modelType=
"PMStatus">
<properties>
<property name="name" type="string" value="A
pproved"/>
<property name="modelType" type="string" val
ue="PMStatus"/>
</properties>
</vpumlModel>
<vpumlModel id="Pt6oIVSGAqACBQML" modelType=
"PMStatus">
<properties>
<property name="name" type="string" value="P
lanned"/>
<property name="modelType" type="string" val
ue="PMStatus"/>
</properties>
</vpumlModel>
<vpumlModel id="vt6oIVSGAqACBQMM" modelType=
"PMStatus">
<properties>
<property name="name" type="string" value="D
esigned"/>
<property name="modelType" type="string" val
ue="PMStatus"/>
</properties>
</vpumlModel>
<vpumlModel id="ft6oIVSGAqACBQMN" modelType=
"PMStatus">
<properties>

```

```

<property name="name" type="string" value="I
mplemented"/>
<property name="modelType" type="string" val
ue="PMStatus"/>
</properties>
</vpumlModel>
<vpumlModel id="ft6oIVSGAqACBQMO" modelType=
"PMStatus">
<properties>
<property name="name" type="string" value="T
ested"/>
<property name="modelType" type="string" val
ue="PMStatus"/>
</properties>
</vpumlModel>
<vpumlModel id="_t6oIVSGAqACBQMP" modelType=
"PMStatus">
<properties>
<property name="name" type="string" value="D
eployed"/>
<property name="modelType" type="string" val
ue="PMStatus"/>
</properties>
</vpumlModel>
</vpumlChildModels>
</vpumlModel>
<vpumlModel id="Bd6oIVSGAqACBQMR" modelType=
"PMDifficultyContainer">
<properties>
<property name="name" type="string" value=""
/>
<property name="modelType" type="string" val
ue="PMDifficultyContainer"/>
</properties>
<vpumlChildModelRefs>
<modelRef value="1d6oIVSGAqACBQMT"/>
<modelRef value="Nd6oIVSGAqACBQMU"/>
<modelRef value="Nd6oIVSGAqACBQMV"/>
</vpumlChildModelRefs>
<vpumlChildModels>
<vpumlModel id="1d6oIVSGAqACBQMT" modelType=
"PMDifficulty">
<properties>
<property name="name" type="string" value="E
asy"/>
<property name="modelType" type="string" val
ue="PMDifficulty"/>
</properties>
</vpumlModel>
<vpumlModel id="Nd6oIVSGAqACBQMU" modelType=
"PMDifficulty">
<properties>
<property name="name" type="string" value="M
edium"/>
<property name="modelType" type="string" val
ue="PMDifficulty"/>
</properties>
</vpumlModel>
<vpumlModel id="Nd6oIVSGAqACBQMV" modelType=
"PMDifficulty">
<properties>
<property name="name" type="string" value="D
ifficult"/>
<property name="modelType" type="string" val
ue="PMDifficulty"/>
</properties>
</vpumlModel>
</vpumlChildModels>
</vpumlModel>
<vpumlModel id="zd6oIVSGAqACBQMX" modelType=
"PMPriorityContainer">
<properties>
<property name="name" type="string" value=""
/>
<property name="modelType" type="string" val
ue="PMPriorityContainer"/>
</properties>
<vpumlChildModelRefs>
<modelRef value="Hd6oIVSGAqACBQMZ"/>
<modelRef value="Hd6oIVSGAqACBQMa"/>
<modelRef value="Hd6oIVSGAqACBQMb"/>
</vpumlChildModelRefs>
<vpumlChildModels>
<vpumlModel id="Hd6oIVSGAqACBQMZ" modelType=
"PMPriority">
<properties>
<property name="name" type="string" value="L
ow"/>
<property name="modelType" type="string" val
ue="PMPriority"/>
</properties>
</vpumlModel>
<vpumlModel id="Hd6oIVSGAqACBQMa" modelType=
"PMPriority">
<properties>
<property name="name" type="string" value="M
edium"/>
<property name="modelType" type="string" val
ue="PMPriority"/>
</properties>
</vpumlModel>
<vpumlModel id="Hd6oIVSGAqACBQMb" modelType=
"PMPriority">
<properties>
<property name="name" type="string" value="H
igh"/>
<property name="modelType" type="string" val
ue="PMPriority"/>
</properties>
</vpumlModel>
</vpumlChildModels>
</vpumlModel>
<vpumlModel id="U96oIVSGAqACBQMd" modelType=
"PMVersionContainer">
<properties>
<property name="name" type="string" value=""
/>
<property name="modelType" type="string" val
ue="PMVersionContainer"/>
</properties>
<vpumlChildModelRefs>
<modelRef value="896oIVSGAqACBQMf"/>
<modelRef value="C96oIVSGAqACBQMg"/>
</vpumlChildModelRefs>
<vpumlChildModels>
<vpumlModel id="896oIVSGAqACBQMf" modelType=
"PMVersion">
<properties>
<property name="name" type="string" value="1
.0"/>
<property name="modelType" type="string" val
ue="PMVersion"/>
</properties>
</vpumlModel>
<vpumlModel id="C96oIVSGAqACBQMg" modelType=
"PMVersion">
<properties>

```

```

<property name="name" type="string" value="2
.0"/>
<property name="modelType" type="string" val
ue="PMVersion"/>
</properties>
</vpumlModel>
</vpumlChildModels>
</vpumlModel>
<vpumlModel id="q96oIVSGAqACBQMi" modelType=
"PMIterationContainer">
<properties>
<property name="name" type="string" value=""
/>
<property name="modelType" type="string" val
ue="PMIterationContainer"/>
</properties>
<vpumlChildModelRefs>
<modelRef value="e96oIVSGAqACBQMk"/>
<modelRef value="e96oIVSGAqACBQMl"/>
</vpumlChildModelRefs>
<vpumlChildModels>
<vpumlModel id="e96oIVSGAqACBQMk" modelType=
"PMIteration">
<properties>
<property name="name" type="string" value="I
teration 1"/>
<property name="modelType" type="string" val
ue="PMIteration"/>
</properties>
</vpumlModel>
<vpumlModel id="e96oIVSGAqACBQMl" modelType=
"PMIteration">
<properties>
<property name="name" type="string" value="I
teration 2"/>
<property name="modelType" type="string" val
ue="PMIteration"/>
</properties>
</vpumlModel>
</vpumlChildModels>
</vpumlModel>
<vpumlModel id="Z96oIVSGAqACBQMn" modelType=
"PMPhaseContainer">
<properties>
<property name="name" type="string" value=""
/>
<property name="modelType" type="string" val
ue="PMPhaseContainer"/>
</properties>
<vpumlChildModelRefs>
<modelRef value="d96oIVSGAqACBQMp"/>
<modelRef value="996oIVSGAqACBQMq"/>
<modelRef value="996oIVSGAqACBQMr"/>
<modelRef value="996oIVSGAqACBQMs"/>
</vpumlChildModelRefs>
<vpumlChildModels>
<vpumlModel id="d96oIVSGAqACBQMp" modelType=
"PMPhase">
<properties>
<property name="name" type="string" value="I
nception"/>
<property name="modelType" type="string" val
ue="PMPhase"/>
</properties>
</vpumlModel>
<vpumlModel id="996oIVSGAqACBQMq" modelType=
"PMPhase">
<properties>
<property name="name" type="string" value="E
laboration"/>
<property name="modelType" type="string" val
ue="PMPhase"/>
</properties>
</vpumlModel>
<vpumlModel id="996oIVSGAqACBQMr" modelType=
"PMPhase">
<properties>
<property name="name" type="string" value="C
onstruction"/>
<property name="modelType" type="string" val
ue="PMPhase"/>
</properties>
</vpumlModel>
<vpumlModel id="996oIVSGAqACBQMs" modelType=
"PMPhase">
<properties>
<property name="name" type="string" value="T
ransition"/>
<property name="modelType" type="string" val
ue="PMPhase"/>
</properties>
</vpumlModel>
</vpumlChildModels>
</vpumlModel>
<vpumlModel id="j96oIVSGAqACBQMu" modelType=
"PMDisciplineContainer">
<properties>
<property name="name" type="string" value=""
/>
<property name="modelType" type="string" val
ue="PMDisciplineContainer"/>
</properties>
<vpumlChildModelRefs>
<modelRef value="v96oIVSGAqACBQMw"/>
<modelRef value="v96oIVSGAqACBQMx"/>
<modelRef value="f96oIVSGAqACBQMy"/>
<modelRef value="f96oIVSGAqACBQMz"/>
<modelRef value="f96oIVSGAqACBQMO"/>
<modelRef value="f96oIVSGAqACBQMl"/>
</vpumlChildModelRefs>
<vpumlChildModels>
<vpumlModel id="v96oIVSGAqACBQMw" modelType=
"PMDiscipline">
<properties>
<property name="name" type="string" value="B
usiness Modeling"/>
<property name="modelType" type="string" val
ue="PMDiscipline"/>
</properties>
</vpumlModel>
<vpumlModel id="v96oIVSGAqACBQMx" modelType=
"PMDiscipline">
<properties>
<property name="name" type="string" value="R
equirements"/>
<property name="modelType" type="string" val
ue="PMDiscipline"/>
</properties>
</vpumlModel>
<vpumlModel id="f96oIVSGAqACBQMy" modelType=
"PMDiscipline">
<properties>
<property name="name" type="string" value="A
nalysis & Design"/>
<property name="modelType" type="string" val
ue="PMDiscipline"/>
</properties>

```

```

</vpumlModel>
<vpumlModel id="f96oIVSGAqACBQMz" modelType="
PMDiscipline">
<properties>
<property name="name" type="string" value="I
mplementation"/>
<property name="modelType" type="string" val
ue="PMDiscipline"/>
</properties>
</vpumlModel>
<vpumlModel id="f96oIVSGAqACBQM0" modelType="
PMDiscipline">
<properties>
<property name="name" type="string" value="T
esting"/>
<property name="modelType" type="string" val
ue="PMDiscipline"/>
</properties>
</vpumlModel>
<vpumlModel id="f96oIVSGAqACBQM1" modelType="
PMDiscipline">
<properties>
<property name="name" type="string" value="Q
A & Maintenance"/>
<property name="modelType" type="string" val
ue="PMDiscipline"/>
</properties>
</vpumlModel>
</vpumlChildModels>
</vpumlModel>
<vpumlModel id="0D6oIVSGAqACBQM3" modelType="
PMAuthorContainer">
<properties>
<property name="name" type="string" value=""
/>
<property name="modelType" type="string" val
ue="PMAuthorContainer"/>
<property modelRef="0D6oIVSGAqACBQM4" name="
defaultValue" type="modelRef"/>
</properties>
<vpumlChildModelRefs>
<modelRef value="0D6oIVSGAqACBQM4"/>
</vpumlChildModelRefs>
<vpumlChildModels>
<argument name="Chris" xmi.id="0D6oIVSGAqACB
QM4">
<XMI.extension xmi.extender="Visual Paradigm
for UML"/>
</argument>
</vpumlChildModels>
</vpumlModel>
<vpumlModel id="hD6oIVSGAqACBQNC" modelType="
BRKeywords">
<properties>
<property name="name" type="string" value=""
/>
<property name="modelType" type="string" val
ue="BRKeywords"/>
<property htmlValue="" name="documentation"
type="htmlString" value=""/>
<property name="pmAuthor" type="string" valu
e="Chris"/>
<property name="pmCreateDateTime" type="stri
ng" value="1282116254753"/>
<property name="pmLastModified" type="string
" value="1282116883364"/>
</properties>
<vpumlChildModelRefs>
<modelRef value="TD6oIVSGAqACBQNM"/>
<modelRef value="zD6oIVSGAqACBQNV"/>
<modelRef value="zD6oIVSGAqACBQNe"/>
<modelRef value="LD6oIVSGAqACBQnN"/>
<modelRef value="LD6oIVSGAqACBQnW"/>
<modelRef value="LD6oIVSGAqACBQn5"/>
<modelRef value="rD6oIVSGAqACBQOC"/>
<modelRef value="rD6oIVSGAqACBQOL"/>
<modelRef value="bD6oIVSGAqACBQOU"/>
<modelRef value="bD6oIVSGAqACBQOd"/>
</vpumlChildModelRefs>
<vpumlChildModels>
<vpumlModel id="TD6oIVSGAqACBQNM" modelType="
BRKeyword">
<properties>
<property name="name" type="string" value="m
ust"/>
<property name="modelType" type="string" val
ue="BRKeyword"/>
<property htmlValue="" name="documentation"
type="htmlString" value=""/>
<property name="pmAuthor" type="string" valu
e="Chris"/>
<property name="pmCreateDateTime" type="stri
ng" value="1282116254770"/>
<property name="pmLastModified" type="string
" value="1282116883365"/>
</properties>
</vpumlModel>
<vpumlModel id="zD6oIVSGAqACBQNV" modelType="
BRKeyword">
<properties>
<property name="name" type="string" value="m
ust be"/>
<property name="modelType" type="string" val
ue="BRKeyword"/>
<property htmlValue="" name="documentation"
type="htmlString" value=""/>
<property name="pmAuthor" type="string" valu
e="Chris"/>
<property name="pmCreateDateTime" type="stri
ng" value="1282116254771"/>
<property name="pmLastModified" type="string
" value="1282116883365"/>
</properties>
</vpumlModel>
<vpumlModel id="zD6oIVSGAqACBQNe" modelType="
BRKeyword">
<properties>
<property name="name" type="string" value="i
f"/>
<property name="modelType" type="string" val
ue="BRKeyword"/>
<property htmlValue="" name="documentation"
type="htmlString" value=""/>
<property name="pmAuthor" type="string" valu
e="Chris"/>
<property name="pmCreateDateTime" type="stri
ng" value="1282116254771"/>
<property name="pmLastModified" type="string
" value="1282116883365"/>
</properties>
</vpumlModel>
<vpumlModel id="LD6oIVSGAqACBQnN" modelType="
BRKeyword">
<properties>
<property name="name" type="string" value="w
hen"/>
<property name="modelType" type="string" val
ue="BRKeyword"/>

```

```

<property htmlValue="" name="documentation"
type="htmlString" value=""/>
<property name="pmAuthor" type="string" value="Chris"/>
<property name="pmCreateDateTime" type="string" value="1282116254772"/>
<property name="pmLastModified" type="string" value="1282116883365"/>
</properties>
</vpumlModel>
<vpumlModel id="LD6oIVSGAqACBQNw" modelType="BRKeyword">
<properties>
<property name="name" type="string" value="must not"/>
<property name="modelType" type="string" value="BRKeyword"/>
<property htmlValue="" name="documentation" type="htmlString" value=""/>
<property name="pmAuthor" type="string" value="Chris"/>
<property name="pmCreateDateTime" type="string" value="1282116254772"/>
<property name="pmLastModified" type="string" value="1282116883366"/>
</properties>
</vpumlModel>
<vpumlModel id="LD6oIVSGAqACBQN5" modelType="BRKeyword">
<properties>
<property name="name" type="string" value="only"/>
<property name="modelType" type="string" value="BRKeyword"/>
<property htmlValue="" name="documentation" type="htmlString" value=""/>
<property name="pmAuthor" type="string" value="Chris"/>
<property name="pmCreateDateTime" type="string" value="1282116254772"/>
<property name="pmLastModified" type="string" value="1282116883366"/>
</properties>
</vpumlModel>
<vpumlModel id="rD6oIVSGAqACBQOC" modelType="BRKeyword">
<properties>
<property name="name" type="string" value="may"/>
<property name="modelType" type="string" value="BRKeyword"/>
<property htmlValue="" name="documentation" type="htmlString" value=""/>
<property name="pmAuthor" type="string" value="Chris"/>
<property name="pmCreateDateTime" type="string" value="1282116254773"/>
<property name="pmLastModified" type="string" value="1282116883366"/>
</properties>
</vpumlModel>
<vpumlModel id="rD6oIVSGAqACBQOL" modelType="BRKeyword">
<properties>
<property name="name" type="string" value="may not"/>
<property name="modelType" type="string" value="BRKeyword"/>
<property htmlValue="" name="documentation" type="htmlString" value=""/>
<property name="pmAuthor" type="string" value="Chris"/>
<property name="pmCreateDateTime" type="string" value="1282116254773"/>
<property name="pmLastModified" type="string" value="1282116883366"/>
</properties>
</vpumlModel>
<vpumlModel id="bD6oIVSGAqACBQOU" modelType="BRKeyword">
<properties>
<property name="name" type="string" value="needed"/>
<property name="modelType" type="string" value="BRKeyword"/>
<property htmlValue="" name="documentation" type="htmlString" value=""/>
<property name="pmAuthor" type="string" value="Chris"/>
<property name="pmCreateDateTime" type="string" value="1282116254773"/>
<property name="pmLastModified" type="string" value="1282116883366"/>
</properties>
</vpumlModel>
<vpumlModel id="bD6oIVSGAqACBQOd" modelType="BRKeyword">
<properties>
<property name="name" type="string" value="needed not"/>
<property name="modelType" type="string" value="BRKeyword"/>
<property htmlValue="" name="documentation" type="htmlString" value=""/>
<property name="pmAuthor" type="string" value="Chris"/>
<property name="pmCreateDateTime" type="string" value="1282116254774"/>
<property name="pmLastModified" type="string" value="1282116883367"/>
</properties>
</vpumlModel>
</vpumlChildModels>
</vpumlModel>
<vpumlModel id="Ecj4IVSGAqACBR33" modelType="ModelRelationshipContainer">
<properties>
<property name="name" type="string" value="relationships"/>
<property name="modelType" type="string" value="ModelRelationshipContainer"/>
<property htmlValue="" name="documentation" type="htmlString" value=""/>
<property name="pmAuthor" type="string" value="Chris"/>
<property name="pmCreateDateTime" type="string" value="1282116883336"/>
<property name="pmLastModified" type="string" value="1282116883391"/>
</properties>
<vpumlChildModelRefs>
<modelRef value="kcj4IVSGAqACBR4A"/>
<modelRef value="Ocj4IVSGAqACBR4J"/>
</vpumlChildModelRefs>
<vpumlChildModels>
<vpumlModel id="kcj4IVSGAqACBR4A" modelType="ModelRelationshipContainer">

```

```

<properties>
<property name="name" type="string" value="Message"/>
<property name="modelType" type="string" value="ModelRelationshipContainer"/>
<property htmlValue="" name="documentation" type="htmlString" value=""/>
<property name="pmAuthor" type="string" value="Chris"/>
<property name="pmCreateDateTime" type="string" value="1282116883337"/>
<property name="pmLastModified" type="string" value="1282306649534"/>
</properties>
<vpumlChildModelRefs>
<modelRef value="91UUIVSGAqACBSUh"/>
<modelRef value="h72UIVSGAqACBSZU"/>
<modelRef value="GmHUIVSGAqACBSjF"/>
<modelRef value="iSA0IVSGAqACBSl7"/>
<modelRef value="Okk0IVSGAqACBSpx"/>
<modelRef value="jnM0IVSGAqACBSsq"/>
<modelRef value="OcS0IVSGAqACBSv6"/>
<modelRef value="ph60IVSGAqACBSyH"/>
<modelRef value="Wau0IVSGAqACBSln"/>
<modelRef value="WBB0IVSGAqACBS4f"/>
<modelRef value="QQV0IVSGAqACBS8P"/>
<modelRef value="wKLMIVSGAqACBT00"/>
<modelRef value="bB.sIVSGAqACBT_v"/>
<modelRef value="SIjsIVSGAqACBUa9"/>
<modelRef value="TrsIVSGAqACBUiJ"/>
<modelRef value="p9fsIVSGAqACBUmM"/>
<modelRef value="Ga4cIVSGAqACBUuI"/>
<modelRef value="HB8cIVSGAqACBUyx"/>
<modelRef value="P4RcIVSGAqACBU2l"/>
<modelRef value="TBZcIVSGAqACBU5j"/>
<modelRef value="a1VcIVSGAqACBU_w"/>
<modelRef value="jUdcIVSGAqACBVc2"/>
<modelRef value="TGvcIVSGAqACBVJc"/>
<modelRef value="OpI8IVSGAqACBVNA"/>
</vpumlChildModelRefs>
</vpumlModel>
<vpumlModel id="0cj4IVSGAqACBR4J" modelType="ModelRelationshipContainer">
<properties>
<property name="name" type="string" value="InteractionDiagramDurationConstraint"/>
<property name="modelType" type="string" value="ModelRelationshipContainer"/>
<property htmlValue="" name="documentation" type="htmlString" value=""/>
<property name="pmAuthor" type="string" value="Chris"/>
<property name="pmCreateDateTime" type="string" value="1282116883339"/>
<property name="pmLastModified" type="string" value="1282120783350"/>
</properties>
</vpumlModel>
</vpumlChildModels>
</vpumlModel>
<vpumlModel id="srjYXlSGAqACBQPT" modelType="TaggedValueDefinition">
<properties>
<property name="name" type="string" value="Complexity"/>
<property name="modelType" type="string" value="TaggedValueDefinition"/>
<property htmlValue="" name="documentation" type="htmlString" value=""/>
<property name="pmAuthor" type="string" value="Chris"/>
<property name="pmCreateDateTime" type="string" value="1281948851534"/>
<property name="pmLastModified" type="string" value="1282116883368"/>
<property name="defaultValue" type="string" value=""/>
<property name="defaultHTMLValue" type="string" value=""/>
<property name="type" type="int" value="2"/>
<property modelRef="QrjYXlSGAqACBQO1" name="tagDefStereotype" type="modelRef"/>
<property name="enumerationValues" type="models">
<vpumlModels>
<argument name="" xmi.id="8rjYXlSGAqACBQQA">
<XMI.extension xmi.extender="Visual Paradigm for UML">
<type value="Name Only"/>
</XMI.extension>
</argument>

```

```

<argument name="" xmi.id="8rjYXlSGAqACBQQJ">
<XMI.extension xmi.extender="Visual Paradigm
for UML">
<type value="Initial"/>
</XMI.extension>
</argument>
<argument name="" xmi.id="8rjYXlSGAqACBQQS">
<XMI.extension xmi.extender="Visual Paradigm
for UML">
<type value="Base"/>
</XMI.extension>
</argument>
<argument name="" xmi.id="CrjYXlSGAqACBQQb">
<XMI.extension xmi.extender="Visual Paradigm
for UML">
<type value="Complete"/>
</XMI.extension>
</argument>
<argument name="" xmi.id="CrjYXlSGAqACBQQk">
<XMI.extension xmi.extender="Visual Paradigm
for UML">
<type value="Deferred"/>
</XMI.extension>
</argument>
</vpumlModels>
</property>
</properties>
</vpumlModel>
<vpumlModel id="CrjYXlSGAqACBQQt" modelType=
"TaggedValueDefinition">
<properties>
<property name="name" type="string" value="I
mplementation Status"/>
<property name="modelType" type="string" val
ue="TaggedValueDefinition"/>
<property htmlValue="" name="documentation"
type="htmlString" value=""/>
<property name="pmAuthor" type="string" valu
e="Chris"/>
<property name="pmCreateDateTime" type="stri
ng" value="1281948851536"/>
<property name="pmLastModified" type="string
" value="1282116883368"/>
<property name="defaultValue" type="string"
value=""/>
<property name="defaultHTMLValue" type="stri
ng" value=""/>
<property name="type" type="int" value="2"/>
<property modelRef="QrjYXlSGAqACBQ01" name="
tagDefStereotype" type="modelRef"/>
<property name="enumerationValues" type="mod
els">
<vpumlModels>
<argument name="" xmi.id="CrjYXlSGAqACBQQ2">
<XMI.extension xmi.extender="Visual Paradigm
for UML">
<type value="Scheduled"/>
</XMI.extension>
</argument>
<argument name="" xmi.id="irjYXlSGAqACBQQ_ ">
<XMI.extension xmi.extender="Visual Paradigm
for UML">
<type value="Started"/>
</XMI.extension>
</argument>
<argument name="" xmi.id="irjYXlSGAqACBQRI">
<XMI.extension xmi.extender="Visual Paradigm
for UML">
<type value="Partially Complete"/>
</XMI.extension>
</argument>
<argument name="" xmi.id="irjYXlSGAqACBQRR">
<XMI.extension xmi.extender="Visual Paradigm
for UML">
<type value="Complete"/>
</XMI.extension>
</argument>
<argument name="" xmi.id="irjYXlSGAqACBQRa">
<XMI.extension xmi.extender="Visual Paradigm
for UML">
<type value="Partially Deferred"/>
</XMI.extension>
</argument>
</vpumlModels>
</property>
</properties>
</vpumlModel>
<vpumlModel id="SrjYXlSGAqACBQRj" modelType=
"TaggedValueDefinition">
<properties>
<property name="name" type="string" value="P
reconditions"/>
<property name="modelType" type="string" val
ue="TaggedValueDefinition"/>
<property htmlValue="" name="documentation"
type="htmlString" value=""/>
<property name="pmAuthor" type="string" valu
e="Chris"/>
<property name="pmCreateDateTime" type="stri
ng" value="1281948851538"/>
<property name="pmLastModified" type="string
" value="1282116883369"/>
<property name="defaultValue" type="string"
value=""/>
<property name="defaultHTMLValue" type="stri
ng" value=""/>
<property name="type" type="int" value="4"/>
<property modelRef="QrjYXlSGAqACBQ01" name="
tagDefStereotype" type="modelRef"/>
</properties>
</vpumlModel>
<vpumlModel id="SrjYXlSGAqACBQRs" modelType=
"TaggedValueDefinition">
<properties>
<property name="name" type="string" value="P
ost-conditions"/>
<property name="modelType" type="string" val
ue="TaggedValueDefinition"/>
<property htmlValue="" name="documentation"
type="htmlString" value=""/>
<property name="pmAuthor" type="string" valu
e="Chris"/>
<property name="pmCreateDateTime" type="stri
ng" value="1281948851538"/>
<property name="pmLastModified" type="string
" value="1282116883369"/>
<property name="defaultValue" type="string"
value=""/>
<property name="defaultHTMLValue" type="stri
ng" value=""/>
<property name="type" type="int" value="4"/>
<property modelRef="QrjYXlSGAqACBQ01" name="
tagDefStereotype" type="modelRef"/>
</properties>
</vpumlModel>
<vpumlModel id="yrjYXlSGAqACBQR1" modelType=
"TaggedValueDefinition">
<properties>

```

```

<property name="name" type="string" value="A
uthor"/>
<property name="modelType" type="string" val
ue="TaggedValueDefinition"/>
<property htmlValue="" name="documentation"
type="htmlString" value=""/>
<property name="pmAuthor" type="string" valu
e="Chris"/>
<property name="pmCreateDateTime" type="stri
ng" value="1281948851539"/>
<property name="pmLastModified" type="string
" value="1282116883369"/>
<property name="defaultValue" type="string"
value=""/>
<property name="defaultHTMLValue" type="stri
ng" value=""/>
<property name="type" type="int" value="0"/>
<property modelRef="QrjYXlSGAqACBQ01" name="
tagDefStereotype" type="modelRef"/>
</properties>
</vpumlModel>
<vpumlModel id="yrjYXlSGAqACBQR." modelType=
"TaggedValueDefinition">
<properties>
<property name="name" type="string" value="A
ssumptions"/>
<property name="modelType" type="string" val
ue="TaggedValueDefinition"/>
<property htmlValue="" name="documentation"
type="htmlString" value=""/>
<property name="pmAuthor" type="string" valu
e="Chris"/>
<property name="pmCreateDateTime" type="stri
ng" value="1281948851539"/>
<property name="pmLastModified" type="string
" value="1282116883369"/>
<property name="defaultValue" type="string"
value=""/>
<property name="defaultHTMLValue" type="stri
ng" value=""/>
<property name="type" type="int" value="4"/>
<property modelRef="QrjYXlSGAqACBQ01" name="
tagDefStereotype" type="modelRef"/>
</properties>
</vpumlModel>
<vpumlModel id="0IDYXlSGAqACBQAZ" modelType=
"TaggedValueDefinition">
<properties>
<property name="name" type="string" value="I
D"/>
<property name="modelType" type="string" val
ue="TaggedValueDefinition"/>
<property htmlValue="" name="documentation"
type="htmlString" value=""/>
<property name="pmAuthor" type="string" valu
e="Chris"/>
<property name="pmCreateDateTime" type="stri
ng" value="1281948844299"/>
<property name="pmLastModified" type="string
" value="1282116883371"/>
<property name="defaultValue" type="string"
value=""/>
<property name="defaultHTMLValue" type="stri
ng" value=""/>
<property name="type" type="int" value="0"/>
<property modelRef="IF9YXlSGAqACBQAE" name="
tagDefStereotype" type="modelRef"/>
</properties>
</vpumlModel>
<vpumlModel id="MIDYXlSGAqACBQAI" modelType=
"TaggedValueDefinition">
<properties>
<property name="name" type="string" value="s
ource"/>
<property name="modelType" type="string" val
ue="TaggedValueDefinition"/>
<property htmlValue="" name="documentation"
type="htmlString" value=""/>
<property name="pmAuthor" type="string" valu
e="Chris"/>
<property name="pmCreateDateTime" type="stri
ng" value="1281948844300"/>
<property name="pmLastModified" type="string
" value="1282116883371"/>
<property name="defaultValue" type="string"
value=""/>
<property name="defaultHTMLValue" type="stri
ng" value=""/>
<property name="type" type="int" value="0"/>
<property modelRef="IF9YXlSGAqACBQAE" name="
tagDefStereotype" type="modelRef"/>
</properties>
</vpumlModel>
<vpumlModel id="MIDYXlSGAqACBQAr" modelType=
"TaggedValueDefinition">
<properties>
<property name="name" type="string" value="k
ind"/>
<property name="modelType" type="string" val
ue="TaggedValueDefinition"/>
<property htmlValue="" name="documentation"
type="htmlString" value=""/>
<property name="pmAuthor" type="string" valu
e="Chris"/>
<property name="pmCreateDateTime" type="stri
ng" value="1281948844301"/>
<property name="pmLastModified" type="string
" value="1282116883372"/>
<property name="defaultValue" type="string"
value=""/>
<property name="defaultHTMLValue" type="stri
ng" value=""/>
<property name="type" type="int" value="2"/>
<property modelRef="IF9YXlSGAqACBQAE" name="
tagDefStereotype" type="modelRef"/>
<property name="enumerationValues" type="mod
els">
<vpumlModels>
<argument name="" xmi.id="gYDYXlSGAqACBQA1">
<XMI.extension xmi.extender="Visual Paradigm
for UML">
<type value="Functional"/>
</XMI.extension>
</argument>
<argument name="" xmi.id="_YDYXlSGAqACBQA.">
<XMI.extension xmi.extender="Visual Paradigm
for UML">
<type value="Performance"/>
</XMI.extension>
</argument>
<argument name="" xmi.id="A4DYXlSGAqACBQBH">
<XMI.extension xmi.extender="Visual Paradigm
for UML">
<type value="Interface"/>
</XMI.extension>
</argument>
</vpumlModels>
</property>

```



```

</properties>
</vpumlModel>
<vpumlModel id="A4DYXlSGAqACBQBQ" modelType=
"TaggedValueDefinition">
<properties>
<property name="name" type="string" value="v
erifyMethod"/>
<property name="modelType" type="string" val
ue="TaggedValueDefinition"/>
<property htmlValue="" name="documentation"
type="htmlString" value=""/>
<property name="pmAuthor" type="string" valu
e="Chris"/>
<property name="pmCreateDateTime" type="stri
ng" value="1281948844480"/>
<property name="pmLastModified" type="string
" value="1282116883372"/>
<property name="defaultValue" type="string"
value=""/>
<property name="defaultHTMLValue" type="stri
ng" value=""/>
<property name="type" type="int" value="2"/>
<property modelRef="IF9YXlSGAqACBQAE" name="
tagDefStereotype" type="modelRef"/>
<property name="enumerationValues" type="mod
els">
<vpumlModels>
<argument name="" xmi.id="A4DYXlSGAqACBQBZ">
<XMI.extension xmi.extender="Visual Paradigm
for UML">
<type value="Analysis"/>
</XMI.extension>
</argument>
<argument name="" xmi.id="g4DYXlSGAqACBQBi">
<XMI.extension xmi.extender="Visual Paradigm
for UML">
<type value="Demonstration"/>
</XMI.extension>
</argument>
<argument name="" xmi.id="g4DYXlSGAqACBQBr">
<XMI.extension xmi.extender="Visual Paradigm
for UML">
<type value="Inspection"/>
</XMI.extension>
</argument>
<argument name="" xmi.id="Q4DYXlSGAqACBQB0">
<XMI.extension xmi.extender="Visual Paradigm
for UML">
<type value="Test"/>
</XMI.extension>
</argument>
</vpumlModels>
</property>
</properties>
</vpumlModel>
<vpumlModel id="Q4DYXlSGAqACBQB9" modelType=
"TaggedValueDefinition">
<properties>
<property name="name" type="string" value="r
isk"/>
<property name="modelType" type="string" val
ue="TaggedValueDefinition"/>
<property htmlValue="" name="documentation"
type="htmlString" value=""/>
<property name="pmAuthor" type="string" valu
e="Chris"/>
<property name="pmCreateDateTime" type="stri
ng" value="1281948844482"/>
<property name="pmLastModified" type="string
" value="1282116883381"/>
<property name="defaultValue" type="string"
value=""/>
<property name="defaultHTMLValue" type="stri
ng" value=""/>
<property name="type" type="int" value="2"/>
<property modelRef="IF9YXlSGAqACBQAE" name="
tagDefStereotype" type="modelRef"/>
<property name="enumerationValues" type="mod
els">
<vpumlModels>
<argument name="" xmi.id="w4DYXlSGAqACBQCG">
<XMI.extension xmi.extender="Visual Paradigm
for UML">
<type value="High"/>
</XMI.extension>
</argument>
<argument name="" xmi.id="w4DYXlSGAqACBQCP">
<XMI.extension xmi.extender="Visual Paradigm
for UML">
<type value="Medium"/>
</XMI.extension>
</argument>
<argument name="" xmi.id="w4DYXlSGAqACBQCY">
<XMI.extension xmi.extender="Visual Paradigm
for UML">
<type value="Low"/>
</XMI.extension>
</argument>
</vpumlModels>
</property>
</properties>
</vpumlModel>
<vpumlModel id="w4DYXlSGAqACBQCh" modelType=
"TaggedValueDefinition">
<properties>
<property name="name" type="string" value="s
tatus"/>
<property name="modelType" type="string" val
ue="TaggedValueDefinition"/>
<property htmlValue="" name="documentation"
type="htmlString" value=""/>
<property name="pmAuthor" type="string" valu
e="Chris"/>
<property name="pmCreateDateTime" type="stri
ng" value="1281948844483"/>
<property name="pmLastModified" type="string
" value="1282116883381"/>
<property name="defaultValue" type="string"
value=""/>
<property name="defaultHTMLValue" type="stri
ng" value=""/>
<property name="type" type="int" value="2"/>
<property modelRef="IF9YXlSGAqACBQAE" name="
tagDefStereotype" type="modelRef"/>
<property name="enumerationValues" type="mod
els">
<vpumlModels>
<argument name="" xmi.id="I4DYXlSGAqACBQCq">
<XMI.extension xmi.extender="Visual Paradigm
for UML">
<type value="Proposed"/>
</XMI.extension>
</argument>
<argument name="" xmi.id="I4DYXlSGAqACBQCz">
<XMI.extension xmi.extender="Visual Paradigm
for UML">
<type value="Approved"/>

```

```

</XMI.extension>
</argument>
<argument name="" xmi.id="I4DYXlSGAqACBQC8">
<XMI.extension xmi.extender="Visual Paradigm
for UML">
<type value="Rejected"/>
</XMI.extension>
</argument>
<argument name="" xmi.id="o4DYXlSGAqACBQDF">
<XMI.extension xmi.extender="Visual Paradigm
for UML">
<type value="Deferred"/>
</XMI.extension>
</argument>
<argument name="" xmi.id="o4DYXlSGAqACBQDO">
<XMI.extension xmi.extender="Visual Paradigm
for UML">
<type value="Implemented"/>
</XMI.extension>
</argument>
<argument name="" xmi.id="Y4DYXlSGAqACBQDX">
<XMI.extension xmi.extender="Visual Paradigm
for UML">
<type value="Mandatory"/>
</XMI.extension>
</argument>
<argument name="" xmi.id="Y4DYXlSGAqACBQDg">
<XMI.extension xmi.extender="Visual Paradigm
for UML">
<type value="Obsolete"/>
</XMI.extension>
</argument>
</vpumlModels>
</property>
</properties>
</vpumlModel>
</vpumlChildModels>
</XMI.extension>
<UML:Stereotype name="UseCase" xmi.id="QrjYX
lSGAqACBQ01">
<XMI.extension xmi.extender="Visual Paradigm
for UML">
<baseType value="UseCase"/>
<taggedValueDefinitions>
<taggedValueDefinition htmlValue="" name="Le
vel" type="2" value="">
<value id="MrjYXlSGAqACBQ04" value="Summary"
/>
<value id="srjYXlSGAqACBQPB" value="User"/>
<value id="srjYXlSGAqACBQPK" value="Subfunct
ion"/>
</taggedValueDefinition>
<taggedValueDefinition htmlValue="" name="Co
mplexity" type="2" value="">
<value id="crjYXlSGAqACBQPc" value="Low"/>
<value id="crjYXlSGAqACBQP1" value="Medium"
/>
<value id="crjYXlSGAqACBQPu" value="High"/>
</taggedValueDefinition>
<taggedValueDefinition htmlValue="" name="Us
e Case Status" type="2" value="">
<value id="8rjYXlSGAqACBQQA" value="Name
Only"/>
<value id="8rjYXlSGAqACBQQJ" value="Initial"
/>
<value id="8rjYXlSGAqACBQQS" value="Base"/>
<value id="CrjYXlSGAqACBQQb" value="Complete"
/>
<value id="CrjYXlSGAqACBQQk" value="Deferred"
/>
</taggedValueDefinition>
<taggedValueDefinition htmlValue="" name="Im
plementation Status" type="2" value="">
<value id="CrjYXlSGAqACBQQ2" value="Schedule
d"/>
<value id="irjYXlSGAqACBQQ_" value="Started"
/>
<value id="irjYXlSGAqACBQRI" value="Partiall
y Complete"/>
<value id="irjYXlSGAqACBQRR" value="Complete"
/>
<value id="irjYXlSGAqACBQRa" value="Partiall
y Deferred"/>
</taggedValueDefinition>
<taggedValueDefinition htmlValue="" name="Pr
econditions" type="4" value="">
<taggedValueDefinition htmlValue="" name="Po
st-conditions" type="4" value="">
<taggedValueDefinition htmlValue="" name="Au
thor" type="0" value="">
<taggedValueDefinition htmlValue="" name="As
sumptions" type="4" value="">
</taggedValueDefinitions>
</XMI.extension>
</UML:Stereotype>
<UML:Stereotype name="requirement" xmi.id="I
F9YXlSGAqACBQAE">
<XMI.extension xmi.extender="Visual Paradigm
for UML">
<baseType value="Requirement"/>
<taggedValueDefinitions>
<taggedValueDefinition htmlValue="" name="Te
xt" type="3" value="">
<taggedValueDefinition htmlValue="" name="ID
" type="0" value="">
<taggedValueDefinition htmlValue="" name="so
urce" type="0" value="">
<taggedValueDefinition htmlValue="" name="ki
nd" type="2" value="">
<value id="gYDYXlSGAqACBQA1" value="Function
al"/>
<value id="_YDYXlSGAqACBQA." value="Performa
nce"/>
<value id="A4DYXlSGAqACBQBH" value="Interfac
e"/>
</taggedValueDefinition>
<taggedValueDefinition htmlValue="" name="ve
rifyMethod" type="2" value="">
<value id="A4DYXlSGAqACBQBZ" value="Analysis"
/>
<value id="g4DYXlSGAqACBQBi" value="Demonstr
ation"/>
<value id="g4DYXlSGAqACBQBr" value="Inspecti
on"/>
<value id="Q4DYXlSGAqACBQB0" value="Test"/>
</taggedValueDefinition>
<taggedValueDefinition htmlValue="" name="ri
sk" type="2" value="">
<value id="w4DYXlSGAqACBQCG" value="High"/>
<value id="w4DYXlSGAqACBQCP" value="Medium"
/>
<value id="w4DYXlSGAqACBQCY" value="Low"/>
</taggedValueDefinition>
<taggedValueDefinition htmlValue="" name="st
atus" type="2" value="">
<value id="I4DYXlSGAqACBQCq" value="Proposed"
/>

```

```

<value id="I4DYXlSGAqACBQCz" value="Approved
"/>
<value id="I4DYXlSGAqACBQC8" value="Rejected
"/>
<value id="o4DYXlSGAqACBQDF" value="Deferred
"/>
<value id="o4DYXlSGAqACBQDO" value="Implemen
ted"/>
<value id="Y4DYXlSGAqACBQDX" value="Mandator
y"/>
<value id="Y4DYXlSGAqACBQDg" value="Obsolete
"/>
</taggedValueDefinition>
</taggedValueDefinitions>
</XMI.extension>
</UML:Stereotype>
<UML:Collaboration name="Sequence
Diagram2" xmi.id="J9BkIVSGAqACBSIb">
<XMI.extension xmi.extender="Visual Paradigm
for UML">
<modelType xmi.value="Frame"/>
<type xmi.value="sd"/>
</XMI.extension>
<UML:Namespace.ownedElement>
<UML:Actor isAbstract="false" isLeaf="false"
isRoot="false" name="Tamias" visibility="pu
blic" xmi.id="wErkIVSGAqACBSJk">
<XMI.extension xmi.extender="Visual Paradigm
for UML">
<businessModel xmi.value="false"/>
</XMI.extension>
</UML:Actor>
<UML:ClassifierRole multiplicity="Unspecifie
d" name="" xmi.id="HkHkIVSGAqACBSLK">
<XMI.extension xmi.extender="Visual Paradigm
for UML">
<active xmi.value="false"/>
<stopped xmi.value="false"/>
<multiObject xmi.value="false"/>
<activations>
<activation type="false" xmi.id="UNUUIVSGAqA
CBSVN"/>
<activation type="false" xmi.id="cwVsIVSGAqA
CBUV9"/>
</activations>
</XMI.extension>
</UML:ClassifierRole>
<UML:ClassifierRole multiplicity="Unspecifie
d" name="k" xmi.id="PtvkIVSGAqACBSNV">
<XMI.extension xmi.extender="Visual Paradigm
for UML">
<active xmi.value="false"/>
<stopped xmi.value="false"/>
<multiObject xmi.value="false"/>
<activations>
<activation type="false" xmi.id="YWHUIVSGAqA
CBSKI"/>
<activation type="false" xmi.id="DIjsIVSGAqA
CBUcR"/>
<activation type="false" xmi.id="oDfsIVSGAqA
CBUnm"/>
<activation type="false" xmi.id="uERcIVSGAqA
CBU4I"/>
</activations>
</XMI.extension>
</UML:ClassifierRole>
<UML:ClassifierRole multiplicity="Unspecifie
d" name="p" xmi.id="QMAUIVSGAqACBSPa">
<XMI.extension xmi.extender="Visual Paradigm
for UML">
<active xmi.value="false"/>
<stopped xmi.value="false"/>
<multiObject xmi.value="false"/>
<activations>
<activation type="false" xmi.id="5cS0IVSGAqA
CBSw5"/>
<activation type="false" xmi.id="gNVcIVSGAqA
CBVBZ"/>
</activations>
</XMI.extension>
</UML:ClassifierRole>
<UML:ClassifierRole multiplicity="Unspecifie
d" name="t" xmi.id="TMOUIVSGAqACBSRu">
<XMI.extension xmi.extender="Visual Paradigm
for UML">
<active xmi.value="false"/>
<stopped xmi.value="false"/>
<multiObject xmi.value="false"/>
<activations>
<activation type="false" xmi.id="yh8cIVSGAqA
CBU0R"/>
</activations>
</XMI.extension>
</UML:ClassifierRole>
<UML:ClassifierRole multiplicity="Unspecifie
d" name="Akyrosi" xmi.id="_YbcIVSGAqACBVHy">
<XMI.extension xmi.extender="Visual Paradigm
for UML">
<active xmi.value="false"/>
<stopped xmi.value="false"/>
<multiObject xmi.value="false"/>
<activations>
<activation type="false" xmi.id="amvcIVSGAqA
CBVLL"/>
</activations>
</XMI.extension>
</UML:ClassifierRole>
</UML:Namespace.ownedElement>
<UML:Collaboration.interaction>
<UML:Interaction name="kodKratasis:kodikos"
xmi.id="4Yb0IVSGAqACBS_T">
<XMI.extension xmi.extender="Visual Paradigm
for UML">
<type xmi.value="CombinedFragment"/>
<operator xmi.value="alt"/>
<operands>
<interactionOperand constraints="kodKratasis
:kodikos" max="" min="" name="Operand" xmi.i
d="H4b0IVSGAqACBS_k">
<messages>
<message xmi.value="Okk0IVSGAqACBSpX"/>
</messages>
</interactionOperand>
<interactionOperand constraints="" max="" mi
n="" name="Operand2" xmi.id="Oeb0IVSGAqACBS_
3">
<messages>
<message xmi.value="OQV0IVSGAqACBS8P"/>
<message xmi.value="jnm0IVSGAqACBSsq"/>
<message xmi.value="ph60IVSGAqACBSyH"/>
<message xmi.value="WBB0IVSGAqACBS4f"/>
<message xmi.value="0cS0IVSGAqACBSv6"/>
<message xmi.value="Wau0IVSGAqACBS1n"/>
</messages>
</interactionOperand>
</operands>
<coveredLifelines>

```

```

<lifeline xmi.value="wErkIVSGAqACBSJk"/>
<lifeline xmi.value="HkHkIVSGAqACBSLK"/>
<lifeline xmi.value="PtvkIVSGAqACBSNV"/>
<lifeline xmi.value="QMAUIVSGAqACBSPa"/>
</coveredLifelines>
</XMI.extension>
<UML:Interaction.message/>
</UML:Interaction>
<UML:Interaction name="CombinedFragment" xmi
.id="V3WMIVSGAqACBTfd">
<XMI.extension xmi.extender="Visual Paradigm
for UML">
<type xmi.value="CombinedFragment"/>
<operator xmi.value="loop"/>
<operands>
<interactionOperand constraints="" max="" mi
n="" name="Operand" xmi.id="L3WMIVSGAqACBTft
">
<messages>
<message xmi.value="iSA0IVSGAqACBS17"/>
<message xmi.value="GmHUIVSGAqACBSjF"/>
</messages>
</interactionOperand>
</operands>
</coveredLifelines>
<lifeline xmi.value="wErkIVSGAqACBSJk"/>
<lifeline xmi.value="HkHkIVSGAqACBSLK"/>
<lifeline xmi.value="PtvkIVSGAqACBSNV"/>
<lifeline xmi.value="QMAUIVSGAqACBSPa"/>
</coveredLifelines>
</XMI.extension>
</UML:Interaction>
<UML:Interaction name="CombinedFragment2" xm
i.id="sDG8IVSGAqACBVue">
<XMI.extension xmi.extender="Visual Paradigm
for UML">
<type xmi.value="CombinedFragment"/>
<operator xmi.value="loop"/>
<operands>
<interactionOperand constraints="" max="" mi
n="" name="Operand" xmi.id="ODG8IVSGAqACBVuu
">
<messages>
<message xmi.value=".TrsIVSGAqACBUiJ"/>
<message xmi.value="Ga4cIVSGAqACBUuI"/>
<message xmi.value="TBZcIVSGAqACBU5j"/>
<message xmi.value="jUdcIVSGAqACBVC2"/>
<message xmi.value="OpI8IVSGAqACBVNA"/>
<message xmi.value="SIjsIVSGAqACBUa9"/>
<message xmi.value="p9fsIVSGAqACBUmM"/>
<message xmi.value="HB8cIVSGAqACBUyx"/>
<message xmi.value="P4RcIVSGAqACBU21"/>
<message xmi.value="a1VcIVSGAqACBU_w"/>
<message xmi.value="TGvcIVSGAqACBVJc"/>
</messages>
<UML:Namespace.ownedElement>
<UML:Interaction name="CombinedFragment3" xm
i.id="u6x8IVSGAqACBV4T">
<XMI.extension xmi.extender="Visual Paradigm
for UML">
<type xmi.value="CombinedFragment"/>
<operator xmi.value="alt"/>
<operands>
<interactionOperand constraints="" max="" mi
n="" name="Operand" xmi.id="d6x8IVSGAqACBV4j
">
<messages>
<message xmi.value="Ga4cIVSGAqACBUuI"/>
<message xmi.value="p9fsIVSGAqACBUmM"/>
</messages>
</interactionOperand>
</operands>
</coveredLifelines>
<lifeline xmi.value="wErkIVSGAqACBSJk"/>
<lifeline xmi.value="HkHkIVSGAqACBSLK"/>
<lifeline xmi.value="PtvkIVSGAqACBSNV"/>
<lifeline xmi.value="QMAUIVSGAqACBSPa"/>
<lifeline xmi.value="TMOUIVSGAqACBSRu"/>
<lifeline xmi.value="_YbcIVSGAqACBVHy"/>
</coveredLifelines>
</XMI.extension>
</UML:Interaction>
</UML:Collaboration.interaction>
</UML:Collaboration>
<argument name="" xmi.id="MrjYX1SGAqACBQO4">
<XMI.extension xmi.extender="Visual Paradigm
for UML">
<type value="Summary"/>
</XMI.extension>
</argument>
<argument name="" xmi.id="srjYX1SGAqACBQPb">

```

```

<XMI.extension xmi.extender="Visual Paradigm
for UML">
<type value="User"/>
</XMI.extension>
</argument>
<argument name="" xmi.id="srjYXlSGAqACBQPK">
<XMI.extension xmi.extender="Visual Paradigm
for UML">
<type value="Subfunction"/>
</XMI.extension>
</argument>
<UML:Message action="91UUIVSGAqACBSUh_type"
name="neaAkyrosi(KodKratasis)" receiver="HkHk
kIVSGAqACBSLK" sender="wErkIVSGAqACBSJk" xmi
.id="91UUIVSGAqACBSUh">
<XMI.extension xmi.extender="Visual Paradigm
for UML">
<number xmi.value="1"/>
<asynchronous xmi.value="false"/>
<to xmi.value="cwVsIVSGAqACBUV9"/>
</XMI.extension>
</UML:Message>
<UML:UninterpretedAction xmi.id="91UUIVSGAqA
CBSUh_type"/>
<UML:Message action="h72UIVSGAqACBSZU_type"
name="k.next(kratisi)" receiver="HkHkIVSGAqA
CBSLK" sender="HkHkIVSGAqACBSLK" xmi.id="h72
UIVSGAqACBSZU">
<XMI.extension xmi.extender="Visual Paradigm
for UML">
<type xmi.value="self"/>
<number xmi.value="1.1"/>
<asynchronous xmi.value="false"/>
<from xmi.value="cwVsIVSGAqACBUV9"/>
<to xmi.value="cwVsIVSGAqACBUV9"/>
</XMI.extension>
</UML:Message>
<UML:UninterpretedAction xmi.id="h72UIVSGAqA
CBSZU_type"/>
<UML:Message action="GmHUIVSGAqACBSjF_type"
name="getKodikos()" receiver="PtvkIVSGAqACBS
NV" sender="HkHkIVSGAqACBSLK" xmi.id="GmHUIV
SGAqACBSjF">
<XMI.extension xmi.extender="Visual Paradigm
for UML">
<number xmi.value="1.2"/>
<asynchronous xmi.value="false"/>
<from xmi.value="cwVsIVSGAqACBUV9"/>
<to xmi.value="YWHUIVSGAqACBSkI"/>
</XMI.extension>
</UML:Message>
<UML:UninterpretedAction xmi.id="GmHUIVSGAqA
CBSjF_type"/>
<UML:Message action="ZSA0IVSGAqACBSms" name=
"Kodikos" receiver="HkHkIVSGAqACBSLK" sender
="PtvkIVSGAqACBSNV" xmi.id="isa0IVSGAqACBS17
">
<XMI.extension xmi.extender="Visual Paradigm
for UML">
<number xmi.value="1.3"/>
<asynchronous xmi.value="false"/>
<from xmi.value="YWHUIVSGAqACBSkI"/>
<to xmi.value="cwVsIVSGAqACBUV9"/>
</XMI.extension>
</UML:Message>
<UML:ReturnAction isAsynchronous="false" nam
e="Return" xmi.id="ZSA0IVSGAqACBSms">
<XMI.extension xmi.extender="Visual Paradigm
for UML">
<iteration xmi.value="false"/>
<recurrenceCondition xmi.value=""/>
<script xmi.value=""/>
<modelType xmi.value="ActionTypeReturn"/>
</XMI.extension>
</UML:ReturnAction>
<UML:Message action="Okk0IVSGAqACBSpX_type"
name="getParastasi()" receiver="PtvkIVSGAqAC
BSNV" sender="HkHkIVSGAqACBSLK" xmi.id="Okk0
IVSGAqACBSpX">
<XMI.extension xmi.extender="Visual Paradigm
for UML">
<number xmi.value="1.4"/>
<asynchronous xmi.value="false"/>
<from xmi.value="cwVsIVSGAqACBUV9"/>
<to xmi.value="YWHUIVSGAqACBSkI"/>
</XMI.extension>
</UML:Message>
<UML:UninterpretedAction xmi.id="Okk0IVSGAqA
CBSpX_type"/>
<UML:Message action="CXM0IVSGAqACBSth" name=
"p" receiver="HkHkIVSGAqACBSLK" sender="Ptvk
IVSGAqACBSNV" xmi.id="jnm0IVSGAqACBSsq">
<XMI.extension xmi.extender="Visual Paradigm
for UML">
<number xmi.value="1.5"/>
<asynchronous xmi.value="false"/>
<from xmi.value="YWHUIVSGAqACBSkI"/>
<to xmi.value="cwVsIVSGAqACBUV9"/>
</XMI.extension>
</UML:Message>
<UML:ReturnAction isAsynchronous="false" nam
e="Return" xmi.id="CXM0IVSGAqACBSth">
<XMI.extension xmi.extender="Visual Paradigm
for UML">
<iteration xmi.value="false"/>
<recurrenceCondition xmi.value=""/>
<script xmi.value=""/>
<modelType xmi.value="ActionTypeReturn"/>
</XMI.extension>
</UML:ReturnAction>
<UML:Message action="0cS0IVSGAqACBSv6_type"
name="getKodikos()" receiver="QMAUIVSGAqACBS
Pa" sender="HkHkIVSGAqACBSLK" xmi.id="0cS0IV
SGAqACBSv6">
<XMI.extension xmi.extender="Visual Paradigm
for UML">
<number xmi.value="1.6"/>
<asynchronous xmi.value="false"/>
<from xmi.value="cwVsIVSGAqACBUV9"/>
<to xmi.value="5cS0IVSGAqACBSw5"/>
</XMI.extension>
</UML:Message>
<UML:UninterpretedAction xmi.id="0cS0IVSGAqA
CBSv6_type"/>
<UML:Message action="h60IVSGAqACBSzE" name=
"kodikosPar" receiver="HkHkIVSGAqACBSLK" sen
der="QMAUIVSGAqACBSPa" xmi.id="ph60IVSGAqACB
SyH">
<XMI.extension xmi.extender="Visual Paradigm
for UML">
<number xmi.value="1.7"/>
<asynchronous xmi.value="false"/>
<from xmi.value="5cS0IVSGAqACBSw5"/>
<to xmi.value="cwVsIVSGAqACBUV9"/>
</XMI.extension>
</UML:Message>
<UML:ReturnAction isAsynchronous="false" nam
e="Return" xmi.id="_h60IVSGAqACBSzE">

```

```

<XMI.extension xmi.extender="Visual Paradigm
for UML">
<iteration xmi.value="false"/>
<recurrenceCondition xmi.value=""/>
<script xmi.value=""/>
<modelType xmi.value="ActionTypeReturn"/>
</XMI.extension>
</UML:ReturnAction>
<UML:Message action="Wau0IVSGAqACBS1n_type"
name="getWra()" receiver="QMAUIVSGAqACBSPa"
sender="HkHkIVSGAqACBSLK" xmi.id="Wau0IVSGAq
ACBS1n">
<XMI.extension xmi.extender="Visual Paradigm
for UML">
<number xmi.value="1.8"/>
<asynchronous xmi.value="false"/>
<from xmi.value="cwVsIVSGAqACBUV9"/>
<to xmi.value="5cS0IVSGAqACBSw5"/>
</XMI.extension>
</UML:Message>
<UML:UninterpretedAction xmi.id="Wau0IVSGAqA
CBS1n_type"/>
<UML:Message action="rBB0IVSGAqACBS5i" name=
"wraPar" receiver="HkHkIVSGAqACBSLK" sender=
"QMAUIVSGAqACBSPa" xmi.id="WBB0IVSGAqACBS4f"
>
<XMI.extension xmi.extender="Visual Paradigm
for UML">
<number xmi.value="1.9"/>
<asynchronous xmi.value="false"/>
<from xmi.value="5cS0IVSGAqACBSw5"/>
<to xmi.value="cwVsIVSGAqACBUV9"/>
</XMI.extension>
</UML:Message>
<UML:ReturnAction isAsynchronous="false" nam
e="Return" xmi.id="rBB0IVSGAqACBS5i">
<XMI.extension xmi.extender="Visual Paradigm
for UML">
<iteration xmi.value="false"/>
<recurrenceCondition xmi.value=""/>
<script xmi.value=""/>
<modelType xmi.value="ActionTypeReturn"/>
</XMI.extension>
</UML:ReturnAction>
<UML:Message action="UN10IVSGAqACBS.P" name=
"kodikosPar,
wraPar" receiver="wErkIVSGAqACBSJk" sender="
HkHkIVSGAqACBSLK" xmi.id="QOV0IVSGAqACBS8P">
<XMI.extension xmi.extender="Visual Paradigm
for UML">
<number xmi.value="2"/>
<asynchronous xmi.value="false"/>
<from xmi.value="cwVsIVSGAqACBUV9"/>
</XMI.extension>
</UML:Message>
<UML:ReturnAction isAsynchronous="false" nam
e="Return" xmi.id="UN10IVSGAqACBS.P">
<XMI.extension xmi.extender="Visual Paradigm
for UML">
<iteration xmi.value="false"/>
<recurrenceCondition xmi.value=""/>
<script xmi.value=""/>
<modelType xmi.value="ActionTypeReturn"/>
</XMI.extension>
</UML:ReturnAction>
<UML:Message action="wKLMIVSGAqACBT00_type"
name="Epivevaiosi()" receiver="HkHkIVSGAqACB
SLK" sender="wErkIVSGAqACBSJk" xmi.id="wKLMIV
SGAqACBT00">

```

```

<XMI.extension xmi.extender="Visual Paradigm
for UML">
<number xmi.value="3"/>
<asynchronous xmi.value="false"/>
<to xmi.value="UNUUIVSGAqACBSVN"/>
</XMI.extension>
</UML:Message>
<UML:UninterpretedAction xmi.id="wKLMIVSGAqA
CBT00_type"/>
<UML:Message action="bB.sIVSGAqACBT_V_type"
name="next(kratisi)" receiver="HkHkIVSGAqACB
SLK" sender="HkHkIVSGAqACBSLK" xmi.id="bB.sI
VSGAqACBT_V">
<XMI.extension xmi.extender="Visual Paradigm
for UML">
<type xmi.value="self"/>
<number xmi.value="3.1"/>
<asynchronous xmi.value="false"/>
<from xmi.value="UNUUIVSGAqACBSVN"/>
<to xmi.value="UNUUIVSGAqACBSVN"/>
</XMI.extension>
</UML:Message>
<UML:UninterpretedAction xmi.id="bB.sIVSGAqA
CBT_V_type"/>
<UML:Message action="SIjsIVSGAqACBUa9_type"
name="getKratisi()" receiver="PtvkIVSGAqACBS
NV" sender="HkHkIVSGAqACBSLK" xmi.id="SIjsIV
SGAqACBUa9">
<XMI.extension xmi.extender="Visual Paradigm
for UML">
<number xmi.value="3.2"/>
<asynchronous xmi.value="false"/>
<from xmi.value="UNUUIVSGAqACBSVN"/>
<to xmi.value="DIjsIVSGAqACBUcR"/>
</XMI.extension>
</UML:Message>
<UML:UninterpretedAction xmi.id="SIjsIVSGAqA
CBUa9_type"/>
<UML:Message action="fTrsIVSGAqACBUjb" name=
"Kodikos" receiver="HkHkIVSGAqACBSLK" sender
="PtvkIVSGAqACBSNV" xmi.id=".TrsIVSGAqACBUiJ
">
<XMI.extension xmi.extender="Visual Paradigm
for UML">
<number xmi.value="3.3"/>
<asynchronous xmi.value="false"/>
<from xmi.value="DIjsIVSGAqACBUcR"/>
<to xmi.value="UNUUIVSGAqACBSVN"/>
</XMI.extension>
</UML:Message>
<UML:ReturnAction isAsynchronous="false" nam
e="Return" xmi.id="fTrsIVSGAqACBUjb">
<XMI.extension xmi.extender="Visual Paradigm
for UML">
<iteration xmi.value="false"/>
<recurrenceCondition xmi.value=""/>
<script xmi.value=""/>
<modelType xmi.value="ActionTypeReturn"/>
</XMI.extension>
</UML:ReturnAction>
<UML:Message action="p9fsIVSGAqACBUmM_type"
name="getTheseisKrat()" receiver="PtvkIVSGAq
ACBSNV" sender="HkHkIVSGAqACBSLK" xmi.id="p9
fsIVSGAqACBUmM">
<XMI.extension xmi.extender="Visual Paradigm
for UML">
<number xmi.value="3.4"/>
<asynchronous xmi.value="false"/>
<from xmi.value="UNUUIVSGAqACBSVN"/>

```

```

<to xmi.value="oDfsIVSGAqACBUnm"/>
</XMI.extension>
</UML:Message>
<UML:UninterpretedAction xmi.id="p9fsIVSGAqACBUm_type"/>
<UML:Message action="3a4cIVSGAqACBUvg" name="theseisKrat" receiver="HkHkIVSGAqACBSLK" sender="PtvkIVSGAqACBSNV" xmi.id="Ga4cIVSGAqACBUuI">
<XMI.extension xmi.extender="Visual Paradigm for UML">
<number xmi.value="3.5"/>
<asynchronous xmi.value="false"/>
<from xmi.value="oDfsIVSGAqACBUnm"/>
<to xmi.value="UNUUIVSGAqACBSVN"/>
</XMI.extension>
</UML:Message>
<UML:ReturnAction isAsynchronous="false" name="Return" xmi.id="3a4cIVSGAqACBUvg">
<XMI.extension xmi.extender="Visual Paradigm for UML">
<iteration xmi.value="false"/>
<recurrenceCondition xmi.value=""/>
<script xmi.value=""/>
<modelType xmi.value="ActionTypeReturn"/>
</XMI.extension>
</UML:ReturnAction>
<UML:Message action="HB8cIVSGAqACBUyx_type" name="setKatastasi()" receiver="TMoUIVSGAqACBSRu" sender="HkHkIVSGAqACBSLK" xmi.id="HB8cIVSGAqACBUyx">
<XMI.extension xmi.extender="Visual Paradigm for UML">
<number xmi.value="3.6"/>
<asynchronous xmi.value="false"/>
<from xmi.value="UNUUIVSGAqACBSVN"/>
<to xmi.value="yh8cIVSGAqACBU0R"/>
</XMI.extension>
</UML:Message>
<UML:UninterpretedAction xmi.id="HB8cIVSGAqACBUyx_type"/>
<UML:Message action="P4RcIVSGAqACBU2l_type" name="getParastasi()" receiver="PtvkIVSGAqACBSNV" sender="HkHkIVSGAqACBSLK" xmi.id="P4RcIVSGAqACBU2l">
<XMI.extension xmi.extender="Visual Paradigm for UML">
<number xmi.value="3.7"/>
<asynchronous xmi.value="false"/>
<from xmi.value="UNUUIVSGAqACBSVN"/>
<to xmi.value="uERcIVSGAqACBU4I"/>
</XMI.extension>
</UML:Message>
<UML:UninterpretedAction xmi.id="P4RcIVSGAqACBU2l_type"/>
<UML:Message action="shZcIVSGAqACBU7E" name="p" receiver="HkHkIVSGAqACBSLK" sender="PtvkIVSGAqACBSNV" xmi.id="TBZcIVSGAqACBU5j">
<XMI.extension xmi.extender="Visual Paradigm for UML">
<number xmi.value="3.8"/>
<asynchronous xmi.value="false"/>
<from xmi.value="uERcIVSGAqACBU4I"/>
<to xmi.value="UNUUIVSGAqACBSVN"/>
</XMI.extension>
</UML:Message>
<UML:ReturnAction isAsynchronous="false" name="Return" xmi.id="shZcIVSGAqACBU7E">
<XMI.extension xmi.extender="Visual Paradigm for UML">
<iteration xmi.value="false"/>
<recurrenceCondition xmi.value=""/>
<script xmi.value=""/>
<modelType xmi.value="ActionTypeReturn"/>
</XMI.extension>
</UML:ReturnAction>
<UML:Message action="c0dcIVSGAqACBVEd" name="wra" receiver="HkHkIVSGAqACBSLK" sender="QMAUIVSGAqACBSPa" xmi.id="jUdcIVSGAqACBVC2">
<XMI.extension xmi.extender="Visual Paradigm for UML">
<number xmi.value="3.10"/>
<asynchronous xmi.value="false"/>
<from xmi.value="gNVcIVSGAqACBVBZ"/>
<to xmi.value="UNUUIVSGAqACBSVN"/>
</XMI.extension>
</UML:Message>
<UML:ReturnAction isAsynchronous="false" name="Return" xmi.id="c0dcIVSGAqACBVEd">
<XMI.extension xmi.extender="Visual Paradigm for UML">
<iteration xmi.value="false"/>
<recurrenceCondition xmi.value=""/>
<script xmi.value=""/>
<modelType xmi.value="ActionTypeReturn"/>
</XMI.extension>
</UML:ReturnAction>
<UML:Message action="TGvcIVSGAqACBVJc_type" name="<<create>>(wra)" receiver="_YbcIVSGAqACBVHy" sender="HkHkIVSGAqACBSLK" xmi.id="TGvcIVSGAqACBVJc">
<XMI.extension xmi.extender="Visual Paradigm for UML">
<number xmi.value="3.11"/>
<asynchronous xmi.value="false"/>
<from xmi.value="UNUUIVSGAqACBSVN"/>
<to xmi.value="amvcIVSGAqACBVLL"/>
</XMI.extension>
</UML:Message>
<UML:UninterpretedAction xmi.id="TGvcIVSGAqACBVJc_type"/>
<UML:Message action="OpI8IVSGAqACBVNA_type" name="AddAkyrosi" receiver="HkHkIVSGAqACBSLK" sender="HkHkIVSGAqACBSLK" xmi.id="OpI8IVSGAqACBVNA">
<XMI.extension xmi.extender="Visual Paradigm for UML">
<type xmi.value="self"/>
<number xmi.value="3.12"/>
<asynchronous xmi.value="false"/>
<from xmi.value="UNUUIVSGAqACBSVN"/>
<to xmi.value="UNUUIVSGAqACBSVN"/>
</XMI.extension>

```

```

</UML:Message>
<UML:UninterpretedAction xmi.id="OpI8IVSGAqA
CBVNA_type"/>
</UML:Namespace.ownedElement>
</UML:Model>
<UML:Diagram diagramType="InteractionDiagram
" documentation="" name="Sequence
Diagram2" toolName="Visual Paradigm for
UML" xmi.id="K9BkIVSGAqACBSIY">
<XMI.extension xmi.extender="Visual Paradigm
for UML">
<properties>
<apmCreateDateTime value="1282117340116"/>
<apmLastModified value="1289849229624"/>
<ashowPackageNameStyle value="0"/>
<ashowConnectorName value="0"/>
<apaintConnectorThroughLabel value="0"/>
<ashowActivityStateNodeCaption value="0"/>
<aeditable value="true"/>
<afollowDiagramParentElement value="true"/>
<aautoFitShapesSize value="false"/>
<ashowTaggedValues value="false"/>
<ashowStereotypes value="true"/>
<ashowConstraints value="false"/>
<aalignToGrid value="true"/>
<agridVisible value="false"/>
<agridWidth value="10"/>
<agridHeight value="10"/>
<agridColor value="Cr:192,192,192,255"/>
<adiagramBackground value="Cr:255,255,255,25
5"/>
<aconnectorStyle value="1"/>
<aconnectorLineJumps value="0"/>
<aconnectorLineJumpsSize value="0"/>
<aconnectionPointStyle value="0"/>
<aconnectorLabelOrientation value="0"/>
<amodelElementNameAlignment value="4"/>
<azoomRatio value="1.0"/>
<amaximized value="true"/>
<ax value="0"/>
<ay value="0"/>
<awidth value="1151"/>
<aheight value="637"/>
<a_globalPaletteOption value="true"/>
<a_alwaysShowReferenceAndSubdiagramResource
value="false"/>
<ainitializeDiagramForCreate value="true"/>
<ashowDefaultPackage value="false"/>
<ashowParameterNamesInMessageOperationSignat
ure value="true"/>
<ashowArgumentsInMessagesOperationSignature
value="true"/>
<ashowMessagesOperationSignature value="fals
e"/>
<ainteractionLifeLinePresentationOption valu
e="0"/>
<a_rootFrame value="J9BkIVSGAqACBSIb"/>
<ashowSequenceNumbers value="true"/>
<ashowActivations value="true"/>
<adisplayLifeLinesAsRobustnessAnalysisIcon v
alue="0"/>
<ashowMessageStereotype value="true"/>
<asequenceNumberHandling value="0"/>
<asequenceNumbering value="0"/>
<a_requestRecalculateSequenceNumbers value="
false"/>
</properties>
</XMI.extension>
<UML:Diagram.element>
<UML:DiagramElement geometry="60,40,60,910"
preferredShapeType="InteractionActor" subjec
t="wErkIVSGAqACBSJk" xmi.id="QErkIVSGAqACBSJ
j">
<XMI.extension xmi.extender="Visual Paradigm
for UML">
<captionBounds value="62,75,56,15"/>
<captionVisible value="true"/>
<captionSide value="12"/>
<properties>
<abackground value="Cr:122,207,245,255"/>
<aforeground value="Cr:0,0,0,255"/>
<azOrder value="54"/>
<aselectable value="true"/>
<arequestResetCaption value="false"/>
<arequestResetCaptionSize value="false"/>
<adisplayStereotypeIcon value="false"/>
<aoverrideAppearanceWithStereotypeIcon value
="true"/>
<arequestSetSizeOption value="0"/>
<aparentConnectorHeaderLength value="40"/>
<aparentConnectorLineLength value="10"/>
<aconnectToPoint value="true"/>
<aparentConnectorDTheta value="0.0"/>
<aconnectionPointType value="2"/>
<amodelElementNameAlignment value="9"/>
<arequestDefaultSize value="false"/>
<arequestFitSize value="false"/>
<arequestFitSizeFromCenter value="false"/>
<aparentFrame value="J9BkIVSGAqACBSIb"/>
</properties>
</XMI.extension>
<elementFill color1="Cr:122,207,245,255" col
or2="" style="1" transparency="0" type="1"/>
<elementFont bold="false" color="Cr:0,0,0,25
5" italic="false" name="Dialog" size="11" st
yle="0"/>
<elementLine color="Cr:0,0,0,255" style="1"
transparency="0" weight="1.0"/>
</UML:DiagramElement>
<UML:DiagramElement geometry="210,40,80,910"
preferredShapeType="InteractionLifeLine" su
bject="HkHkIVSGAqACBSLk" xmi.id="HkHkIVSGAqA
CBSLJ">
<XMI.extension xmi.extender="Visual Paradigm
for UML">
<captionBounds value="210,40,80,15"/>
<captionVisible value="true"/>
<captionSide value="12"/>
<properties>
<abackground value="Cr:122,207,245,255"/>
<aforeground value="Cr:0,0,0,255"/>
<azOrder value="52"/>
<aselectable value="true"/>
<arequestResetCaption value="false"/>
<arequestResetCaptionSize value="false"/>
<adisplayStereotypeIcon value="false"/>
<aoverrideAppearanceWithStereotypeIcon value
="true"/>
<arequestSetSizeOption value="0"/>
<aparentConnectorHeaderLength value="40"/>
<aparentConnectorLineLength value="10"/>
<aconnectToPoint value="true"/>
<aparentConnectorDTheta value="0.0"/>
<aconnectionPointType value="2"/>
<amodelElementNameAlignment value="9"/>
<arequestDefaultSize value="false"/>
<arequestFitSize value="false"/>
<arequestFitSizeFromCenter value="false"/>

```



```

<aparentFrame value="J9BkIVSGAqACBSIb"/>
<ashowClassifier value="true"/>
</properties>
</XMI.extension>
<elementFill color1="Cr:122,207,245,255" color2="" style="1" transparency="0" type="1"/>
<elementFont bold="false" color="Cr:0,0,0,255" italic="false" name="Dialog" size="11" style="0"/>
<elementLine color="Cr:0,0,0,255" style="1" transparency="0" weight="1.0"/>
</UML:DiagramElement>
<UML:DiagramElement geometry="380,40,60,910" preferredShapeType="InteractionLifeLine" subject="PtvkIVSGAqACBSNV" xmi.id="PtvkIVSGAqACBSNV">
<XMI.extension xmi.extender="Visual Paradigm for UML">
<captionBounds value="380,40,60,15"/>
<captionVisible value="true"/>
<captionSide value="12"/>
<properties>
<abackground value="Cr:122,207,245,255"/>
<aforeground value="Cr:0,0,0,255"/>
<azOrder value="50"/>
<aselectable value="true"/>
<arequestResetCaption value="false"/>
<arequestResetCaptionSize value="false"/>
<adisplayStereotypeIcon value="false"/>
<aoverrideAppearanceWithStereotypeIcon value="true"/>
<arequestSetSizeOption value="0"/>
<aparentConnectorHeaderLength value="40"/>
<aparentConnectorLineLength value="10"/>
<aconnectToPoint value="true"/>
<aparentConnectorDTheta value="0.0"/>
<aconnectionPointType value="2"/>
<amodelElementNameAlignment value="9"/>
<arequestDefaultSize value="false"/>
<arequestFitSize value="false"/>
<arequestFitSizeFromCenter value="false"/>
<aparentFrame value="J9BkIVSGAqACBSIb"/>
<ashowClassifier value="true"/>
</properties>
</XMI.extension>
<elementFill color1="Cr:122,207,245,255" color2="" style="1" transparency="0" type="1"/>
<elementFont bold="false" color="Cr:0,0,0,255" italic="false" name="Dialog" size="11" style="0"/>
<elementLine color="Cr:0,0,0,255" style="1" transparency="0" weight="1.0"/>
</UML:DiagramElement>
<UML:DiagramElement geometry="530,40,60,910" preferredShapeType="InteractionLifeLine" subject="QMAUIVSGAqACBSPa" xmi.id="QMAUIVSGAqACBSPZ">
<XMI.extension xmi.extender="Visual Paradigm for UML">
<captionBounds value="530,40,60,30"/>
<captionVisible value="true"/>
<captionSide value="12"/>
<properties>
<abackground value="Cr:122,207,245,255"/>
<aforeground value="Cr:0,0,0,255"/>
<azOrder value="48"/>
<aselectable value="true"/>
<arequestResetCaption value="false"/>
<arequestResetCaptionSize value="false"/>
<adisplayStereotypeIcon value="false"/>
<aoverrideAppearanceWithStereotypeIcon value="true"/>
<arequestSetSizeOption value="0"/>
<aparentConnectorHeaderLength value="40"/>
<aparentConnectorLineLength value="10"/>
<aconnectToPoint value="true"/>
<aparentConnectorDTheta value="0.0"/>
<aconnectionPointType value="2"/>
<amodelElementNameAlignment value="9"/>
<arequestDefaultSize value="false"/>
<arequestFitSize value="false"/>
<arequestFitSizeFromCenter value="false"/>
<aparentFrame value="J9BkIVSGAqACBSIb"/>
<ashowClassifier value="true"/>
</properties>
</XMI.extension>
<elementFill color1="Cr:122,207,245,255" color2="" style="1" transparency="0" type="1"/>
<elementFont bold="false" color="Cr:0,0,0,255" italic="false" name="Dialog" size="11" style="0"/>
<elementLine color="Cr:0,0,0,255" style="1" transparency="0" weight="1.0"/>
</UML:DiagramElement>
<UML:DiagramElement geometry="94,130;246,130;" preferredShapeType="Message" subject="91UIVSGAqACBSUh" xmi.id="D1UIVSGAqACBSU8">

```

```

<XMI.extension xmi.extender="Visual Paradigm
for UML">
<captionBounds value="95,110,150,15"/>
<captionVisible value="true"/>
<captionSide value="1"/>
<properties>
<abackground value="Cr:0,0,0,255"/>
<aforeground value="Cr:0,0,0,255"/>
<azOrder value="-1"/>
<aselectable value="true"/>
<arequestResetCaption value="false"/>
<arequestResetCaptionSize value="false"/>
<aconnectorStyle value="3"/>
<aconnectorLineJumps value="4"/>
<aconnectorLabelOrientation value="4"/>
<afromShapeXDiff value="0"/>
<afromShapeYDiff value="0"/>
<atoShapeXDiff value="0"/>
<atoShapeYDiff value="0"/>
<afromConnectType value="0"/>
<atoConnectType value="0"/>
<auseFromShapeCenter value="true"/>
<auseToShapeCenter value="true"/>
<afromPinType value="1"/>
<atoPinType value="1"/>
<arequestRebuild value="false"/>
<ashowConnectorName value="2"/>
<apaintThroughLabel value="2"/>
<afolded value="false"/>
<ashowParameterNamesInMessageOperationSignat
ure value="0"/>
<ashowArgumentsInMessageOperationSignature v
alue="0"/>
<ashowMessageOperationSignature value="0"/>
</properties>
</XMI.extension>
<elementFont bold="false" color="Cr:0,0,0,25
5" italic="false" name="Dialog" size="11" st
yle="0"/>
<elementLine color="Cr:0,0,0,255" style="1"
transparency="0" weight="1.0"/>
</UML:DiagramElement>
<UML:DiagramElement geometry="246,480,8,427"
preferredShapeType="Activation" subject="UN
UIVSGAqACBSVN" xmi.id="UNUIVSGAqACBSVM">
<XMI.extension xmi.extender="Visual Paradigm
for UML">
<captionBounds value="286,120,8,417"/>
<captionVisible value="false"/>
<captionSide value="7"/>
<properties>
<abackground value="Cr:122,207,245,255"/>
<aforeground value="Cr:0,0,0,255"/>
<azOrder value="44"/>
<aselectable value="true"/>
<arequestResetCaption value="false"/>
<arequestResetCaptionSize value="false"/>
<adisplayStereotypeIcon value="false"/>
<aoverrideAppearanceWithStereotypeIcon value
="true"/>
<arequestSetSizeOption value="0"/>
<aparentConnectorHeaderLength value="40"/>
<aparentConnectorLineLength value="10"/>
<aconnectToPoint value="true"/>
<aparentConnectorDTheta value="0.0"/>
<aconnectionPointType value="2"/>
<amodelElementNameAlignment value="9"/>
<arequestDefaultSize value="false"/>
<arequestFitSize value="false"/>
<arequestFitSizeFromCenter value="false"/>
</properties>
</XMI.extension>
<elementFill color1="Cr:122,207,245,255" col
or2="" style="1" transparency="0" type="1"/>
<elementFont bold="false" color="Cr:0,0,0,25
5" italic="false" name="Dialog" size="11" st
yle="0"/>
<elementLine color="Cr:0,0,0,255" style="1"
transparency="0" weight="1.0"/>
</UML:DiagramElement>
<UML:DiagramElement geometry="254,140;300,14
0;300,160;254,160;" preferredShapeType="Mess
age" subject="h72UIVSGAqACBSZU" xmi.id="h72U
IVSGAqACBSZv">
<XMI.extension xmi.extender="Visual Paradigm
for UML">
<captionBounds value="300,140,107,15"/>
<captionVisible value="true"/>
<captionSide value="1"/>
<properties>
<abackground value="Cr:0,0,0,255"/>
<aforeground value="Cr:0,0,0,255"/>
<azOrder value="-1"/>
<aselectable value="true"/>
<arequestResetCaption value="false"/>
<arequestResetCaptionSize value="false"/>
<aconnectorStyle value="3"/>
<aconnectorLineJumps value="4"/>
<aconnectorLabelOrientation value="4"/>
<afromShapeXDiff value="0"/>
<afromShapeYDiff value="0"/>
<atoShapeXDiff value="0"/>
<atoShapeYDiff value="0"/>
<afromConnectType value="0"/>
<atoConnectType value="0"/>
<auseFromShapeCenter value="true"/>
<auseToShapeCenter value="true"/>
<afromPinType value="1"/>
<atoPinType value="1"/>
<arequestRebuild value="false"/>
<ashowConnectorName value="2"/>
<apaintThroughLabel value="2"/>
<afolded value="false"/>
<ashowParameterNamesInMessageOperationSignat
ure value="0"/>
<ashowArgumentsInMessageOperationSignature v
alue="0"/>
<ashowMessageOperationSignature value="0"/>
</properties>
</XMI.extension>
<elementFont bold="false" color="Cr:0,0,0,25
5" italic="false" name="Dialog" size="11" st
yle="0"/>
<elementLine color="Cr:0,0,0,255" style="1"
transparency="0" weight="1.0"/>
</UML:DiagramElement>
<UML:DiagramElement geometry="254,170;406,17
0;" preferredShapeType="Message" subject="Gm
HUIVSGAqACBSjF" xmi.id="mmHUIVSGAqACBSjg">
<XMI.extension xmi.extender="Visual Paradigm
for UML">
<captionBounds value="278,150,103,15"/>
<captionVisible value="true"/>
<captionSide value="1"/>
<properties>
<abackground value="Cr:0,0,0,255"/>
<aforeground value="Cr:0,0,0,255"/>
<azOrder value="-1"/>

```

```

<aselectable value="true"/>
<arequestResetCaption value="false"/>
<arequestResetCaptionSize value="false"/>
<aconnectorStyle value="3"/>
<aconnectorLineJumps value="4"/>
<aconnectorLabelOrientation value="4"/>
<afromShapeXDiff value="0"/>
<afromShapeYDiff value="0"/>
<atoShapeXDiff value="0"/>
<atoShapeYDiff value="0"/>
<afromConnectType value="0"/>
<atoConnectType value="0"/>
<auseFromShapeCenter value="true"/>
<auseToShapeCenter value="true"/>
<afromPinType value="1"/>
<atoPinType value="1"/>
<arequestRebuild value="false"/>
<ashowConnectorName value="2"/>
<apaintThroughLabel value="2"/>
<afolded value="false"/>
<ashowParameterNamesInMessageOperationSignature value="0"/>
<ashowArgumentsInMessageOperationSignature value="0"/>
<ashowMessageOperationSignature value="0"/>
</properties>
</XMI.extension>
<elementFont bold="false" color="Cr:0,0,0,255" italic="false" name="Dialog" size="11" style="0"/>
<elementLine color="Cr:0,0,0,255" style="1" transparency="0" weight="1.0"/>
</UML:DiagramElement>
<UML:DiagramElement geometry="406,160,8,130" preferredShapeType="Activation" subject="YWHUIVSGAqACBSkI" xmi.id="YWHUIVSGAqACBSkH">
<XMI.extension xmi.extender="Visual Paradigm for UML">
<captionBounds value="486,160,8,130"/>
<captionVisible value="false"/>
<captionSide value="7"/>
<properties>
<abackground value="Cr:122,207,245,255"/>
<aforeground value="Cr:0,0,0,255"/>
<azOrder value="40"/>
<aselectable value="true"/>
<arequestResetCaption value="false"/>
<arequestResetCaptionSize value="false"/>
<adisplayStereotypeIcon value="false"/>
<aoverrideAppearanceWithStereotypeIcon value="true"/>
<arequestSetSizeOption value="0"/>
<aparentConnectorHeaderLength value="40"/>
<aparentConnectorLineLength value="10"/>
<aconnectToPoint value="true"/>
<aparentConnectorDTheta value="0.0"/>
<aconnectionPointType value="2"/>
<amodelElementNameAlignment value="9"/>
<arequestDefaultSize value="false"/>
<arequestFitSize value="false"/>
<arequestFitSizeFromCenter value="false"/>
</properties>
</XMI.extension>
<elementFill color1="Cr:122,207,245,255" color2="" style="1" transparency="0" type="1"/>
<elementFont bold="false" color="Cr:0,0,0,255" italic="false" name="Dialog" size="11" style="0"/>
<elementLine color="Cr:0,0,0,255" style="1" transparency="0" weight="1.0"/>
</UML:DiagramElement>
<UML:DiagramElement geometry="406,190;254,190;" preferredShapeType="Message" subject="iSA0IVSGAqACBS17" xmi.id="SSA0IVSGAqACBSmW">
<XMI.extension xmi.extender="Visual Paradigm for UML">
<captionBounds value="290,170,81,15"/>
<captionVisible value="true"/>
<captionSide value="1"/>
<properties>
<abackground value="Cr:0,0,0,255"/>
<aforeground value="Cr:0,0,0,255"/>
<azOrder value="-1"/>
<aselectable value="true"/>
<arequestResetCaption value="false"/>
<arequestResetCaptionSize value="false"/>
<aconnectorStyle value="3"/>
<aconnectorLineJumps value="4"/>
<aconnectorLabelOrientation value="4"/>
<afromShapeXDiff value="0"/>
<afromShapeYDiff value="0"/>
<atoShapeXDiff value="0"/>
<atoShapeYDiff value="0"/>
<afromConnectType value="0"/>
<atoConnectType value="0"/>
<auseFromShapeCenter value="true"/>
<auseToShapeCenter value="true"/>
<afromPinType value="1"/>
<atoPinType value="1"/>
<arequestRebuild value="false"/>
<ashowConnectorName value="2"/>
<apaintThroughLabel value="2"/>
<afolded value="false"/>
<ashowParameterNamesInMessageOperationSignature value="0"/>
<ashowArgumentsInMessageOperationSignature value="0"/>
<ashowMessageOperationSignature value="0"/>
</properties>
</XMI.extension>
<elementFont bold="false" color="Cr:0,0,0,255" italic="false" name="Dialog" size="11" style="0"/>
<elementLine color="Cr:0,0,0,255" style="1" transparency="0" weight="1.0"/>
</UML:DiagramElement>
<UML:DiagramElement geometry="254,230;406,230;" preferredShapeType="Message" subject="Okk0IVSGAqACBSpX" xmi.id="ukk0IVSGAqACBSpy">
<XMI.extension xmi.extender="Visual Paradigm for UML">
<captionBounds value="275,210,110,15"/>
<captionVisible value="true"/>
<captionSide value="1"/>
<properties>
<abackground value="Cr:0,0,0,255"/>
<aforeground value="Cr:0,0,0,255"/>
<azOrder value="-1"/>
<aselectable value="true"/>
<arequestResetCaption value="false"/>
<arequestResetCaptionSize value="false"/>
<aconnectorStyle value="3"/>
<aconnectorLineJumps value="4"/>
<aconnectorLabelOrientation value="4"/>
<afromShapeXDiff value="0"/>
<afromShapeYDiff value="0"/>
<atoShapeXDiff value="0"/>

```

```

<atoShapeYDiff value="0"/>
<afromConnectType value="0"/>
<atoConnectType value="0"/>
<auseFromShapeCenter value="true"/>
<auseToShapeCenter value="true"/>
<afromPinType value="1"/>
<atoPinType value="1"/>
<arequestRebuild value="false"/>
<ashowConnectorName value="2"/>
<apaintThroughLabel value="2"/>
<afolded value="false"/>
<ashowParameterNamesInMessageOperationSignature value="0"/>
<ashowArgumentsInMessageOperationSignature value="0"/>
<ashowMessageOperationSignature value="0"/>
</properties>
</XMI.extension>
<elementFont bold="false" color="Cr:0,0,0,255" italic="false" name="Dialog" size="11" style="0"/>
<elementLine color="Cr:0,0,0,255" style="1" transparency="0" weight="1.0"/>
</UML:DiagramElement>
<UML:DiagramElement geometry="406,280;254,280;" preferredShapeType="Message" subject="jnM0IVSGAqACBSsq" xmi.id="TnM0IVSGAqACBStF">
<XMI.extension xmi.extender="Visual Paradigm for UML">
<captionBounds value="306,260,48,15"/>
<captionVisible value="true"/>
<captionSide value="1"/>
<properties>
<abackground value="Cr:0,0,0,255"/>
<aforeground value="Cr:0,0,0,255"/>
<azOrder value="-1"/>
<aselectable value="true"/>
<arequestResetCaption value="false"/>
<arequestResetCaptionSize value="false"/>
<aconnectorStyle value="3"/>
<aconnectorLineJumps value="4"/>
<aconnectorLabelOrientation value="4"/>
<afromShapeXDiff value="0"/>
<afromShapeYDiff value="0"/>
<atoShapeXDiff value="0"/>
<atoShapeYDiff value="0"/>
<afromConnectType value="0"/>
<atoConnectType value="0"/>
<auseFromShapeCenter value="true"/>
<auseToShapeCenter value="true"/>
<afromPinType value="1"/>
<atoPinType value="1"/>
<arequestRebuild value="false"/>
<ashowConnectorName value="2"/>
<apaintThroughLabel value="2"/>
<afolded value="false"/>
<ashowParameterNamesInMessageOperationSignature value="0"/>
<ashowArgumentsInMessageOperationSignature value="0"/>
<ashowMessageOperationSignature value="0"/>
</properties>
</XMI.extension>
<elementFont bold="false" color="Cr:0,0,0,255" italic="false" name="Dialog" size="11" style="0"/>
<elementLine color="Cr:0,0,0,255" style="1" transparency="0" weight="1.0"/>
</UML:DiagramElement>
<UML:DiagramElement geometry="556,300,8,91" preferredShapeType="Activation" subject="5cS0IVSGAqACBSw5" xmi.id="5cS0IVSGAqACBSw4">
<XMI.extension xmi.extender="Visual Paradigm for UML">
<captionBounds value="706,280,8,90"/>
<captionVisible value="false"/>
<captionSide value="7"/>
<properties>
<abackground value="Cr:122,207,245,255"/>
<aforeground value="Cr:0,0,0,255"/>
<azOrder value="36"/>
<aselectable value="true"/>
<arequestResetCaption value="false"/>
<arequestResetCaptionSize value="false"/>
<adisplayStereotypeIcon value="false"/>
<aoverrideAppearanceWithStereotypeIcon value="true"/>
<arequestSetSizeOption value="0"/>
<aparentConnectorHeaderLength value="40"/>
<aparentConnectorLineLength value="10"/>
<aconnectToPoint value="true"/>
<aparentConnectorDTheta value="0.0"/>
<aconnectionPointType value="2"/>

```

```

<amodelElementNameAlignment value="9"/>
<arequestDefaultSize value="false"/>
<arequestFitSize value="false"/>
<arequestFitSizeFromCenter value="false"/>
</properties>
</XMI.extension>
<elementFill color1="Cr:122,207,245,255" color2="" style="1" transparency="0" type="1"/>
<elementFont bold="false" color="Cr:0,0,0,255" italic="false" name="Dialog" size="11" style="0"/>
<elementLine color="Cr:0,0,0,255" style="1" transparency="0" weight="1.0"/>
</UML:DiagramElement>
<UML:DiagramElement geometry="556,330;254,330;" preferredShapeType="Message" subject="ph60IVSGAqACBSyH" xmi.id="Zh60IVSGAqACBSyi">
<XMI.extension xmi.extender="Visual Paradigm for UML">
<captionBounds value="357,310,96,15"/>
<captionVisible value="true"/>
<captionSide value="1"/>
<properties>
<abackground value="Cr:0,0,0,255"/>
<aforeground value="Cr:0,0,0,255"/>
<azOrder value="-1"/>
<aselectable value="true"/>
<arequestResetCaption value="false"/>
<arequestResetCaptionSize value="false"/>
<connectorStyle value="3"/>
<connectorLineJumps value="4"/>
<connectorLabelOrientation value="4"/>
<fromShapeXDiff value="0"/>
<fromShapeYDiff value="0"/>
<toShapeXDiff value="0"/>
<toShapeYDiff value="0"/>
<fromConnectType value="0"/>
<toConnectType value="0"/>
<useFromShapeCenter value="true"/>
<useToShapeCenter value="true"/>
<fromPinType value="1"/>
<toPinType value="1"/>
<arequestRebuild value="false"/>
<showConnectorName value="2"/>
<paintThroughLabel value="2"/>
<afolded value="false"/>
<showParameterNamesInMessageOperationSignature value="0"/>
<showArgumentsInMessageOperationSignature value="0"/>
<showMessageOperationSignature value="0"/>
</properties>
</XMI.extension>
<elementFont bold="false" color="Cr:0,0,0,255" italic="false" name="Dialog" size="11" style="0"/>
<elementLine color="Cr:0,0,0,255" style="1" transparency="0" weight="1.0"/>
</UML:DiagramElement>
<UML:DiagramElement geometry="556,380;254,380;" preferredShapeType="Message" subject="WB0IVSGAqACBS4f" xmi.id="2BB0IVSGAqACBS46">
<XMI.extension xmi.extender="Visual Paradigm for UML">
<captionBounds value="367,360,76,15"/>
<captionVisible value="true"/>
<captionSide value="1"/>
<properties>
<abackground value="Cr:0,0,0,255"/>
<aforeground value="Cr:0,0,0,255"/>
<azOrder value="-1"/>
<aselectable value="true"/>
<arequestResetCaption value="false"/>
<arequestResetCaptionSize value="false"/>
<connectorStyle value="3"/>
<connectorLineJumps value="4"/>
<connectorLabelOrientation value="4"/>
<fromShapeXDiff value="0"/>
<fromShapeYDiff value="0"/>
<toShapeXDiff value="0"/>
<toShapeYDiff value="0"/>
<fromConnectType value="0"/>
<toConnectType value="0"/>
<useFromShapeCenter value="true"/>
<useToShapeCenter value="true"/>
<fromPinType value="1"/>
<toPinType value="1"/>
<arequestRebuild value="false"/>
<showConnectorName value="2"/>
<paintThroughLabel value="2"/>
<afolded value="false"/>
<showParameterNamesInMessageOperationSignature value="0"/>

```

```

<ashowArgumentsInMessageOperationSignature value="0"/>
<ashowMessageOperationSignature value="0"/>
</properties>
</XMI.extension>
<elementFont bold="false" color="Cr:0,0,0,255" italic="false" name="Dialog" size="11" style="0"/>
<elementLine color="Cr:0,0,0,255" style="1" transparency="0" weight="1.0"/>
</UML:DiagramElement>
<UML:DiagramElement geometry="246,410;94,410" preferredShapeType="Message" subject="OQV0IVSGAqACBS8P" xmi.id="uQV0IVSGAqACBS8q">
<XMI.extension xmi.extender="Visual Paradigm for UML">
<captionBounds value="108,390,128,15"/>
<captionVisible value="true"/>
<captionSide value="1"/>
<properties>
<abackground value="Cr:0,0,0,255"/>
<aforeground value="Cr:0,0,0,255"/>
<azOrder value="-1"/>
<aselectable value="true"/>
<arequestResetCaption value="false"/>
<arequestResetCaptionSize value="false"/>
<aconnectorStyle value="3"/>
<aconnectorLineJumps value="4"/>
<aconnectorLabelOrientation value="4"/>
<afromShapeXDiff value="0"/>
<afromShapeYDiff value="0"/>
<atoShapeXDiff value="0"/>
<atoShapeYDiff value="0"/>
<afromConnectType value="0"/>
<atoConnectType value="0"/>
<auseFromShapeCenter value="true"/>
<auseToShapeCenter value="true"/>
<afromPinType value="1"/>
<atoPinType value="1"/>
<arequestRebuild value="false"/>
<ashowConnectorName value="2"/>
<apaintThroughLabel value="2"/>
<afolded value="false"/>
<ashowParameterNamesInMessageOperationSignature value="0"/>
<ashowArgumentsInMessageOperationSignature value="0"/>
<ashowMessageOperationSignature value="0"/>
</properties>
</XMI.extension>
<elementFont bold="false" color="Cr:0,0,0,255" italic="false" name="Dialog" size="11" style="0"/>
<elementLine color="Cr:0,0,0,255" style="1" transparency="0" weight="1.0"/>
</UML:DiagramElement>
<UML:DiagramElement geometry="70,200,510,244" preferredShapeType="CombinedFragment" subject="4Yb0IVSGAqACBS_T" xmi.id="4Yb0IVSGAqACBS_S">
<XMI.extension xmi.extender="Visual Paradigm for UML">
<captionBounds value="0,0,23,18"/>
<captionVisible value="true"/>
<captionSide value="12"/>
<properties>
<abackground value="Cr:122,207,245,0"/>
<aforeground value="Cr:0,0,0,255"/>
<azOrder value="38"/>
<aselectable value="true"/>
<arequestResetCaption value="false"/>
<arequestResetCaptionSize value="false"/>
<adisplayStereotypeIcon value="false"/>
<aoverrideAppearanceWithStereotypeIcon value="true"/>
<arequestSetSizeOption value="0"/>
<aparentConnectorHeaderLength value="40"/>
<aparentConnectorLineLength value="10"/>
<aconnectToPoint value="true"/>
<aparentConnectorDTheta value="0.0"/>
<aconnectionPointType value="2"/>
<amodelElementNameAlignment value="9"/>
<arequestDefaultSize value="false"/>
<arequestFitSize value="false"/>
<arequestFitSizeFromCenter value="false"/>
</properties>
</XMI.extension>
<elementFill color1="Cr:122,207,245,0" color2="" style="1" transparency="100" type="1"/>
<elementFont bold="false" color="Cr:0,0,0,255" italic="false" name="Dialog" size="11" style="0"/>
<elementLine color="Cr:0,0,0,255" style="1" transparency="0" weight="1.0"/>
</UML:DiagramElement>
<UML:DiagramElement geometry="70,218,510,50" preferredShapeType="InteractionOperand" subject="H4b0IVSGAqACBS_k" xmi.id="74b0IVSGAqACBS_j">
<XMI.extension xmi.extender="Visual Paradigm for UML">
<captionBounds value="80,218,660,50"/>
<captionVisible value="false"/>
<captionSide value="7"/>
<properties>
<abackground value="Cr:122,207,245,255"/>
<aforeground value="Cr:0,0,0,255"/>
<azOrder value="8"/>
<aselectable value="true"/>
<arequestResetCaption value="false"/>
<arequestResetCaptionSize value="false"/>
<adisplayStereotypeIcon value="false"/>
<aoverrideAppearanceWithStereotypeIcon value="true"/>
<arequestSetSizeOption value="0"/>
<aparentConnectorHeaderLength value="40"/>
<aparentConnectorLineLength value="10"/>
<aconnectToPoint value="true"/>
<aparentConnectorDTheta value="0.0"/>
<aconnectionPointType value="2"/>
<amodelElementNameAlignment value="9"/>
<arequestDefaultSize value="false"/>
<arequestFitSize value="false"/>
<arequestFitSizeFromCenter value="false"/>
</properties>
</XMI.extension>
<elementFill color1="Cr:122,207,245,255" color2="" style="1" transparency="0" type="1"/>
<elementFont bold="false" color="Cr:0,0,0,255" italic="false" name="Dialog" size="11" style="0"/>
<elementLine color="Cr:0,0,0,255" style="1" transparency="0" weight="1.0"/>
</UML:DiagramElement>
<UML:DiagramElement geometry="70,268,510,176" preferredShapeType="InteractionOperand" subject="OEb0IVSGAqACBS_3" xmi.id="OEb0IVSGAqACBS_2">

```

```

<XMI.extension xmi.extender="Visual Paradigm
for UML">
<captionBounds value="70,268,510,176"/>
<captionVisible value="false"/>
<captionSide value="7"/>
<properties>
<abackground value="Cr:122,207,245,255"/>
<aforeground value="Cr:0,0,0,255"/>
<azOrder value="10"/>
<aselectable value="true"/>
<arequestResetCaption value="false"/>
<arequestResetCaptionSize value="false"/>
<adisplayStereotypeIcon value="false"/>
<aoverrideAppearanceWithStereotypeIcon value
="true"/>
<arequestSetSizeOption value="0"/>
<aparentConnectorHeaderLength value="40"/>
<aparentConnectorLineLength value="10"/>
<aconnectToPoint value="true"/>
<aparentConnectorDTheta value="0.0"/>
<aconnectionPointType value="2"/>
<amodelElementNameAlignment value="9"/>
<arequestDefaultSize value="false"/>
<arequestFitSize value="false"/>
<arequestFitSizeFromCenter value="false"/>
</properties>
</XMI.extension>
<elementFill color1="Cr:122,207,245,255" col
or2="" style="1" transparency="0" type="1"/>
<elementFont bold="false" color="Cr:0,0,0,25
5" italic="false" name="Dialog" size="11" st
yle="0"/>
<elementLine color="Cr:0,0,0,255" style="1"
transparency="0" weight="1.0"/>
</UML:DiagramElement>
<UML:DiagramElement geometry="50,130,560,330
" preferredShapeType="CombinedFragment" subj
ect="V3WMIVSGAqACBTfd" xmi.id="V3WMIVSGAqACB
Tfc">
<XMI.extension xmi.extender="Visual Paradigm
for UML">
<captionBounds value="0,0,33,18"/>
<captionVisible value="true"/>
<captionSide value="12"/>
<properties>
<abackground value="Cr:122,207,245,0"/>
<aforeground value="Cr:0,0,0,255"/>
<azOrder value="42"/>
<aselectable value="true"/>
<arequestResetCaption value="false"/>
<arequestResetCaptionSize value="false"/>
<adisplayStereotypeIcon value="false"/>
<aoverrideAppearanceWithStereotypeIcon value
="true"/>
<arequestSetSizeOption value="0"/>
<aparentConnectorHeaderLength value="40"/>
<aparentConnectorLineLength value="10"/>
<aconnectToPoint value="true"/>
<aparentConnectorDTheta value="0.0"/>
<aconnectionPointType value="2"/>
<amodelElementNameAlignment value="9"/>
<arequestDefaultSize value="false"/>
<arequestFitSize value="false"/>
<arequestFitSizeFromCenter value="false"/>
</properties>
</XMI.extension>
<elementFill color1="Cr:122,207,245,0" color
2="" style="1" transparency="100" type="1"/>
<elementFont bold="false" color="Cr:0,0,0,25
5" italic="false" name="Dialog" size="11" st
yle="0"/>
<elementLine color="Cr:0,0,0,255" style="1"
transparency="0" weight="1.0"/>
</UML:DiagramElement>
<UML:DiagramElement geometry="50,148,560,312
" preferredShapeType="InteractionOperand" su
bject="L3WMIVSGAqACBTft" xmi.id="L3WMIVSGAqA
CBTfs">
<XMI.extension xmi.extender="Visual Paradigm
for UML">
<captionBounds value="0,0,20,50"/>
<captionVisible value="false"/>
<captionSide value="7"/>
<properties>
<abackground value="Cr:122,207,245,255"/>
<aforeground value="Cr:0,0,0,255"/>
<azOrder value="12"/>
<aselectable value="true"/>
<arequestResetCaption value="false"/>
<arequestResetCaptionSize value="false"/>
<adisplayStereotypeIcon value="false"/>
<aoverrideAppearanceWithStereotypeIcon value
="true"/>
<arequestSetSizeOption value="0"/>
<aparentConnectorHeaderLength value="40"/>
<aparentConnectorLineLength value="10"/>
<aconnectToPoint value="true"/>
<aparentConnectorDTheta value="0.0"/>
<aconnectionPointType value="2"/>
<amodelElementNameAlignment value="9"/>
<arequestDefaultSize value="false"/>
<arequestFitSize value="false"/>
<arequestFitSizeFromCenter value="false"/>
</properties>
</XMI.extension>
<elementFill color1="Cr:122,207,245,255" col
or2="" style="1" transparency="0" type="1"/>
<elementFont bold="false" color="Cr:0,0,0,25
5" italic="false" name="Dialog" size="11" st
yle="0"/>
<elementLine color="Cr:0,0,0,255" style="1"
transparency="0" weight="1.0"/>
</UML:DiagramElement>
<UML:DiagramElement geometry="94,480;246,480
;" preferredShapeType="Message" subject="wKL
MIVSGAqACBT00" xmi.id="IKLMIVSGAqACBT0p">
<XMI.extension xmi.extender="Visual Paradigm
for UML">
<captionBounds value="123,460,95,15"/>
<captionVisible value="true"/>
<captionSide value="1"/>
<properties>
<abackground value="Cr:0,0,0,255"/>
<aforeground value="Cr:0,0,0,255"/>
<azOrder value="-1"/>
<aselectable value="true"/>
<arequestResetCaption value="false"/>
<arequestResetCaptionSize value="false"/>
<aconnectorStyle value="3"/>
<aconnectorLineJumps value="4"/>
<aconnectorLabelOrientation value="4"/>
<afromShapeXDiff value="0"/>
<afromShapeYDiff value="0"/>
<atoShapeXDiff value="0"/>
<atoShapeYDiff value="0"/>
<afromConnectType value="0"/>
<atoConnectType value="0"/>

```

```

<auseFromShapeCenter value="true"/>
<auseToShapeCenter value="true"/>
<afromPinType value="1"/>
<atoPinType value="1"/>
<arequestRebuild value="false"/>
<ashowConnectorName value="2"/>
<apaintThroughLabel value="2"/>
<afolded value="false"/>
<ashowParameterNamesInMessageOperationSignature value="0"/>
<ashowArgumentsInMessageOperationSignature value="0"/>
<ashowMessageOperationSignature value="0"/>
</properties>
</XMI.extension>
<elementFont bold="false" color="Cr:0,0,0,255" italic="false" name="Dialog" size="11" style="0"/>
<elementLine color="Cr:0,0,0,255" style="1" transparency="0" weight="1.0"/>
</UML:DiagramElement>
<UML:DiagramElement geometry="254,500;330,500;330,520;254,520;" preferredShapeType="Message" subject="bB.sIVSGAqACBT_V" xmi.id="7B.sIVSGAqACBT_w"/>
<XMI.extension xmi.extender="Visual Paradigm for UML">
<captionBounds value="340,500,98,15"/>
<captionVisible value="true"/>
<captionSide value="1"/>
<properties>
<abackground value="Cr:0,0,0,255"/>
<aforeground value="Cr:0,0,0,255"/>
<azOrder value="-1"/>
<aselectable value="true"/>
<arequestResetCaption value="false"/>
<arequestResetCaptionSize value="false"/>
<aconnectorStyle value="3"/>
<aconnectorLineJumps value="4"/>
<aconnectorLabelOrientation value="4"/>
<afromShapeXDiff value="0"/>
<afromShapeYDiff value="0"/>
<atoShapeXDiff value="0"/>
<atoShapeYDiff value="0"/>
<afromConnectType value="0"/>
<atoConnectType value="0"/>
<auseFromShapeCenter value="true"/>
<auseToShapeCenter value="true"/>
<afromPinType value="1"/>
<atoPinType value="1"/>
<arequestRebuild value="false"/>
<ashowConnectorName value="2"/>
<apaintThroughLabel value="2"/>
<afolded value="false"/>
<ashowParameterNamesInMessageOperationSignature value="0"/>
<ashowArgumentsInMessageOperationSignature value="0"/>
<ashowMessageOperationSignature value="0"/>
</properties>
</XMI.extension>
<elementFont bold="false" color="Cr:0,0,0,255" italic="false" name="Dialog" size="11" style="0"/>
<elementLine color="Cr:0,0,0,255" style="1" transparency="0" weight="1.0"/>
</UML:DiagramElement>
<UML:DiagramElement geometry="246,130,8,292" preferredShapeType="Activation" subject="cwVsIVSGAqACBUV9" xmi.id="cwVsIVSGAqACBUV8">
<XMI.extension xmi.extender="Visual Paradigm for UML">
<captionBounds value="286,130,8,292"/>
<captionVisible value="false"/>
<captionSide value="7"/>
<properties>
<abackground value="Cr:122,207,245,255"/>
<aforeground value="Cr:0,0,0,255"/>
<azOrder value="34"/>
<aselectable value="true"/>
<arequestResetCaption value="false"/>
<arequestResetCaptionSize value="false"/>
<adisplayStereotypeIcon value="false"/>
<aoverrideAppearanceWithStereotypeIcon value="true"/>
<arequestSetSizeOption value="0"/>
<aparentConnectorHeaderLength value="40"/>
<aparentConnectorLineLength value="10"/>
<aconnectToPoint value="true"/>
<aparentConnectorDTheta value="0.0"/>
<aconnectionPointType value="2"/>
<amodelElementNameAlignment value="9"/>
<arequestDefaultSize value="false"/>
<arequestFitSize value="false"/>
<arequestFitSizeFromCenter value="false"/>
</properties>
</XMI.extension>
<elementFill color1="Cr:122,207,245,255" color2="" style="1" transparency="0" type="1"/>
<elementFont bold="false" color="Cr:0,0,0,255" italic="false" name="Dialog" size="11" style="0"/>
<elementLine color="Cr:0,0,0,255" style="1" transparency="0" weight="1.0"/>
</UML:DiagramElement>
<UML:DiagramElement geometry="254,540;406,540;" preferredShapeType="Message" subject="SIjsIVSGAqACBUa9" xmi.id="yIjsIVSGAqACBUbY">
<XMI.extension xmi.extender="Visual Paradigm for UML">
<captionBounds value="283,520,95,15"/>
<captionVisible value="true"/>
<captionSide value="1"/>
<properties>
<abackground value="Cr:0,0,0,255"/>
<aforeground value="Cr:0,0,0,255"/>
<azOrder value="-1"/>
<aselectable value="true"/>
<arequestResetCaption value="false"/>
<arequestResetCaptionSize value="false"/>
<aconnectorStyle value="3"/>
<aconnectorLineJumps value="4"/>
<aconnectorLabelOrientation value="4"/>
<afromShapeXDiff value="0"/>
<afromShapeYDiff value="0"/>
<atoShapeXDiff value="0"/>
<atoShapeYDiff value="0"/>
<afromConnectType value="0"/>
<atoConnectType value="0"/>
<auseFromShapeCenter value="true"/>
<auseToShapeCenter value="true"/>
<afromPinType value="1"/>
<atoPinType value="1"/>
<arequestRebuild value="false"/>
<ashowConnectorName value="2"/>
<apaintThroughLabel value="2"/>

```



```

<afolded value="false"/>
<ashowParameterNamesInMessageOperationSignature value="0"/>
<ashowArgumentsInMessageOperationSignature value="0"/>
<ashowMessageOperationSignature value="0"/>
</properties>
</XMI.extension>
<elementFont bold="false" color="Cr:0,0,0,255" italic="false" name="Dialog" size="11" style="0"/>
<elementLine color="Cr:0,0,0,255" style="1" transparency="0" weight="1.0"/>
</UML:DiagramElement>
<UML:DiagramElement geometry="406,540,8,40" preferredShapeType="Activation" subject="DIjsIVSGAqACBUcR" xmi.id="DIjsIVSGAqACBUcQ">
<XMI.extension xmi.extender="Visual Paradigm for UML">
<captionBounds value="486,540,8,40"/>
<captionVisible value="false"/>
<captionSide value="7"/>
<properties>
<abackground value="Cr:122,207,245,255"/>
<aforeground value="Cr:0,0,0,255"/>
<azOrder value="32"/>
<aselectable value="true"/>
<arequestResetCaption value="false"/>
<arequestResetCaptionSize value="false"/>
<adisplayStereotypeIcon value="false"/>
<aoverrideAppearanceWithStereotypeIcon value="true"/>
<arequestSetSizeOption value="0"/>
<aparentConnectorHeaderLength value="40"/>
<aparentConnectorLineLength value="10"/>
<aconnectToPoint value="true"/>
<aparentConnectorDTheta value="0.0"/>
<aconnectionPointType value="2"/>
<amodelElementNameAlignment value="9"/>
<arequestDefaultSize value="false"/>
<arequestFitSize value="false"/>
<arequestFitSizeFromCenter value="false"/>
</properties>
</XMI.extension>
<elementFill color1="Cr:122,207,245,255" color2="" style="1" transparency="0" type="1"/>
<elementFont bold="false" color="Cr:0,0,0,255" italic="false" name="Dialog" size="11" style="0"/>
<elementLine color="Cr:0,0,0,255" style="1" transparency="0" weight="1.0"/>
</UML:DiagramElement>
<UML:DiagramElement geometry="406,570;254,570;" preferredShapeType="Message" subject="TrsIVSGAqACBUiJ" xmi.id="BTrsIVSGAqACBUik">
<XMI.extension xmi.extender="Visual Paradigm for UML">
<captionBounds value="290,550,81,15"/>
<captionVisible value="true"/>
<captionSide value="1"/>
<properties>
<abackground value="Cr:0,0,0,255"/>
<aforeground value="Cr:0,0,0,255"/>
<azOrder value="-1"/>
<aselectable value="true"/>
<arequestResetCaption value="false"/>
<arequestResetCaptionSize value="false"/>
<aconnectorStyle value="3"/>
<aconnectorLineJumps value="4"/>
<aconnectorLabelOrientation value="4"/>
<afromShapeXDiff value="0"/>
<afromShapeYDiff value="0"/>
<atoShapeXDiff value="0"/>
<atoShapeYDiff value="0"/>
<afromConnectType value="0"/>
<atoConnectType value="0"/>
<auseFromShapeCenter value="true"/>
<auseToShapeCenter value="true"/>
<afromPinType value="1"/>
<atoPinType value="1"/>
<arequestRebuild value="false"/>
<ashowConnectorName value="2"/>
<apaintThroughLabel value="2"/>
<afolded value="false"/>
<ashowParameterNamesInMessageOperationSignature value="0"/>
<ashowArgumentsInMessageOperationSignature value="0"/>
<ashowMessageOperationSignature value="0"/>
</properties>
</XMI.extension>

```

```

<elementFont bold="false" color="Cr:0,0,0,25
5" italic="false" name="Dialog" size="11" st
yle="0"/>
<elementLine color="Cr:0,0,0,255" style="1"
transparency="0" weight="1.0"/>
</UML:DiagramElement>
<UML:DiagramElement geometry="406,610,8,40"
preferredShapeType="Activation" subject="oDf
sIVSGAqACBUnm" xmi.id="oDfsIVSGAqACBUnl">
<XML:extension xmi.extender="Visual Paradigm
for UML">
<captionBounds value="486,610,8,40"/>
<captionVisible value="false"/>
<captionSide value="7"/>
<properties>
<abackground value="Cr:122,207,245,255"/>
<aforeground value="Cr:0,0,0,255"/>
<azOrder value="30"/>
<aselectable value="true"/>
<arequestResetCaption value="false"/>
<arequestResetCaptionSize value="false"/>
<adisplayStereotypeIcon value="false"/>
<aoverrideAppearanceWithStereotypeIcon value
="true"/>
<arequestSetSizeOption value="0"/>
<aparentConnectorHeaderLength value="40"/>
<aparentConnectorLineLength value="10"/>
<aconnectToPoint value="true"/>
<aparentConnectorDTheta value="0.0"/>
<aconnectionPointType value="2"/>
<amodelElementNameAlignment value="9"/>
<arequestDefaultSize value="false"/>
<arequestFitSize value="false"/>
<arequestFitSizeFromCenter value="false"/>
</properties>
</XML:extension>
<elementFill color1="Cr:122,207,245,255" col
or2="" style="1" transparency="0" type="1"/>
<elementFont bold="false" color="Cr:0,0,0,25
5" italic="false" name="Dialog" size="11" st
yle="0"/>
<elementLine color="Cr:0,0,0,255" style="1"
transparency="0" weight="1.0"/>
</UML:DiagramElement>
<UML:DiagramElement geometry="406,640;254,64
0;" preferredShapeType="Message" subject="Ga
4cIVSGAqACBUuI" xmi.id="ma4cIVSGAqACBUuJ">
<XML:extension xmi.extender="Visual Paradigm
for UML">
<captionBounds value="282,620,96,15"/>
<captionVisible value="true"/>
<captionSide value="1"/>
<properties>
<abackground value="Cr:0,0,0,255"/>
<aforeground value="Cr:0,0,0,255"/>
<azOrder value="-1"/>
<aselectable value="true"/>
<arequestResetCaption value="false"/>
<arequestResetCaptionSize value="false"/>
<aconnectorStyle value="3"/>
<aconnectorLineJumps value="4"/>
<aconnectorLabelOrientation value="4"/>
<afromShapeXDiff value="0"/>
<afromShapeYDiff value="0"/>
<atoShapeXDiff value="0"/>
<atoShapeYDiff value="0"/>
<afromConnectType value="0"/>
<atoConnectType value="0"/>
<auseFromShapeCenter value="true"/>
<auseToShapeCenter value="true"/>
<afromPinType value="1"/>
<atoPinType value="1"/>
<arequestRebuild value="false"/>
<ashowConnectorName value="2"/>
<apaintThroughLabel value="2"/>
<afolded value="false"/>
<ashowParameterNamesInMessageOperationSignat
ure value="0"/>
<ashowArgumentsInMessageOperationSignature v
alue="0"/>
<ashowMessageOperationSignature value="0"/>
</properties>
</XML:extension>
<elementFont bold="false" color="Cr:0,0,0,25
5" italic="false" name="Dialog" size="11" st
yle="0"/>
<elementLine color="Cr:0,0,0,255" style="1"
transparency="0" weight="1.0"/>
</UML:DiagramElement>
<UML:DiagramElement geometry="711,670,8,12"
preferredShapeType="Activation" subject="yh8
cIVSGAqACBU0R" xmi.id="yh8cIVSGAqACBU0Q">
<auseToShapeCenter value="true"/>
<afromPinType value="1"/>
<atoPinType value="1"/>
<arequestRebuild value="false"/>
<ashowConnectorName value="2"/>
<apaintThroughLabel value="2"/>
<afolded value="false"/>
<ashowParameterNamesInMessageOperationSignat
ure value="0"/>
<ashowArgumentsInMessageOperationSignature v
alue="0"/>
<ashowMessageOperationSignature value="0"/>
</properties>
</XML:extension>
<elementFont bold="false" color="Cr:0,0,0,25
5" italic="false" name="Dialog" size="11" st
yle="0"/>
<elementLine color="Cr:0,0,0,255" style="1"
transparency="0" weight="1.0"/>
</UML:DiagramElement>
<UML:DiagramElement geometry="254,670;711,67
0;" preferredShapeType="Message" subject="HB
8cIVSGAqACBUyx" xmi.id="nB8cIVSGAqACBUzm">
<XML:extension xmi.extender="Visual Paradigm
for UML">
<captionBounds value="428,650,109,15"/>
<captionVisible value="true"/>
<captionSide value="1"/>
<properties>
<abackground value="Cr:0,0,0,255"/>
<aforeground value="Cr:0,0,0,255"/>
<azOrder value="-1"/>
<aselectable value="true"/>
<arequestResetCaption value="false"/>
<arequestResetCaptionSize value="false"/>
<aconnectorStyle value="3"/>
<aconnectorLineJumps value="4"/>
<aconnectorLabelOrientation value="4"/>
<afromShapeXDiff value="0"/>
<afromShapeYDiff value="0"/>
<atoShapeXDiff value="0"/>
<atoShapeYDiff value="0"/>
<afromConnectType value="0"/>
<atoConnectType value="0"/>
<auseFromShapeCenter value="true"/>
<auseToShapeCenter value="true"/>
<afromPinType value="1"/>
<atoPinType value="1"/>
<arequestRebuild value="false"/>
<ashowConnectorName value="2"/>
<apaintThroughLabel value="2"/>
<afolded value="false"/>
<ashowParameterNamesInMessageOperationSignat
ure value="0"/>
<ashowArgumentsInMessageOperationSignature v
alue="0"/>
<ashowMessageOperationSignature value="0"/>
</properties>
</XML:extension>
<elementFont bold="false" color="Cr:0,0,0,25
5" italic="false" name="Dialog" size="11" st
yle="0"/>
<elementLine color="Cr:0,0,0,255" style="1"
transparency="0" weight="1.0"/>
</UML:DiagramElement>
<UML:DiagramElement geometry="711,670,8,12"
preferredShapeType="Activation" subject="yh8
cIVSGAqACBU0R" xmi.id="yh8cIVSGAqACBU0Q">

```

```

<XMI.extension xmi.extender="Visual Paradigm
for UML">
<captionBounds value="0,0,8,12"/>
<captionVisible value="false"/>
<captionSide value="7"/>
<properties>
<abackground value="Cr:122,207,245,255"/>
<aforeground value="Cr:0,0,0,255"/>
<azOrder value="28"/>
<aselectable value="true"/>
<arequestResetCaption value="false"/>
<arequestResetCaptionSize value="false"/>
<adisplayStereotypeIcon value="false"/>
<aoverrideAppearanceWithStereotypeIcon value
="true"/>
<arequestSetSizeOption value="0"/>
<aparentConnectorHeaderLength value="40"/>
<aparentConnectorLineLength value="10"/>
<aconnectToPoint value="true"/>
<aparentConnectorDTheta value="0.0"/>
<aconnectionPointType value="2"/>
<amodelElementNameAlignment value="9"/>
<arequestDefaultSize value="false"/>
<arequestFitSize value="false"/>
<arequestFitSizeFromCenter value="false"/>
</properties>
</XMI.extension>
<elementFill color1="Cr:122,207,245,255" col
or2="" style="1" transparency="0" type="1"/>
<elementFont bold="false" color="Cr:0,0,0,25
5" italic="false" name="Dialog" size="11" st
yle="0"/>
<elementLine color="Cr:0,0,0,255" style="1"
transparency="0" weight="1.0"/>
</UML:DiagramElement>
<UML:DiagramElement geometry="254,710;406,71
0;" preferredShapeType="Message" subject="P4
RcIVSGAqACBU21" xmi.id="v4RcIVSGAqACBU3A">
<XMI.extension xmi.extender="Visual Paradigm
for UML">
<captionBounds value="275,690,110,15"/>
<captionVisible value="true"/>
<captionSide value="1"/>
<properties>
<abackground value="Cr:0,0,0,255"/>
<aforeground value="Cr:0,0,0,255"/>
<azOrder value="-1"/>
<aselectable value="true"/>
<arequestResetCaption value="false"/>
<arequestResetCaptionSize value="false"/>
<aconnectorStyle value="3"/>
<aconnectorLineJumps value="4"/>
<aconnectorLabelOrientation value="4"/>
<afromShapeXDiff value="0"/>
<afromShapeYDiff value="0"/>
<atoShapeXDiff value="0"/>
<atoShapeYDiff value="0"/>
<afromConnectType value="0"/>
<atoConnectType value="0"/>
<auseFromShapeCenter value="true"/>
<auseToShapeCenter value="true"/>
<afromPinType value="1"/>
<atoPinType value="1"/>
<arequestRebuild value="false"/>
<ashowConnectorName value="2"/>
<apaintThroughLabel value="2"/>
<afolded value="false"/>
<ashowParameterNamesInMessageOperationSignat
ure value="0"/>
<showArgumentsInMessageOperationSignature v
alue="0"/>
<showMessageOperationSignature value="0"/>
</properties>
</XMI.extension>
<elementFont bold="false" color="Cr:0,0,0,25
5" italic="false" name="Dialog" size="11" st
yle="0"/>
<elementLine color="Cr:0,0,0,255" style="1"
transparency="0" weight="1.0"/>
</UML:DiagramElement>
<UML:DiagramElement geometry="406,710,8,40"
preferredShapeType="Activation" subject="uER
cIVSGAqACBU4I" xmi.id="uERcIVSGAqACBU4H">
<XMI.extension xmi.extender="Visual Paradigm
for UML">
<captionBounds value="486,710,8,40"/>
<captionVisible value="false"/>
<captionSide value="7"/>
<properties>
<abackground value="Cr:122,207,245,255"/>
<aforeground value="Cr:0,0,0,255"/>
<azOrder value="26"/>
<aselectable value="true"/>
<arequestResetCaption value="false"/>
<arequestResetCaptionSize value="false"/>
<adisplayStereotypeIcon value="false"/>
<aoverrideAppearanceWithStereotypeIcon value
="true"/>
<arequestSetSizeOption value="0"/>
<aparentConnectorHeaderLength value="40"/>
<aparentConnectorLineLength value="10"/>
<aconnectToPoint value="true"/>
<aparentConnectorDTheta value="0.0"/>
<aconnectionPointType value="2"/>
<amodelElementNameAlignment value="9"/>
<arequestDefaultSize value="false"/>
<arequestFitSize value="false"/>
<arequestFitSizeFromCenter value="false"/>
</properties>
</XMI.extension>
<elementFill color1="Cr:122,207,245,255" col
or2="" style="1" transparency="0" type="1"/>
<elementFont bold="false" color="Cr:0,0,0,25
5" italic="false" name="Dialog" size="11" st
yle="0"/>
<elementLine color="Cr:0,0,0,255" style="1"
transparency="0" weight="1.0"/>
</UML:DiagramElement>
<UML:DiagramElement geometry="406,740;254,74
0;" preferredShapeType="Message" subject="TB
ZcIVSGAqACBU5j" xmi.id="zBZcIVSGAqACBU5.">
<XMI.extension xmi.extender="Visual Paradigm
for UML">
<captionBounds value="306,720,48,15"/>
<captionVisible value="true"/>
<captionSide value="1"/>
<properties>
<abackground value="Cr:0,0,0,255"/>
<aforeground value="Cr:0,0,0,255"/>
<azOrder value="-1"/>
<aselectable value="true"/>
<arequestResetCaption value="false"/>
<arequestResetCaptionSize value="false"/>
<aconnectorStyle value="3"/>
<aconnectorLineJumps value="4"/>
<aconnectorLabelOrientation value="4"/>
<afromShapeXDiff value="0"/>
<afromShapeYDiff value="0"/>

```

```

<atoShapeXDiff value="0"/>
<atoShapeYDiff value="0"/>
<afromConnectType value="0"/>
<atoConnectType value="0"/>
<auseFromShapeCenter value="true"/>
<auseToShapeCenter value="true"/>
<afromPinType value="1"/>
<atoPinType value="1"/>
<arequestRebuild value="false"/>
<ashowConnectorName value="2"/>
<apaintThroughLabel value="2"/>
<afolded value="false"/>
<ashowParameterNamesInMessageOperationSignature value="0"/>
<ashowArgumentsInMessageOperationSignature value="0"/>
<ashowMessageOperationSignature value="0"/>
</properties>
</XMI.extension>
<elementFont bold="false" color="Cr:0,0,0,255" italic="false" name="Dialog" size="11" style="0"/>
<elementLine color="Cr:0,0,0,255" style="1" transparency="0" weight="1.0"/>
</UML:DiagramElement>
<UML:DiagramElement geometry="254,780;556,780;" preferredShapeType="Message" subject="alVcIVSGAqACBU_w" xmi.id="61VcIVSGAqACBVAL">
<XMI.extension xmi.extender="Visual Paradigm for UML">
<captionBounds value="363,760,84,15"/>
<captionVisible value="true"/>
<captionSide value="1"/>
<properties>
<abackground value="Cr:0,0,0,255"/>
<aforeground value="Cr:0,0,0,255"/>
<azOrder value="-1"/>
<aselectable value="true"/>
<arequestResetCaption value="false"/>
<arequestResetCaptionSize value="false"/>
<aconnectorStyle value="3"/>
<aconnectorLineJumps value="4"/>
<aconnectorLabelOrientation value="4"/>
<afromShapeXDiff value="0"/>
<afromShapeYDiff value="0"/>
<atoShapeXDiff value="0"/>
<atoShapeYDiff value="0"/>
<afromConnectType value="0"/>
<atoConnectType value="0"/>
<auseFromShapeCenter value="true"/>
<auseToShapeCenter value="true"/>
<afromPinType value="1"/>
<atoPinType value="1"/>
<arequestRebuild value="false"/>
<ashowConnectorName value="2"/>
<apaintThroughLabel value="2"/>
<afolded value="false"/>
<ashowParameterNamesInMessageOperationSignature value="0"/>
<ashowArgumentsInMessageOperationSignature value="0"/>
<ashowMessageOperationSignature value="0"/>
</properties>
</XMI.extension>
<elementFont bold="false" color="Cr:0,0,0,255" italic="false" name="Dialog" size="11" style="0"/>
<elementLine color="Cr:0,0,0,255" style="1" transparency="0" weight="1.0"/>
</UML:DiagramElement>
<UML:DiagramElement geometry="556,780,8,40" preferredShapeType="Activation" subject="gNvcIVSGAqACBVBZ" xmi.id="gNvcIVSGAqACBVBZ">
<XMI.extension xmi.extender="Visual Paradigm for UML">
<captionBounds value="706,780,8,40"/>
<captionVisible value="false"/>
<captionSide value="7"/>
<properties>
<abackground value="Cr:122,207,245,255"/>
<aforeground value="Cr:0,0,0,255"/>
<azOrder value="24"/>
<aselectable value="true"/>
<arequestResetCaption value="false"/>
<arequestResetCaptionSize value="false"/>
<adisplayStereotypeIcon value="false"/>
<aoverrideAppearanceWithStereotypeIcon value="true"/>
<arequestSetSizeOption value="0"/>
<aparentConnectorHeaderLength value="40"/>
<aparentConnectorLineLength value="10"/>
<aconnectToPoint value="true"/>
<aparentConnectorDTheta value="0.0"/>
<aconnectionPointType value="2"/>
<amodelElementNameAlignment value="9"/>
<arequestDefaultSize value="false"/>
<arequestFitSize value="false"/>
<arequestFitSizeFromCenter value="false"/>
</properties>
</XMI.extension>
<elementFill color1="Cr:122,207,245,255" color2="" style="1" transparency="0" type="1"/>
<elementFont bold="false" color="Cr:0,0,0,255" italic="false" name="Dialog" size="11" style="0"/>
<elementLine color="Cr:0,0,0,255" style="1" transparency="0" weight="1.0"/>
</UML:DiagramElement>
<UML:DiagramElement geometry="556,810;254,810;" preferredShapeType="Message" subject="jUdcIVSGAqACBVC2" xmi.id="TUdcIVSGAqACBVD2">
<XMI.extension xmi.extender="Visual Paradigm for UML">
<captionBounds value="373,790,65,15"/>
<captionVisible value="true"/>
<captionSide value="1"/>
<properties>
<abackground value="Cr:0,0,0,255"/>
<aforeground value="Cr:0,0,0,255"/>
<azOrder value="-1"/>
<aselectable value="true"/>
<arequestResetCaption value="false"/>
<arequestResetCaptionSize value="false"/>
<aconnectorStyle value="3"/>
<aconnectorLineJumps value="4"/>
<aconnectorLabelOrientation value="4"/>
<afromShapeXDiff value="0"/>
<afromShapeYDiff value="0"/>
<atoShapeXDiff value="0"/>
<atoShapeYDiff value="0"/>
<afromConnectType value="0"/>
<atoConnectType value="0"/>
<auseFromShapeCenter value="true"/>
<auseToShapeCenter value="true"/>
<afromPinType value="1"/>
<atoPinType value="1"/>
<arequestRebuild value="false"/>
<ashowConnectorName value="2"/>

```

```

<apaintThroughLabel value="2"/>
<afolded value="false"/>
<ashowParameterNamesInMessageOperationSignature value="0"/>
<ashowArgumentsInMessageOperationSignature value="0"/>
<ashowMessageOperationSignature value="0"/>
</properties>
</XMI.extension>
<elementFont bold="false" color="Cr:0,0,0,255" italic="false" name="Dialog" size="11" style="0"/>
<elementLine color="Cr:0,0,0,255" style="1" transparency="0" weight="1.0"/>
</UML:DiagramElement>
<UML:DiagramElement geometry="820,40,60,910" preferredShapeType="InteractionLifeLine" subject="_YbcIVSGAqACBVHy" xmi.id="_YbcIVSGAqACBVHx">
<XMI.extension xmi.extender="Visual Paradigm for UML">
<captionBounds value="820,40,60,15"/>
<captionVisible value="true"/>
<captionSide value="12"/>
<properties>
<abackground value="Cr:122,207,245,255"/>
<aforeground value="Cr:0,0,0,255"/>
<azOrder value="22"/>
<aselectable value="true"/>
<arequestResetCaption value="false"/>
<arequestResetCaptionSize value="false"/>
<adisplayStereotypeIcon value="false"/>
<aoverrideAppearanceWithStereotypeIcon value="true"/>
<arequestSetSizeOption value="0"/>
<aparentConnectorHeaderLength value="40"/>
<aparentConnectorLineLength value="10"/>
<aconnectToPoint value="true"/>
<aparentConnectorDTheta value="0.0"/>
<aconnectionPointType value="2"/>
<amodelElementNameAlignment value="9"/>
<arequestDefaultSize value="false"/>
<arequestFitSize value="false"/>
<arequestFitSizeFromCenter value="false"/>
<aparentFrame value="J9BkIVSGAqACBSIb"/>
<ashowClassifier value="true"/>
</properties>
</XMI.extension>
<elementFill color1="Cr:122,207,245,255" color2="" style="1" transparency="0" type="1"/>
<elementFont bold="false" color="Cr:0,0,0,255" italic="false" name="Dialog" size="11" style="0"/>
<elementLine color="Cr:0,0,0,255" style="1" transparency="0" weight="1.0"/>
</UML:DiagramElement>
<UML:DiagramElement geometry="254,850,846,850;" preferredShapeType="Message" subject="TgvcIVSGAqACBVJc" xmi.id="zGvcIVSGAqACBVJ3">
<XMI.extension xmi.extender="Visual Paradigm for UML">
<captionBounds value="486,830,129,15"/>
<captionVisible value="true"/>
<captionSide value="1"/>
<properties>
<abackground value="Cr:0,0,0,255"/>
<aforeground value="Cr:0,0,0,255"/>
<azOrder value="-1"/>
<aselectable value="true"/>
<arequestResetCaption value="false"/>
<arequestResetCaptionSize value="false"/>
<aconnectorStyle value="3"/>
<aconnectorLineJumps value="4"/>
<aconnectorLabelOrientation value="4"/>
<afromShapeXDiff value="0"/>
<afromShapeYDiff value="0"/>
<atoShapeXDiff value="0"/>
<atoShapeYDiff value="0"/>
<afromConnectType value="0"/>
<atoConnectType value="0"/>
<auseFromShapeCenter value="true"/>
<auseToShapeCenter value="true"/>
<afromPinType value="1"/>
<atoPinType value="1"/>
<arequestRebuild value="false"/>
<ashowConnectorName value="2"/>
<apaintThroughLabel value="2"/>
<afolded value="false"/>
<ashowParameterNamesInMessageOperationSignature value="0"/>
<ashowArgumentsInMessageOperationSignature value="0"/>
<ashowMessageOperationSignature value="0"/>
</properties>
</XMI.extension>
<elementFont bold="false" color="Cr:0,0,0,255" italic="false" name="Dialog" size="11" style="0"/>
<elementLine color="Cr:0,0,0,255" style="1" transparency="0" weight="1.0"/>
</UML:DiagramElement>
<UML:DiagramElement geometry="846,850,8,12" preferredShapeType="Activation" subject="amvcIVSGAqACBVLK" xmi.id="qmvcIVSGAqACBVLK">
<XMI.extension xmi.extender="Visual Paradigm for UML">
<captionBounds value="0,0,8,12"/>
<captionVisible value="false"/>
<captionSide value="7"/>
<properties>
<abackground value="Cr:122,207,245,255"/>
<aforeground value="Cr:0,0,0,255"/>
<azOrder value="14"/>
<aselectable value="true"/>
<arequestResetCaption value="false"/>
<arequestResetCaptionSize value="false"/>
<adisplayStereotypeIcon value="false"/>
<aoverrideAppearanceWithStereotypeIcon value="true"/>
<arequestSetSizeOption value="0"/>
<aparentConnectorHeaderLength value="40"/>
<aparentConnectorLineLength value="10"/>
<aconnectToPoint value="true"/>
<aparentConnectorDTheta value="0.0"/>
<aconnectionPointType value="2"/>
<amodelElementNameAlignment value="9"/>
<arequestDefaultSize value="false"/>
<arequestFitSize value="false"/>
<arequestFitSizeFromCenter value="false"/>
</properties>
</XMI.extension>
<elementFill color1="Cr:122,207,245,255" color2="" style="1" transparency="0" type="1"/>
<elementFont bold="false" color="Cr:0,0,0,255" italic="false" name="Dialog" size="11" style="0"/>
<elementLine color="Cr:0,0,0,255" style="1" transparency="0" weight="1.0"/>

```

```

</UML:DiagramElement>
<UML:DiagramElement geometry="254,870;330,870;330,890;254,890;" preferredShapeType="Message" subject="OpI8IVSGAqACBVNA" xmi.id="upI8IVSGAqACBVNB">
<XMI.extension xmi.extender="Visual Paradigm for UML">
<captionBounds value="330,870,103,15"/>
<captionVisible value="true"/>
<captionSide value="1"/>
<properties>
<abackground value="Cr:0,0,0,255"/>
<aforeground value="Cr:0,0,0,255"/>
<azOrder value="-1"/>
<aselectable value="true"/>
<arequestResetCaption value="false"/>
<arequestResetCaptionSize value="false"/>
<connectorStyle value="3"/>
<aconnectorLineJumps value="4"/>
<aconnectorLabelOrientation value="4"/>
<afromShapeXDiff value="0"/>
<afromShapeYDiff value="0"/>
<atoShapeXDiff value="0"/>
<atoShapeYDiff value="0"/>
<afromConnectType value="0"/>
<atoConnectType value="0"/>
<auseFromShapeCenter value="true"/>
<auseToShapeCenter value="true"/>
<afromPinType value="1"/>
<atoPinType value="1"/>
<arequestRebuild value="false"/>
<ashowConnectorName value="2"/>
<apaintThroughLabel value="2"/>
<afolded value="false"/>
<ashowParameterNamesInMessageOperationSignature value="0"/>
<ashowArgumentsInMessageOperationSignature value="0"/>
<ashowMessageOperationSignature value="0"/>
</properties>
</XMI.extension>
<elementFont bold="false" color="Cr:0,0,0,255" italic="false" name="Dialog" size="11" style="0"/>
<elementLine color="Cr:0,0,0,255" style="1" transparency="0" weight="1.0"/>
</UML:DiagramElement>
<UML:DiagramElement geometry="40,490,850,440" preferredShapeType="CombinedFragment" subject="sDG8IVSGAqACBVue" xmi.id="sDG8IVSGAqACBVud">
<XMI.extension xmi.extender="Visual Paradigm for UML">
<captionBounds value="0,0,33,18"/>
<captionVisible value="true"/>
<captionSide value="12"/>
<properties>
<abackground value="Cr:122,207,245,0"/>
<aforeground value="Cr:0,0,0,255"/>
<azOrder value="20"/>
<aselectable value="true"/>
<arequestResetCaption value="false"/>
<arequestResetCaptionSize value="false"/>
<adisplayStereotypeIcon value="false"/>
<aoverrideAppearanceWithStereotypeIcon value="true"/>
<arequestSetSizeOption value="0"/>
<aparentConnectorHeaderLength value="40"/>
<aparentConnectorLineLength value="10"/>
<aconnectToPoint value="true"/>
<aparentConnectorDTheta value="0.0"/>
<aconnectionPointType value="2"/>
<amodelElementNameAlignment value="9"/>
<arequestDefaultSize value="false"/>
<arequestFitSize value="false"/>
<arequestFitSizeFromCenter value="false"/>
</properties>
</XMI.extension>
<elementFill color1="Cr:122,207,245,255" color2="" style="1" transparency="0" type="1"/>
<elementFont bold="false" color="Cr:0,0,0,255" italic="false" name="Dialog" size="11" style="0"/>
<elementLine color="Cr:0,0,0,255" style="1" transparency="0" weight="1.0"/>
</UML:DiagramElement>
<UML:DiagramElement geometry="50,590,820,330" preferredShapeType="CombinedFragment" subject="u6x8IVSGAqACBV4T" xmi.id="O6x8IVSGAqACBV4S">
<XMI.extension xmi.extender="Visual Paradigm for UML">
<captionBounds value="0,0,23,18"/>
<captionVisible value="true"/>
<captionSide value="12"/>
<properties>
<abackground value="Cr:122,207,245,0"/>
<aforeground value="Cr:0,0,0,255"/>
<azOrder value="16"/>
<aconnectToPoint value="true"/>
<aparentConnectorDTheta value="0.0"/>
<aconnectionPointType value="2"/>
<amodelElementNameAlignment value="9"/>
<arequestDefaultSize value="false"/>
<arequestFitSize value="false"/>
<arequestFitSizeFromCenter value="false"/>
</properties>
</XMI.extension>
<elementFill color1="Cr:122,207,245,0" color2="" style="1" transparency="100" type="1"/>
<elementFont bold="false" color="Cr:0,0,0,255" italic="false" name="Dialog" size="11" style="0"/>
<elementLine color="Cr:0,0,0,255" style="1" transparency="0" weight="1.0"/>
</UML:DiagramElement>
<UML:DiagramElement geometry="40,508,850,422" preferredShapeType="InteractionOperand" subject="ODG8IVSGAqACBVuu" xmi.id="ODG8IVSGAqACBVut">
<XMI.extension xmi.extender="Visual Paradigm for UML">
<captionBounds value="0,0,20,50"/>
<captionVisible value="false"/>
<captionSide value="7"/>
<properties>
<abackground value="Cr:122,207,245,255"/>
<aforeground value="Cr:0,0,0,255"/>
<azOrder value="18"/>
<aselectable value="true"/>
<arequestResetCaption value="false"/>
<arequestResetCaptionSize value="false"/>
<adisplayStereotypeIcon value="false"/>
<aoverrideAppearanceWithStereotypeIcon value="true"/>
<arequestSetSizeOption value="0"/>
<aparentConnectorHeaderLength value="40"/>
<aparentConnectorLineLength value="10"/>
<aconnectToPoint value="true"/>
<aparentConnectorDTheta value="0.0"/>
<aconnectionPointType value="2"/>
<amodelElementNameAlignment value="9"/>
<arequestDefaultSize value="false"/>
<arequestFitSize value="false"/>
<arequestFitSizeFromCenter value="false"/>
</properties>
</XMI.extension>
<elementFill color1="Cr:122,207,245,255" color2="" style="1" transparency="0" type="1"/>
<elementFont bold="false" color="Cr:0,0,0,255" italic="false" name="Dialog" size="11" style="0"/>
<elementLine color="Cr:0,0,0,255" style="1" transparency="0" weight="1.0"/>
</UML:DiagramElement>
<UML:DiagramElement geometry="50,590,820,330" preferredShapeType="CombinedFragment" subject="u6x8IVSGAqACBV4T" xmi.id="O6x8IVSGAqACBV4S">
<XMI.extension xmi.extender="Visual Paradigm for UML">
<captionBounds value="0,0,23,18"/>
<captionVisible value="true"/>
<captionSide value="12"/>
<properties>
<abackground value="Cr:122,207,245,0"/>
<aforeground value="Cr:0,0,0,255"/>
<azOrder value="16"/>

```

```

<aselectable value="true"/>
<arequestResetCaption value="false"/>
<arequestResetCaptionSize value="false"/>
<adisplayStereotypeIcon value="false"/>
<aoverrideAppearanceWithStereotypeIcon value="true"/>
<arequestSetSizeOption value="0"/>
<aparentConnectorHeaderLength value="40"/>
<aparentConnectorLineLength value="10"/>
<aconnectToPoint value="true"/>
<aparentConnectorDTheta value="0.0"/>
<aconnectionPointType value="2"/>
<amodelElementNameAlignment value="9"/>
<arequestDefaultSize value="false"/>
<arequestFitSize value="false"/>
<arequestFitSizeFromCenter value="false"/>
</properties>
<parentDiagramElement idref="ODG8IVSGAqACBVu
t"/>
</XMI.extension>
<elementFill color1="Cr:122,207,245,0" color
2="" style="1" transparency="100" type="1"/>
<elementFont bold="false" color="Cr:0,0,0,25
5" italic="false" name="Dialog" size="11" st
yle="0"/>
<elementLine color="Cr:0,0,0,255" style="1"
transparency="0" weight="1.0"/>
</UML:DiagramElement>
<UML:DiagramElement geometry="60,650,680,70"
preferredShapeType="CombinedFragment" subje
ct="DKj8IVSGAqACBWFP" xmi.id="DKj8IVSGAqACBW
PO">
<XMI.extension xmi.extender="Visual Paradigm
for UML">
<captionBounds value="0,0,33,18"/>
<captionVisible value="true"/>
<captionSide value="12"/>
<properties>
<abackground value="Cr:122,207,245,0"/>
<aforeground value="Cr:0,0,0,255"/>
<azOrder value="2"/>
<aselectable value="true"/>
<arequestResetCaption value="false"/>
<arequestResetCaptionSize value="false"/>
<adisplayStereotypeIcon value="false"/>
<aoverrideAppearanceWithStereotypeIcon value="true"/>
<arequestSetSizeOption value="0"/>
<aparentConnectorHeaderLength value="40"/>
<aparentConnectorLineLength value="10"/>
<aconnectToPoint value="true"/>
<aparentConnectorDTheta value="0.0"/>
<aconnectionPointType value="2"/>
<amodelElementNameAlignment value="9"/>
<arequestDefaultSize value="false"/>
<arequestFitSize value="false"/>
<arequestFitSizeFromCenter value="false"/>
</properties>
<parentDiagramElement idref="ODG8IVSGAqACBVu
t"/>
</XMI.extension>
<elementFill color1="Cr:122,207,245,0" color
2="" style="1" transparency="100" type="1"/>
<elementFont bold="false" color="Cr:0,0,0,25
5" italic="false" name="Dialog" size="11" st
yle="0"/>
<elementLine color="Cr:0,0,0,255" style="1"
transparency="0" weight="1.0"/>
</UML:DiagramElement>
<UML:DiagramElement geometry="50,608,820,50"
preferredShapeType="InteractionOperand" sub
ject="d6x8IVSGAqACBV4j" xmi.id="d6x8IVSGAqAC
BV4i">
<XMI.extension xmi.extender="Visual Paradigm
for UML">
<captionBounds value="80,605,60,50"/>
<captionVisible value="false"/>
<captionSide value="7"/>
<properties>
<abackground value="Cr:122,207,245,255"/>
<aforeground value="Cr:0,0,0,255"/>
<azOrder value="6"/>
<aselectable value="true"/>
<arequestResetCaption value="false"/>
<arequestResetCaptionSize value="false"/>
<adisplayStereotypeIcon value="false"/>
<aoverrideAppearanceWithStereotypeIcon value="true"/>
<arequestSetSizeOption value="0"/>
<aparentConnectorHeaderLength value="40"/>
<aparentConnectorLineLength value="10"/>
<aconnectToPoint value="true"/>
<aparentConnectorDTheta value="0.0"/>
<aconnectionPointType value="2"/>
<amodelElementNameAlignment value="9"/>
<arequestDefaultSize value="false"/>
<arequestFitSize value="false"/>
<arequestFitSizeFromCenter value="false"/>
</properties>
</XMI.extension>
<elementFill color1="Cr:122,207,245,255" col
or2="" style="1" transparency="0" type="1"/>
<elementFont bold="false" color="Cr:0,0,0,25
5" italic="false" name="Dialog" size="11" st
yle="0"/>
<elementLine color="Cr:0,0,0,255" style="1"
transparency="0" weight="1.0"/>
</UML:DiagramElement>
<UML:DiagramElement geometry="50,658,820,262"
preferredShapeType="InteractionOperand" su
bject="f6x8IVSGAqACBV41" xmi.id="f6x8IVSGAqA
CBV40">
<XMI.extension xmi.extender="Visual Paradigm
for UML">
<captionBounds value="0,0,20,50"/>
<captionVisible value="false"/>
<captionSide value="7"/>
<properties>
<abackground value="Cr:122,207,245,255"/>
<aforeground value="Cr:0,0,0,255"/>
<azOrder value="4"/>
<aselectable value="true"/>
<arequestResetCaption value="false"/>
<arequestResetCaptionSize value="false"/>
<adisplayStereotypeIcon value="false"/>
<aoverrideAppearanceWithStereotypeIcon value="true"/>
<arequestSetSizeOption value="0"/>
<aparentConnectorHeaderLength value="40"/>
<aparentConnectorLineLength value="10"/>
<aconnectToPoint value="true"/>
<aparentConnectorDTheta value="0.0"/>
<aconnectionPointType value="2"/>
<amodelElementNameAlignment value="9"/>
<arequestDefaultSize value="false"/>
<arequestFitSize value="false"/>
<arequestFitSizeFromCenter value="false"/>
</properties>

```

```

</XMI.extension>
<elementFill color1="Cr:122,207,245,255" color2="" style="1" transparency="0" type="1"/>
<elementFont bold="false" color="Cr:0,0,0,255" italic="false" name="Dialog" size="11" style="0"/>
<elementLine color="Cr:0,0,0,255" style="1" transparency="0" weight="1.0"/>
</UML:DiagramElement>
<UML:DiagramElement geometry="60,668,680,52" preferredShapeType="InteractionOperand" subject="PKj8IVSGAqACBWPf" xmi.id="PKj8IVSGAqACBWPf">
<XMI.extension xmi.extender="Visual Paradigm for UML">
<captionBounds value="0,0,20,50"/>
<captionVisible value="false"/>
<captionSide value="7"/>
<properties>
<abackground value="Cr:122,207,245,255"/>
<aforeground value="Cr:0,0,0,255"/>
<azOrder value="0"/>
<aselectable value="true"/>
<arequestResetCaption value="false"/>
<arequestResetCaptionSize value="false"/>
<adisplayStereotypeIcon value="false"/>
<aoverrideAppearanceWithStereotypeIcon value="true"/>

```

```

<arequestSetSizeOption value="0"/>
<aparentConnectorHeaderLength value="40"/>
<aparentConnectorLineLength value="10"/>
<aconnectToPoint value="true"/>
<aparentConnectorDTheta value="0.0"/>
<aconnectionPointType value="2"/>
<amodelElementNameAlignment value="9"/>
<arequestDefaultSize value="false"/>
<arequestFitSize value="false"/>
<arequestFitSizeFromCenter value="false"/>
</properties>
</XMI.extension>
<elementFill color1="Cr:122,207,245,255" color2="" style="1" transparency="0" type="1"/>
<elementFont bold="false" color="Cr:0,0,0,255" italic="false" name="Dialog" size="11" style="0"/>
<elementLine color="Cr:0,0,0,255" style="1" transparency="0" weight="1.0"/>
</UML:DiagramElement>
</UML:DiagramElement>
</UML:Diagram>
</XMI.content>
</XMI>

```



## ΠΑΡΑΡΤΗΜΑ Β'

### Κώδικας Φοιτητή NO.2

```

public class TameioCtrl {

    public void neaAkyrosi(int kwdKratisis){

        while( Kratiseis k = kratiseis.next() ){

            int kwdikos = k.getKwdikos();

            if(kwdKratisis==kwdikos){

                Parastasi p = k.getParastasi();

                int kwdikosPar = p.getKwdikos();

                String wraPar = p.getWra();

                System.out.println("Kwdikos parastasis:"+ kwdikosPar +

                    + "\nWra Parastasis :"+ wraPar);

                }//if

            }//while

        }//neaAkyrosi

    public void Epibebaiosi(){

        while(kratiseis k=kratiseis.next()){

            int kwdikos = k.getKwdikos();

            ArrayList <TheseisParastasi> theseisKrat = k.getTheseisKrat();

            while(TheseisParastasi t = theseisKrat.next()){

                t.serKatastasi("");

            }//while

            Parastasi p = k.getParastasi();

            String wra = p.getWra();

```

```
Akyrosi a = new Akyrosi(wra);  
Add Akyrosi(a);  
} //outer while  
} //Epibebaios  
} //class TameioCtrl
```

## ΠΑΡΑΡΤΗΜΑ Γ'

### Κώδικας Πειράματος

```
package teatro;
```

```
public class Main {
```

```
    public static void main(String[] args) {
```

```
        // ΠΧ01: Δημιουργία Θεάτρου - Αίθουσας Teatro teatro =
        new Teatro("ΟΑΥΜνιΟΝ");
```

```
        // ΠΧ02: Ανέβασμα Εργου & Ορισμός παραστάσεων Ergo ergo = new
        Ergo(Ο Μπακαλόγατος - Αστερία"); teatro.setErgo(ergo);
```

```
        // Ανέβασμα 2 παραστάσεων
        ergo.setParastasi(new Parastasi(1, "18:30", teatro.getTheseisQ));
        ergo.setParastasi(new Parastasi(2, "22:30", teatro.getTheseis()));
```

```
        // ΠΧ03: Ανοιγμα Ταμείου TameioCtrl tamCtrl = new
        TameioCtrl(); tamCtrl.setErgo(ergo);
        ergo.emfanParast();
```

```
        // ΠΧ04: Επαναληπτική διαδικασία κρατήσεων. Εξοδος με 99
        int kwdParast = UserInput.getInteger("\nΔώσε αριθμό Παράστασης(1, 2, 99)=> "] while
        (kwdParast!=99) {
            if (kwdParast==1 || kwdParast==2) {
                tamCtrl.neaKratisi(ergo);
                // Επιλογή παράστασης και εμφάνιση διαθέσιμων θέσεων
                tamCtrl.epilogiParast(kwdParast);
                tamCtrl.epilogiThesewn();
                tamCtrl.telosKratisi();
            }
            else
                System.out.println("Οι κωδικοί παραστάσεων είναι 1 & 2. Εξοδος με 99"); if
                (kwdParast!=99) { ergo.emfanParast();
                    kwdParast = UserInput.getInteger("Αυ^ αριθμό Παράστασης(1, 2, 99)=> ");
```

```

    }
}

// Εμφάνιση κρατήσεων
tamCtrl.emfanKratiseisO;
}
package teatro; import
java.util.ArrayList; import
Java.util.Iterator;

public class TameioCtrl {

    private static int arithmosKratisis = 1; private
    Ergo ergo; private Kratisi trexKratisi;
    private ArrayList <Kratisi> kratiseis = new ArrayList(12); private
    Parastasi trexParast;

    public TameioCtrl() { }

    public void setErgo(Ergo ergo) {
        this.ergo=ergo;
    }

    public void neaKratisi(Ergo ergo) {
        trexKratisi= new Kratisi(arithmosKratisis++);
    }

    public void epilogiParast(int kwdParast) {
        trexParast = ergo.getParastasi(kwdParast);
        System.out.ρπηϊλη("\ηΕττ.ιλέγετοα η παράσταση " + kwdParast);
        trexParast.emfanTheseis();
        trexKratisi.setParastasi(trexParast);
    }

    public void epilogiThesewn() {
        int arThes = UserInput.getInteger("Α(Λαε αριθμό θέσης(0-9)=> "); while
        (arThes!=99) { if (arThes>=-1 && arThes<10) { if

```

```

(trexParast.getKatastasiThesis(arThes) == ") {
trexParast.setKatastasiThesis(arThes, 'K');
trexKratysi.addThesiParast(trexParast.getThesiParast(arThes));
    else
        System.out.println("Η θέση " + arThes + " είναι κατειλημμένη");
    }
    arThes = UserInput.getInteger("AU^ αριθμό θέσης(0-9)=> ");
}

public void telosKratysiQ {
    kratiseis.add(trexKratysi);
    System.out.println(" --- Τέλος κράτησης..... ");
}

public void emfanKratiseis() {
    System.out.println("\n===== ΕΜΦΑΝΙΣΗ ΚΡΑΤΗΣΕΩΝ =====");
    Iterator itr = kratiseis.iterator(); while
    (itr.hasNext()) {
        Kratysi kratysi = (Kratysi) itr.next();
        kratysi.emfanKratiseisO;
    }
}

package teatro; import
java.util.ArrayList;

public class Teatro {

    private String epwonymia; private
    Aithousa aith; private Ergo ergo;

    public Teatro(String epwn) {
        this.epwonymia = epwn; this.aith = new
        Aithousa(IO);
    }
    public void setErgo(Ergo ergo) {
        this.ergo = ergo;
        System.out.println("\nGEATPO " + epwonymia);
        System.out.println("\nEPrO:" + ergo.getTitlos());
    }
}

```

```
public ArrayList getTheseis() {
    return aith.getTheseisQ;
}

public void emfanAithousa() {
    aith.emfanTheseis();
}
}

package teatro; import
Java.util.*;

public class Ergo {

    private String titlos;
    private ArrayList <Parastasi> parastaseis = new ArrayList(3);

    public Ergo(String titlos) {
        this.titlos=titlos;
    }

    public void setParastasi(Parastasi parast) {
        parastaseis.add(parast);
    }

    public void emfanParast() {
        System.out.println("\nΟι παραστάσεις:"); Iterator itr
        = parastaseis.iterator(); while (itr.hasNext()) {
            Parastasi parast = (Parastasi) itr.next();
            System.out.println(parast);
        }
    }

    public Parastasi getParastasi(int kwdikos) {
```

```

    Iterator itr = parastaseis.iterator(); while
    (itr.hasNext()) {
        Parastasi parast = (Parastasi) itr.next();
        if (parast.getKwdikos()==kwdikos) return
            parast;
    }
    return null;
}

public String getTitlos() {
    return titlos;
}
}

package teatro; import
Java.util.*;

public class Parastasi {

    private int kwdikos; private
    String wra;
    private ArrayList<ThesiParastasis> theseisParast;

    public Parastasi(int kwdikos, String wra, ArrayList theseis) {
        this.kwdikos = kwdikos; this.wra =
        wra;
        theseisParast = new ArrayList(theseis.size()); for (int
        i=0; i<theseis.size(); i++) {

            Thesi thesi = (Thesi)theseis.get(i);
            ThesiParastasis thesiParast = new ThesiParastasis(thesi)
            theseisParast.add(thesiParast);
        }
    }

    public void emfanTheseis() {
        System.out.println("\nΔιαθέσιμες θέσεις παράστασης"); for (int i=0; i<theseisParast.size(); i++) {
        System.out.println(theseisParast.get(i));
    }
}

```

```

    }
    }

    public int getKwdikos() {
        return kwdikos;
    }

    public ThesiParastasis getThesiParast(int t) {
        return theseisParast.get(t);
    }

    public char getKatastasiThesis(int t) {
        ThesiParastasis thesiParast = theseisParast.get(t); return thesiParast.getKatastasi();
    }

    public void setKatastasiThesis(int t, char k) {
        ThesiParastasis thesiParast = theseisParast.get(t); thesiParast. setKatastasi(k);
    }

    public String toString() {
        return ("Παράσταση:" + kwdikos + " " + "Ωρα:" + wra + " ");
    }
}

package teatro; import
Java.util.*;

public class Kratisi {

    private int kwdikos = 0; private
    Parastasi parast;
    private ArrayList <ThesiParastasis> theseisKrat = new ArrayList(10);

    public Kratisi (int arithmoskratisis) {
        this.kwdikos ^arithmoskratisis;
    }

    public void setParastasi(Parastasi parast) {

```



```

    this.parast = parast;
}

public void addThesiParast(ThesiParastasis thesiParast) {
    this.theseisKrat.add(thesiParast);
}

public void emfanKratasisQ {
    System.out.println("\ηΑριθμός κράτησης:" + kwdikos + " " + parast);
    System.out.println(" Θέσεις κράτησης:"); Iterator itr = theseisKrat.iterator();
    while (itr.hasNext()) {
        ThesiParastasis thesParast = (ThesiParastasis) itr.next();
        System.out.println(thesParast);
    }
}

package teatro; import
Java.util.*;
public class ThesiParastasis {

    private Thesi thesi;
    private char katastasi;

    public ThesiParastasis(Thesi thesi) {
        this.thesi = thesi;
        this.katastasi = ' ';
    }

    public void setDiathesimi(int i) {
        katastasi = ' ';
    }

    public void setKatillimeni(int i) {
        katastasi = 'K';
    }

    public char getKatastasi() {
        return katastasi;
    }
}

```

```
public void setKatastasi(char k) {  
    katastasi = k;  
}  
public int getArithmos() {  
    return thesi.getArithmos();  
}  
public String toString() {  
    return (" Arithmos θέσης:" + getArithmosQ + " [" + katastasi + "]");  
}  
}
```