ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΙΓΑΙΟΥ

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΑΚΩΝ ΚΑΙ ΕΠΙΚΟΙΝΩΝΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

ΠΡΟΠΤΥΧΙΑΚΟ ΠΡΟΓΡΑΜΜΑ ΣΠΟΥΔΩΝ
ΜΑΘΗΜΑ: ΤΕΧΝΟΛΟΓΙΑ ΛΟΓΙΣΜΙΚΟΥ

ΤΥΠΙΚΗ ΠΡΟΔΙΑΓΡΑΦΗ

Διδάσκων:

Γ. Χαραλαμπίδης, Επ. Καθηγητής

Στόχοι

- Εξήγηση του λόγου για τον οποίο οι τεχνικές τυπικής προδιαγραφής βοηθούν στον εντοπισμό προβλημάτων στις απαιτήσεις συστήματος
- Περιγραφής της χρήσης αλγεβρικών τεχνικών για τον ορισμό προδιαγραφών διασύνδεσης
- Περιγραφή του τρόπου χρήσης των τυπικών τεχνικών βάσει μοντέλων για την προδιαγραφή της συμπεριφοράς

Περιεχόμενα

- Τυπική προδιαγραφή στη διαδικασία παραγωγής λογισμικού
- Προδιαγραφή διασύνδεσης υποσυστημάτων
- Προδιαγραφή συμπεριφοράς

Τυπικές μέθοδοι

- Η τυπική προδιαγραφή είναι μέρος μιας πιο γενικής συλλογής τεχνικών γνωστών ως "τυπικών μεθόδων".
- Όλες αυτές οι τεχνικές βασίζονται στη μαθηματική αναπαράσταση και ανάλυση του λογισμικού.
- Στις τυπικές μεθόδους συγκαταλέγονται οι εξής:
  - Τυπική προδιαγραφή
  - Ανάλυση και απόδειξη της προδιαγραφής
  - Μετασχηματιστική ανάπτυξη
  - Επαλήθευση του προγράμματος

Αποδοχή των τυπικών μεθόδων

- Αντίθετα με τις προβλέψεις, οι τυπικές μέθοδοι δεν έχουν επικρατήσει ως τεχνικές ανάπτυξης λογισμικού.
  - Στη βελτίωση της ποιότητας των συστημάτων είχαν επιτυχία και άλλες μέθοδοι τεχνολογίας λογισμικού. Έτσι η αναγκαιότητα των τυπικών μεθόδων περιορίστηκε.
  - Οι αλλαγές της αγοράς έχουν καταστήσει ως κομβικό παράγοντα το χρόνο διάθεσης του λογισμικού στην αγορά αντί της παραγωγής ενός προϊόντος με μικρό πλήθος σφαλμάτων. Οι τυπικές μέθοδοι δεν μειώνουν το χρόνο διάθεσης του λογισμικού στην αγορά.
  - Η εμβέλεια των τυπικών μεθόδων είναι περιορισμένη. Δεν είναι κατάλληλες για την προδιαγραφή διασυνδέσεων χρήστη και της αλληλεπίδρασης του λογισμικού με το χρήστη.
  - Οι τυπικές μέθοδοι δύσκολα αναπροσαρμόζονται για μεγαλύτερα συστήματα.

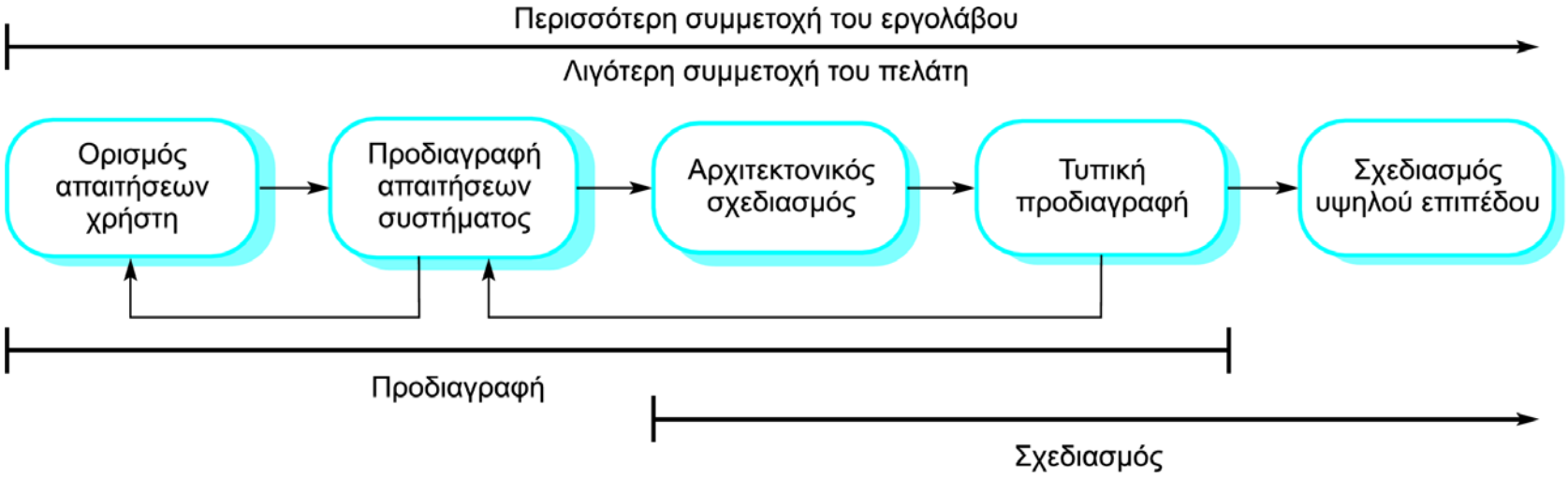
Χρήση των τυπικών μεθόδων

- Τα βασικά οφέλη των τυπικών μεθόδων αφορούν τη μείωση των ελαττωμάτων που υπάρχουν στα συστήματα.
- Συνεπώς, ο βασικός τομέας εφαρμογής τους είναι η τεχνολογία κρίσιμων συστημάτων. Σε αυτόν τον τομέα έχουν υπάρξει πολλά επιτυχημένα έργα στα οποία έχουν χρησιμοποιηθεί με επιτυχία τυπικές μέθοδοι.
- Σε αυτόν τον τομέα, η χρήση τυπικών μεθόδων θα είναι πιθανότατα αποτελεσματική από άποψη κόστους επειδή πρέπει να αποφευχθεί το υψηλό κόστος των αστοχιών των συστημάτων αυτών.

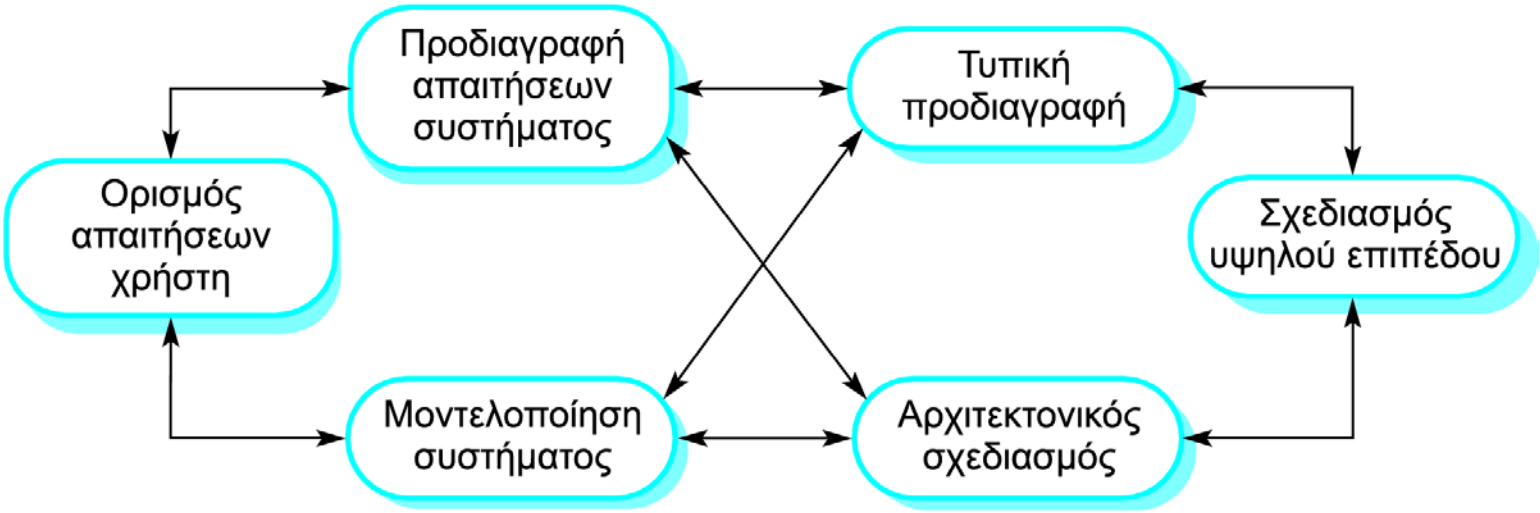
Προδιαγραφή στη διαδικασία παραγωγής λογισμικού

- Η προδιαγραφή και ο σχεδιασμός είναι άρρηκτα αναμεμιγμένες έννοιες.
- Ο αρχιτεκτονικός σχεδιασμός είναι θεμελιώδης στη δόμηση μιας προδιαγραφής, αλλά και σε ολόκληρη τη διαδικασία εξαγωγής προδιαγραφών.
- Οι τυπικές προδιαγραφές εκφράζονται με μαθηματική σημειογραφία η οποία έχει συγκεκριμένο λεξιλόγιο, συντακτική δομή και σημασιολογία.

Προδιαγραφή και σχεδιασμός



Προδιαγραφή στη διαδικασία παραγωγής λογισμικού



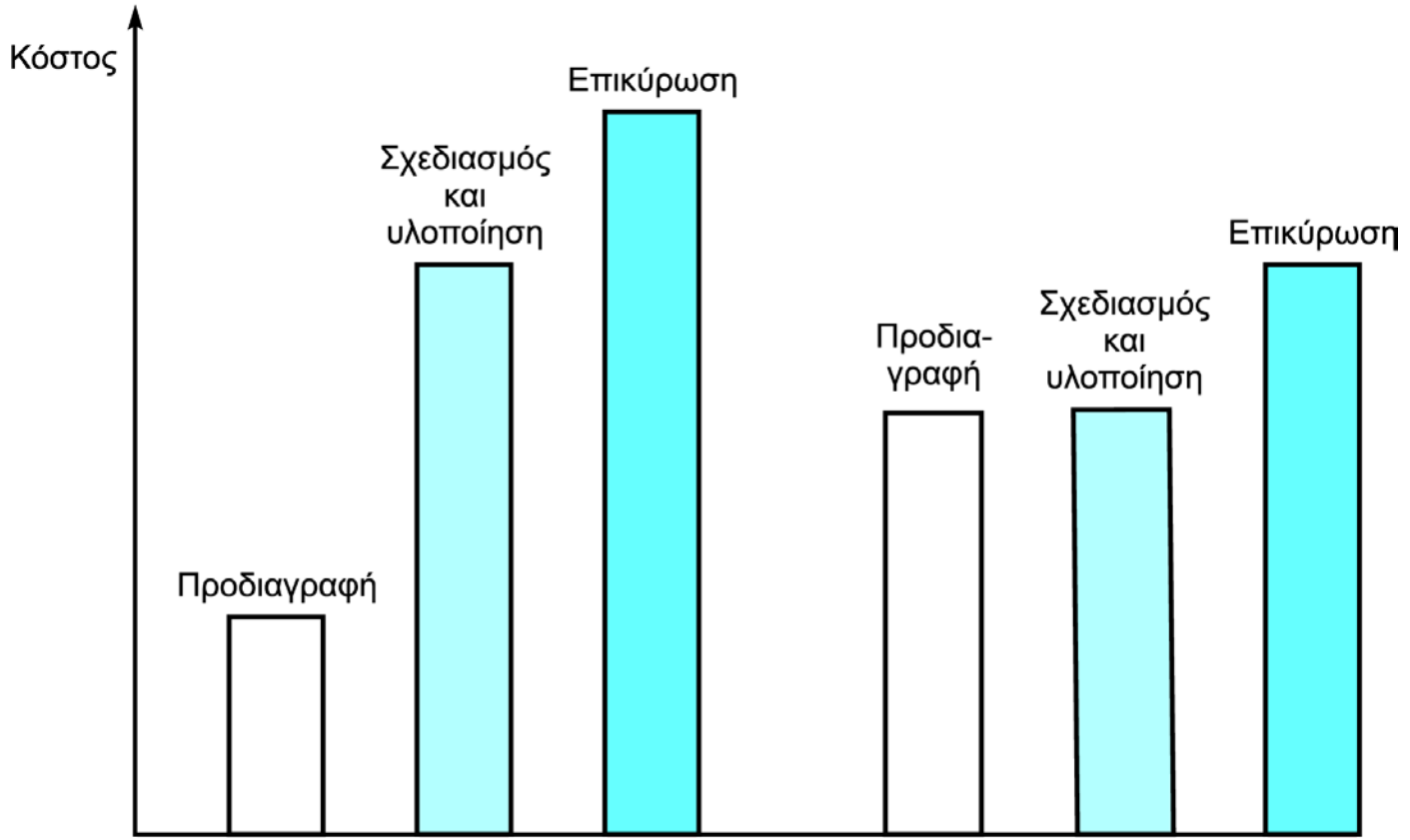
Χρήση της τυπικής προδιαγραφής

- Η τυπική προδιαγραφή προϋποθέτει περισσότερο κόπο στις αρχικές φάσεις ανάπτυξης λογισμικού.
- Αυτό μειώνει τα σφάλματα στις απαιτήσεις αφού υποχρεώνει να γίνεται λεπτομερής ανάλυσή τους.
- Έτσι μπορούν να ανακαλύπτονται και να τακτοποιούνται ασυνέπειες και ημιτελή στοιχεία.
- Συνεπώς γίνεται εξοικονόμηση σε πολλούς τομείς καθώς μειώνεται ο όγκος της επανεπεξεργασίας εξαιτίας προβλημάτων στις απαιτήσεις.

Κατανομή κόστους

- Η χρήση της τυπικής προδιαγραφής σημαίνει ότι αλλάζει η κατανομή του κόστους ενός έργου
  - Υπάρχουν μεγαλύτερα προκαταβολικά έξοδα αφού αφιερώνεται περισσότερος χρόνος και εργασία στην ανάπτυξη της προδιαγραφής.
  - Αλλά τα έξοδα υλοποίησης και επικύρωσης θα μειωθούν αφού η διαδικασία της προδιαγραφής ελαττώνει τα σφάλματα και τις διφορούμενες ερμηνείες των απαιτήσεων.

Κόστος ανάπτυξης λογισμικού με τυπική προδιαγραφή



Τεχνικές προδιαγραφής

- Αλγεβρική προδιαγραφή
  - Το σύστημα περιγράφεται με βάση τις λειτουργίες του και τις μεταξύ τους σχέσεις.
- Προδιαγραφή βάσει μοντέλου
  - Το σύστημα περιγράφεται με βάση ένα μοντέλο κατάστασης του συστήματος το οποίο δημιουργείται με τη χρήση μαθηματικών δομών όπως σύνολα και ακολουθίες. Οι λειτουργίες του συστήματος ορίζονται με βάση το πώς τροποποιούν την κατάσταση του συστήματος.

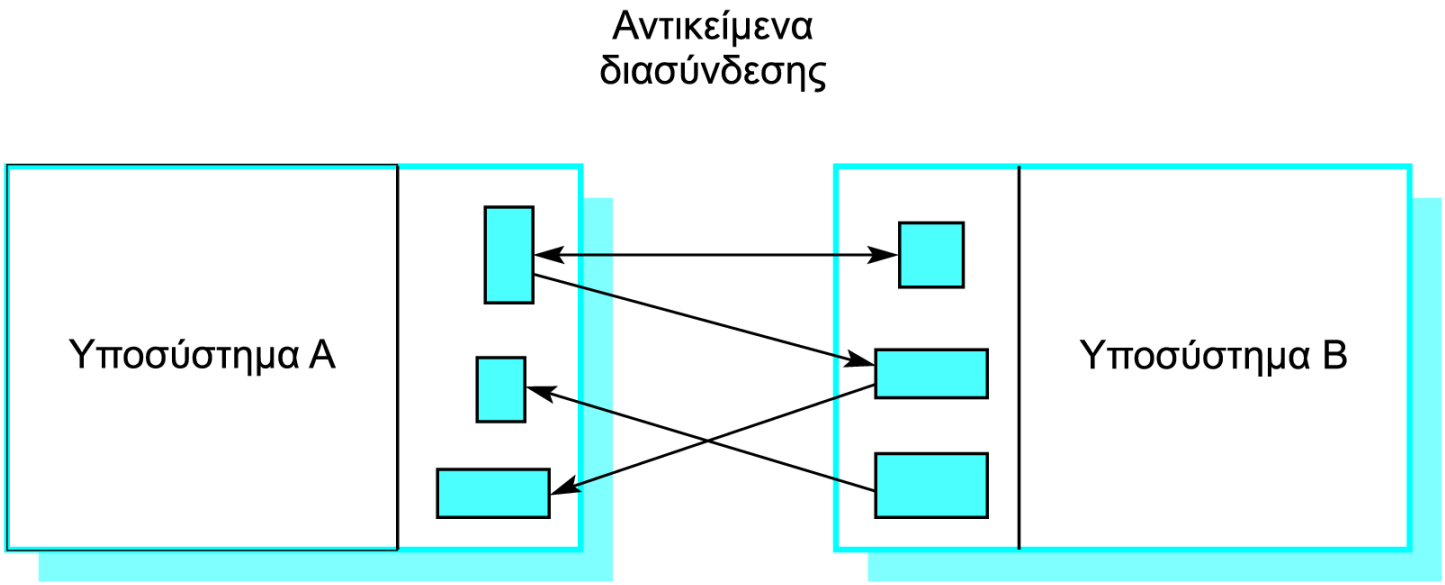
Γλώσσες τυπικής προδιαγραφής

	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)

Προδιαγραφές διασύνδεσης

- Τα μεγάλα συστήματα συνήθως αποδομούνται σε υποσυστήματα τα οποία διαθέτουν πλήρως ορισμένες διασυνδέσεις για τη μεταξύ τους επικοινωνία.
- Η προδιαγραφή των διασυνδέσεων των υποσυστημάτων επιτρέπει την ανεξάρτητη ανάπτυξη κάθε υποσυστήματος.
- Οι διασυνδέσεις ορίζονται με τη μορφή αφηρημένων τύπων δεδομένων ή κλάσεων αντικειμένων.
- Η αλγεβρική προσέγγιση της τυπικής προδιαγραφής ταιριάζει απόλυτα στην προδιαγραφή διασυνδέσεων αφού επικεντρώνεται στις καθορισμένες λειτουργίες ενός αντικειμένου.

Διασυνδέσεις υποσυστημάτων



Η δομή μιας αλγεβρικής προδιαγραφής

<ΟΝΟΜΑ ΠΡΟΔΙΑΓΡΑΦΗΣ>

sort <όνομα>
imports <ΛΙΣΤΑ ΟΝΟΜΑΤΩΝ ΠΡΟΔΙΑΓΡΑΦΩΝ>

Άτυπη περιγραφή του είδους και των λειτουργιών του

Υπογραφές λειτουργιών, που ορίζουν τα ονόματα και τους τύπους των παραμέτρων για τις λειτουργίες του οριζόμενου είδους

Αξιώματα που ορίζουν τις λειτουργίες του είδους

Στοιχεία μιας προδιαγραφής

- Εισαγωγή
  - Δηλώνει το είδος (sort — το όνομα του τύπου) της οντότητας που προδιαγράφεται και δηλώνει άλλες προδιαγραφές που χρησιμοποιούνται.
- Περιγραφή
  - Περιγράφονται άτυπα οι λειτουργίες.
- Υπογραφή
  - Καθορίζει τη σύνταξη των λειτουργιών της διασύνδεσης και των παραμέτρων τους.
- Αξιώματα
  - Ορίζεται η σημασιολογία των λειτουργιών μέσω ενός συνόλου αξιωμάτων που χαρακτηρίζουν τη συμπεριφορά.

Συστηματική αλγεβρική προδιαγραφή

- Οι αλγεβρικές προδιαγραφές ενός συστήματος μπορούν να αναπτυχθούν με συστηματικό τρόπο
  - Δόμηση της προδιαγραφής
  - Ονομασία προδιαγραφής
  - Επιλογή λειτουργιών
  - Άτυπη προδιαγραφή λειτουργιών
  - Ορισμός σύνταξης
  - Ορισμός αξιωμάτων

Λειτουργίες προδιαγραφής

- Λειτουργίες κατασκευής. Δημιουργούν οντότητες του τύπου που ορίζεται στην προδιαγραφή.
- Λειτουργίες επιθεώρησης. Υπολογίζουν οντότητες του τύπου που ορίζεται στην προδιαγραφή.
- Για να καθορίσουμε τη συμπεριφορά, πρέπει να ορίσουμε τις λειτουργίες επιθεώρησης κάθε λειτουργίας κατασκευής.

Λειτουργίας μιας λίστας ADT

- Λειτουργίες κατασκευής για τη δόμηση και ταξινόμηση της λίστας
  - Create, Cons και Tail.
- Λειτουργίες επιθεώρησης οι οποίες δέχονται ως παράμετρο την ταξινομημένη λίστα και επιστρέφουν κάποια άλλη ταξινομημένη τιμή
  - Head και Length
- Η μέθοδος Tail μπορεί να οριστεί από τις πιο απλές μεθόδους κατασκευής Create και Cons. Δεν απαιτείται ο ορισμός των Head και Length με τη βοήθεια της Tail.

Προδιαγραφή λίστας

LIST (Elem)

sort List
imports INTEGER

Ορίζει μια λίστα (List), όπου στοιχεία (Elem) προστίθενται στο τέλος και αφαιρούνται από την αρχή. Οι λειτουργίες είναι η Create (δημιουργία), που δημιουργεί μια κενή λίστα, η Cons (κατασκευή), που δημιουργεί μια νέα λίστα με ένα μέλος, η Length (μήκος), που υπολογίζει το μέγεθος της λίστας, η Head (κεφαλή), που επιστρέφει το πρώτο στοιχείο της λίστας, και η Tail (ουρά), που δημιουργεί μια λίστα αφαιρώντας την κεφαλή από την τρέχουσα λίστα. Το Undefined (αόριστο) αντιπροσωπεύει μια μη ορισμένη τιμή τύπου Elem.

Create → List
Cons (List, Elem) → List
Head (List) → Elem
Length (List) → Integer
Tail (List) → List

Head (Create) = Undefined **exception** (empty list)
Head (Cons (L, v)) = **if** L = Create **then** v **else** Head (L)
Length (Create) = 0
Length (Cons (L, v)) = Length (L) + 1
Tail (Create) = Create
Tail (Cons (L, v)) = **if** L = Create **then** Create **else** Cons (Tail (L), v)

Αναδρομή στις προδιαγραφές

- Οι λειτουργίες ορίζονται πολλές φορές με αναδρομικό τρόπο.
- Tail (Cons (L, v)) = if L = Create then Create
  else Cons (Tail (L), v).
  - Cons ([5, 7], 9) = [5, 7, 9]
  - Tail ([5, 7, 9]) = Tail (Cons ([5, 7], 9)) =
  - Cons (Tail ([5, 7]), 9) = Cons (Tail (Cons ([5], 7)), 9) =
  - Cons (Cons (Tail ([5]), 7), 9) =
  - Cons (Cons (Tail (Cons ([], 5)), 7), 9) =
  - Cons (Cons ([Create], 7), 9) = Cons ([7], 9) = [7, 9]

Προδιαγραφή διασυνδέσεων σε κρίσιμα συστήματα

- Θα εξετάσουμε ένα σύστημα ελέγχου εναέριας κυκλοφορίας, με βάση το οποίο τα αεροσκάφη κινούνται διαμέσου ελεγχόμενων τομέων του εναέριου χώρου.
- Κάθε τομέας μπορεί να περιλαμβάνει έναν αριθμό αεροσκαφών, αλλά αυτά θα πρέπει να έχουν μεταξύ τους κάποια απόσταση για λόγους ασφάλειας.
- Στο παράδειγμα προτείνεται η κατακόρυφη απόσταση των 300 μέτρων.
- Το σύστημα πρέπει να προειδοποιεί τον ελεγκτή αν κάποιο αεροσκάφος προσπαθήσει να λάβει θέση η οποία παραβιάζει αυτόν τον περιορισμό.

Ένα αντικείμενο τομέα

- Οι κρίσιμες λειτουργίες οι οποίες εφαρμόζονται στο αντικείμενο που αντιπροσωπεύει έναν ελεγχόμενο τομέα είναι
  - Enter (είσοδος). Προσθέτει ένα αεροσκάφος στον ελεγχόμενο εναέριο χώρο
  - Leave (αποχώρηση). Αφαιρεί ένα αεροσκάφος από τον ελεγχόμενο εναέριο χώρο
  - Move (μετακίνηση). Μετακινεί ένα αεροσκάφος από ένα ύψος σε κάποιο άλλο
  - Lookup (αναζήτηση). Με δεδομένο το αναγνωριστικό ενός αεροσκάφους, επιστρέφει το τρέχον ύψος του

Απλούστερες λειτουργίες

- Μερικές φορές, για την απλούστευση της προδιαγραφής είναι απαραίτητος ο ορισμός επιπλέον λειτουργιών.
- Κατόπιν όλες οι υπόλοιπες λειτουργίες μπορούν να οριστούν με τη χρήση αυτών των απλούστερων λειτουργιών.
- Απλούστερες λειτουργίες
  - Create (δημιουργία). Προκαλεί τη δημιουργία ενός κενού στιγμιοτύπου τομέα
  - Put (τοποθέτηση). Προσθέτει ένα αεροσκάφος στον τομέα χωρίς έλεγχο περιορισμών ασφάλειας
  - In-space (στο χώρο). Προσδιορίζει αν ένα δεδομένο αεροσκάφος βρίσκεται στο συγκεκριμένο τομέα
  - Occupied (κατειλημμένο). Με δεδομένο ένα ύψος, προσδιορίζει αν υπάρχει κάποιο αεροσκάφος σε απόσταση 300 μέτρων από το συγκεκριμένο ύψος.

Προδιαγραφή τομέα (1)

SECTOR

sort Sector
imports INTEGER, BOOLEAN

Enter	-προσθέτει ένα αεροσκάφος στον τομέα αν ικανοποιούνται οι συνθήκες ασφάλειας
Leave	-αφαιρεί ένα αεροσκάφος από τον τομέα
Move	-μετακινεί ένα αεροσκάφος από ένα ύψος σε ένα άλλο, αν αυτό είναι ασφαλές να γίνει
Lookup	-βρίσκει το ύψος ενός αεροσκάφους στον τομέα
Create	-δημιουργεί έναν κενό τομέα
Put	-προσθέτει ένα αεροσκάφος σε έναν τομέα χωρίς ελέγχους περιορισμών
In-space	-ελέγχει αν ένα αεροσκάφος είναι ήδη σε έναν τομέα
Occupied	-ελέγχει αν είναι διαθέσιμο ένα καθορισμένο ύψος

Enter (Sector, Call-sign, Height) → Sector
Leave (Sector, Call-sign) → Sector
Move (Sector, Call-sign, Height) → Sector
Lookup (Sector, Call-sign) → Height

Create → Sector
Put (Sector, Call-sign, Height) → Sector
In-space (Sector, Call-sign) → Boolean
Occupied (Sector, Height) → Boolean

Προδιαγραφή τομέα (2)

```
Enter (S, CS, H) =
        if      In-space (S, CS) then S exception (Αεροσκάφος ήδη στον τομέα)
        elseif  Occupied (S, H) then S exception (Διένεξη ύψους)
        else    Put (S, CS, H)

Leave (Create, CS) = Create exception (Αεροσκάφος δεν είναι στον τομέα)
Leave (Put (S, CS1, H1), CS) =
        if CS = CS1 then S else Put (Leave (S, CS), CS1, H1)

Move (S, CS, H) =
        if      S = Create then Create exception (Δεν υπάρχει αεροσκάφος στον
                                                  τομέα)
        elseif  not In-space (S, CS) then S exception (Το αεροσκάφος δεν είναι στον
                                                  τομέα)
        elseif  Occupied (S, H) then S exception (Διένεξη ύψους)
        else    Put (Leave (S, CS), CS, H)

-- NO-HEIGHT είναι μια σταθερά που υποδηλώνει ότι δεν μπορεί να επιστραφεί ένα έγκυρο
-- ύψος
Lookup (Create, CS) = NO-HEIGHT exception (Το αεροσκάφος δεν είναι στον τομέα)
Lookup (Put (S, CS1, H1), CS) =
        if CS = CS1 then H1 else Lookup (S, CS)

Occupied (Create, H) = false
Occupied (Put (S, CS1, H1), H) =
        if (H1 > H and H1 – H <= 300) or (H > H1 and H – H1<= 300) then true
        else Occupied (S, H)

In-space (Create, CS) = false
In-space (Put (S, CS1, H1), CS) =
        if CS = CS1 then true else In-space (S, CS)
```

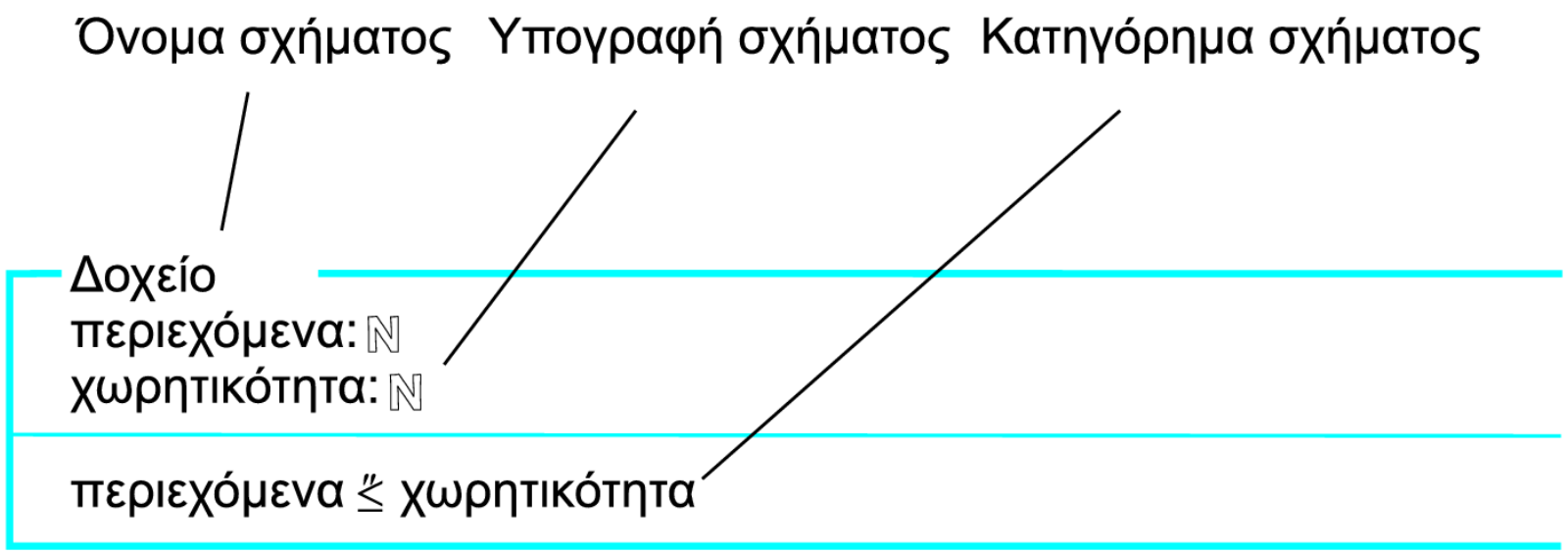
Σχόλια για την προδιαγραφή

- Χρησιμοποιήστε τις βασικές λειτουργίες Create και Put για να καθορίσετε άλλες.
- Ορίστε τις λειτουργίες Occupied και In-space με χρήση των Create και Put και χρησιμοποιήστε τις για να πραγματοποιήσετε ελέγχους των ορισμών άλλων λειτουργιών.
- Όλες οι λειτουργίες που επιφέρουν αλλαγές στον τομέα πρέπει να πληρούν το κριτήριο ασφάλειας.

Προδιαγραφή συμπεριφοράς

- Η αλγεβρική προδιαγραφή μπορεί να γίνει δύσχρηστη όταν οι λειτουργίες των αντικειμένων εξαρτώνται από την κατάσταση των αντικειμένων.
- Η προδιαγραφή βάσει μοντέλου αποκαλύπτει την κατάσταση του συστήματος και ορίζει τις λειτουργίες με βάση τις αλλαγές στην κατάσταση αυτή.
- Η σημειογραφία Z είναι μια ώριμη τεχνική προδιαγραφής βάσει μοντέλου. Συνδυάζει τυπική και άτυπη περιγραφή και χρησιμοποιεί επισημάνσεις με γραφικά στοιχεία κατά την παρουσίαση προδιαγραφών.

Η δομή ενός σχήματος της σημειογραφίας Z



Μοντελοποίηση της αντλίας ινσουλίνης

- Το σχήμα Z που αναφέρεται στην αντλία ινσουλίνης δηλώνει διάφορες μεταβλητές κατάστασης στις οποίες συμπεριλαμβάνονται:
  - Μεταβλητές εισόδου όπως οι switch? (ο διακόπτης λειτουργίας), InsulinReservoir? (η τρέχουσα ποσότητα ινσουλίνης στο δοχείο) και Reading? (η ένδειξη του αισθητήρα)
  - Μεταβλητές εξόδου όπως οι alarm! (συναγερμός του συστήματος), display1!, display2! (οι ενδείξεις πάνω στην αντλία) και dose! (η δόση ινσουλίνης που πρόκειται να χορηγηθεί).

Αναλλοίωτες συνθήκες σχήματος

- Σε κάθε σχήμα Z υπάρχει ένα αναλλοίωτο τμήμα στο οποίο ορίζονται συνθήκες που ισχύουν πάντα.
- Για το σχήμα της αντλίας ινσουλίνης ισχύει πάντα ότι
  - Η δόση πρέπει να είναι μικρότερη ή ίση με τη χωρητικότητα του δοχείου της ινσουλίνης
  - Καμία δόση δεν μπορεί να υπερβαίνει τις 4 μονάδες ινσουλίνης, ενώ η συνολική δόση που χορηγείται σε μια συγκεκριμένη χρονική περίοδο δεν μπορεί να υπερβαίνει τις 25 μονάδες. Αυτός είναι περιορισμός ασφάλειας.
  - Η μεταβλητή display2! δείχνει το ποσό της ινσουλίνης που πρέπει να χορηγηθεί.

Σχήμα για την αντλία ινσουλίνης

INSULIN_PUMP_STATE

//Ορισμός συσκευής εισόδου

switch?: (off, manual, auto)
ManualDeliveryButton?: ℕ
Reading?: ℕ
HardwareTest?: (OK, batterylow, pumpfail, sensorfail, deliveryfail)
InsulinReservoir?: (present, notpresent)
Needle?: (present, notpresent)
clock?: TIME

//Ορισμός συσκευής εξόδου
alarm! = (on, off)
display1!, string
display2!: string
clock!: TIME
dose!: ℕ

// Μεταβλητές κατάστασης που χρησιμοποιούνται για τον υπολογισμό της δόσης
status: (running, warning, error)
r0, r1, r2: ℕ
capacity, insulin_available : ℕ
max_daily_dose, max_single_dose, minimum_dose: ℕ
safemin, safemax: ℕ
CompDose, cumulative_dose: ℕ

Αναλλοίωτες συνθήκες κατάστασης

r2 = Reading?
dose! ≤ insulin_available
insulin_available ≤ capacity

// Η συνολική δόση ινσουλίνης που χορηγήθηκε μηδενίζεται μία φορά το 24ωρο
clock? = 000000 $\Rightarrow$ cumulative_dose = 0

// Αν η συνολική δόση υπερβεί το όριο τότε αναστέλλεται η λειτουργία
cumulative_dose ≥ max_daily_dose $\wedge$ status = error $\Rightarrow$
display1! = “Υπέρβαση ημερήσιας δόσης”

// Παράμετροι διευθέτησης αντλίας
capacity = 100 $\wedge$ safemin = 6 $\wedge$ safemax = 14
max_daily_dose = 25 $\wedge$ max_single_dose = 4 $\wedge$ minimum_dose = 1

display2! = nat_to_string (dose!)
clock! = clock?

Υπολογισμός δόσης

- Η αντλία ινσουλίνης υπολογίζει το ποσό της απαιτούμενης ινσουλίνης από τη σύγκριση της τρέχουσας ένδειξης με τις δύο προηγούμενες.
- Αν από αυτές προκύπτει ότι το επίπεδο του σακχάρου αυξάνεται, τότε χορηγείται ινσουλίνη.
- Αποθηκεύονται πληροφορίες σχετικές με τη συνολική δόση που χορηγήθηκε οι οποίες επιτρέπουν την εφαρμογή της αναλλοίωτης συνθήκης του ελέγχου ασφάλειας.
- Να σημειωθεί ότι αυτή η αναλλοίωτη συνθήκη ισχύει πάντα - δεν χρειάζεται η επανάληψή της στον υπολογισμό της δόσης.

Σχήμα RUN (1)

RUN

ΔINSULIN_PUMP_STATE

switch? = auto _

status = running $\vee$ status = warning

insulin_available ≥ max_single_dose

cumulative_dose < max_daily_dose

// Η δόση ινσουλίνης υπολογίζεται ανάλογα με το επίπεδο σακχάρου στο αίμα

(SUGAR_LOW $\vee$ SUGAR_OK $\vee$ SUGAR_HIGH)

// 1. Αν η δόση που υπολογίστηκε είναι μηδέν, δεν χορηγείται ινσουλίνη

CompDose = 0 $\Rightarrow$ dose! = 0

$\vee$

// 2. Αν χορηγηθεί η δόση που υπολογίστηκε θα γίνει υπέρβαση της μέγιστης ημερήσιας δόσης, οπότε η χορηγούμενη δόση ορίζεται ως ίση με τη διαφορά μεταξύ της μέγιστης επιτρεπόμενης ημερήσιας δόσης και του συνόλου των δόσεων που έχουν χορηγηθεί μέχρι τώρα

CompDose + cumulative_dose > max_daily_dose $\Rightarrow$ alarm! = on $\wedge$ status' = warning $\wedge$ dose! = max_daily_dose-cumulative_dose

Σχήμα RUN (2)

// 3. Η κανονική κατάσταση. Χορηγείται η δόση που υπολογίστηκε αν δεν υπερβαίνει τη μέγιστη δόση. Αν η δόση που υπολογίστηκε είναι πολύ μεγάλη, η χορηγούμενη δόση περιορίζεται στη μέγιστη επιτρεπόμενη δόση.

CompDose + cumulative_dose < max_daily_dose ⇒

(CompDose ≤ max_single_dose ⇒ dose! = CompDose

∨

CompDose > max_single_dose ⇒ dose! = max_single_dose)

insulin_available′ = insulin_available – dose!

cumulative_dose′ = cumulative_dose + dose!

insulin_available ≤ max_single_dose * 4 ⇒ status′ = warning ∧

display1! = “Χαμηλή ινσουλίνη”

r1' = r2

r0' = r1

Σχήμα Sugar OK

SUGAR_OK

$r2 \geq safemin \vee r2 \leq safemax$

// το επίπεδο σακχάρου είναι σταθερό ή πέφτει

$r2 \leq r1 \Rightarrow CompDose = 0$

$\vee$

// το επίπεδο σακχάρου αυξάνεται αλλά ο ρυθμός αύξησης μειώνεται

$r2 > r1 \wedge (r2 - r1) < (r1 - r0) \Rightarrow CompDose = 0$

$\vee$

// το επίπεδο σακχάρου ανεβαίνει και ο ρυθμός αύξησης αυξάνεται:
// υπολογισμός δόσης. Αν η υπολογισμένη δόση στρογγυλοποιείται στο
// μηδέν, χορηγείται μια ελάχιστη δόση

$r2 > r1 \wedge (r2 - r1) \geq (r1 - r0) \wedge (round\ ((r2 - r1) / 4) = 0) \Rightarrow$
$CompDose = minimum\_dose$

$\vee$

$r2 > r1 \wedge (r2 - r1) \geq (r1 - r0) \wedge (round\ ((r2 - r1) / 4) > 0) \Rightarrow$
$CompDose = round\ ((r2 - r1) / 4)$

Κύρια σημεία

- Οι μέθοδοι της τυπικής προδιαγραφής συστημάτων συμπληρώνουν τις άτυπες τεχνικές προδιαγραφής των απαιτήσεων.
- Οι τυπικές προδιαγραφές είναι ακριβείς και μονοσήμαντες. Ξεκαθαρίζουν αμφίβολα σημεία των προδιαγραφών.
- Η τυπική προδιαγραφή επιβάλλει ανάλυση των απαιτήσεων του συστήματος σε πρώιμο στάδιο. Η διόρθωση σφαλμάτων σε αυτό το στάδιο έχει μικρότερο κόστος από την τροποποίηση ενός συστήματος που έχει παραδοθεί.
- Οι τεχνικές της τυπικής προδιαγραφής εφαρμόζονται κυρίως στην ανάπτυξη κρίσιμων συστημάτων και προτύπων.

Κύρια σημεία

- Οι αλγεβρικές τεχνικές είναι κατάλληλες για την προδιαγραφή διασυνδέσεων, όπου η διασύνδεση ορίζεται ως ένα σύνολο κλάσεων αντικειμένων.
- Οι τεχνικές βάσει μοντέλου μοντελοποιούν το σύστημα χρησιμοποιώντας σύνολα και συναρτήσεις. Αυτό απλοποιεί ορισμένους τύπους προδιαγραφών συμπεριφοράς.
- Στην προδιαγραφή βάσει μοντέλου οι λειτουργίες ορίζονται μέσω προσυνθηκών και μετασυνθηκών που αφορούν την κατάσταση του συστήματος.