ΕΠΛ362: Τεχνολογία Λογισμικού ΙΙ

(μετάφραση στα ελληνικά των διαφανειών του βιβλίου Software Engineering, 9/E, Ian Sommerville, 2011)

Ενότητα 8 (κεφάλαιο 21) – Θεματοκεντρική Τεχνολογία Λογισμικού

Οι διαφάνειες αυτές έχουν συμπληρωματικό και επεξηγηματικό χαρακτήρα και σε καμία περίπτωση δεν υποκαθιστούν το βιβλίο

Γιώργος Α. Παπαδόπουλος
Τμήμα Πληροφορικής
Πανεπιστήμιο Κύπρου

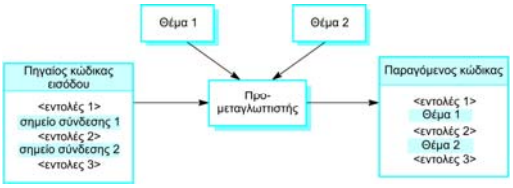
Περιεχόμενα

- ✧ Διαχωρισμός των ζητουμένων.
- ✧ Θέματα, σημεία σύνδεσης και σημειοτομές.
- ✧ Τεχνολογία λογισμικού με θέματα.

Θεματοκεντρική ανάπτυξη λογισμικού (α)

- ✧ Μία προσέγγιση ανάπτυξης λογισμικού η οποία βασίζεται σε ένα νέο τύπο αφαίρεσης – το θέμα.
- ✧ Χρησιμοποιείται μαζί με άλλες προσεγγίσεις – συνήθως την αντικειμενοστρεφή τεχνολογία λογισμικού.
- ✧ Τα θέματα ενθυλακώνουν λειτουργικές δυνατότητες (τα ζητούμενα) που διασταυρώνονται και συνυπάρχουν με άλλες λειτουργικές δυνατότητες.
- ✧ Τα θέματα περιλαμβάνουν έναν ορισμό για το σημείο στο οποίο πρέπει να συμπεριληφθούν στο πρόγραμμα (με ύφανση), καθώς και κώδικα που υλοποιεί το εγκάρσιο (cross cutting) ζητούμενο.
- ✧ Ο κώδικας των ζητουμένων προέρχεται από επαναχρησιμοποίηση, ενώ το νέο σύστημα παράγεται από τον προμεταγλωττιστή θεμάτων.

Θεματοκεντρική ανάπτυξη λογισμικού (β)



Διαχωρισμός των ζητουμένων

- ✧ Σύμφωνα με την αρχή του διαχωρισμού των ζητουμένων, το λογισμικό πρέπει να είναι οργανωμένο έτσι ώστε κάθε στοιχείο του προγράμματος να κάνει ένα και μοναδικό πράγμα.
- ✧ Έτσι κάθε στοιχείο του προγράμματος θα είναι κατανοητό χωρίς να παίζουν ρόλο οι αναφορές σε άλλα στοιχεία.
- ✧ Τα αφηρημένα στοιχεία του προγράμματος (υπορουτίνες, διεργασίες, αντικείμενα, κλπ.) υποστηρίζουν το διαχωρισμό των ζητουμένων.

Ζητούμενα

- ✧ Τα ζητούμενα δεν είναι ζητήματα που αφορούν το πρόγραμμα, αλλά απεικονίζουν τις απαιτήσεις του συστήματος και τις προτεραιότητες που έχουν θέσει οι ενδιαφερόμενοι.
  - Παραδείγματα ζητουμένων αποτελούν η απόδοση, η προστασία από εξωτερικούς κινδύνους, συγκεκριμένες λειτουργικές δυνατότητες, κλπ.
- ✧ Όταν σε ένα πρόγραμμα είναι σαφής ο διαχωρισμός των ζητουμένων, υπάρχει η δυνατότητα σαφούς ανίχνευσης των απαιτήσεων μέχρι την υλοποίησή τους.
- ✧ Τα κύρια ζητούμενα είναι τα λειτουργικά ζητούμενα που σχετίζονται με τον πρωταρχικό σκοπό του συστήματος, ενώ τα δευτερεύοντα ζητούμενα είναι τα λειτουργικά ζητούμενα που αντιστοιχούν σε μη λειτουργικές απαιτήσεις και απαιτήσεις σχετικές με την ποιότητα εξυπηρέτησης.

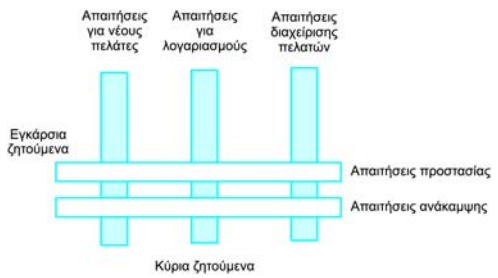
Ζητούμενα των ενδιαφερομένων

- Λειτουργικά ζητούμενα που σχετίζονται με συγκεκριμένες λειτουργικές δυνατότητες οι οποίες πρέπει να συμπεριλαμβάνονται στο σύστημα.
- Ζητούμενα σχετικά με την ποιότητα εξυπηρέτησης, τα οποία αφορούν μη λειτουργική συμπεριφορά ενός συστήματος.
- Ζητούμενα πολιτικής, που σχετίζονται με τις συνολικές πολιτικές οι οποίες διέπουν τη χρήση του συστήματος.
- Ζητούμενα συστήματος, τα οποία αφορούν ιδιότητες του συστήματος ως σύνολο, όπως η δυνατότητα συντήρησης ή διευθέτησής του.
- Εταιρικά ζητούμενα, που σχετίζονται με στόχους και προτεραιότητες της εταιρείας όπως η παραγωγή ενός συστήματος μέσα στα όρια του προϋπολογισμού, η χρήση υπαρχόντων πόρων λογισμικού ή η διατήρηση της φήμης μίας εταιρείας.

Εγκάρσια ζητούμενα (α)

- Τα εγκάρσια ζητούμενα είναι εκείνα των οποίων η υλοποίηση επηρεάζει πολλά συστατικά στοιχεία ενός προγράμματος.
- Αυτό επιφέρει δυσκολίες όταν πρέπει να πραγματοποιηθούν αλλαγές σε ένα συγκεκριμένο ζητούμενο – ο κώδικας που πρέπει να τροποποιηθεί δεν βρίσκεται σε μία θέση, αλλά σε διάφορες θέσεις ενός συστήματος.
- Τα εγκάρσια ζητούμενα οδηγούν σε φαινόμενα ανάμιξης και διασποράς.

Εγκάρσια ζητούμενα (β)



Ανάμιξη διαχείρισης χώρου προσωρινής αποθήκευσης και κώδικα συγχρονισμού

```
synchronized void put (SensorRecord rec )
{
// Check that there is space in the buffer; wait if not
if ( numberOfEntries == bufsize)
wait () ;
// Add record at end of buffer
store [back] = new SensorRecord (rec.sensorId, rec.sensorVal) ;
back = back + 1 ;
// If at end of buffer, next entry is at the beginning
if (back == bufsize)
back = 0 ;
numberOfEntries = numberOfEntries + 1 ;
// indicate that buffer is available
notify () ;
} // put
```


Διασπορά μεθόδων οι οποίες υλοποιούν δευτερεύοντα ζητούμενα

Θέματα, σημεία σύνδεσης και σημειοτομές

- ✧ Θέμα είναι μία προγραμματιστική αφαίρεση η οποία υλοποιεί ένα ζητούμενο. Περιέχει πληροφορίες για τη θέση στην οποία πρέπει να συμπεριληφθεί στο πρόγραμμα.
- ✧ Σημείο σύνδεσης είναι μία θέση του προγράμματος στην οποία μπορεί να συμπεριληφθεί (να υφανθεί) ένα θέμα.
- ✧ Οι σημειοτομές ορίζουν πού (σε ποια σημεία σύνδεσης) θα συμπεριληφθούν τα θέματα στο πρόγραμμα.

Ορολογία που χρησιμοποιείται στη θεματοκεντρική ανάπτυξη λογισμικού

Όρος	Ορισμός
Συμβουλή (advice).	Ο κώδικας που υλοποιεί ένα ζητούμενο.
Θέμα (aspect).	Μία προγραμματιστική αφαίρεση η οποία ορίζει ένα εγκάρσιο ζητούμενο. Περιλαμβάνει τον ορισμό μίας σημειοτομής και τη συμβουλή που σχετίζεται με αυτό το ζητούμενο.
Σημείο σύνδεσης (join point).	Συμβάν κατά την εκτέλεση ενός προγράμματος το οποίο θα μπορούσε να προκαλέσει την εκτέλεση της συμβουλής που σχετίζεται με κάποιο θέμα.
Μοντέλο σημείων σύνδεσης.	Το σύνολο των συμβάντων στα οποία μπορεί να αναφέρεται μία σημειοτομή.
Σημειοτομή (pointcut).	Μία δήλωση η οποία συμπεριλαμβάνεται σε ένα θέμα και ορίζει τα σημεία σύνδεσης στα οποία πρέπει να εκτελεστεί η αντίστοιχη συμβουλή θέματος.
Ύφανση (weaving).	Η ενσωμάτωση κώδικα συμβουλής στα καθορισμένα σημεία σύνδεσης από ένα μηχανισμό ύφανσης θεμάτων.

Περιγραφή θέματος για πιστοποίηση ταυτότητας

```
aspect authentication
{
 before: call(public void update* (..)) // σημειοτομή
 {
  // η συμβουλή που πρέπει να εκτελεστεί όταν το θέμα
  // "υφανθεί" στο εκτελούμενο σύστημα
  int tries = 0;
  string userPassword = Password.Get( tries );
  while(tries < 3 && userPassword != thisUser.password( ) )
  {
   // επιτρέπει 3 προσπάθειες καταχώρισης του κωδικού πρόσβασης
   tries = tries + 1;
   userPassword = Password.Get( tries );
  }
  if (userPassword != thisUser.password( )) then
      // αν ο κωδικός είναι λάθος, ο χρήστης θεωρείται
     // ότι ξέχασε να αποσυνδεθεί
     System.Logout (thisUser.uid);
 }
} // authentication
```


AspectJ – μοντέλο σημείων σύνδεσης

- ✧ Συμβάντα κλήσεων.
  - Κλήσεις σε μία μέθοδο ή συνάρτηση κατασκευής.
- ✧ Συμβάντα εκτέλεσης.
  - Η εκτέλεση μίας μεθόδου ή συνάρτησης κατασκευής.
- ✧ Συμβάντα απόδοσης αρχικών τιμών.
  - Απόδοση αρχικών τιμών σε κλάση ή αντικείμενο.
- ✧ Συμβάντα δεδομένων.
  - Προσπέλαση ή ενημέρωση ενός πεδίου.
- ✧ Συμβάντα εξαιρέσεων.
  - Ο χειρισμός μίας εξαίρεσης.

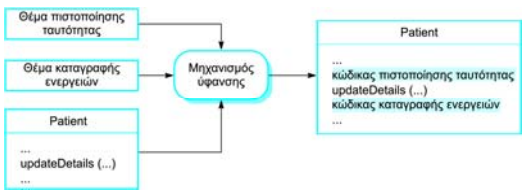
Σημειοτομές

- ✧ Μία σημειοτομή προσδιορίζει τα συγκεκριμένα συμβάντα με τα οποία πρέπει να συσχετιστεί μία συμβουλή.
- ✧ Παραδείγματα θεματικών πλαισίων στα οποία μπορεί να υφανθεί μία συμβουλή.
  - Πριν την εκτέλεση μίας συγκεκριμένης μεθόδου.
  - Μετά την επιστροφή μίας μεθόδου, είτε υπό κανονικές συνθήκες είτε με την παραγωγή κάποιας εξαίρεσης.
  - Με κάθε τροποποίηση ενός πεδίου αντικειμένου.

Ύφανση θεμάτων (α)

- ✧ Οι προμεταγλωττιστές επεξεργάζονται τον πηγαίο κώδικα και υφαίνουν τα θέματα στις καθορισμένες σημειοτομές στο πρόγραμμα.
- ✧ Υπάρχουν τρεις προσεγγίσεις ύφανσης θεμάτων.
  - Προεπεξεργασία του πηγαίου κώδικα.
  - Ύφανση κατά το χρόνο σύνδεσης.
  - Δυναμική ύφανση κατά το χρόνο εκτέλεσης.

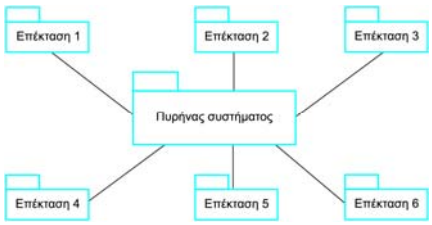
Ύφανση θεμάτων (β)



Τεχνολογία λογισμικού με θέματα

- Τα θέματα εμφανίστηκαν ως δομές γλώσσας προγραμματισμού αλλά, επειδή η πραγματική προέλευση της έννοιας των ζητούμενων είναι οι απαιτήσεις συστήματος, μπορεί να υιοθετηθεί μία θεματοκεντρική προσέγγιση σε όλα τα στάδια της διαδικασίας ανάπτυξης ενός συστήματος.
- Η αρχιτεκτονική ενός θεματοκεντρικού συστήματος βασίζεται γύρω από έναν πυρήνα και τις διάφορες επεκτάσεις.
- Ο πυρήνας υλοποιεί τα πρωτεύοντα ζητούμενα. Οι επεκτάσεις υλοποιούν τα δευτερεύοντα και τα εγκάρσια ζητούμενα.

Πυρήνας συστήματος με επεκτάσεις



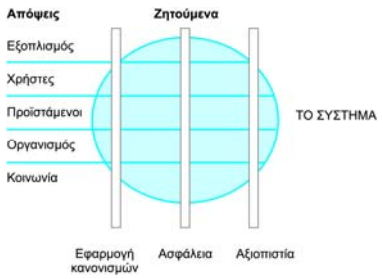
Τύποι επεκτάσεων

- Δευτερεύουσες λειτουργικές επεκτάσεις.
  - Προσθέτουν επιπλέον λειτουργικές δυνατότητες στον πυρήνα του συστήματος.
- Επεκτάσεις πολιτικής.
  - Προσθέτουν λειτουργικές δυνατότητες με σκοπό την υποστήριξη κάποιας εταιρικής πολιτικής, όπως είναι η πολιτική προστασίας από εξωτερικούς κινδύνους.
- Επεκτάσεις ποιότητας εξυπηρέτησης.
  - Προσθέτουν λειτουργικές δυνατότητες οι οποίες βοηθούν στην επίτευξη των απαιτήσεων εξυπηρέτησης.
- Επεκτάσεις υποδομής.
  - Προσθέτουν λειτουργικές δυνατότητες με στόχο την υποστήριξη της υλοποίησης ενός συστήματος σε κάποια συγκεκριμένη πλατφόρμα.

Τεχνολογία απαιτήσεων βάσει ζητουμένων

- ✧ Μία προσέγγιση της τεχνολογίας απαιτήσεων η οποία επικεντρώνεται στα ζητούμενα που θέτει ο πελάτης και είναι συνεπής με τη θεματοκεντρική ανάπτυξη λογισμικού.
- ✧ Οι απόψεις είναι ένας τρόπος διαχωρισμού των ζητουμένων κάθε ενδιαφερομένου.
- ✧ Οι απόψεις αντιπροσωπεύουν τις απαιτήσεις ομάδων ενδιαφερομένων που σχετίζονται μεταξύ τους.
- ✧ Τα εγκάρσια ζητούμενα είναι ζητούμενα που επισημαίνονται σε κάθε άποψη.

Απόψεις και ζητούμενα



Απόψεις για ένα σύστημα αποθήκης

1. Χρήστες υπηρεσιών έκτακτης ανάγκης
1.1 Εντοπισμός εξοπλισμού καθορισμένου τύπου (π.χ. ανυψωτικοί μηχανισμοί)
1.2 Προβολή εξοπλισμού που είναι διαθέσιμος σε μία συγκεκριμένη αποθήκη
1.3 Εξαγωγή εξοπλισμού
1.4 Εισαγωγή εξοπλισμού
1.5 Διευθέτηση μεταφοράς εξοπλισμού σε έκτακτο περιστατικό
1.6 Υποβολή αναφοράς ζημιών
1.7 Εντοπισμός αποθήκης κοντά στο επείγον περιστατικό

2. Δημιουργοί σχεδίων αντιμετώπισης έκτακτων αναγκών
2.1 Εντοπισμός εξοπλισμού καθορισμένου τύπου
1.2 Προβολή εξοπλισμού που είναι διαθέσιμος σε μία συγκεκριμένη τοποθεσία
2.3 Προσθήκη και αφαίρεση εξοπλισμού από μία αποθήκη
2.4 Μεταφορά εξοπλισμού από τη μία αποθήκη στην άλλη
2.6 Παραγγελία νέου εξοπλισμού

3. Προσωπικό συντήρησης
3.1 Εισαγωγή/εξαγωγή εξοπλισμού για συντήρηση
3.2 Προβολή διαθέσιμου εξοπλισμού σε κάθε αποθήκη
3.3 Εντοπισμός εξοπλισμού καθορισμένου τύπου
3.4 Προβολή χρονοδιαγράμματος συντήρησης για ένα στοιχείο εξοπλισμού
3.5 Προβολή ιστορικού συντήρησης για ένα στοιχείο εξοπλισμού
3.6 Εμφάνιση όλων των ειδών που χρειάζονται συντήρηση σε μία αποθήκη

Απαιτήσεις διαθεσιμότητας για το σύστημα αποθήκης

AV.1 Σε τοποθεσία διαφορετική από αυτή του κύριου συστήματος, πρέπει να υπάρχει διαθέσιμο ένα εφεδρικό σύστημα σε κατάσταση αναμονής.

Σκεπτικό: Το περιστατικό έκτακτης ανάγκης μπορεί να επηρεάσει την τοποθεσία του κύριου συστήματος.

AV.1.1 Όλες οι συναλλαγές πρέπει να καταγράφονται τόσο στην τοποθεσία του κύριου συστήματος όσο και στην απομακρυσμένη τοποθεσία του εφεδρικού συστήματος.

Σκεπτικό: Αυτό επιτρέπει την αναπαραγωγή των συναλλαγών και τη συνέπεια των βάσεων δεδομένων του συστήματος.

AV.1.2 Το σύστημα πρέπει να στέλνει πληροφορίες κατάστασης στο κέντρο ελέγχου κάθε πέντε λεπτά.

Σκεπτικό: Αν το κύριο σύστημα δεν είναι διαθέσιμο, οι χειριστές στο κέντρο ελέγχου θα μπορούν να μεταφερθούν στο εφεδρικό σύστημα.

Σύστημα αποθήκης – απαιτήσεις πυρήνα

- C.1 Το σύστημα θα επιτρέπει σε εξουσιοδοτημένους χρήστες να προβάλλουν την περιγραφή οποιουδήποτε στοιχείου εξοπλισμού που υπάρχει στην αποθήκη των υπηρεσιών έκτακτης ανάγκης.
- C.2 Το σύστημα θα περιλαμβάνει μία λειτουργία αναζήτησης η οποία θα επιτρέπει σε εξουσιοδοτημένους χρήστες να αναζητούν συγκεκριμένα στοιχεία ή τύπους εξοπλισμού σε μία μεμονωμένη αποθήκη ή γενικά.

Σύστημα αποθήκης – απαιτήσεις επεκτάσεων

- E1.1 Ένας εξουσιοδοτημένος χρήστης θα μπορεί να κάνει παραγγελίες σε διαπιστευμένους προμηθευτές για αντικατάσταση στοιχείων του εξοπλισμού.
- E1.1.1 Όταν παραγγέλλεται κάποιο στοιχείο εξοπλισμού, θα πρέπει να αποδίδεται σε μία συγκεκριμένη αποθήκη και να επισημαίνεται ως «επί παραγγελία».

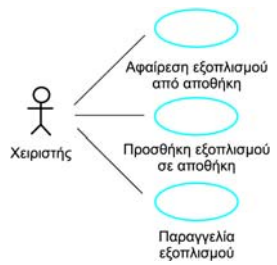
Θεματοκεντρικός σχεδιασμός και προγραμματισμός

- ✧ Θεματοκεντρικός σχεδιασμός.
  - Η διαδικασία σχεδιασμού ενός συστήματος με χρήση θεμάτων για την υλοποίηση των εγκάρσιων ζητούμενων και των επεκτάσεων που προσδιορίζονται κατά τη διαδικασία εξαγωγής απαιτήσεων.
- ✧ Θεματοκεντρικός προγραμματισμός.
  - Η υλοποίηση ενός θεματοκεντρικού σχεδιασμού με χρήση μίας θεματοκεντρικής γλώσσας προγραμματισμού όπως είναι η AspectJ.

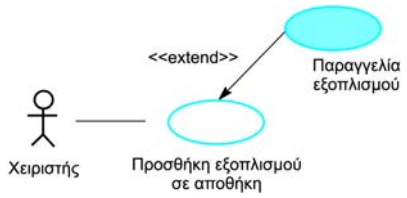
Περιπτώσεις χρήσης

- ✧ Οι περιπτώσεις χρήσης μπορούν να αποτελέσουν τη βάση για τη θεματοκεντρική τεχνολογία λογισμικού.
- ✧ Κάθε περίπτωση χρήσης αντιπροσωπεύει ένα θέμα.
  - Οι επεκτάσεις περιπτώσεων χρήσης ταιριάζουν απόλυτα με το αρχιτεκτονικό μοντέλο συστημάτων το οποίο αποτελείται από τον πυρήνα και διάφορες επεκτάσεις.
- ✧ Οι Jacobsen και Ng επεκτείνουν την έννοια των περιπτώσεων χρήσης με την εισαγωγή νέων εννοιών όπως είναι τα τεμάχια περιπτώσεων χρήσης και οι υπομονάδες περιπτώσεων χρήσης.

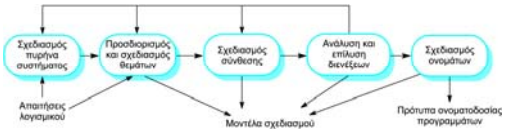
Περιπτώσεις χρήσης για το σύστημα διαχείρισης αποθήκης



Μία επέκταση περίπτωσης χρήσης για το σύστημα διαχείρισης αποθήκης



Μία γενικευμένη διαδικασία θεματοκεντρικού σχεδιασμού



Δραστηριότητες σχεδιασμού (α)

- *Σχεδιασμός πυρήνα συστήματος*.
  - Σχεδιάζεται μία αρχιτεκτονική για την υποστήριξη της λειτουργικότητας του πυρήνα του συστήματος Εδώ πρέπει να λαμβάνονται υπόψη απαιτήσεις για ποιότητα εξυπηρέτησης, απόδοση, αξιοπιστία, κλπ. Οι οποίες όμως θα υλοποιηθούν με τη μορφή επεκτάσεων.
- *Προσδιορισμός και σχεδιασμός θεμάτων.*
  - Ξεκινώντας με τις επεκτάσεις που έχουν προσδιορισθεί στις απαιτήσεις του συστήματος, πρέπει να τις αναλύσετε για να δείτε αν αποτελούν θέματα από μόνες τους ή αν πρέπει να διασπαστούν σε ένα αριθμό από θέματα, τα οποία μετά μπορείτε να τα σχεδιάσετε ξεχωριστά.
- *Σχεδιασμός σύνθεσης*.
  - Εδώ αναλύετε τον πυρήνα του συστήματος και τους σχεδιασμούς των θεμάτων για να διαπιστώσετε που ακριβώς πρέπει να συνδυαστούν τα θέματα με τον πυρήνα. Ουσιαστικά, σε αυτό το στάδιο προσδιορίζετε τα σημεία σύνδεσης ενός προγράμματος όπου θα υφανθούν τα θέματα.

Δραστηριότητες σχεδιασμού (β)

✧ *Ανάλυση και επίλυση διενέξεων.*

- Διενέξεις προκύπτουν όταν διαφορετικά θέματα σε μία ή περισσότερες σημειοτομές καθορίζουν ότι πρέπει να συντεθούν στο ίδιο σημείο του προγράμματος.
- Ή, επειδή το κάθε θέμα, επειδή σχεδιάζεται ανεξάρτητα από τα άλλα, ενσωματώνει παραδοχές για τη λειτουργικότητα του πυρήνα οι οποίες πιθανόν να μην ισχύουν όταν συνδυάζονται πολλά θέματα.

✧ *Σχεδιασμός ονομάτων.*

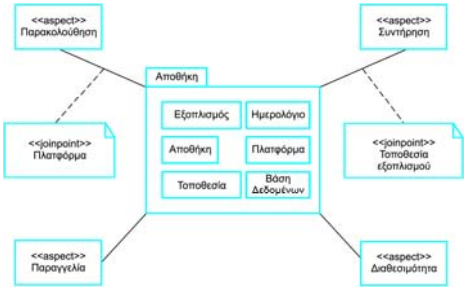
- Εδώ ορίζονται πρότυπα για την ονομασία των οντοτήτων στο πρόγραμμα. Κάτι τέτοιο είναι απαραίτητο για να αποφευχθεί το πρόβλημα των αθέλητων σημειοτομών. Αυτές παρουσιάζονται όταν, σε κάποιο σημείο σύνδεσης του προγράμματος, το όνομα ταιριάζει με κάποιο υπόδειγμα σημειοτομής χωρίς να υπάρχει πρόθεση ύφανσης συμβουλής στη συγκεκριμένη περίπτωση.

Επεκτάσεις σε UML

✧ Για την έκφραση ενός θεματοκεντρικού σχεδιασμού σε γλώσσα UML απαιτούνται:

- Ένα μέσο μοντελοποίησης θεμάτων με χρήση των στερεοτύπων της UML.
- Ένα μέσο καθορισμού των σημείων σύνδεσης στα οποία η συμβουλή κάθε θέματος πρόκειται να συντεθεί με τον πυρήνα του συστήματος.

Ένα μοντέλο θεματοκεντρικού σχεδιασμού



Μέρος μοντέλου θέματος

Δήλωση επέκτασης

- Στη μέθοδο `viewItem`, μετά την κλήση της μεθόδου `getItemInfo`, πρέπει να συμπεριληφθεί μία κλήση της μεθόδου `displayHistory` για την εμφάνιση του ιστορικού συντήρησης.

Επαλήθευση και επικύρωση

- Η διαδικασία της εξασφάλισης ότι ένα πρόγραμμα ικανοποιεί τις προδιαγραφές του (επαλήθευση) και ότι ικανοποιεί τις πραγματικές ανάγκες των ενδιαφερομένων (επικύρωση).
- Όπως και άλλα συστήματα, τα θεματοκεντρικά συστήματα μπορούν να θεωρηθούν στις δοκιμές ως «μαύρα κουτιά» και οι προδιαγραφές τους θα χρησιμοποιηθούν για την επινόηση των δοκιμών.
- Ωστόσο, οι επιθεωρήσεις του κώδικα και οι δοκιμές «λευκού κουτιού», οι οποίες βασίζονται στον πηγαίο κώδικα του προγράμματος, είναι προβληματικές.
- Με τα θέματα εμφανίζονται επιπλέον δυσκολίες στις δοκιμές.

Προβλήματα στην επιθεώρηση προγραμμάτων

- Για να είναι αποτελεσματική η επιθεώρηση ενός προγράμματος (το οποίο έχει γραφεί σε μία συμβατική γλώσσα προγραμματισμού), ο αναγνώστης πρέπει να μπορεί να το διαβάσει από αριστερά προς τα δεξιά και από επάνω προς τα κάτω.
- Τα θέματα καθιστούν αυτή την ανάγνωση αδύνατη αφού το πρόγραμμα έχει τη μορφή ιστού και όχι ενός σειριακού εγγράφου. Ο αναγνώστης δεν μπορεί να διακρίνει από τον πηγαίο κώδικα σε ποιο σημείο θα υφανθεί και θα εκτελεστεί ένα θέμα.
- Η «επιπεδοποίηση» ενός θεματοκεντρικού προγράμματος ώστε να διευκολυνθεί η ανάγνωσή του είναι πρακτικά ανέφικτη.

Δοκιμές λευκού κουτιού

- Ο στόχος των δοκιμών λευκού κουτιού είναι η αξιοποίηση της γνώσης του πηγαίου κώδικα στο σχεδιασμό δοκιμών οι οποίες θα παρέχουν κάποιο επίπεδο κάλυψης του προγράμματος – για παράδειγμα, κάθε λογικός κλάδος του προγράμματος πρέπει να εκτελείται τουλάχιστον μία φορά.
- Προβλήματα με τα θέματα.
  - Πώς μπορεί να χρησιμοποιηθεί η γνώση του πηγαίου κώδικα στην παραγωγή δοκιμών;
  - Τι ακριβώς σημαίνει ο όρος «κάλυψη δοκιμών»;

Προβλήματα με τα θέματα

- Η δημιουργία γραφήματος ροής για ένα πρόγραμμα με θέματα δεν είναι εφικτή. Επομένως είναι δύσκολο να σχεδιαστούν με συστηματικό τρόπο δοκιμές που εξασφαλίζουν ότι εκτελούνται όλοι οι συνδυασμοί του βασικού κώδικα και των θεμάτων.
- Τι ακριβώς σημαίνει ο όρος «κάλυψη δοκιμών»;
  - Μήπως εννοούμε την εκτέλεση του κώδικα κάθε θέματος τουλάχιστον μία φορά;
  - Μήπως εννοούμε ότι ο κώδικας ενός θέματος πρέπει να εκτελεστεί τουλάχιστον μία φορά σε κάθε σημείο σύνδεσης το οποίο καθορίζεται στη σημειοτομή του θέματος;
  - Ή μήπως κάτι άλλο;

Δυσκολίες που επιφέρουν τα θέματα στις δοκιμές

- Με ποιον τρόπο πρέπει να καθορίζονται τα θέματα έτσι ώστε να είναι δυνατή η παραγωγή δοκιμών τους;
- Πώς μπορεί να δοκιμαστεί ένα θέμα ανεξάρτητα από τον πυρήνα στον οποίο θα υφανθεί;
- Πώς μπορούν να δοκιμαστούν οι περιπτώσεις διενέξεων μεταξύ θεμάτων;
- Με ποιον τρόπο μπορούν να σχεδιαστούν δοκιμές έτσι ώστε να εκτελούνται όλα τα σημεία σύνδεσης ενός προγράμματος και να εφαρμόζονται οι κατάλληλες δοκιμές θεμάτων;

Κύρια σημεία (α)

- Το κύριο πλεονέκτημα μίας θεματοκεντρικής προσέγγισης στην ανάπτυξη λογισμικού είναι ότι υποστηρίζει διαχωρισμό των ζητουμένων. Η αναπαράσταση των εγκάρσιων ζητουμένων με τη μορφή θεμάτων επιτρέπει την ανεξάρτητη κατανόηση, επαναχρησιμοποίηση και τροποποίησή τους.
- Ανάμιξη έχουμε όταν μία υπομονάδα υλοποιεί πολλές διαφορετικές απαιτήσεις του συστήματος. Διασπορά παρατηρείται όταν η υλοποίηση ενός μόνο ζητουμένου «διασπείρεται» σε πολλά συστατικά στοιχεία ενός προγράμματος.
- Τα θέματα περιλαμβάνουν σημειοτομές – δηλώσεις οι οποίες ορίζουν πού ακριβώς θα υφανθεί στο πρόγραμμα το θέμα, και συμβουλές – κώδικα που υλοποιεί τα κύρια ζητούμενα των ενδιαφερομένων και επεκτάσεις που υλοποιούν τα δευτερεύοντα ζητούμενα.

Κύρια σημεία (β)

- Ο διαχωρισμός των ζητουμένων επιτυγχάνεται μέσω του σχεδιασμού ενός συστήματος έτσι ώστε να αποτελείται από ένα πυρήνα ο οποίος υλοποιεί τα κύρια ζητούμενα και επεκτάσεις που υλοποιούν τα δευτερεύοντα.
- Ο προσδιορισμός των ζητουμένων μπορεί να γίνει με χρήση μίας προσέγγισης εξαγωγής απαιτήσεων βάσει απόψεων, η οποία επιτρέπει την εξαγωγή των απαιτήσεων από τους ενδιαφερόμενους και τον ρητό προσδιορισμό εγκάρσιων θεμάτων που αφορούν ποιότητα εξυπηρέτησης και πολιτικές.
- Η μετάβαση από τις απαιτήσεις στο σχεδιασμό είναι εφικτή με τον προσδιορισμό περιπτώσεων χρήσης, καθεμιά από τις οποίες αντιπροσωπεύει ένα ζητούμενο κάποιου ενδιαφερομένου.
- Τα προβλήματα στις επιθεωρήσεις και την παραγωγή δοκιμών για θεματοκεντρικά προγράμματα αποτελούν σημαντικό εμπόδιο στην υιοθέτηση της θεματοκεντρικής ανάπτυξης λογισμικού.