ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΑΤΤΙΚΗΣ

ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ΤΥΠΙΚΗ ΠΡΟΔΙΑΓΡΑΦΗ WEB ΥΠΗΡΕΣΙΩΝ ΓΙΑ ΠΡΩΤΟΚΟΛΛΑ ONLINE ΕΠΙΚΟΙΝΩΝΙΑΣ: Η ΠΕΡΙΠΤΩΣΗ ΤΟΥ "ROCKET.CHAT"

ΔΗΜΗΤΡΙΟΣ ΝΙΑΒΗΣ
Α.Μ. 141070

Εισηγητής: ΚΩΝΣΤΑΝΤΙΝΟΣ ΜΠΑΡΛΑΣ, Ε.ΔΙ.Π.

(Κενό φύλλο)

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ΤΥΠΙΚΗ ΠΡΟΔΙΑΓΡΑΦΗ WEB ΥΠΗΡΕΣΙΩΝ ΓΙΑ ΠΡΩΤΟΚΟΛΛΑ ONLINE ΕΠΙΚΟΙΝΩΝΙΑΣ: Η ΠΕΡΙΠΤΩΣΗ ΤΟΥ “ROCKET.CHAT”

ΔΗΜΗΤΡΙΟΣ ΝΙΑΒΗΣ
Α.Μ. 141070

Εισηγητής:

ΚΩΝΣΤΑΝΤΙΝΟΣ ΜΠΑΡΛΑΣ, Ε.ΔΙ.Π.

Konstantinos Barlas
Digitally signed by Konstantinos Barlas
Date: 2024.04.10 19:22:42 +03'00'

Εξεταστική Επιτροπή:

ΣΤΑΥΡΟΣ ΦΑΤΟΥΡΟΣ, ΑΝΑΠΛΗΡΩΤΗΣ ΚΑΘΗΓΗΤΗΣ

Stavros Fatouros
Digitally signed by Stavros Fatouros
Date: 2024.04.10 12:15:54 +03'00'

ΧΡΗΣΤΟΣ ΤΡΟΥΣΣΑΣ, ΕΠΙΚΟΥΡΟΣ ΚΑΘΗΓΗΤΗΣ

Christos Troussas
Digitally signed by Christos Troussas
Date: 2024.04.10 17:30:10 +03'00'

Ημερομηνία εξέτασης /03/2024

ΔΗΛΩΣΗ ΣΥΓΓΡΑΦΕΑ ΜΕΤΑΠΤΥΧΙΑΚΗΣ ΕΡΓΑΣΙΑΣ

Ο κάτωθι υπογεγραμμένος Νιαβής Δημήτριος του Γεωργίου, με αριθμό μητρώου 141070 φοιτητής του Τμήματος Μηχανικών Πληροφορικής και Υπολογιστών της Σχολής Μηχανικών του Πανεπιστημίου Δυτικής Αττικής, δηλώνω ότι:

«Βεβαιώνω ότι είμαι συγγραφέας αυτής της Διπλωματικής εργασίας και κάθε βοήθεια την οποία είχα για την προετοιμασία της, είναι πλήρως αναγνωρισμένη και αναφέρεται στην εργασία. Επίσης, οι όποιες πηγές από τις οποίες έκανα χρήση δεδομένων, ιδεών ή λέξεων, είτε ακριβώς είτε παραφρασμένες, αναφέρονται στο σύνολό τους, με πλήρη αναφορά στους συγγραφείς, τον εκδοτικό οίκο ή το περιοδικό, συμπεριλαμβανομένων και των πηγών που ενδεχομένως χρησιμοποιήθηκαν από το διαδίκτυο. Επίσης, βεβαιώνω ότι αυτή η εργασία έχει συγγραφεί από μένα αποκλειστικά και αποτελεί προϊόν πνευματικής ιδιοκτησίας τόσο δικής μου, όσο και του Ιδρύματος.
Παράβαση της ανωτέρω ακαδημαϊκής μου ευθύνης αποτελεί ουσιώδη λόγο για την ανάκληση του πτυχίου μου».

Ο Δηλών

ΔΗΜΗΤΡΗΣ ΝΙΑΒΗΣ
141070

ΕΥΧΑΡΙΣΤΙΕΣ

Η παρούσα διπλωματική εργασία ολοκληρώθηκε μετά από επίμονες προσπάθειες, σε ένα ενδιαφέρον γνωστικό αντικείμενο, όπως αυτό της επεξεργασίας κειμένου. Την προσπάθειά μου αυτή υποστήριξε ο επιβλέπων καθηγητής μου, τον οποίο θα ήθελα να ευχαριστήσω.
Ακόμα θα ήθελα να ευχαριστήσω την οικογένειά μου για τη συμπαράσταση κατά τη διάρκεια των σπουδών μου.

ΠΕΡΙΛΗΨΗ

Η παρούσα διπλωματική εργασία ασχολείται με την μελέτη για την εφαρμογή Τυπικών Μεθόδων στη μοντελοποίηση των υπηρεσιών web της πλατφόρμας επικοινωνίας ‘rocket.chat’. Το μοντέλο που παράγεται περιγράφει το σχεδιασμό των υπηρεσιών σε μορφή αλγεβρικών οντοτήτων, επιτρέποντας έτσι την βαθύτερη κατανόηση τους και θωρακίζοντας τες από σχεδιαστικά σφάλματα.

ABSTRACT

The present thesis deals with the study of the application of Standard Methods in the modeling of the web services of the communication platform 'rocket.chat'. The model produced describes the design of the services in the form of algebraic entities, thus allowing their deeper understanding and shielding them from design errors.

ΕΠΙΣΤΗΜΟΝΙΚΗ ΠΕΡΙΟΧΗ:

- Επιστήμη των Υπολογιστών
- Βελτίωση, Απόδειξη Θεωρημάτων και Έλεγχος Μοντέλων

ΛΕΞΕΙΣ ΚΛΕΙΔΙΑ: Z, Τυπικές μέθοδοι, rocket.chat

ΠΕΡΙΕΧΟΜΕΝΑ

ΚΑΤΑΛΟΓΟΣ ΕΙΚΟΝΩΝ

ΚΑΤΑΛΟΓΟΣ ΠΙΝΑΚΩΝ

ΚΕΦΑΛΑΙΟ 1

ΕΙΣΑΓΩΓΗ

Η μοντελοποίηση των υπηρεσιών ενός συστήματος επικοινωνίας όπως η πλατφόρμα Rocket.Chat είναι ένα σημαντικό βήμα για την κατανόηση, τον σχεδιασμό και την ανάπτυξη του. Η Rocket.Chat αποτελεί μία πλατφόρμα ανοικτού κώδικα που παρέχει υπηρεσίες συνομιλίας και επικοινωνίας για οργανισμούς και ομάδες.

Σε αυτό το πλαίσιο, η μοντελοποίηση των υπηρεσιών της Rocket.Chat μπορεί να γίνει με τη χρήση Τυπικών Μεθόδων μοντελοποίησης. Οι Τυπικές Μεθόδοι αναφέρονται σε προσεγγίσεις που χρησιμοποιούνται ευρέως στην επιστήμη της πληροφορικής και της μηχανικής λογισμικού για να περιγράψουν, να αναλύσουν και να σχεδιάσουν συστήματα.

Αρχικά, μια συνοπτική εισαγωγή στη Rocket.Chat ως πλατφόρμα επικοινωνίας θα παρουσιαστεί. Στη συνέχεια, θα εξεταστούν τα βασικά συστατικά της, όπως οι υπηρεσίες συνομιλίας, η διαχείριση χρηστών και οι δυνατότητες προσαρμογής.

Η μοντελοποίηση μπορεί να γίνει με χρήση διαγραμμάτων UML (Unified Modeling Language) για την απεικόνιση της δομής και των σχέσεων μεταξύ των κομματιών του συστήματος. Επιπλέον, η χρήση γλωσσών προδιαγραφής όπως η Z μπορεί να επιτρέψει την ακριβή περιγραφή των προδιαγραφών και των λειτουργιών του συστήματος.

Η μοντελοποίηση μπορεί να επιτρέψει στην ομάδα ανάπτυξης να έχει μια σαφή εικόνα του συστήματος πριν από την υλοποίηση, βοηθώντας στην πρόληψη σφαλμάτων και στη βελτιστοποίηση της απόδοσης. Επιπλέον, η μοντελοποίηση μπορεί να χρησιμεύσει ως εργαλείο για την επικοινωνία μεταξύ των μελών της ομάδας ανάπτυξης και με άλλα ενδιαφερόμενα μέρη, ενισχύοντας τη συνεννόηση και τη συνεργασία.

Συνολικά, η μοντελοποίηση των υπηρεσιών της Rocket.Chat με τη χρήση τυπικών μεθόδων μπορεί να συμβάλλει στην ομαλή ανάπτυξη και λειτουργία του συστήματος, ενισχύοντας την κατανόηση και την ασφάλεια του.

1.1 ΕΙΣΑΓΩΓΗ

Οι Τυπικές Μέθοδοι που χρησιμοποιούνται στην ανάπτυξη συστημάτων υπολογιστών είναι τεχνικές βασισμένες σε μαθηματικά για την περιγραφή των ιδιοτήτων του συστήματος. Οι Τυπικές Μέθοδοι παρέχουν πλαίσια εντός των οποίων οι άνθρωποι μπορούν να προσδιορίσουν, να αναπτύξουν και να επαληθεύσουν συστήματα με συστηματικό και όχι ad hoc τρόπο. (Wing, 1990)

Μία μέθοδος είναι τυπική εάν έχει μια σωστή μαθηματική βάση, που συνήθως δίνεται από μια τυπική γλώσσα προδιαγραφών. Αυτή η βάση παρέχει τα μέσα για τον ακριβή καθορισμό των εννοιών όπως η συνοχή και η πληρότητα και πιο σχετικά, η προδιαγραφή, η υλοποίηση και η ορθότητα. Παρέχει τα μέσα απόδειξης ότι μια προδιαγραφή είναι πραγματοποιήσιμη, αποδεικνύοντας ότι ένα σύστημα έχει εφαρμοστεί σωστά και αποδεικνύοντας τις ιδιότητες ενός συστήματος χωρίς απαραίτητα να το τρέχει για να προσδιορίσει τη συμπεριφορά του. (Wing, 1990)

Μία τυπική μέθοδος ασχολείται επίσης και με έναν αριθμό πραγματιστικών ζητημάτων: ποιος τη χρησιμοποιεί, σε τι χρησιμοποιείται, πότε χρησιμοποιείται και πώς χρησιμοποιείται. πιο συχνά οι σχεδιαστές συστημάτων χρησιμοποιούν Τυπικές Μεθόδους για να καθορίσουν τις επιθυμητές συμπεριφορές και δομικές ιδιότητες ενός συστήματος. Ωστόσο, οποιοσδήποτε εμπλέκεται σε οποιοδήποτε στάδιο ανάπτυξης συστήματος μπορεί να χρησιμοποιήσει Τυπικές Μεθόδους. Μπορούν να χρησιμοποιηθούν σε μια αρχική δήλωση των απαιτήσεων ενός πελάτη, μέσω σχεδιασμού συστήματος, υλοποίησης, δοκιμών, εντοπισμού σφαλμάτων, συντήρησης, επαλήθευσης και αξιολόγησης. Οι Τυπικής Μέθοδοι χρησιμοποιούνται για την αποκάλυψη ασαφειών, ατελειών και ασυνεπειών σε ένα σύστημα. Εάν χρησιμοποιηθούν νωρίς στη διαδικασία ανάπτυξης του συστήματος, μπορούν να αποκαλύψουν ελαττώματα σχεδιασμού που διαφορετικά θα μπορούσαν να ανακαλυφθούν μόνο στις δαπανηρές φάσεις δοκιμών και εντοπισμού σφαλμάτων. Εάν χρησιμοποιηθούν αργότερα, μπορούν να βοηθήσουν στον προσδιορισμό της ορθότητας μιας υλοποίησης συστήματος και της ισοδυναμίας των διαφορετικών υλοποιήσεων (Wing, 1990)

Για να είναι μια μέθοδος τυπική, πρέπει να έχει μια καλά καθορισμένη μαθηματική βάση. Δεν χρειάζεται να αντιμετωπίσει ρεαλιστικές σκέψεις, αλλά τότε θα ήταν μια αρκετά άχρηστη μέθοδος. Ως εκ τούτου, μια Τυπική Μέθοδος θα πρέπει να συνοδεύεται από ένα σύνολο κατευθυντήριων γραμμών, που λέει στους χρήστες υπό ποιες συνθήκες μπορεί και πρέπει να εφαρμοστεί η μέθοδος και πόσο αποτελεσματικά να την εφαρμόσουν. (Wing, 1990)

Ένα απτό προϊόν της εφαρμογής μιας τυπικής μεθόδου είναι μια (τυπική) προδιαγραφή. Μια προδιαγραφή χρησιμεύει ως «συμβόλαιο» μεταξύ του πελάτη και του υλοποιητή. Μια προδιαγραφή χρησιμεύει ως πολύτιμο κομμάτι τεκμηρίωσης και ως μέσο επικοινωνίας μεταξύ των πελατών, των

προσδιοριστών και των υλοποιητών. Λόγω της μαθηματικής βάσης μιας τυπικής μεθόδου, οι τυπικές προδιαγραφές είναι πιο ακριβείς και συνήθως πιο συνοπτικές από τις μη-τυπικές. (Wing, 1990)

Δεδομένου ότι μια Τυπική Μέθοδος είναι μια μέθοδος, όχι απλώς ένα πρόγραμμα ή μια γλώσσα υπολογιστή, μπορεί να έχει ή να μην έχει υποστήριξη εργαλείων. Εάν η σύνταξη της γλώσσας προδιαγραφών μιας Τυπικής Μεθόδου είναι σαφής, τότε θα ήταν εύκολο να παρέχουμε τυπικά εργαλεία ανάλυσης σύνταξης για τυπικές προδιαγραφές. Εάν η σημασιολογία της γλώσσας είναι επαρκώς περιορισμένη, μπορούν να πραγματοποιηθούν διάφοροι βαθμοί σημασιολογικής ανάλυσης και με μηχανικά βοηθήματα. Έτσι, οι τυπικές προδιαγραφές έχουν το πρόσθετο πλεονέκτημα σε σχέση με τις άτυπες ότι είναι επιδεκτικές σε μηχανική ανάλυση και χειρισμό. (Wing, 1990)

1.2 ΙΔΙΟΤΗΤΕΣ ΤΩΝ ΠΡΟΔΙΑΓΡΑΦΩΝ:

Κάθε γλώσσα προδιαγραφών θα πρέπει να ορίζεται έτσι ώστε κάθε καλά διαμορφωμένη προδιαγραφή γραμμένη στη γλώσσα να είναι ξεκάθαρη.

Ανεπίσημα, μια προδιαγραφή είναι σαφής εάν και μόνο εάν έχει ακριβώς ένα νόημα. Αυτή η βασική ιδιότητα των τυπικών προδιαγραφών σημαίνει ότι οποιαδήποτε γλώσσα προδιαγραφών που βασίζεται ή ενσωματώνει μια φυσική γλώσσα (όπως τα αγγλικά) δεν είναι τυπική, καθώς οι φυσικές γλώσσες είναι εγγενώς διφορούμενες. Σημαίνει επίσης ότι μια γλώσσα οπτικών προδιαγραφών που επιτρέπει πολλαπλές ερμηνείες ενός πλαισίου και/ή ενός βέλους είναι λανθασμένα καθορισμένη και, ως εκ τούτου, δεν είναι τυπική. (Wing, 1990)

Μια άλλη επιθυμητή ιδιότητα των προδιαγραφών είναι η συνέπεια.

Ανεπίσημα, μια προδιαγραφή είναι συνεπής εάν και μόνο εάν η συγκεκριμένη και το σύνολο της δεν είναι κενό. Όσον αφορά τα προγράμματα, η συνέπεια είναι σημαντική γιατί σημαίνει ότι υπάρχει κάποια υλοποίηση που θα ικανοποιεί τις προδιαγραφές. Βλέποντας μια προδιαγραφή ως ένα σύνολο γεγονότων, η συνέπεια συνεπάγεται ότι δεν μπορούμε να αντλήσουμε τίποτα αντιφατικό από την προδιαγραφή. Εάν θέτουμε μια ερώτηση με βάση μια σταθερή προδιαγραφή, δεν θα λάβουμε αμοιβαία αποκλειστικές απαντήσεις. Είναι προφανές ότι θέλουμε να έχουμε συνεπείς προδιαγραφές, αφού μια ασυνεπής προδιαγραφή σημαίνει ότι δεν έχουμε καθόλου γνώση, καθώς η προδιαγραφή αναιρεί σε μια περίπτωση αυτό που ισχυρίζεται σε μια άλλη. (Wing, 1990)

Οι προδιαγραφές δεν χρειάζεται να είναι "πλήρεις" με την έννοια που χρησιμοποιείται στη μαθηματική λογική, αν και ορισμένες ιδιότητες "σχετικής πληρότητας" μπορεί να είναι επιθυμητές (π.χ. επαρκής πληρότητα μιας αλγεβρικής προδιαγραφής). Στην πράξη, συνήθως πρέπει να αντιμετωπίζουμε ελλιπείς προδιαγραφές. Γιατί; Οι προσδιοριστές ενδέχεται να αφήσουν σκόπιμα

ορισμένα πράγματα απροσδιόριστα, δίνοντας στον υλοποιητή κάποια ελευθερία να επιλέξει μεταξύ διαφορετικών δομών δεδομένων και αλγορίθμων. Επίσης, οι προσδιοριστές δεν μπορούν να προβλέψουν ρεαλιστικά όλα τα πιθανά σενάρια στα οποία θα εκτελεστεί ένα σύστημα, και έτσι, ίσως άθελά τους, έχουν αφήσει κάποια πράγματα απροσδιόριστα. Τέλος, οι προσδιοριστές αναπτύσσουν προδιαγραφές σταδιακά και επαναληπτικά, ίσως ως απάντηση στις μεταβαλλόμενες απαιτήσεις των πελατών, και ως εκ τούτου εργάζονται πιο συχνά με ημιτελή προϊόντα, παρά με τελικά. (Wing, 1990)

Υπάρχει μια λεπτή ισορροπία μεταξύ του να λες αρκετά και να λες πάρα πολλά σε μια προδιαγραφή. Οι προσδιοριστές θέλουν να πουν αρκετά, ώστε οι υλοποιητές να μην επιλέγουν μη αποδεκτές υλοποιήσεις. Οι προσδιοριστές είναι υπεύθυνοι για τη μη παραβίαση. Οποιαδήποτε ανεπάρκεια των προδιαγραφών θα πρέπει να είναι σκόπιμη ατελή. Από την άλλη πλευρά, το να πούμε πάρα πολλά μπορεί να αφήσει ελάχιστη ελευθερία σχεδιασμού για τον υλοποιητή. Μια προδιαγραφή που υπερκαθορίζει είναι ένοχη για μεροληψία υλοποίησης. Ανεπίσημα, μια προδιαγραφή έχει μεροληψία υλοποίησης εάν καθορίζει εξωτερικά μη παρατηρήσιμες ιδιότητες των ειδικών της. Ως εκ τούτου, θέτει περιττούς περιορισμούς στις ιδιαιτερότητές του. Για παράδειγμα, μια προδιαγραφή συνόλου που παρακολουθεί τη σειρά εισαγωγής των στοιχείων της έχει μεροληψία υλοποίησης, π.χ., προς μια αναπαράσταση λίστας διατεταγμένων και έναντι αναπαράστασης πίνακα κατακερματισμού. (Wing, 1990)

1.3 ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ:

Τα χαρακτηριστικά των τυπικών μεθόδων, όπως το αν η γλώσσα είναι γραφική ή όχι ή αν η υποκείμενη λογική της είναι πρώτης τάξης ή όχι, επηρεάζουν τον τρόπο με τον οποίο οι χρήστες την εφαρμόζουν. Δεν είναι το αντικείμενο αυτής της εργασίας να δώσει μια πλήρη ταξινόμηση όλων των πιθανών χαρακτηριστικών μιας μεθόδου ούτε να ταξινομήσει εξαντλητικά όλες τις μεθόδους σύμφωνα με αυτά τα χαρακτηριστικά. Αντίθετα, δίνουμε μια μερική λίστα διαφορετικών χαρακτηριστικών, σημειώνοντας ότι μια μέθοδος αντικατοπτρίζει συνήθως έναν συνδυασμό πολλών διαφορετικών. (Wing, 1990).

Η Z είναι Τυπική Μέθοδος που βασίζεται στη θεωρία συνόλων, αν και η Z μπορεί να χρησιμοποιηθεί τόσο σε στυλ προσανατολισμένα στο μοντέλο όσο και σε στυλ προσανατολισμού ιδιοτήτων. Η κατάσταση του πίνακα μοντελοποιείται με μια μερική αντιστοίχιση από κλειδιά σε τιμές (το X+Y υποδηλώνει ένα σύνολο μερικών αντιστοιχίσεων από το σύνολο X στο σύνολο Y μια μερική αντιστοίχιση σχετίζεται κάθε μέλος του X με το πολύ ένα μέλος του Y.) Σύμφωνα με τη σύμβαση, οι μη αρχικοποιημένες μεταβλητές στη Z αντιπροσωπεύουν την κατάσταση πριν από την εκτέλεση μιας πράξης και οι μεταβλητές εκκίνησης για την κατάσταση μετά. θα χρησιμοποιήσουμε την ίδια σύμβαση

στις προδιαγραφές VDM και Larch. Υπάρχουν τέσσερις λειτουργίες στον πίνακα, INIT, INSERT, LOOKUP και DELETE. Το INIT προετοιμάζει τον πίνακα συμβόλων ώστε να είναι κενός. Το INSERT τροποποιεί τον πίνακα προσθέτοντας μια νέα δέσμευση στο st, στην περίπτωση που το κλειδί k δεν βρίσκεται ήδη στον τομέα του st. Το LOOKUP απαιτεί το κλειδί k να βρίσκεται στον τομέα της αντιστοίχισης, επιστρέφει την τιμή στην οποία αντιστοιχίζεται το k και δεν αλλάζει την κατάσταση του πίνακα συμβόλων (st = st). Το DELETE απαιτεί επίσης το κλειδί k να βρίσκεται στον τομέα της αντιστοίχισης και τροποποιεί τον πίνακα διαγράφοντας τη σύνδεση που σχετίζεται με το k από το st (είναι ένας τελεστής αφαίρεσης τομέα). Ο ελεγκτής αποδείξεων B έχει χρησιμοποιηθεί για την απόδειξη θεωρημάτων με βάση τις προδιαγραφές Z. (Wing, 1990)

Το VDM υποστηρίζει ένα στυλ προδιαγραφών προσανατολισμένο στο μοντέλο. Το VDM ορίζει ένα σύνολο ενσωματωμένων τύπων δεδομένων, π.χ. σύνολα, λίστες και αντιστοιχίσεις, τα οποία χρησιμοποιούν οι προσδιοριστές για να ορίσουν άλλους τύπους. Οι συμπεριφορές των λειτουργιών INIT, INSERT, LOOKUP και DELETE είναι οι ίδιες με αυτές που καθορίζονται στην προδιαγραφή Z. Ωστόσο, οι προϋποθέσεις, που καθορίζονται στις προκαταρκτικές ρήτρες, είναι σαφείς και χωριστές από τις εκ των υστέρων προϋποθέσεις, που καθορίζονται στις μεταρρύθμιση. Προϋπόθεση για μια λειτουργία είναι ένα κατηγόρημα που πρέπει να διατηρείται στην κατάσταση σε κάθε επίκληση της πράξης. εάν δεν ισχύει, τότε η συμπεριφορά της λειτουργίας είναι απροσδιόριστη. Μια μετασυνθήκη είναι ένα κατηγόρημα που διατηρείται στην κατάσταση κατά την επιστροφή. Οι πελάτες μιας λειτουργίας είναι υπεύθυνοι για την ικανοποίηση των προϋποθέσεων και ο εκτελεστής της είναι υπεύθυνος για την εγγύηση της μεταγενέστερης συνθήκης. Το γεγονός ότι το LOOKUP δεν τροποποιεί τον πίνακα συμβόλων (άρα st = st), αλλά το INSERT και το DELETE, καθορίζεται χρησιμοποιώντας το rd (για πρόσβαση "μόνο για ανάγνωση") αντί του wr (για πρόσβαση "εγγραφής και ανάγνωσης") στο δήλωση των μεταβλητών εξωτερικής κατάστασης στις οποίες έχει πρόσβαση κάθε λειτουργία. (Wing, 1990)

Το Larch είναι μια μέθοδος προσανατολισμένη στην ιδιότητα που συνδυάζει αξιωματικές και αλγεβρικές προδιαγραφές σε μια προδιαγραφή "δύο επιπέδων". Το αξιωματικό στοιχείο καθορίζει συμπεριφορά που εξαρτάται από την κατάσταση, π.χ. παρενέργειες και έκτακτος τερματισμός προγραμμάτων. Η αλγεβρική συνιστώσα προσδιορίζει τις ανεξάρτητες από την κατάσταση ιδιότητες των δεδομένων στα οποία έχουν πρόσβαση τα προγράμματα. (Wing, 1990)

Το πρώτο κομμάτι της προδιαγραφής Larch, που ονομάζεται προδιαγραφή διεπαφής, μοιάζει με τις προδιαγραφές Z και VDM. Για κάθε πράξη, οι ρήτρες απαιτήσεων και διασφαλίσεων προσδιορίζουν τις προ και τις μεταγενέστερες προϋποθέσεις της. Ο όρος τροποποιήσεις παραθέτει εκείνα τα αντικείμενα των οποίων η τιμή ενδέχεται να αλλάξει ως αποτέλεσμα της εκτέλεσης της λειτουργίας. Ως εκ τούτου, η αναζήτηση δεν επιτρέπεται να αλλάξει την κατάσταση του ορίσματος του πίνακα συμβόλων, αλλά επιτρέπεται η εισαγωγή και η διαγραφή. Μια διαφορά μεταξύ Larch και VDM (και

Larch και Z), είναι ότι εάν η γλώσσα προγραμματισμού της επιλογής μας υποστηρίζει χειρισμό εξαιρέσεων, οι διεπαφές θα καθορίζουν εάν και υπό ποιες συνθήκες μια λειτουργία σηματοδοτεί εξαιρέσεις. Για παράδειγμα, θα μπορούσαμε να αφαιρέσουμε τον όρο “απαιτεί” και, αντ' αυτού, να χρησιμοποιήσουμε έναν ειδικό όρο σημάτων στη μεταγενέστερη συνθήκη του για να καθορίσουμε ότι ένα σήμα θα πρέπει να ανυψωθεί στην περίπτωση που το κλειδί k βρίσκεται ήδη στον πίνακα συμβόλων. (Wing, 1990)

Το δεύτερο κομμάτι της προδιαγραφής Larch, που ονομάζεται χαρακτηριστικό, μοιάζει με αλγεβρική προδιαγραφή. Περιέχει ένα σύνολο δηλώσεων συμβόλων συνάρτησης και ένα σύνολο εξισώσεων που καθορίζουν τη σημασία των συμβόλων συνάρτησης. Οι εξισώσεις καθορίζουν μια σχέση ισοδυναμίας σε ταξινομημένους όρους. Τα αντικείμενα του τύπου δεδομένων πίνακα συμβόλων που καθορίζονται στην προδιαγραφή διεπαφής κυμαίνονται σε τιμές που υποδηλώνονται με τους όρους της ταξινόμησης S. Η ρήτρα που δημιουργείται δηλώνει ότι όλες οι τιμές του πίνακα συμβόλων μπορούν να αναπαρασταθούν με όρους που αποτελούνται αποκλειστικά από τα δύο σύμβολα συναρτήσεων, emp και add. Αυτή η ρήτρα ορίζει έναν επαγωγικό κανόνα συμπερασμάτων και είναι χρήσιμος για την απόδειξη ιδιοτήτων για όλες τις τιμές του πίνακα συμβόλων. Η ρήτρα κατατμημένη προσθέτει περισσότερες ισοδυναμίες μεταξύ των όρων. Διαισθητικά δηλώνει ότι δύο όροι είναι ίσοι εάν δεν μπορούν να διακριθούν από καμία από τις συναρτήσεις που αναφέρονται στην πρόταση. Στο παράδειγμα, θα μπορούσαμε να χρησιμοποιήσουμε αυτήν την ιδιότητα για να δείξουμε ότι η σειρά εισαγωγής διαφορετικών ζευγών κλειδιού-τιμής σε έναν πίνακα συμβόλων δεν έχει σημασία, δηλ. η εισαγωγή είναι ανταλλακτική. Η ρήτρα εξαίρεσης τεκμηριώνει την απουσία των δεξιών πλευρών των εξισώσεων για rem(emp) και find(emp). οι ρήτρες απαιτήσεων και σημάτων στην προδιαγραφή διεπαφής ασχολούνται με αυτές τις "τιμές σφάλματος". Οι ρήτρες μετατροπής και εξαίρεσης μαζί παρέχουν έναν τρόπο να δηλώσουμε ότι αυτή η αλγεβρική προδιαγραφή είναι επαρκώς πλήρης. (Wing, 1990)

Υπάρχουν αναλυτές σύνταξης για τα χαρακτηριστικά και τις διεπαφές Larch. Το Larch Prover έχει χρησιμοποιηθεί για την εκτέλεση σημασιολογικής ανάλυσης των χαρακτηριστικών του Larch.

Τα σύμβολα συναρτήσεων που ορίζονται από τον χρήστη σε ένα χαρακτηριστικό Larch είναι ακριβώς αυτά που χρησιμοποιούνται στις προ-και μετά τις συνθήκες της προδιαγραφής διεπαφής. έχουν τον ίδιο ρόλο με τα ενσωματωμένα σύμβολα όπως το U και χρησιμοποιούνται στις προδιαγραφές Z και VDM. Σε αντίθεση με το Z και το VDM, το Larch δεν συνοδεύεται από ειδική ενσωματωμένη σημείωση ούτε ενσωματωμένους τύπους. Το πλεονέκτημα είναι ότι ο χρήστης δεν χρειάζεται να μάθει κάποιο ειδικό λεξιλόγιο για αυτές τις έννοιες και είναι ελεύθερος να εισάγει όποια σύμβολα επιθυμεί, δίνοντάς τους ακριβώς το νόημα που είναι κατάλληλο για το συγκεκριμένο και το σύνολο. Ακριβώς και μόνο εκείνες οι ιδιότητες ενός τύπου δεδομένων που καθορίζονται πρέπει να δηλώνονται ρητά και να ικανοποιούνται από μια υλοποίηση. Το μειονέκτημα είναι ότι ο χρήστης μπορεί συχνά να χρειάζεται να παρέχει ένα

μεγάλο σύνολο συμβόλων που ορίζονται από τον χρήστη, καθώς και τις εξισώσεις που ορίζουν τη σημασία τους. Εφόσον μοντελοποιήσαμε πίνακες συμβόλων σε Z και VDM με όρους πεπερασμένων αντιστοιχίσεων, δεν χρειαζόταν να δηλώνουμε ρητά ότι η εισαγωγή είναι ανταλλακτική, καθώς αυτή είναι μια ιδιότητα αντιστοιχίσεων, δηλαδή, αυτή η ιδιότητα ήρθε "δωρεάν". Το Εγχειρίδιο Larch χρησιμεύει ως συμβιβασμός μεταξύ των δύο άκρων: παρέχει μια βιβλιοθήκη χαρακτηριστικών που καθορίζουν πολλές γενικές και κοινώς χρησιμοποιούμενες έννοιες, π.χ. ιδιότητες πεπερασμένων αντιστοιχίσεων, μερικές τάξεις, σύνολα και ακολουθίες. (Wing, 1990)

ΚΕΦΑΛΑΙΟ 2

2.1 ΠΡΩΤΟΚΟΛΛΟ ΕΠΙΚΟΙΝΩΝΙΑΣ

Ως πρωτόκολλο επικοινωνίας ορίζεται ένα σύνολο κανόνων για την ανταλλαγή πληροφοριών μεταξύ δύο μερών, οι οποίοι γίνονται αποδεκτοί και από τα δύο μέρη μιας επικοινωνίας. Ένα πρωτόκολλο επικοινωνίας είναι ένα σύνολο κανόνων στο οποίο βασίζεται η επικοινωνία συσκευών (συνήθως, αλλά όχι απαραίτητα υπολογιστών) σε ένα δίκτυο. Οι κανόνες αυτοί καθορίζουν τη μορφή, το χρονοδιάγραμμα και την αλληλουχία της μετάδοσης πληροφοριών στο δίκτυο. Μπορούν επίσης να παρέχουν έλεγχο σφαλμάτων και διόρθωση σφαλμάτων κατά τη μετάδοση των πληροφοριών. Υπάρχουν διάφορα πρωτόκολλα επικοινωνίας, τα οποία συχνά προκαλούν σύγχυση στους χρήστες.

Προτού πραγματοποιηθεί επιτυχής μετάδοση, οι δικτυωμένες συσκευές επικοινωνίας πρέπει να συμφωνήσουν σε πολλές φυσικές πτυχές των δεδομένων που πρόκειται να ανταλλάσσονται. Οι κανόνες που ορίζουν τις μεταδόσεις δεδομένων ονομάζονται «πρωτόκολλα». (Rouse, 2023)

Υπάρχουν πολλές ιδιότητες μιας μετάδοσης που μπορεί να ορίσει ένα πρωτόκολλο. Για παράδειγμα, οι ιδιότητες που αντιμετωπίζονται με πρωτόκολλα μπορεί να περιλαμβάνουν (Rouse, 2023):

- Packet size - Μέγεθος πακέτου.
- Transmission speed. - Ταχύτητα μετάδοσης.
- Error correction types - Τύποι διόρθωσης σφαλμάτων.
- Handshaking and synchronization techniques - Τεχνικές χειραψίας και συγχρονισμού.
- Address mapping - Χαρτογράφηση διεύθυνσης.
- Acknowledgment processes - Διαδικασίες αναγνώρισης.
- Flow control - Έλεγχος ροής.
- Packet sequence controls - Στοιχεία ελέγχου ακολουθίας πακέτων.
- Routing - Δρομολόγηση.
- Address formatting - Μορφοποίηση διεύθυνσης.

2.2 ΤΟ ΠΡΩΤΟΚΟΛΛΟ ΕΠΙΚΟΙΝΩΝΙΑΣ TFTP

Το TFTP (Trivial File Transfer Protocol) είναι ένα απλό πρωτόκολλο για τη μετακίνηση αρχείων μεταξύ μηχανών. Έχει σχεδιαστεί για να είναι μικρό και εύκολο στην εφαρμογή του, επομένως δεν διαθέτει τα περισσότερα από τα συνηθισμένα χαρακτηριστικά του FTP. Το πρωτόκολλο υποστηρίζει μόνο την ανάγνωση και εγγραφή αρχείων από/προς έναν απομακρυσμένο διακομιστή. Δεν μπορεί να παραθέσει καταλόγους και προς το παρόν δεν υποστηρίζει έλεγχο ταυτότητας χρήστη. Το TFTP είναι ένα πρωτόκολλο με αυστηρούς περιορισμούς μεταφοράς δεδομένων. Όταν παρουσιαστεί σφάλμα, η τρέχουσα μεταφορά διακόπτεται και η σύνδεση τερματίζεται, επομένως είναι απαραίτητο να πραγματοποιήσετε τη σύνδεση και να ξεκινήσετε ξανά τη μεταφορά. Η επικοινωνία μεταξύ διακομιστή και πελάτη θα περιγραφεί σε επίπεδο ανταλλαγής πακέτων. Ανάλογα με τον τύπο των δεδομένων μέσα σε ένα πακέτο, τα πακέτα μπορούν να υποδιαιρεθούν σε τύπους που συνοψίζονται στον Πίνακα 1.1. (Šimoňák, 2012)

Packet type	Description
RRQ	Read request
WRQ	Write request
D1	Data 1 (first packet of a file)
DL	Data L (last packet of a file)
DN	Data N (N-th packet of a file)
ACK0	Answer after the request for writing to the server was received
ACK1	Answer after receiving the first packet of the file
ACKN	Answer after receiving the N-th packet of the file
ERR	Error

Πίνακας 1.1: TFTP PACKET TYPES

Όλα τα πακέτα της επικοινωνίας εξυπηρετούν έναν από τους παρακάτω σκοπούς:

• Για να μεταφέρετε το (τμήματα του) αρχείου, δηλαδή πακέτα δεδομένων (D1, DL, DN).

• Για τον έλεγχο της μεταφοράς, δηλαδή των πακέτων ελέγχου (ACK0, ACK1, ACKN, ERR, RRQ, WRQ).

Η βασική λειτουργία TFTP περιλαμβάνει την ανάγνωση ενός αρχείου από τον διακομιστή και την εγγραφή ενός αρχείου στον διακομιστή, αντίστοιχα. Ας περιγράψουμε αυτές τις δραστηριότητες με περισσότερες λεπτομέρειες (Šimoňák, 2012).

2.3 ΑΝΑΓΝΩΣΗ ΕΝΟΣ ΑΡΧΕΙΟΥ ΑΠΟ ΤΟ SERVER

Ανάγνωση ενός αρχείου από τον διακομιστή

Η λειτουργία ξεκινά με την αποστολή του πακέτου RRQ από τον πελάτη. Ο διακομιστής μπορεί

να απαντήσει σε αυτό το αίτημα με τρεις τρόπους:

• Με αποστολή ERR, εάν το αρχείο που ζητήθηκε δεν υπάρχει ή δεν μπορεί να το διαβάσει

αρχείο.

• Με αποστολή D1 – θετική απάντηση στο αίτημα και ένα πρώτο πακέτο του αρχείου.

• Με αποστολή DL – μια θετική απάντηση επίσης, αλλά και ένα σήμα, ότι αυτό είναι το μόνο

πακέτο του αρχείου.

Όταν ένας πελάτης λάβει το πακέτο ERR, η ανάγνωση του αρχείου τερματίζεται και μπορεί να στείλει το νέο του αίτημα. Ο πελάτης απαντά στο D1 και στο DL (που λαμβάνεται ως το πρώτο μετά το πακέτο RRQ) στέλνοντας το πακέτο ACK1. Εάν ο διακομιστής λάβει το πακέτο ACK1 μετά την αποστολή του DL, τερματίζει τη σύνδεση και είναι έτοιμος για το επόμενο αίτημα. Στην περίπτωση που ο διακομιστής λάβει ACK1 μετά την αποστολή του D1, αποκρίνεται με το επόμενο τμήμα του αρχείου με τη μορφή πακέτου DN ή DL, όπου το τελευταίο από τα δύο χρησιμοποιείται, όταν πρόκειται να σταλεί το τελευταίο μέρος του αρχείου. Ο πελάτης απαντά στο λαμβανόμενο DN από το πακέτο ACKN, όπου N είναι ο αριθμός του πακέτου, και στο πακέτο DL (όταν δεν είναι μόνο το πακέτο του αρχείου), απαντά επίσης με DN (Šimoňák, 2012).

2.4 ΕΓΓΡΑΦΗ ΕΝΟΣ ΑΡΧΕΙΟΥ ΣΤΟ SERVER

Η λειτουργία εγγραφής είναι πολύ παρόμοια με αυτή της ανάγνωσης, με μια σημαντική εξαίρεση: μετά τη λήψη της αίτησης εγγραφής με τη μορφή του πακέτου WRQ, ο διακομιστής απαντά από το ACK0, το οποίο είναι ένας τύπος πακέτου που προορίζεται ειδικά για αυτόν τον σκοπό. Η υπόλοιπη επικοινωνία εκτελείται αναλογικά με τη λειτουργία ανάγνωσης που εξηγήθηκε παραπάνω (Šimoňák, 2012).

2.5 ΤΥΠΙΚΗ ΠΡΟΔΙΑΓΡΑΦΗ

Οι τυπικές προδιαγραφές χρησιμοποιούν μαθηματικό συμβολισμό για να περιγράψουν με ακριβή τρόπο τις ιδιότητες που πρέπει να έχει ένα πληροφοριακό σύστημα, χωρίς να περιορίζουν χωρίς λόγο τον τρόπο με τον οποίο επιτυγχάνονται αυτές οι ιδιότητες. Περιγράφουν τι πρέπει να κάνει το σύστημα χωρίς να λένε πώς θα γίνει. Αυτή η αφαίρεση καθιστά τις τυπικές προδιαγραφές χρήσιμες στη διαδικασία ανάπτυξης ενός συστήματος υπολογιστή, επειδή επιτρέπουν σε ερωτήσεις σχετικά με το τι κάνει το σύστημα να απαντώνται με ακρίβεια, χωρίς να χρειάζεται να ξεμπερδέψουμε τις πληροφορίες από μια μάζα λεπτομερούς κώδικα προγράμματος ή να μαντέψουμε τη σημασία των φράσεων μέσα σε μια ανακριβή πεζογραφική περιγραφή. Μια τυπική προδιαγραφή μπορεί να χρησιμεύσει ως ένα ενιαίο, αξιόπιστο σημείο αναφοράς για εκείνους που διερευνούν τις ανάγκες του πελάτη, εκείνους που εφαρμόζουν προγράμματα για την ικανοποίηση αυτών των αναγκών, εκείνους που δοκιμάζουν τα αποτελέσματα και εκείνους που γράφουν εγχειρίδια οδηγιών για το σύστημα. Επειδή είναι ανεξάρτητο από τον κώδικα του προγράμματος, μια τυπική προδιαγραφή ενός συστήματος μπορεί να ολοκληρωθεί νωρίς στην ανάπτυξή του, αν και μπορεί να χρειαστεί να αλλάξει καθώς η ομάδα σχεδιασμού αποκτά κατανόηση και εξελίσσονται οι αντιληπτές ανάγκες του πελάτη, μπορεί να είναι ένα πολύτιμο μέσο για την προώθηση μιας κοινής κατανόησης μεταξύ όλων όσων ασχολούνται με το σύστημα. Ένας τρόπος με τον οποίο η μαθηματική σημειογραφία μπορεί να οδηγήσει στην επίτευξη αυτών των στόχων είναι μέσω της χρήσης μαθηματικών τύπων δεδομένων για τη μοντελοποίηση των δεδομένων σε ένα σύστημα. Αυτοί οι τύποι δεδομένων δεν είναι προσανατολισμένοι στην αναπαράσταση υπολογιστή, αλλά υπακούουν σε μια πλούσια συλλογή μαθηματικών νόμων που καθιστούν δυνατό να συλλογιστούμε αποτελεσματικά τον τρόπο με τον οποίο θα συμπεριφέρεται ένα συγκεκριμένο σύστημα. Χρησιμοποιούμε τον συμβολισμό της λογικής κατηγορήματος για να περιγράψουμε αφηρημένα το αποτέλεσμα κάθε λειτουργίας του συστήματός μας, όπου με αυτό τον τρόπο έχουμε τη δυνατότητα να συλλογιστούμε τη συμπεριφορά της. (Spivey, 1998)

Ένα κύριο συστατικό της Z είναι ένας τρόπος αποσύνθεσης μιας προδιαγραφής σε μικρά κομμάτια που ονομάζονται σχήματα. Διαχωρίζοντας την προδιαγραφή σε σχήματα, μπορούμε να την παρουσιάσουμε κομμάτι-κομμάτι. Κάθε κομμάτι μπορεί να συνδεθεί με ένα σχόλιο που εξηγεί ανεπίσημα τη σημασία των τυπικών μαθηματικών. Στη Z, τα σχήματα χρησιμοποιούνται για να περιγράψουν τόσο στατικές όσο και δυναμικές πτυχές ενός συστήματος. (Spivey, 1998)

Οι στατικές πτυχές περιλαμβάνουν:

- τις καταστάσεις που μπορεί να περιέχει
- τις αμετάβλητες σχέσεις που διατηρούνται καθώς το σύστημα μετακινείται από κατάσταση σε κατάσταση.

Οι δυναμικές πτυχές περιλαμβάνουν:

- τις λειτουργίες που είναι δυνατές
- σχέση μεταξύ εισροών και εκροών
- τις αλλαγές κατάστασης που συμβαίνουν.

(Spivey, 1998)

Επιπλέον θα δούμε πώς η γλώσσα σχήματος επιτρέπει σε διαφορετικές πτυχές ενός συστήματος να περιγραφεί χωριστά, στη συνέχεια να συσχετίζει και να συνδυάζει. Η εξέλιξη μιας ενιαίας διαδικασίας σε ένα πλήρες σύστημα μπορεί να περιγραφεί μεμονωμένα και στη συνέχεια να σχετίζεται με την εξέλιξη του συστήματος στο σύνολό του. Κατασκευάζοντας μια ακολουθία προδιαγραφών, καθεμία από τις οποίες περιέχει αρκετές λεπτομέρειες, μπορούμε τελικά να φτάσουμε σε ένα πρόγραμμα με την πεποίθηση ότι είναι ικανοποιητικό σύμφωνα με την προδιαγραφή.

(Spivey, 1998)

2.6 ΤΥΠΙΚΕΣ ΜΕΘΟΔΟΙ

1. Η τυπική προδιαγραφή αποτελεί μέρος μιας γενικότερης συλλογής τεχνικών που είναι γνωστές ως Τυπικές Μέθοδοι.
2. Όλες αυτές οι τεχνικές βασίζονται στη μαθηματική αναπαράσταση και ανάλυση του λογισμικού.
3. Οι Τυπικές Μεθόδους περιλαμβάνουν:
    - Τυπική προδιαγραφή.
    - Ανάλυση και τεκμηρίωση προδιαγραφών.
    - Μετασχηματιστική ανάπτυξη.
    - Επικύρωση προγράμματος.

(Sommerville, 2011)

Χρήση των Τυπικών Μεθόδων

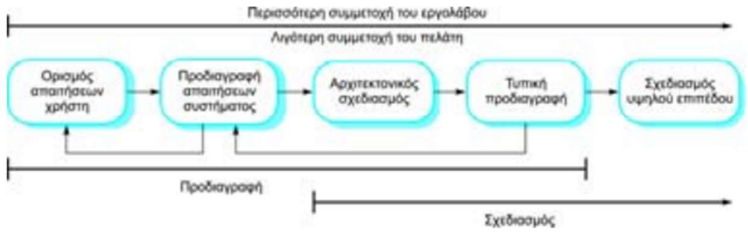
- Τα κύρια πλεονεκτήματα της τυπικής μεθόδου σχετίζονται με τη μείωση των ελαττωμάτων που υπάρχουν στο σύστημα.

- Ως εκ τούτου, ο κύριος τομέας εφαρμογής είναι η μηχανική κρίσιμων συστημάτων. Υπάρχουν πολλά επιτυχημένα έργα σε αυτόν τον τομέα στα οποία έχουν χρησιμοποιηθεί τυπικές μέθοδοι.
- Η χρήση των τυπικών μεθόδων σε αυτόν τον τομέα είναι πιθανό να είναι οικονομικά αποδοτική, καθώς πρέπει να αποφευχθεί το υψηλό κόστος λόγω ελαττωμάτων του συστήματος.

(Sommerville, 2011)

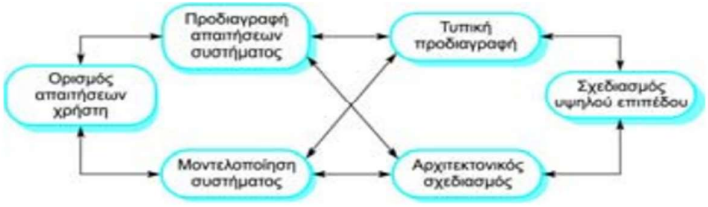
Προδιαγραφές στη διαδικασία παραγωγής λογισμικού

- Οι προδιαγραφές και ο σχεδιασμός είναι έννοιες άρρηκτα συνδεδεμένες.
- Ο αρχιτεκτονικός σχεδιασμός αποτελεί τη βάση όχι μόνο για την δόμηση των προδιαγραφών, αλλά και για ολόκληρη τη διαδικασία ανάδειξης προδιαγραφών.
- Οι τυπικές προδιαγραφές εκφράζονται σε μαθηματικό συμβολισμό η οποία έχει συγκεκριμένο λεξικό, συντακτική δομή και σημασιολογία.



Εικόνα 1 Προδιαγραφή και σχεδιασμός

(Sommerville, 2011)



Εικόνα 2 Προδιαγραφή στη διαδικασία παραγωγής λογισμικού

Χρήση της τυπικής προδιαγραφής

- Οι τυπικές προδιαγραφές απαιτούν μεγαλύτερη προσπάθεια στα αρχικά στάδια.
- Απαιτούν λεπτομερή ανάλυση των απαιτήσεων και έτσι μειώνουν τα λάθη στις απαιτήσεις.
- Οι ασυνέπειές και τα στοιχεία που λείπουν μπορούν έτσι να εντοπιστούν και να αντιμετωπιστούν.
- Αυτό μειώνει την επανεπεξεργασία που προκαλείται από προβλήματα απαιτήσεων και μπορεί να οδηγήσει σε εξοικονόμηση πόρων σε πολλούς τομείς.

(Sommerville, 2011)

Τεχνικές προδιαγραφής:

- Αλγεβρική προδιαγραφή.
- Τα συστήματα ορίζονται από την άποψη των λειτουργιών τους και των διαλειτουργικών τους σχέσεων.
- Προδιαγραφή βάσει μοντέλου.
- Ένα σύστημα περιγράφεται με βάση ένα μοντέλο κατάστασης του συστήματος όπως σύνολα και ακολουθίες. Οι λειτουργίες του συστήματος ορίζονται με βάση τον τρόπο με τον οποίο μεταβάλλεται η κατάσταση του συστήματος.

	Ακολουθιακή	**Ταυτόχρονη**
Αλγεβρική	Larch (Guttag κ.ά., 1993) OBJ (Futatsugi κ.ά., 1985)	Lotos (Bolognesi και Brinksma, 1987)
Βάσει μοντέλου	Z (Spivey, 1992) VDM (Jones, 1980) B (Wordsworth, 1996)	CSP (Hoare, 1985) Δίκτυα Petri (Peterson, 1981)

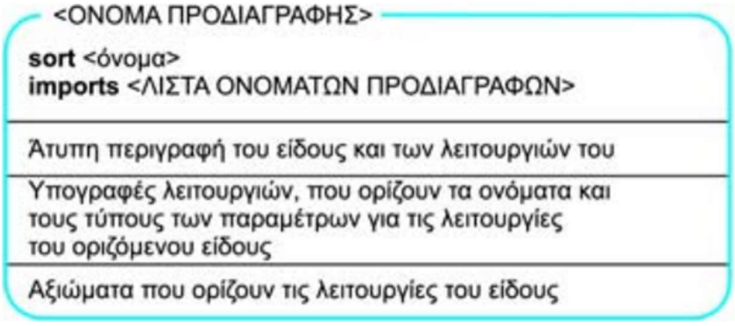
Εικόνα 3 Γλώσσες τυπικής προδιαγραφής: (Sommerville, 2011)

Προδιαγραφές διεπαφής:

- Τα μεγάλα συστήματα συχνά αναλύονται σε υποσυστήματα και διαθέτουν επακριβώς καθορισμένες διεπαφές για την επικοινωνία μεταξύ των υποσυστημάτων.

- Οι προδιαγραφές διεπαφών υποσυστημάτων επιτρέπουν την ανεξάρτητη ανάπτυξη κάθε υποσυστήματος.
- Οι διεπαφές ορίζονται με τη μορφή αφηρημένων τύπων δεδομένων ή κλάσεων αντικειμένων.
- Η αλγεβρική προσέγγιση των τυπικών προδιαγραφών είναι κατάλληλη για τις προδιαγραφές διεπαφών, καθώς εστιάζει στην καθορισμένη λειτουργικότητα των αντικειμένων.

(Sommerville, 2011)



Εικόνα 4 Η δομή μίας αλγεβρικής προδιαγραφής:

Στοιχεία των προδιαγραφών:

- Εισαγωγή.
- Υποδεικνύει τον καθορισμένο τύπο οντότητας (ταξινόμηση – όνομα τύπου) και καθορίζει άλλες ιδιότητες που χρησιμοποιούνται.
- Περιγραφή.
- Περιγράφει ανεπίσημα τη λειτουργικότητα.
- Υπογραφή.
- Καθορίζει τη σύνταξη της συνάρτησης διασύνδεσης και των παραμέτρων της.
- Χαρακτηρίστηκα.
- Καθορίζει τη σημασιολογία μια συνάρτησης μέσω ενός συνόλου αξιωμάτων που χαρακτηρίζουν τη συμπεριφορά της.

(Sommerville, 2011)

Συστηματικές αλγεβρικές ιδιότητες:

- Οι αλγεβρικές προδιαγραφές των συστημάτων μπορούν να αναπτυχθούν με συστηματικό τρόπο.
- Δόμηση της προδιαγραφής.
- Όνομα της προδιαγραφής.
- Επιλογή συναρτήσεων.
- Άτυπη προδιαγραφή των συναρτήσεων.
- Ορισμός του συντακτικού.
- Ορισμός αξιωμάτων.

(Sommerville, 2011)

Λειτουργίες προδιαγραφής:

- Συνάρτηση παραγωγής. Δημιουργεί περιουσιακά στοιχεία του τύπου που ορίζεται στην προδιαγραφή.
- Λειτουργία επιθεώρησης. Υπολογίζει οντότητες του τύπου που ορίζεται στην προδιαγραφή.
- Για την εξειδίκευση της διαδικασίας πρέπει να οριστεί μια λειτουργία επιθεώρησης για κάθε λειτουργία παραγωγής.

(Sommerville, 2011)

Προσδιορισμός των διεπαφών με κρίσιμα συστήματα:

- Σκεφτείτε ένα σύστημα ελέγχου εναέριας κυκλοφορίας όπου τα αεροσκάφη κινούνται σε ελεγχόμενους τομείς του εναέριου χώρου.
- Κάθε τομέας μπορεί να περιέχει μεγάλο αριθμό αεροσκαφών, αλλά για λόγους ασφαλείας πρέπει να απέχουν μεταξύ τους.
- Σε αυτό το παράδειγμα, προτείνεται κατακόρυφος διαχωρισμός 300 μέτρων.
- Εάν ένα αεροσκάφος επιχειρήσει να λάβει θέση που παραβιάζει αυτό το όριο, το σύστημα πρέπει να ειδοποιήσει το κέντρο ελέγχου εναέριας κυκλοφορίας.

(Sommerville, 2011)

Αντικείμενα τομέα:

Οι βασικές ιδιότητες που ισχύουν για τα αντικείμενα που αντιπροσωπεύουν ελεγχόμενους τομείς είναι οι εξής:

- Enter (είσοδος). Προσθήκη αεροσκαφών σε ελεγχόμενο εναέριο χώρο.
- Leave (αποχώρηση). Το αεροσκάφος εγκαταλείπει τον ελεγχόμενο εναέριο χώρο.
- Move (μετακίνηση). Μετακινεί ένα αεροσκάφος από ένα υψόμετρο σε ένα άλλο.
- Lookup (αναζήτηση). Η αναζήτηση επιστρέφει το τρέχον υψόμετρο όταν δίνεται η ταυτότητα του αεροσκάφους.

(Sommerville, 2011)

Απλούστερες λειτουργίες

- Για την απλούστευση των προδιαγραφών μπορεί να χρειαστεί να οριστούν πρόσθετες συναρτήσεις.
- Όλες οι άλλες συναρτήσεις μπορούν στη συνέχεια να οριστούν χρησιμοποιώντας αυτές τις απλούστερες συναρτήσεις.
- Απλούστερες συναρτήσεις:
    1. Create (δημιουργία). Δημιουργίας ενός κενού στιγμιότυπου τομέα.
    2. Put (τοποθέτηση). Τοποθετεί αεροσκάφη στον τομέα χωρίς να ελέγξει τους περιορισμούς ασφαλείας.
    3. In-space (στο χώρο). Προσδιορίζει αν το συγκεκριμένο αεροσκάφος βρίσκεται στον συγκεκριμένο τομέα.
    4. Occupied (κατειλημμένο). Εάν έχει καθοριστεί υψόμετρο, προσδιορίζει εάν το αεροσκάφος βρίσκεται σε απόσταση 300 μέτρων από το καθορισμένο υψόμετρο.

(Sommerville, 2011)

Χαρακτηριστικά συμπεριφοράς

- Οι αλγεβρικές προδιαγραφές καθίστανται άχρηστες όταν η λειτουργία ενός αντικειμένου εξαρτάται από την κατάσταση του αντικειμένου.
- Οι προδιαγραφές που βασίζονται σε μοντέλα περιγράφουν την κατάσταση του συστήματος και ορίζουν λειτουργίες με βάση τις αλλαγές σε αυτή την κατάσταση.
- Η Z-notation είναι μία ώριμη τεχνική προδιαγραφών βασισμένων σε μοντέλα. Επιπλέον συνδυάζει τυπικές και άτυπες επισημάνσεις και χρησιμοποιεί επισημάνσεις με γραφικά στοιχεία για την παρουσίαση των προδιαγραφών.

Εικόνα 5 Η δομή ενός σχήματος της σημειογραφίας Z:

(Sommerville, 2011)

Βασικά σημεία:

- Οι μέθοδοι της τυπικής προδιαγραφής συστημάτων συμπληρώνουν τις άτυπες μεθόδους προδιαγραφής των απαιτήσεων.
- Οι τυπικές προδιαγραφές είναι ακριβείς και σαφείς. Αποσαφηνίζουν τις ασάφειες στις προδιαγραφές.
- Οι τυπικές προδιαγραφές απαιτούν την ανάλυση των απαιτήσεων του συστήματος σε πρώιμο στάδιο. Η διόρθωση σφαλμάτων σε αυτό το στάδιο είναι λιγότερο δαπανηρή από τη διόρθωση του παραδοτέου συστήματος.
- Οι τεχνικές ορισμού τυπικών προδιαγραφών εφαρμόζονται κυρίως στην ανάπτυξη κρίσιμων συστημάτων και προτύπων.
- Οι αλγεβρικές τεχνικές είναι κατάλληλες για προδιαγραφές διεπαφών, όπου η διεπαφή ορίζεται ως ένα σύνολο κλάσεων αντικειμένων.

- Οι τεχνικές που βασίζονται σε μοντέλα χρησιμοποιούν σύνολα και συναρτήσεις για την μοντελοποίηση συστημάτων. Αυτό απλοποιεί ορισμένες προδιαγραφές συμπεριφοράς.

Σε μια προδιαγραφή βασισμένη στο μοντέλο, οι συναρτήσεις ορίζονται με προϋποθέσεις και μετασυνθήκες που σχετίζονται με την κατάσταση του συστήματος. (Sommerville, 2011).

2.6 ΣΧΕΤΙΚΑ ΜΕ ΤΗΝ ΤΥΠΙΚΗ ΠΡΟΔΙΑΓΡΑΦΗ

Η τυπική προδιαγραφή είναι μια αυστηρή και συστηματική προσέγγιση για την επεξήγηση της συμπεριφοράς και των ιδιοτήτων ενός συστήματος λογισμικού ή στοιχείου χρησιμοποιώντας μαθηματικές σημειώσεις και γλώσσες. Χρησιμεύει ως τυπική και ξεκάθαρη αναπαράσταση των απαιτήσεων, του σχεδιασμού και της λειτουργικότητας ενός συστήματος. Ο πρωταρχικός στόχος των τυπικών προδιαγραφών είναι η μείωση της ασάφειας, η διασφάλιση της ορθότητας, η διευκόλυνση της επαλήθευσης και η βελτίωση της επικοινωνίας μεταξύ των ενδιαφερομένων που εμπλέκονται στην ανάπτυξη λογισμικού. Ακολουθούν ορισμένες βασικές πτυχές και πλεονεκτήματα των τυπικών προδιαγραφών:

1. Οι τυπικές γλώσσες προδιαγραφών, όπως οι Z, VDM (Μέθοδος Ανάπτυξης Βιέννης) και Alloy, παρέχουν έναν ακριβή και ξεκάθαρο τρόπο περιγραφής της συμπεριφοράς του συστήματος. Αυτή η ακρίβεια βοηθά στην εξάλειψη παρεξηγήσεων και ασάφειών στις απαιτήσεις και στα έγγραφα σχεδιασμού.
2. Τυπικές σημειώσεις: Οι τυπικές προδιαγραφές χρησιμοποιούν συχνά μαθηματικές σημειώσεις και σύμβολα για να αναπαραστήσουν τα στοιχεία του συστήματος, τις ιδιότητές τους και τις αλληλεπιδράσεις τους. Αυτές οι σημειώσεις επιτρέπουν τη συνοπτική και σαφή έκφραση σύνθετων ιδεών.
3. Αφαίρεση: Οι τυπικές προδιαγραφές υποστηρίζουν διάφορα επίπεδα αφαίρεσης, επιτρέποντας στους μηχανικούς λογισμικού να συλλαμβάνουν απαιτήσεις συστήματος υψηλού επιπέδου καθώς και λεπτομέρειες υλοποίησης χαμηλού επιπέδου. Αυτή η αφαίρεση διευκολύνει τον διαχωρισμό των ανησυχιών και επιτρέπει τη σταδιακή βελτίωση.
4. Με βάση το μοντέλο: Οι τυπικές προδιαγραφές χρησιμοποιούνται συχνά σε μια προσέγγιση ανάπτυξης που βασίζεται σε μοντέλο. Ένα μοντέλο του συστήματος δημιουργείται χρησιμοποιώντας τυπικούς συμβολισμούς και αυτό το μοντέλο μπορεί να αναλυθεί, να τελειοποιηθεί και να μετασχηματιστεί για να εξαχθεί η τελική υλοποίηση.
5. Επαλήθευση: Ένα από τα κύρια πλεονεκτήματα των τυπικών προδιαγραφών είναι η δυνατότητα μαθηματικής επαλήθευσης των ιδιοτήτων του συστήματος, όπως η ορθότητα, η ασφάλεια και η

ζωτικότητα. Τα εργαλεία και οι τεχνικές τυπικών μεθόδων μπορούν να αποδείξουν ή να διαψεύσουν την ικανοποίηση αυτών των ιδιοτήτων, οδηγώντας σε πιο αξιόπιστα συστήματα.
6. Έλεγχος συνέπειας: Οι τυπικές προδιαγραφές επιτρέπουν τον αυτοματοποιημένο έλεγχο συνέπειας, διασφαλίζοντας ότι διαφορετικά μέρη του συστήματος ή απαιτήσεις δεν έρχονται σε αντίθεση μεταξύ τους.
7. Τεκμηρίωση: Οι τυπικές προδιαγραφές χρησιμεύουν ως ζωντανή τεκμηρίωση για ένα έργο λογισμικού. Παρέχουν μια σαφή αναφορά για προγραμματιστές, δοκιμαστές και άλλα ενδιαφερόμενα μέρη σε όλο τον κύκλο ζωής ανάπτυξης λογισμικού.
8. Έγκαιρη ανίχνευση ζητημάτων: Οι τυπικές μέθοδοι μπορούν να βοηθήσουν στον εντοπισμό ζητημάτων και ελαττωμάτων στη φάση του σχεδιασμού ή των απαιτήσεων, επιτρέποντας τον έγκαιρο μετριασμό, ο οποίος είναι συχνά πιο οικονομικός από την αντιμετώπιση θεμάτων αργότερα στη διαδικασία ανάπτυξης.
9. Επικοινωνία: Οι τυπικές προδιαγραφές γεφυρώνουν το χάσμα μεταξύ των ειδικών του τομέα, των μηχανικών λογισμικού και άλλων ενδιαφερόμενων μερών. Παρέχουν μια κοινή γλώσσα και κατανόηση της συμπεριφοράς του συστήματος.
10. Σύνθετα συστήματα: Οι τυπικές προδιαγραφές είναι ιδιαίτερα πολύτιμες για συστήματα υψηλής κρισιμότητας, όπου η ορθότητα και η ασφάλεια είναι πρωταρχικής σημασίας, όπως η αεροδιαστημική, οι ιατρικές συσκευές και τα συστήματα αυτοκινήτου
(Keith D. Cooper, 2012), (Spivey, 1998).

Προκλήσεις και προβληματισμοί:

Ενώ μια τυπική προδιαγραφή προσφέρει πολλά πλεονεκτήματα, παρουσιάζει επίσης ορισμένες προκλήσεις:

- Καμπύλη εκμάθησης: Οι Τυπικές Μέθοδοι και σημειώσεις μπορεί να είναι περίπλοκες και μπορεί να απαιτούν εκπαίδευση για τους προγραμματιστές ώστε να γίνουν ικανοί στη χρήση τους.
- Υποστήριξη εργαλείων: Η αποτελεσματική χρήση Τυπικών Μεθόδων συχνά βασίζεται σε εξειδικευμένα εργαλεία και η διαθεσιμότητα τέτοιων εργαλείων μπορεί να ποικίλλει.
- Κόστος: Η ανάπτυξη τυπικών προδιαγραφών και η εκτέλεση τυπικής επαλήθευσης μπορεί να είναι πιο χρονοβόρα και δαπανηρή σε σύγκριση με τις άτυπες μεθόδους.
- Επεκτασιμότητα: Για μεγάλα και πολύπλοκα συστήματα, η δημιουργία ολοκληρωμένων τυπικών προδιαγραφών μπορεί να είναι πρόκληση.
- Πλήρης τυπική επαλήθευση: Η επίτευξη 100% τυπικής επαλήθευσης δεν είναι πάντα εφικτή λόγω της πολυπλοκότητας των συστημάτων του πραγματικού κόσμου.

Συνοπτικά, η τυπική προδιαγραφή είναι μια ισχυρή τεχνική που μας βοηθά ώστε να βελτιώσουμε τη ποιότητα και την αξιοπιστία των συστημάτων λογισμικού παρέχοντας μια μαθηματικά ακριβή περιγραφή της συμπεριφοράς του συστήματος. Αν και μπορεί να μην είναι χρήσιμο για όλα τα έργα, είναι ένα πολύτιμο εργαλείο για κρίσιμα συστήματα και εκείνα με αυστηρές απαιτήσεις για ορθότητα και ασφάλεια (Keith D. Cooper, 2012), (Spivey, 1998).

2.7 ΑΚΟΛΟΥΘΟΥΝ ΟΡΙΣΜΕΝΑ ΒΑΣΙΚΑ ΣΗΜΕΙΑ ΣΧΕΤΙΚΑ ΜΕ ΤΗ ΣΗΜΕΙΩΣΗ Z:

- Μαθηματική βάση: Ο συμβολισμός Z βασίζεται στη θεωρία συνόλων και στη λογική κατηγορήματος πρώτης τάξης, καθιστώντας τον έναν μαθηματικά αυστηρό τρόπο περιγραφής συστημάτων. Παρέχει έναν τυπικό και ακριβή τρόπο προσδιορισμού της συμπεριφοράς και των ιδιοτήτων ενός συστήματος.
- Σημείωση: Ο συμβολισμός Z χρησιμοποιεί έναν συνδυασμό μαθηματικών συμβόλων και αγγλικών λέξεων για να καθορίσει τη δομή και τον τρόπο συμπεριφοράς ενός συστήματος. Χρησιμοποιεί σχήματα για να αναπαραστήσει διαφορετικές πτυχές ενός συστήματος, όπως κατάσταση, λειτουργίες και περιορισμούς.
- Τύποι προδιαγραφών: Ο συμβολισμός Z χρησιμοποιείται τόσο για αφηρημένες προδιαγραφές όσο και για συγκεκριμένες προδιαγραφές. Οι αφηρημένες προδιαγραφές επικεντρώνονται στη συμπεριφορά και τις ιδιότητες υψηλού επιπέδου ενός συστήματος, ενώ οι συγκεκριμένες προδιαγραφές παρέχουν λεπτομερείς περιγραφές των στοιχείων του συστήματος.
- Επαλήθευση: Ένας από τους αρχικούς σκοπούς του συμβολισμού Z είναι η υποστήριξη της τυπικής επαλήθευσης. Επιτρέπει τη μαθηματική απόδειξη ιδιοτήτων, όπως η ορθότητα και η ασφάλεια, ενός συστήματος. Η τυπική επαλήθευση βοηθά στον εντοπισμό και την εξάλειψη σφαλμάτων νωρίς στη διαδικασία ανάπτυξης λογισμικού.
- Υποστήριξη εργαλείων: Έχουν αναπτυχθεί διάφορα εργαλεία και περιβάλλοντα ανάπτυξης για την υποστήριξη του συμβολισμού Z. Αυτά τα εργαλεία βοηθούν στη σύνταξη, την ανάλυση και την επαλήθευση των προδιαγραφών Z.
- Βιομηχανικές Εφαρμογές: Ο συμβολισμός Z έχει χρησιμοποιηθεί σε βιομηχανίες όπως η αεροδιαστημική, οι μεταφορές και η χρηματοδότηση για τον προσδιορισμό και την επαλήθευση κρίσιμων συστημάτων όπου η ορθότητα είναι υψίστης σημασίας.
- Σύνθετα συστήματα: Ενώ η σημείωση Z είναι ιδιαίτερα κατάλληλη για τον επεξήγηση και την επαλήθευση πολύπλοκων και κρίσιμων για την ασφάλεια συστημάτων, μπορεί να χρησιμοποιηθεί σε ένα αρκετά μεγάλο αριθμό συστημάτων λογισμικού και υλικού.

(Spivey, 1998)

Συνοπτικά, ο συμβολισμός Z είναι μια τυπική Γλώσσα προδιαγραφών με ισχυρή μαθηματική βάση που χρησιμοποιείται για τον προσδιορισμό και την επαλήθευση της ορθότητας λογισμικού και πολύπλοκων συστημάτων. Διαδραματίζει κρίσιμο ρόλο στη εξασφάλιση της αξιοπιστίας και της ασφάλειας των κρίσιμων συστημάτων (Spivey, 1998).

2.8 ΣΧΕΤΙΚΑ ΜΕ ΤΗΝ ΕΠΑΛΗΘΕΥΣΗ

Η επαλήθευση, στο πλαίσιο της μηχανικής λογισμικού και των Τυπικών Μεθόδων, αναφέρεται στη διαδικασία αυστηρής και συστηματικής αξιολόγησης εάν ένα σύστημα λογισμικού ή ένα στοιχείο πληροί τις καθορισμένες απαιτήσεις του και συμπεριφέρεται σωστά σύμφωνα με ένα δεδομένο σύνολο ιδιοτήτων ή προδιαγραφών. Η επαλήθευση είναι ένα κρίσιμο βήμα στην ανάπτυξη λογισμικού, καθώς διασφαλίζει ότι το λογισμικό είναι αξιόπιστο, ασφαλές και απαλλαγμένο από κρίσιμα ελαττώματα. Ακολουθούν ορισμένες βασικές πτυχές και μέθοδοι επαλήθευσης λογισμικού (Keith D. Cooper, 2012), (Spivey, 1998):

1. Τυπική επαλήθευση: Αυτή η προσέγγιση χρησιμοποιεί μαθηματικές τεχνικές και Τυπικές Μεθόδους για να αποδείξει την ορθότητα του λογισμικού σε σχέση με μια τυπική προδιαγραφή. Η τυπική επαλήθευση μπορεί να περιλαμβάνει έλεγχο μοντέλου, απόδειξη θεωρημάτων και στατική ανάλυση για την επαλήθευση ιδιοτήτων όπως η ορθότητα του προγράμματος, η ασφάλεια και η ζωτικότητα

2. Δοκιμές: Αν και δεν είναι τόσο αυστηρές όσο οι Τυπικές Μέθοδοι, οι δοκιμές είναι μια κοινή τεχνική επαλήθευσης. Περιλαμβάνει την εκτέλεση του λογισμικού με διάφορες εισόδους για να ελεγχθεί εάν συμπεριφέρεται όπως αναμένεται και να αποκαλυφθούν ελαττώματα. Συνήθως χρησιμοποιούνται τεχνικές όπως η δοκιμή μονάδων, η δοκιμή ενοποίησης και η δοκιμή συστήματος.

3. Έλεγχος μοντέλου: Ο έλεγχος μοντέλου είναι μια αυτοματοποιημένη τεχνική για την επαλήθευση συστημάτων πεπερασμένων καταστάσεων. Εξερευνά συστηματικά όλες τις πιθανές καταστάσεις ενός μοντέλου για να ελέγξει εάν μια δεδομένη ιδιότητα ισχύει. Ο έλεγχος μοντέλου είναι ιδιαίτερα χρήσιμος για την επαλήθευση ταυτόχρονων και αντιδραστικών συστημάτων.

4. Απόδειξη Θεωρήματος: Η απόδειξη θεωρημάτων περιλαμβάνει τη χρήση τυπικής λογικής και μαθηματικών αποδείξεων για να διαπιστωθεί η ορθότητα του λογισμικού. Οι διαδραστικοί αποδείκτες θεωρημάτων, όπως οι Coq και Isabelle, επιτρέπουν στους προγραμματιστές να προσδιορίζουν τυπικά τις ιδιότητες και να κατασκευάζουν τυπικές αποδείξεις ορθότητας.

5. Στατική ανάλυση: Τα εργαλεία στατικής ανάλυσης εξετάζουν τον πηγαίο κώδικα ή άλλες αναπαραστάσεις του λογισμικού χωρίς να το εκτελούν. Μπορούν να εντοπίσουν πιθανά ζητήματα,

όπως σφάλματα τύπου, μηδενικές παραπομπές δείκτη και ευπάθειες ασφαλείας, μέσω ανάλυσης κώδικα.

6. Τυπικές προδιαγραφές: Η επαλήθευση ξεκινά συχνά με την τυπική προδιαγραφή των απαιτήσεων και των ιδιοτήτων του συστήματος. Οι τυπικές προδιαγραφές χρησιμοποιούν μαθηματικούς συμβολισμούς, όπως τον συμβολισμό Z, για να περιγράψουν με ακρίβεια τη συμπεριφορά του συστήματος, τις ιδιότητες ασφαλείας και τα αμετάβλητα.

7. Αυτοματοποιημένα εργαλεία: Διάφορα αυτοματοποιημένα εργαλεία και πλαίσια επαλήθευσης λογισμικού, υποστηρίζουν διαφορετικές τεχνικές επαλήθευσης. Αυτά τα εργαλεία μπορούν να βοηθήσουν στην αυτοματοποίηση της διαδικασίας επαλήθευσης και να την κάνουν πιο προσιτή στους προγραμματιστές.

8. Ιδιότητες ορθότητας: Η επαλήθευση μπορεί να αξιολογήσει διάφορες ιδιότητες ορθότητας, συμπεριλαμβανομένης της λειτουργικής ορθότητας (αν το λογισμικό εκτελεί την προβλεπόμενη λειτουργία του), της ασφάλειας (αν αποφεύγει ανεπιθύμητες συμπεριφορές) και της ζωηρότητας (αν συνεχίζει να σημειώνει πρόοδο).

9. Αναθεώρηση και επιθεωρήσεις κώδικα: Η αναθεώρηση και οι επιθεωρήσεις ανθρώπινου κώδικα περιλαμβάνουν έμπειρους προγραμματιστές που εξετάζουν έγγραφα κώδικα και σχεδιασμού για να εντοπίσουν πιθανά ελαττώματα και ασυνέπειες. Αν και δεν είναι τόσο τυπική όσο άλλες μέθοδοι, η αναθεώρηση κώδικα μπορεί να είναι εξαιρετικά αποτελεσματική.

10. Τυπικές Μέθοδοι σε κρίσιμα συστήματα: Η επαλήθευση είναι ιδιαίτερα σημαντική σε κρίσιμα συστήματα, όπως η αεροδιαστημική, η υγειονομική περίθαλψη, η αυτοκινητοβιομηχανία και τα οικονομικά, όπου οι αστοχίες λογισμικού μπορεί να έχουν σοβαρές επιπτώσεις.
(Keith D. Cooper, 2012), (Spivey, 1998).

Προκλήσεις και προβληματισμοί:

- Πολυπλοκότητα: Η επαλήθευση γίνεται πιο δύσκολη καθώς αυξάνεται η πολυπλοκότητα του λογισμικού ή του συστήματος.
- Επεκτασιμότητα: Ορισμένες τεχνικές επαλήθευσης ενδέχεται να μην επεκταθούν καλά σε μεγάλα και πολύπλοκα συστήματα.
- Επιλογή εργαλείου: Η επιλογή των σωστών εργαλείων και τεχνικών επαλήθευσης για ένα συγκεκριμένο έργο μπορεί να είναι κρίσιμη.
- Κόστος επαλήθευσης: Οι προσπάθειες επαλήθευσης μπορεί να είναι χρονοβόρες και δαπανηρές και τα οφέλη πρέπει να σταθμίζονται έναντι των πιθανών κινδύνων και συνεπειών της αποτυχίας λογισμικού.
- Μερική επαλήθευση: Η επίτευξη επαλήθευσης 100% μπορεί να μην είναι πάντα εφικτή, επομένως μπορεί να χρειαστεί να γίνουν αντισταθμίσεις.

Συνοπτικά, η επαλήθευση λογισμικού είναι ένα κρίσιμο μέρος της διαδικασίας ανάπτυξης λογισμικού, με στόχο τη διασφάλιση της αξιοπιστίας, της ασφάλειας και της ορθότητας των συστημάτων λογισμικού. Περιλαμβάνει μια σειρά τεχνικών, από Τυπικές Μεθόδους έως δοκιμές, και θα πρέπει να είναι προσαρμοσμένη στις συγκεκριμένες ανάγκες και απαιτήσεις του έργου λογισμικού (Keith D. Cooper, 2012), (Spivey, 1998).

ΚΕΦΑΛΑΙΟ 3

3.1 ΣΧΕΤΙΚΑ ΜΕ ΤΗΝ ΕΦΑΡΜΟΓΗ ROCKET.CHAT

Σχετικά με το Rocket.Chat, πρόκειται για μια προσαρμόσιμη πλατφόρμα ανοιχτού κώδικα, έχει σχεδιαστεί για με αυστηρά πρότυπα προστασίας δεδομένων. Επιπλέον προσφέρει κρυπτογράφηση από άκρο σε άκρο, έλεγχο ταυτότητας δύο παραγόντων και ελέγχους πρόσβασης. Σχετικά με τον τρόπο λειτουργίας της, έχει την δυνατότητα να ενσωματώνει πολλαπλά κανάλια, όπως Ζωντανή συνομιλία, email και μέσα κοινωνικής δικτύωσης, βελτιώνοντας την αποτελεσματικότητα. (Rocket.Chat, n.d.).

3.2 ΣΧΕΤΙΚΑ ΜΕ ΤΗΝ ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΤΗΣ ΠΛΑΤΦΟΡΜΑΣ ROCKET.CHAT

Το Rocket.Chat είναι μια πλατφόρμα συνομιλίας και ανταλλαγής μηνυμάτων ανοιχτού κώδικα σε πραγματικό χρόνο, γνωστή για την ευελιξία και την επεκτασιμότητα της. Η αρχιτεκτονική του είναι χτισμένη γύρω από διάφορα στοιχεία και τεχνολογίες για να παρέχει δυνατότητες επικοινωνίας σε πραγματικό χρόνο. Ακολουθεί μια επισκόπηση της αρχιτεκτονικής Rocket.Chat (Rocket.Chat, n.d.), (Rocket.Chat, https://github.com/RocketChat/Rocket.Chat, n.d.).:

1. Frontend:

Διασύνδεση Ιστού: Η διεπαφή ιστού του Rocket.Chat έχει δημιουργηθεί χρησιμοποιώντας τεχνολογίες όπως HTML, CSS και JavaScript (συνήθως με το πλαίσιο Meteor).
Εφαρμογές για υπολογιστές και κινητές συσκευές: Το Rocket.Chat παρέχει εφαρμογές για επιτραπέζιους υπολογιστές και κινητές συσκευές για διάφορες πλατφόρμες, συμπεριλαμβανομένων των Windows, macOS, Linux, iOS και Android.

2. Backend:

Node.js: Η πλευρά του διακομιστή του Rocket.Chat υλοποιείται κυρίως στο Node.js, το οποίο επιτρέπει τον ασύγχρονο προγραμματισμό που βασίζεται σε συμβάντα.
Meteor: Το Meteor είναι ένα πλήρες πλαίσιο JavaScript πλήρους στοίβας που χρησιμοποιείται για την κατασκευή εφαρμογών ιστού σε πραγματικό χρόνο. Το Rocket.Chat χρησιμοποίησε αρχικά το Meteor εκτενώς, αλλά σταδιακά κινήθηκε προς μια πιο αρθρωτή αρχιτεκτονική.

3. Επικοινωνία σε πραγματικό χρόνο:

WebSocket: Το Rocket.Chat χρησιμοποιεί το WebSocket για επικοινωνία σε πραγματικό χρόνο μεταξύ των πελατών και του διακομιστή, διασφαλίζοντας χαμηλή καθυστέρηση και αποτελεσματικές ενημερώσεις.

DDP (Distributed Data Protocol): Το DDP είναι ένα πρωτόκολλο χτισμένο πάνω από το WebSocket που χρησιμοποιεί το Rocket.Chat για να συγχρονίσει τις αλλαγές δεδομένων μεταξύ των πελατών και του διακομιστή.

4. Αποθήκευση δεδομένων:

MongoDB: Το Rocket.Chat χρησιμοποιεί το MongoDB ως το κύριο σύστημα βάσης δεδομένων του για την αποθήκευση δεδομένων χρήστη, ιστορικού συνομιλιών και ρυθμίσεων διαμόρφωσης. Η MongoDB είναι μια βάση δεδομένων NoSQL γνωστή για την ευελιξία και την επεκτασιμότητα της.

5. Ενσωματώσεις και Επεκτασιμότητα:

Εφαρμογές Rocket.Chat: Το Rocket.Chat επιτρέπει την ανάπτυξη και εγκατάσταση προσαρμοσμένων εφαρμογών και ενσωματώσεων, επεκτείνοντας τη λειτουργικότητά του. Οι εφαρμογές μπορούν να γραφτούν σε JavaScript και να αλληλεπιδράσουν με τον διακομιστή Rocket.Chat μέσω API.

6. Έλεγχος ταυτότητας και ασφάλεια:

OAuth: Το Rocket.Chat υποστηρίζει έλεγχο ταυτότητας βάσει OAuth, επιτρέποντας στους χρήστες να συνδεθούν χρησιμοποιώντας τους υπάρχοντες λογαριασμούς τους από υπηρεσίες όπως το Google, το GitHub ή το LDAP.

Κρυπτογράφηση από άκρο σε άκρο: Το Rocket.Chat προσφέρει κρυπτογράφηση από άκρο σε άκρο για ασφαλή ανταλλαγή μηνυμάτων μεταξύ των χρηστών.

7. Επεκτασιμότητα και υψηλή διαθεσιμότητα:

Το Rocket.Chat μπορεί να κλιμακωθεί οριζόντια για να χειριστεί αυξημένα φορτία. Υποστηρίζει τη χρήση εξισορροπητών φορτίου και κατανεμημένων ρυθμίσεων βάσης δεδομένων.

Η υψηλή διαθεσιμότητα μπορεί να επιτευχθεί με την ανάπτυξη του Rocket.Chat σε πολλούς διακομιστές και με τη χρήση της αναπαραγωγής της βάσης δεδομένων.

(Rocket.Chat, n.d.), (Rocket.Chat, https://github.com/RocketChat/Rocket.Chat, n.d.).

ΚΕΦΑΛΑΙΟ 4

4.1 ΜΟΝΤΕΛΟΠΟΙΗΣΗ ΤΗΣ ΕΠΙΚΟΙΝΩΝΙΑΣ CLIENT – SERVER

Η μοντελοποίηση της επικοινωνίας πελάτη και διακομιστή στο Rocket.Chat περιλαμβάνει τον προσδιορισμό του τρόπου με τον οποίο η εφαρμογή πελάτη και ο διακομιστής Rocket.Chat ανταλλάσσουν πληροφορίες και αλληλεπιδρούν μεταξύ τους. Αυτή η επικοινωνία συνήθως ακολουθεί ένα μοτίβο αιτήματος-απόκρισης και βασίζεται σε ένα σύνολο καθορισμένων API και πρωτοκόλλων. Ακολουθεί μια εξήγηση βήμα προς βήμα για το πώς λειτουργεί αυτή η επικοινωνία με λέξεις:

1. Εκκίνηση της εφαρμογής από τον πελάτη:

Η εφαρμογή πελάτη ξεκινά την επικοινωνία δημιουργώντας μια σύνδεση με τον διακομιστή Rocket.Chat. Αυτό μπορεί να γίνει χρησιμοποιώντας μια σύνδεση WebSocket ή HTTP, σύμφωνα με τις ανάγκες και τις δυνατότητες των πελατών.

2. Ταυτοποίηση:

Προτού μπορέσει να πραγματοποιηθεί οποιαδήποτε επικοινωνία, ο πελάτης πρέπει να πιστοποιηθεί με τον διακομιστή. Αυτό συνήθως περιλαμβάνει την αποστολή διαπιστευτηρίων, όπως όνομα χρήστη και κωδικό πρόσβασης, διακριτικό API ή χρήση ροής OAuth. Μετά τον έλεγχο ταυτότητας, ο διακομιστής παρέχει μια περίοδο λειτουργίας ή διακριτικό που χρησιμοποιεί ο πελάτης για επόμενα αιτήματα.

3. Αίτημα για τα δεδομένα του χρήστη:

Μετά τον έλεγχο ταυτότητας, ο πελάτης μπορεί να ζητήσει πληροφορίες σχετικά με τον χρήστη, όπως στοιχεία προφίλ, κανάλια στα οποία ανήκουν ή τις επαφές του. Στέλνει αιτήματα (GET) σε συγκεκριμένα τελικά σημεία API, υποδεικνύοντας τον τύπο των πληροφοριών που χρειάζεται.

4. Δημιουργία ή σύνδεση σε κανάλι:

Οι πελάτες μπορούν να δημιουργήσουν νέα κανάλια συνομιλίας ή να συμμετάσχουν σε υπάρχοντα. Για να δημιουργήσετε ένα κανάλι, ο πελάτης στέλνει ένα αίτημα (POST) στο κατάλληλο τελικό σημείο API με το όνομα και τις ρυθμίσεις του καναλιού. Για να συμμετάσχει σε ένα κανάλι, ο πελάτης στέλνει ένα αίτημα που υποδεικνύει το επιθυμητό αναγνωριστικό καναλιού.

5. Αποστολή μηνυμάτων:

Η αποστολή μηνυμάτων αποτελεί βασικό μέρος της επικοινωνίας πελάτη-διακομιστή. Ο πελάτης διατυπώνει ένα μήνυμα, συμπεριλαμβανομένου του κειμένου και τυχόν συνημμένων, και το στέλνει στον διακομιστή Rocket.Chat μέσω αιτήματος (POST) στο τελικό σημείο API του καθορισμένου καναλιού.

6. Λήψη μηνυμάτων:

Ο διακομιστής παρακολουθεί συνεχώς για εισερχόμενα μηνύματα στα κανάλια στα οποία είναι συνδρομητής ο πελάτης. Όταν φθάνει ένα νέο μήνυμα, ο διακομιστής ωθεί τα δεδομένα του μηνύματος στον πελάτη χρησιμοποιώντας WebSocket.

7. Επεξεργασία και διαγραφή μηνυμάτων:

Οι πελάτες έχουν τη δυνατότητα να επεξεργάζονται ή να διαγράφουν τα δικά τους μηνύματα. Για να γίνει αυτό, ο πελάτης στέλνει αιτήματα (PUT ή DELETE), αντίστοιχα, στα κατάλληλα τελικά σημεία του API μηνυμάτων.

8. Ενημερώσεις σε πραγματικό χρόνο:

Το Rocket.Chat παρέχει δυνατότητες σε πραγματικό χρόνο μέσω συνδέσεων WebSocket. Οι πελάτες μπορούν να εγγραφούν σε συγκεκριμένα συμβάντα, όπως νέα μηνύματα ή αλλαγές παρουσίας χρηστών, για να λαμβάνουν ενημερώσεις σε πραγματικό χρόνο χωρίς να απαιτείται συχνή ψηφοφορία.

9. Στοιχεία διαθεσιμότητας:

Οι πελάτες μπορούν να ζητήσουν και να λάβουν πληροφορίες διαθεσιμότητας για άλλους χρήστες (π.χ. εντός σύνδεσης, εκτός σύνδεσης) για να καθορίσουν τη διαθεσιμότητα των επαφών στο δίκτυό τους.

10. Ανέβασμα και λήψεις αρχείων:

Οι πελάτες μπορούν να ανεβάζουν αρχεία και συνημμένα στο Rocket.Chat, τα οποία είναι αποθηκευμένα στον διακομιστή. Οι χρήστες μπορούν να κάνουν λήψη αυτών των αρχείων στέλνοντας κατάλληλα αιτήματα στον διακομιστή.

11. Χειρισμός σφαλμάτων:

Σε όλη τη διαδικασία επικοινωνίας, τόσο ο πελάτης όσο και ο διακομιστής θα πρέπει να χειρίζονται τα σφάλματα. Οι απαντήσεις σφαλμάτων από τον διακομιστή πρέπει να κοινοποιούνται με σαφήνεια στον πελάτη, επιτρέποντάς του να ανταποκρίνεται κατάλληλα.

12. Αποσύνδεση:

Όταν ένας χρήστης αποσυνδέεται από την εφαρμογή πελάτη, ο πελάτης στέλνει ένα αίτημα στον διακομιστή για να τερματίσει τη συνεδρία.

13. Κλείσιμο της σύνδεσης:

Όταν η εφαρμογή-πελάτης έχει ολοκληρωθεί η επικοινωνία με τον διακομιστή ή τερματίζεται, θα πρέπει να κλείσει σωστά τη σύνδεση για να απελευθερώσει πόρους.

Συνολικά, η μοντελοποίηση της επικοινωνίας πελάτη και διακομιστή στο Rocket.Chat περιλαμβάνει έναν συνδυασμό αιτημάτων, επικοινωνίας και ενημερώσεων σε πραγματικό χρόνο για να μπορούν οι χρήστες να ανταλλάσσουν μηνύματα, να συνεργάζονται και να αλληλεπιδρούν απρόσκοπτα στο Rocket.Chat.

(Rocket.Chat, n.d.), (Rocket.Chat, https://github.com/RocketChat/Rocket.Chat, n.d.).

4.2 ΚΩΔΙΚΑΣ ΓΙΑ ΤΟΝ ΤΡΟΠΟ ΛΕΙΤΟΥΡΓΙΑΣ ΤΗΣ ΕΦΑΡΜΟΓΗΣ ΣΤΗΝ ΑΠΟΣΤΟΛΗ ΜΗΝΥΜΑΤΟΣ ΑΠΟ ΤΟ Α ΣΤΟΝ Β

Η λειτουργικότητα της επικοινωνίας Rocket.Chat μεταξύ του Χρήστη Α και του Χρήστη Β:

```
-- Set definitions
Client ::= {ClientA, ClientB}
Message ::= TEXT
Channel ::= CHATROOM

-- State Schema
ChatState ::=
  clients: Client --> State
  messages: Channel --> MESSAGESEQ

-- User States
State ::= offline | online | typing | away

-- Message Sequence
MessageSeq ::= seq × MESSAGE
```

```
-- Initial State
StateInit ::=
  clients = {ClientA ↦ offline, ClientB ↦ offline},
  messages = {}

-- Operations
-- Joining a Channel
JoinChannel ==
  ∀ client: Client @
    (clients' = clients ++ {client ↦ online}) ∧
    (State' = online)

-- Sending a Message
SendMessage ==
  ∀ client: Client; text: TEXT @
    (clients' = clients ++ {client ↦ typing}) ∧
    (∃ seq: seq; content: MESSAGE @
      (messages' = messages ++ {Channel ↦ (seq, (client, text))}))

-- Receiving a Message
ReceiveMessage ==
  ∀ client: Client; message: MESSAGE @
    (clients' = clients ++ {client ↦ online}) ∧
    (∃ seq: seq @
      (messages' = messages ++ {Channel ↦ (seq, message)}))

-- Transition Rules

-- Initial state
Initial ==
  StateInit
```

```
-- Client A joins the channel
JoiningChannelA ==
  (∀ client: Client | client = ClientA) ∧
  JoinChannel
-- Client B joins the channel
JoiningChannelB ==
  (∀ client: Client | client = ClientB) ∧
  JoinChannel

-- Client A sends a message
SendingMessageA ==
  (∀ client: Client | client = ClientA) ∧
  SendMessage("Hello, Client B")

-- Client B receives a message
ReceivingMessageB ==
  (∀ client: Client | client = ClientB) ∧
  ReceiveMessage(Channel ↦ (N+1, "Hello, Client B"))
```

Ορίζουμε σύνολα `Client`, `Message`, and `Channel` για να αντιπροσωπεύουν πελάτες, μηνύματα και κανάλια.

1. Το σχήμα κατάστασης `ChatState` περιλαμβάνει την κατάσταση των πελατών (`State`) και το ιστορικό μηνυμάτων (`MESSAGESEQ`) για κάθε κανάλι (`Channel`).
2. Οι καταστάσεις χρήστη αντιπροσωπεύονται από την απαρίθμηση «Κατάσταση».
3. Οι ακολουθίες μηνυμάτων αντιπροσωπεύονται από το "MessageSeq", όπου κάθε μήνυμα έχει έναν αριθμό σειράς και ένα περιεχόμενο.
4. Το σχήμα «StateInit» καθορίζει την αρχική κατάσταση του συστήματος συνομιλίας.
5. Οι λειτουργίες ('JoinChannel', 'SendMessage', 'ReceiveMessage') περιγράφουν τις ενέργειες που μπορούν να εκτελέσουν οι πελάτες.
6. Οι κανόνες μετάβασης ('Initial', 'JoiningChannelA', 'JoiningChannelB', 'SendingMessageA', 'ReceivingMessageB') δείχνουν πώς το σύστημα μεταβαίνει από τη μια κατάσταση στην άλλη καθώς οι πελάτες αλληλεπιδρούν με το σύστημα συνομιλίας.

4.3 ΕΠΕΞΗΓΗΣΗ ΤΟΥ ΚΩΔΙΚΑ ΓΙΑ ΤΗΝ ΠΕΡΙΓΡΑΦΗ ΛΕΙΤΟΥΡΓΙΑΣ

-- Set definitions

Client ::= {ClientA, ClientB}

Message ::= TEXT

Channel ::= CHATROOM

- Ορίζονται αρχικά τα σύνολα που αντιπροσωπεύουν διαφορετικές οντότητες στο σύστημα συνομιλίας. Αρχικά Το "Client" αντιπροσωπεύει το σύνολο των χρηστών, στο οποίο περιλαμβάνονται ο "ClientA" και ο "ClientB". Το " Message " αντιπροσωπεύει το σύνολο των μηνυμάτων και το " Channel " αντιπροσωπεύει το σύνολο των αιθουσών συνομιλίας.

-- State Schema

ChatState ::=

clients: Client --> State

messages: Channel --> MESSAGESEQ

Ορίζεται ένα σχήμα που ονομάζεται "ChatState" το οποίο περιγράφει τη δομή της κατάστασης του συστήματος συνομιλίας.

Αυτό περιλαμβάνει δύο δεδομένα:

- `clients`: Αντικατοπτρίζει την κατάσταση του κάθε πελάτη, η οποία μπορεί να είναι μία από τις {offline, online, typing, away}.

- `messages`:Συσχετίζει κάθε κανάλι με μία ακολουθία (`MESSAGESEQ`).

-- User States

State ::= offline | online | typing | away

Αντικατοπτρίζει την κατάσταση του κάθε χρήστη στην οποία έχει την δυνατότητα να τοποθετηθεί. Οι χρήστες μπορούν να βρίσκονται σε μία από τις τέσσερις καταστάσεις: "εκτός σύνδεσης", "συνδεδεμένος", "πληκτρολόγηση" , ή "μακριά".

-- Message Sequence

MessageSeq ::= seq × MESSAGE

- Ορίζεται ένα σχήμα που ονομάζεται `MessageSeq`, που αντιπροσωπεύει μια ακολουθία μηνυμάτων. Αποτελείται από ζεύγη ενός αριθμού σειράς («seq») και ενός μηνύματος (`MESSAGE`).

-- Initial State

StateInit ::=

clients = {ClientA ↦ offline, ClientB ↦ offline},

messages = {}

Αυτό το σχήμα, «StateInit», καθορίζει την αρχική κατάσταση του συστήματος συνομιλίας. Δηλώνει ότι τόσο το «ClientA» και το «ClientB» βρίσκονται αρχικά σε κατάσταση «εκτός σύνδεσης» και δεν υπάρχουν μηνύματα στα «messages» του συστήματος.

-- Operations

-- Joining a Channel

op JoinChannel ==

∀ client: Client @

(clients' = clients ++ {client ↦ online}) ∧

(State' = online)

- Αυτή η λειτουργία, «JoinChannel», αντιπροσωπεύει έναν χρήστη που συνδέεται σε ένα κανάλι.

- Ενημερώνει τη κατάσταση των χρηστών ορίζοντας την κατάσταση του κάθε «χρήστη» σε «συνδεδεμένος».

- Ορίζει επίσης την " State' " (κατάσταση του καναλιού) σε "online", υποδεικνύοντας τη μετάβαση της κατάστασης.

```
-- Sending a Message
op SendMessage ==
  ∀ client: Client; text: TEXT @
    (clients' = clients ++ {client ↦ typing}) ∧
    (∃ seq: seq; content: MESSAGE @
      (messages' = messages ++ {Channel ↦ (seq, (client, text))}))
```

- Αυτή η λειτουργία, «Αποστολή μηνύματος», αντιπροσωπεύει έναν χρήστη που στέλνει ένα μήνυμα. Χρειάζεται δύο παραμέτρους, «χρήστης» (ο αποστολέας) και «κείμενο» (το περιεχόμενο του μηνύματος). Μέσα στη λειτουργία:

  - Ενημερώνει την κατάσταση του αποστολέα σε «πληκτρολόγηση».

  - Εισάγει τις μεταβλητές «seq» και «content» για να καθορίσει ένα νέο μήνυμα που θα προστεθεί στην αντιστοίχιση «μηνυμάτων» στο κατάλληλο κανάλι.

```
-- Receiving a Message
op ReceiveMessage ==
  ∀ client: Client; message: MESSAGE @
    (clients' = clients ++ {client ↦ online}) ∧
    (∃ seq: seq @
      (messages' = messages ++ {Channel ↦ (seq, message)}))
```

- Αυτή η λειτουργία, «Λήψη μηνύματος», αντιπροσωπεύει έναν χρήστη που λαμβάνει ένα μήνυμα. Χρειάζεται δύο παραμέτρους, «client» (ο παραλήπτης) και «message» (το ληφθέν μήνυμα). Μέσα στη λειτουργία:

  - Ενημερώνει την κατάσταση του δέκτη σε «online».

  - Εισάγει την μεταβλητή `seq` για να καθορίσει ένα νέο ληφθέν μήνυμα που θα προστεθεί στην αντιστοίχιση «μηνυμάτων» στο κατάλληλο κανάλι.

-- Transition Rules

-- Initial state

[Initial]

init

- Αυτός ο κανόνας μετάβασης, «[Initial]», ορίζει την αρχική κατάσταση του συστήματος συνομιλίας χρησιμοποιώντας το σχήμα «init».

-- Client A joins the channel

[JoiningChannelA]

(∀ client: Client | client = ClientA) ∧

JoinChannel

- Αυτός ο κανόνας μετάβασης, «[JoiningChannelA]», μοντελοποιεί το συμβάν όπου ο πελάτης Α συμμετέχει σε ένα κανάλι. Αυτό το κάνει διασφαλίζοντας ότι μόνο το «ClientA» συμμετέχει στο κανάλι χρησιμοποιώντας το «JoinChannel».

-- Client B joins the channel

[JoiningChannelB]

(∀ client: Client | client = ClientB) ∧

JoinChannel

- Αυτός ο κανόνας μετάβασης, «[JoiningChannelB]», μοντελοποιεί το συμβάν όπου ο πελάτης Β συμμετέχει σε ένα κανάλι. Αυτό το κάνει διασφαλίζοντας ότι μόνο το «ClientB» συμμετέχει στο κανάλι χρησιμοποιώντας το «JoinChannel».

-- Client A sends a message

SendingMessageA ==

(∀ client: Client | client = ClientA) ∧

SendMessage("Hello, Client B")

Παράδειγμα της λειτουργίας «Αποστολή μηνύματος», όπου ο Χρήστης Α αποστέλλει ένα μήνυμα στον Χρήστη Β.

```
-- Client B receives a message
ReceivingMessageB ==
  (∀ client: Client | client = ClientB) ∧
  ReceiveMessage(Channel ↦ (N+1, "Hello, Client B"))
```

Παράδειγμα της λειτουργίας «Λήψη μηνύματος», όπου ο Χρήστης Β λαμβάνει ένα μήνυμα από τον Χρήστη Α.

Αυτοί οι κανόνες μετάβασης περιγράφουν πώς αλλάζει η κατάσταση του συστήματος συνομιλίας καθώς οι πελάτες αλληλοεπιδρούν μαζί του, με βάση τις καθορισμένες λειτουργίες. Κάθε κανόνας καθορίζει μια συνθήκη υπό την οποία επιτρέπεται να πραγματοποιηθεί η λειτουργία και οι επακόλουθες αλλαγές στην κατάσταση του συστήματος.

ΣΥΜΠΕΡΑΣΜΑΤΑ

Ένα σημαντικό πλεονέκτημα μιας επίσημης σημειογραφίας, στην περίπτωση μας η Z, είναι ότι είναι ακριβής και σαφής. Αυτή η εργασία χωρίζεται σε δύο κύρια μέρη. Το πρώτο ασχολείται με τη φύση των επίσημων προδιαγραφών και γιατί πρέπει να χρησιμοποιείται. Επιπλέον, περιέχει μια σύντομη εισαγωγή στη Z σημειογραφία και πώς χρησιμοποιείται. Το δεύτερο μισό της εργασίας ασχολείται με την επεξήγηση της εφαρμογής, Rocket.chat, χρησιμοποιώντας τη Z για το σχεδιασμό και την τεκμηρίωση των υπηρεσιών δικτύου για τον τρόπο λειτουργίας του συστήματος. Τέλος μπορούν να εξαχθούν ορισμένα συμπεράσματα σχετικά με τα πλεονεκτήματα και τα μειονεκτήματα της χρήσης μιας τυπικής προδιαγραφής.

Ο παραπάνω τρόπος ανάλυσης του προβλήματος, με τη χρήση της Z σημειογραφίας και των πραγματικών συμβόλων, έχει ορισμένα πλεονεκτήματα που μπορούν να είναι κατάλληλα σύμφωνα με τις απαιτήσεις και τα χαρακτηριστικά του έργου:

1. **Συνεπής και Κατανοητός Κώδικας:**

- Η χρήση πραγματικών συμβόλων μπορεί να καταστήσει τον κώδικα πιο συνεπή και κατανοητό για τους αναγνώστες που δεν είναι εξοικειωμένοι με την αφαιρετική σύμβαση της Z σημειογραφίας.

2. **Ευανάγνωστος Κώδικας:**

- Η χρήση πραγματικών συμβόλων ενδέχεται να καταστήσει τον κώδικα πιο ευανάγνωστο, ειδικά για όσους δεν έχουν εξειδικευμένη γνώση στην Z σημειογραφία.

3. **Εύκολη Εφαρμογή:**

- Η Z σημειογραφία με πραγματικά σύμβολα μπορεί να είναι πιο εύκολα κατανοητή για ανθρώπους που είναι εξοικειωμένοι με γλώσσες προγραμματισμού και μαθηματικές έννοιες.

4. **Ευέλικτη Προσαρμογή:**

- Η χρήση πραγματικών συμβόλων μπορεί να προσφέρει μεγαλύτερη ευελιξία και προσαρμογή, ειδικά εάν υπάρχει ανάγκη για εξατομικευμένη αναπαράσταση του κώδικα.

Ο παραπάνω τρόπος ανάλυσης του προβλήματος, με τη χρήση της Z σημειογραφίας και των πραγματικών συμβόλων, έχει και ορισμένα μειονεκτήματα όπως:
Ένας περιορισμός των τυπικών μεθόδων είναι ότι μπορούν να χρησιμοποιηθούν μόνο για να αποδειχθεί η ορθότητα ενός συστήματος σε σχέση με μια προδιαγραφή. Επομένως, ακριβώς επειδή ένα πρόγραμμα έχει αποδειχθεί μαθηματικά ότι συμμορφώνεται σύμφωνα με τις προδιαγραφές, δεν υπάρχει καμία εγγύηση ότι η προδιαγραφή από μόνη της είναι σωστή και χωρίς σφάλματα.

ΒΙΒΛΙΟΓΡΑΦΙΑ

Keith D. Cooper, L. T. (2012). *Engineering a Compiler.* Burlington, USA: Elsevier.

Rocket.Chat. (n.d.). *https://github.com/RocketChat/Rocket.Chat*. Retrieved from https://github.com/RocketChat/Rocket.Chat: https://github.com/RocketChat/Rocket.Chat

Rocket.Chat. (n.d.). *Rocket.Chat*. Retrieved from rocket.chat/: https://docs.rocket.chat/

Rouse, M. (2023, July 26). *techopedia*. Retrieved from techopedia: https://www.techopedia.com/definition/25705/communication-protocol

Šimoňák, S. (2012, November 04). Verification of Communication Protocols Based. Košice, Košice, Slovakia.

Sommerville, I. (2011). *Software Engineering, 9th Edition.* Boston, Massachusetts: Pearson Education.

Spivey, J. M. (1998). *The Z Notation:.* Oxford, England: Prentice Hall International (UK) Ltd.

Toyn, I. (1997). *Z Notation.* May: 02.

Wing, J. M. (1990). *A Specifier's Introduction To Formal Method.* Pittsburgh: Institute of Electrical and Electronics Engineers.