

Κεφάλαιο Τέσσερα
-
Πρότυπα, Υποδομές και Αρχιτεκτονικές

Εισαγωγή

- Η μηχανική λογισμικού ακολουθεί καθοδηγήσεις και παραδείγματα από έμπρακτους σχεδιασμούς, έτσι ώστε να καταλήγει στις σωστές αποφάσεις. Υπάρχει δηλαδή συνδυασμός λίγης θεωρίας για τη διαχείριση των συστατικών με πολλή εμπειρία στο σχεδιασμό συστημάτων με επαναχρησιμοποιούμενα συστατικά. Η δημιουργία προϊόντων που ικανοποιούν τεχνικές και μη τεχνικές προϋποθέσεις βασίζεται σ' αυτό το μίγμα. Αυτό προϋποθέτει την ύπαρξη αποτελεσματικών μηχανισμών για την ανταλλαγή πληροφοριών μεταξύ έμπρακτων παραδειγμάτων.
- Στην ουσία της μηχανικής λογισμικού, είναι σημαντική η πλήρης κατανόηση των τεχνικών επαναχρησιμοποίησης εδραιωμένων σχεδιασμών. Η πλήρης αξιοποίηση τέτοιων μεθόδων σχεδιασμού υποθέτει τη χρησιμοποίηση μίας καθιερωμένης μεθοδολογίας που έχει αποδεικτεί εμπειρικά.

4.1 Τύποι Επαναχρησιμοποίησης Επιπέδου Σχεδιασμού

- Η επαναχρησιμοποίηση αποδεδειγμένων εμπειρικά σχεδίων, είναι βασική, αλλά δεν υπάρχει προσέγγιση που να καλύπτει όλα τα επίπεδα διαμελισμού. Υπάρχει μεγάλη διαφορά μεταξύ τού να ταξινομείς μία σειρά αριθμών και να κτίζεις πολύπλοκα αλληλοεπιδρώντα συστήματα.
- Ο σχεδιασμός επαναχρησιμοποίησης μπορεί να γίνει κατανοητός σαν μία προσπάθεια να γίνουν κοινές (*shared*), δηλ. να διαμοιραστούν ορισμένες προοπτικές μίας προσέγγισης δια μέσου διαφόρων μελετών. Τέτοια είδη σχεδιασμών παρατίθενται πιο κάτω με τα επίπεδα χρήσης τους.
 - Διαμοίρασμα συνοχής: γλώσσες προγραμματισμού.
 - Διαμοίρασμα συγκεκριμένων κομματιών επιλύσεως: βιβλιοθήκες.
 - Διαμοίρασμα συμβάσεων: διασυνδέσεις.
 - Διαμοίρασμα αρχιτεκτονικής αλληλεπίδρασης: πρότυπα.
 - Διαμοίρασμα αρχιτεκτονικής υποσυστημάτων: υποδομές.
 - Διαμοίρασμα ολικής δομής: αρχιτεκτονικές.

4.1.1 Διαμοίρασμα Συνοχής: Γλώσσες Προγραμματισμού

- Ένας από τους παλαιότερους τύπους επαναχρησιμοποίησης αποδεδειγμένων μεθόδων είναι η χρησιμοποίησή τους σε γλώσσες προγραμματισμού. Με την κατασκευή της μία γλώσσα προγραμματισμού καθοδηγεί τον προγραμματιστή πώς πρέπει να γίνονται ή να μην γίνονται ορισμένα πράγματα.
 - Για παράδειγμα μία γλώσσα μπορεί να παρουσιάζει δυσκολίες στη δημιουργία δυναμικά μεταβαλλόμενων πινάκων, και στο τέλος ο προγραμματιστής να καταλήξει στο συμπέρασμα ότι δεν χρειάζονται.
- Τελικά οι πλέον επιτυχημένες γλώσσες είναι αυτές που συνδυάζουν την καθοδήγηση με την ελαστικότητα για απρόβλεπτες καταστάσεις.
- Καθώς αυξάνει το μέγεθος και η πολυπλοκότητα, η σημασία της αρχιτεκτονικής κερδίζει έδαφος και οι πιο απαιτητικές γλώσσες γίνονται περισσότερο δημοφιλείς.
- Η καθοδήγηση τέτοιων γλωσσών απλά αποτρέπει πράγματα που μπορούν να προκαλέσουν προβλήματα. Έτσι υπάρχουν στη ανάπτυξη του συστήματος ιδεατές έννοιες, δομές και μέσα σύνθεσης, που είναι:
 - Στατικού τύπου συστήματα με μηχανισμούς παραμετρικού πολυμορφισμού ή οριακού πολυμορφισμού.
 - Συναρτήσεις, ψηλού βαθμού συναρτήσεις και συναρτησιακές συνθέσεις.
 - Ενότητες (closures or blocks).
 - Κατ'ανάγκη εκτέλεση (lazy evaluation).
 - Περιλήψεις διεργασίας (procedural abstractions).
 - Εξαιρέσεις και χειρισμός εξαιρέσεων (exceptions).
 - Τάξεις και διαδοχή εφαρμογών.
 - Δυναμικοί τύποι συστημάτων με πολυμορφισμό.
 - Υποστήριξη διαστήματος ανοικτού αντικειμένου.
 - Αργή σύνδεση (late binding) και απόσπαση κατά τύπο.
 - Υποστήριξη συγχρονισμού ταυτόχρονων διεργασιών.
 - Συστήματα διαμορφωτών και πακέτων (modules and package systems).
- Οι υβριδικές αντικειμενοστρεφείς γλώσσες είναι οι πιο πετυχημένες στο σύγχρονο σπονδυλωτό προγραμματισμό.

4.1.2 Διαμοίρασμα Συγκεκριμένων Κομματιών Επίλυσης: Βιβλιοθήκες

- Την αρχική τάση για γλώσσες που παρέχουν όλες τις πιθανές συναρτήσεις, αντικατέστησαν γλώσσες με διαμορφωτές που μετέθεσαν της συναρτήσεις σε διαμορφωτές-βιβλιοθήκες. Με αυτό τον τρόπο δεν εξελίσσεται η γλώσσα αλλά οι διαμορφωτές.

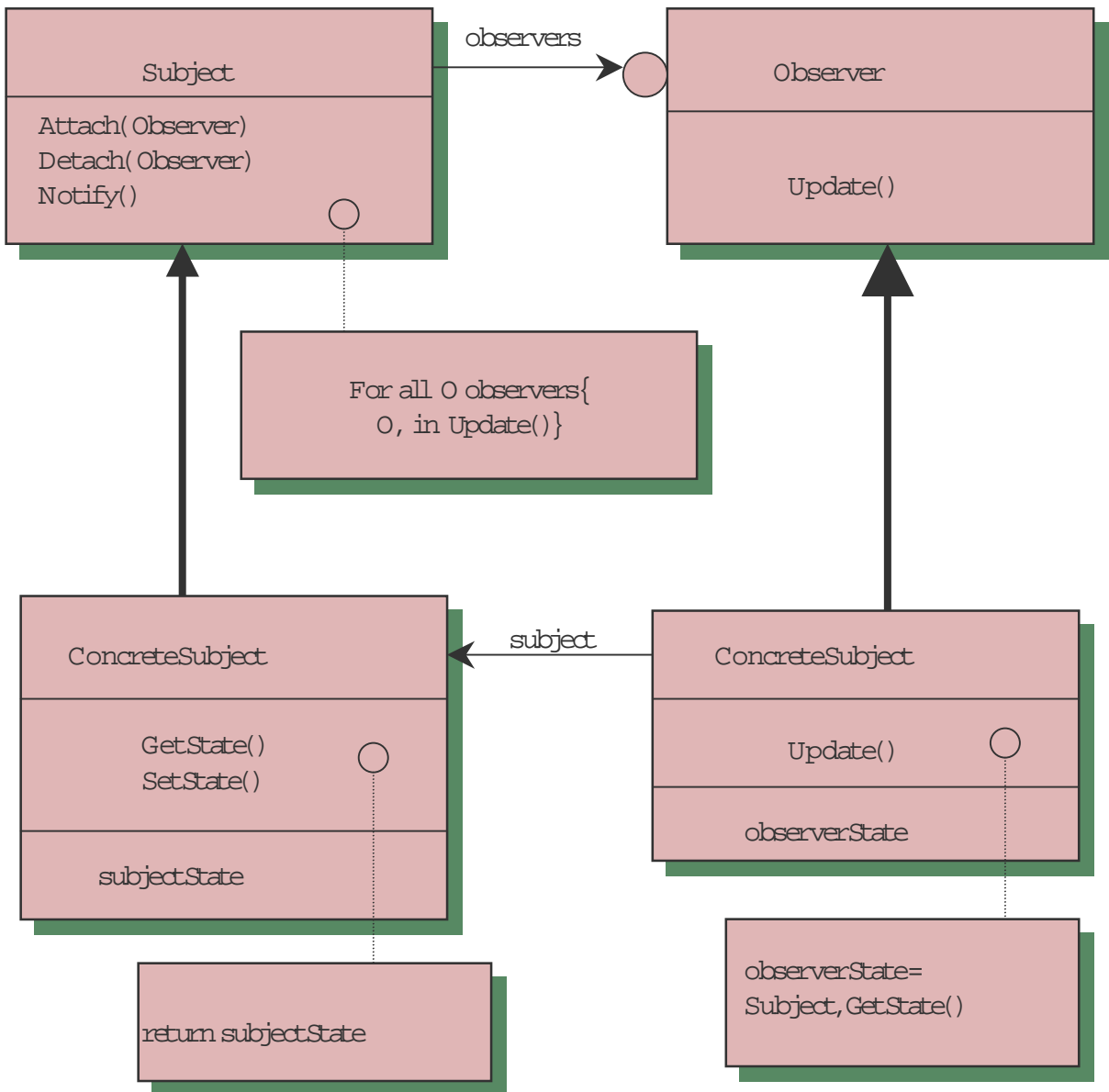
4.1.3 Διαμοίρασμα Συμβάσεων: Διασυνδέσεις

- Οι αυτοτελείς διεργασίες μίας διασύνδεσης μεταξύ πελάτη και εξυπηρετητή δεν έχουν νόημα από μόνες τους εκτός και αν πληρούν κάποιες συμβάσεις.
- Οι διασυνδέσεις και οι συνθήκες τους είναι λογικά οι μικρότερες οντότητες σε ένα σύστημα. Οι διεργασίες διασύνδεσης δεν έχουν νόημα από μόνες τους. Όλες μαζί συνθέτουν τα θεμέλια για αλληλεπίδραση πελάτη και εξυπηρετητή.

4.1.4 Διαμοίρασμα Αρχιτεκτονικής Αλληλεπίδρασης: Πρότυπα.

- Η συστηματική ταξινόμηση της μικρότερης επαναλαμβανόμενης αρχιτεκτονικής σε αντικειμενοστρεφή λογισμικά έχει καταλήξει στην ταξινόμηση προτύπων σύμφωνα με τα ακόλουθα τέσσερα στοιχεία.
 - Όνομα προτύπου.
 - Το πρόβλημα που λύει το πρότυπο περιλαμβανομένων και των συνθηκών εφαρμογής του προτύπου.
 - Η λύση του προβλήματος:
 - Παράγοντες και ο ρόλος τους, ευθύνες, σχέσεις και συνεργασίες.
 - Ο γενικός σχεδιασμός και η εφαρμογή.
 - Επακόλουθα:
 - Αναλογίες χώρου και χρόνου.
 - Θέματα γλώσσας και εφαρμογής.
 - Επενέργειες στην ελαστικότητα (*flexibility*), επέκταση (*extensibility*) και φορητότητα (*portability*).
- Ένα από τα ταξινομηθέντα πρότυπα είναι το *Observer* (βλ. σχ. 41) που ορίζει μία-προς-πολλές εξάρτηση αντικειμένων: όταν ένα αντικείμενο αλλάζει όλες οι εξαρτήσεις ενημερώνονται για την αλλαγή. Το πρότυπο παρουσιάζεται στο διάγραμμα που ακολουθεί. Η ιδέα είναι ότι αν το πρόβλημα απομονωθεί, το κατάλληλο πρότυπο μπορεί να επιλεγεί. Αν δηλαδή οι συνθήκες στις οποίες το πρότυπο προσαρμόζεται απαιτούν όπως ένα υποκείμενο (*Subject*) πρέπει να παρατηρηθεί από *N* παρατηρητές (*observers*) τότε το πρότυπο *Observer* επιλέγεται, η δε μέθοδος *Update()* μπορεί να χρειάζεται επιπλέον στοιχεία για να πληροφορήσει τους παρατηρητές για τις αλλαγές.
- Τα πρότυπα είναι μικροαρχιτεκτονικές που δείχνουν την αλληλεπίδραση μεταξύ αντικειμένων στη λύση ενός προβλήματος. Σε σύγκριση με τις υποδομές είναι:
 - Περισσότερο θεωρητικά και λιγότερο εξειδικευμένα (οι υποδομές είναι εφαρμογές υποσυστημάτων ενώ τα πρότυπα δεν έχουν καμμία άμεση εφαρμογή).
 - Μικρότερα αρχιτεκτονικά στοιχεία από τις υποδομές (μερικά πρότυπα υπάρχουν από το διαμελισμό συγκεκριμένων μεθόδων).

Διαμοίρασμα Αρχιτεκτονικής Αλληλεπίδρασης: Πρότυπα



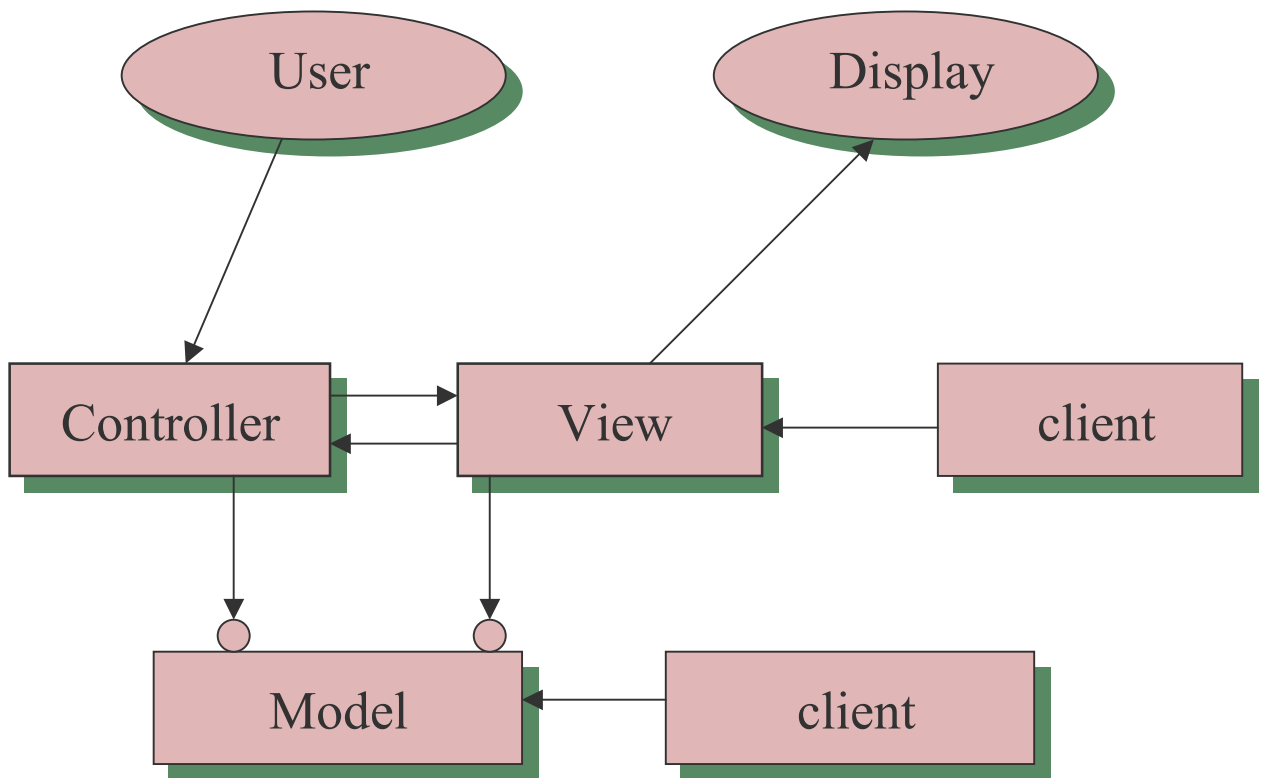
Σχ. 4.1 Το Πρότυπο Observer

4.1.5 Διαμοίρασμα Υποσυστημάτων Αρχιτεκτονικής : Υποδομές

- Οι υποδομές είναι σύνολα συνεργαζόμενων τάξεων που συναρμολογούν ένα επαναχρησιμοποιούμενο συστατικό.
- Δεν εστιάζονται κατ' ανάγκη σε συγκεκριμένες περιοχές εφαρμογών, έχουν όμως συγκεκριμένες έννοιες εφαρμογών, παρόλο που η λειτουργικότητα των τάξεων μπορεί να παραμένει άγνωστη.
- Οι τάξεις πρέπει να είναι ανοικτές για διαδοχή εφαρμογών (υποτάξεις). Παροχή αυτόματων επιλογών για ελάττωση του φόρτου ελαφριάς χρήσης μίας υποδομής.
- Οι παραδοσιακές υποδομές ασχολούνται πλήρως με τις τάξεις και τη διαδοχή τους.
 - Μία εφαρμογή διαδοχής τάξεων τείνει να βασίζεται στη γνώση της λειτουργικότητας των υπερτάξεων και γι' αυτό καλείται διαφανής (whitebox) επαναχρησιμοποίηση.
 - Αντίθετα η σύνθεση αντικειμένων (τάξεων), που βασίζεται στη μεταβίβαση, απλά στηρίζεται στις διασυνδέσεις των εμπλεκόμενων αντικειμένων και γι' αυτό καλείται αδιαφανής (blackbox) επαναχρησιμοποίηση.
- Ο ρόλος των υποδομών: Ρυθμίζουν την αλληλεπίδραση των εμπλεκόμενων τάξεων. Με την παγοποίηση ορισμένων αποφάσεων σχεδιασμού στην υποδομή, οι κρίσιμες πλευρές των διεργασιών μπορούν να σταθεροποιηθούν. Έτσι επιταχύνεται η δημιουργία συγκεκριμένων λύσεων.
- Εφαρμογές με πολλά πρότυπα σε μία υποδομή: το πρότυπο Model View Controller (MVC) μπορεί να διαχωρισθεί στους εξής ρόλους-πρότυπα:
 - Παρατηρητής (Observer).
 - Σύνθετος (Composite).
 - Στρατηγική (Strategy).
- Σ' αυτό το πρότυπο υπάρχουν τρεις βασικές λειτουργικότητες. Η αντιπροσώπευση πληροφοριών (Model), η παρουσίαση πληροφοριών (View) και η ερμηνεία των χειρισμών (Controller). Πολλές παρουσιάσεις όπως text outline, text preview, word count, Flesch reading Ease score, graphics, scrolling- panning- zooming views, κ.α. μπορούν να συσχετισθούν με μία και μόνο αντιπροσώπευση ενώ κάθε παρουσίαση έχει τη δική της ερμηνεία χειρισμών (βλ. σχ. 2).

Διαμοίρασμα Αρχιτεκτονικής Υποσυστημάτων: Υποδομές

- Ο μικρός αριθμός τάξεων (Model, View, Controller) θα οδηγούσε στο συμπέρασμα ότι το MVC είναι πρότυπο και όχι υποδομή. Το πραγματικό MVC είναι πολυπλοκότερο και το πρότυπο Observer συσχετίζει τα πρότυπα Model και View. Το πρότυπο Strategy βοηθά το Controller να ερμηνεύσει τα γεγονότα χρηστών σε διεργασίες του View.
- Μία υποδομή αποτελεί ένα υψηλότερο επίπεδο αρχιτεκτονικής αλλά και υποδομής που ολοκληρώνει χαμηλότερου επιπέδου έννοιες όπως τα πρότυπα. Ο βαθμός επιβολής κανόνων ολοκλήρωσης εξαρτάται από την τεχνολογία υποδομής που χρησιμοποιείται. Αν για παράδειγμα είναι τάξεις με Smalltalk τότε δεν υπάρχουν σημαντικοί περιορισμοί. Αν η υποδομή περιορίζεται από κλειστούς διαμορφωτές ή πακέτα με απρόσιτες εφαρμογές τότε υπάρχουν αρκετοί περιορισμοί ολοκλήρωσής της.



Σχ. 4.2 Η Υποδομή MVC

Διαμοίρασμα Αρχιτεκτονικής Υποσυστημάτων: Υποδομές

- Τρόποι σχεδιασμού υποδομών:
 - Από τη βάση στην κορυφή οδηγούμενοι από τεκμηριωμένα πρότυπα.
 - Από την κορυφή προς τα κάτω οδηγούμενοι από καλά καθορισμένους στόχους. Οι στόχοι περιέχουν αλληλοεπιδράσεις και οντότητες. Οι περιοχές των υποδομών περιέχουν τεχνικές ενώ οι στόχοι παρέχουν λύσεις σε πραγματικές εφαρμογές.

4.1.6 Διαμοίρασμα ολικής δομής: αρχιτεκτονικές

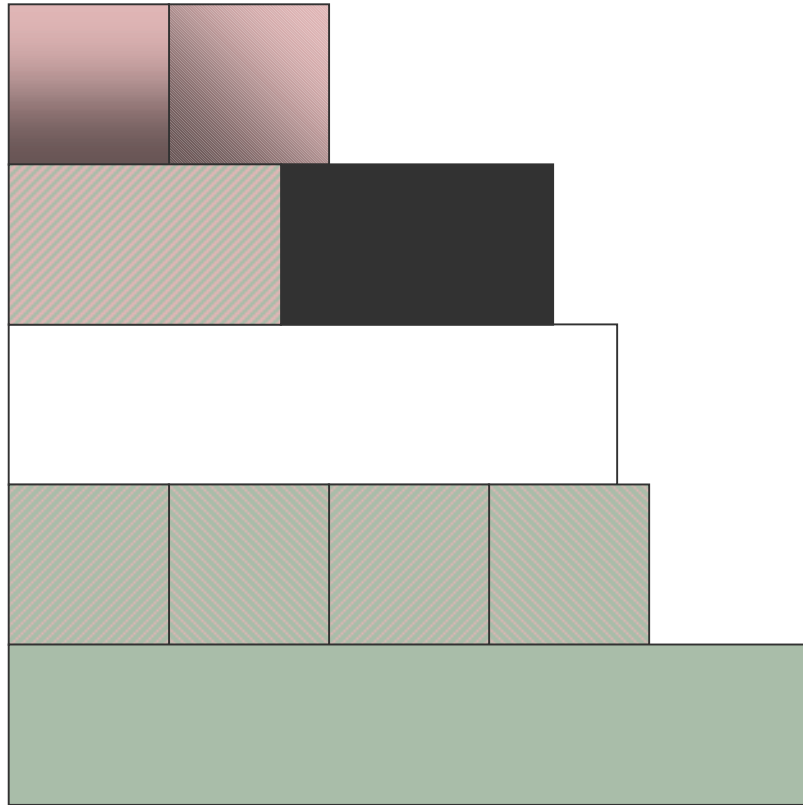
- Βασικές αρχές επιτυχημένων αρχιτεκτονικών:
 - Επίπεδα, αστηρά και μη.
 - Ιεραρχίες και ετεραρχίες.
- Σ' ένα αστηρά διαβαθμισμένο σύστημα κάθε επίπεδο στηρίζεται στις λειτουργίες του αμέσως χαμηλότερου επιπέδου. Οι λειτουργίες κάθε επιπέδου γίνονται εύκολα κατανοητές από τις διασυνδέσεις του ανώτερου και του χαμηλότερου επιπέδου (βλ. σχ 4.3).
- Όμως αυτή η εξάρτηση των επιπέδων από το πιο κάτω επίπεδο περιορίζει σημαντικά τη δυνατότητα επεκτάσεων στο συγκεκριμένο επίπεδο, τη στιγμή που κάποιο άλλο πιο χαμηλό επίπεδο θα μπορούσε να υποστηρίξει την αντίστοιχη επέκταση. Γι' αυτό κάθε επίπεδο ξεχωριστά πρέπει να είναι ανοικτό σε επεκτάσεις. Αυτό μπορεί τελικά να οδηγήσει και στην τακτική της απ' ευθείας πρόσβασης σε ακόμη χαμηλότερα επίπεδα (δηλ. χωρίς ενδιάμεσες διασυνδέσεις, βλ. σχ. 4.4).
- Τέτοια λογική πρόσβασης έχει και η χαλαρή διαβάθμιση, στην οποία όμως κάθε επέκταση στηρίζεται σε ένα και μόνο επίπεδο. Το κύριο μειονέκτημα είναι ότι χάνεται η συνοχή (και ευκολία πρόσβασης) των εφαρμογών του κάθε επιπέδου (βλ. σχ. 4.5).

Διαμοίρασμα ολικής δομής: αρχιτεκτονικές



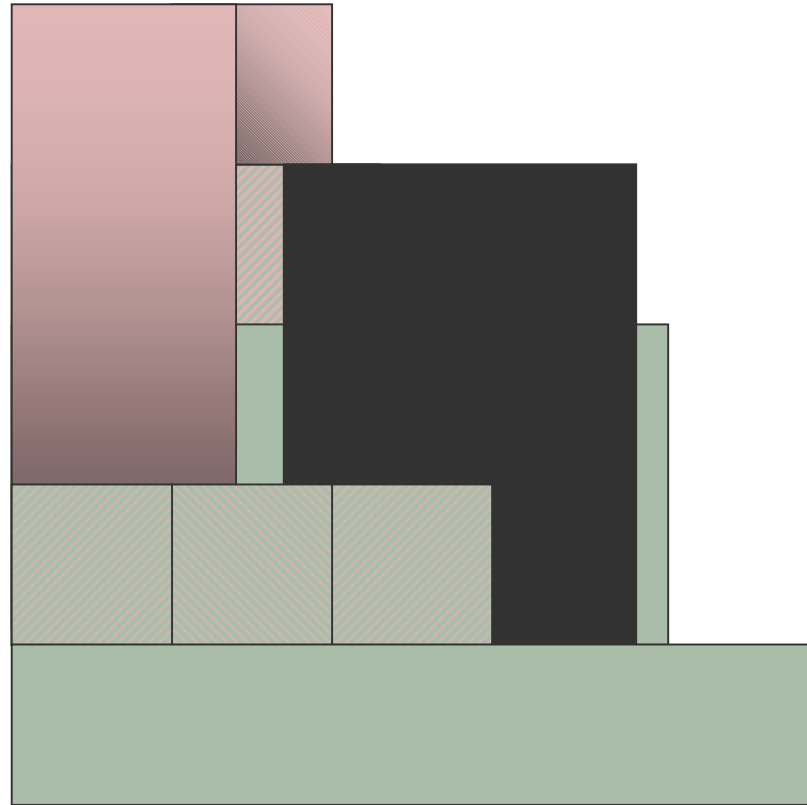
Σχ. 4.3 Αυστηρή αρχιτεκτονική διαβάθμιση.

Διαμοίρασμα ολικής δομής: αρχιτεκτονικές



Σχ. 4.4 Επεκτάσιμη αυστηρή διαβάθμιση.

Διαμοίρασμα ολικής δομής: αρχιτεκτονικές



Σχ. 4.5 Χαλαρή διαβάθμιση.

4.1.7 Συστήματα και Υποσυστήματα – Ιεραρχίες Υποδομών

- Τα πρότυπα είναι μικροαρχιτεκτονικές.
- Οι υποδομές είναι υποσυστήματα αρχιτεκτονικών. Τα περισσότερα μεγάλα συστήματα αποτελούνται από πολλαπλές υποδομές. Στις πολλαπλές αυτές υποδομές βρίσκουμε τις υποδομές τάξεων και σε πίο πάνω ιεραρχικό επίπεδο τις υποδομές υποσυστήματος. Η κάθε υποδομή στο επίπεδο του υποσυστήματος αποτελείται από προκατασκευασμένα συστήματα, που μερικώς συνδυάζουν υποσυστήματα-κλειδιά. Τελικά στα διάφορα επίπεδα ιεράρχισης, υπάρχουν οι ανάγκες των υψηλότερων επιπέδων (όπως νέοι οργανικοί ρόλοι) που πρέπει να ταιριάζουν με τις συγκεκριμένες λεπτομέρειες που διεκπεραιώνουν τα χαμηλότερα επίπεδα (όπως η σειρά εκτέλεσης). Η λύση είναι η δημιουργία μίας δεύτερης τάξης υποδομών που η δομή της ταιριάζει με τις παραδοσιακές υποδομές των τάξεων.

4.2 Διαλειτουργικότητα, εξέλιξη και ανακατασκευή

- Τα συστήματα στη γένεσή τους συνήθως παρουσιάζουν μία συνεκτική αρχιτεκτονική με καθαρούς στόχους στα χαμηλότερα επίπεδα λειτουργικότητας. Στη εξέλιξή τους όμως η αρχιτεκτονική αποσαφινίζεται και στην προσπάθεια υιοθέτησης νέας τεχνολογίας το σύστημα χρειάζεται ανακατασκευή.
- Ένα καθαρό προτέρημα των αρχιτεκτονικών που αποτελούνται από συστατικά είναι η υποστήριξη σε ανακατασκευές συγκεκριμένων περιοχών. Επιπλέον, η ιεραρχική αρχιτεκτονική επιτρέπει την επιλεκτική ανακατασκευή των υποσυστημάτων. Η σύζευξη πρέπει να αποφεύγεται ιδιαίτερα στα υψηλά επίπεδα των υποσυστημάτων.
- Μία διαφορετική αλλά σχετική προοπτική είναι της διαλειτουργικότητας δύο ξεχωριστών συστημάτων. Στο εννοιολογικό αυτό επίπεδο απαιτούνται κοινοί ορισμοί. Στο υποδομικό αυτό επίπεδο απαιτούνται κοινές διασύνδεσεις.
- Τελικά σαν συμπέρασμα πρέπει να πούμε ότι η εισαγωγή κάθε νέας τεχνολογίας προϋποθέτει ικανοποιητική αρχιτεκτονική προσέγγιση όλων των σχετικών επιπέδων.