

**Κεφάλαιο Δεκαέξι**  
-  
**Συναρμολόγηση Συστατικών**

## 16.1 Εικονική Συναρμολόγηση Συστατικών

- Η εικονική συναρμολόγηση περιπτώσεων συστατικών είναι ένας τρόπος απλοποίησης της διεργασίας συναρμολόγησης. Στη JavaBeans για παράδειγμα κάθε bean εκδηλώνει συμπεριφορά που βοηθά στην διασύνδεση περιπτώσεων και καθοδηγεί στη συναρμολόγηση. Τα συστατικά παίρνουν αρχικές τιμές και οι περιπτώσεις συνδέονται με ομοιόμορφη προσέγγιση σε αντικείμενα με `outgoing` και `incoming` διασυνδέσεις.
- Επιπλέον συμπεριφορά προστίθεται με κώδικα-διεργασία που διαταυρώνεται με ένα συμβάν στη διαδρομή αποστολής του και πυροδοτεί ειδικές δραστηριότητες. Τέτοιος αφοσιωμένος κώδικας υπάρχει στη Visual Basic. Στη Borland's Delphi και C++ Builder υπάρχει επίσης διάκριση μεταξύ του κτίζω και χρησιμοποιώ μία εφαρμογή.

## 16.2 Τα Σύνθετα Κείμενα Παραγκωνίζουν την Εικονική Συναρμολόγηση

- Σε σύνθετα κείμενα η ολοκλήρωση της σύνθεσης και της χρήσης είναι ευθύς και φυσική. Τα κείμενα είναι εφαρμογές και η έκδοσή τους συναρμολόγηση περιπτώσεων συστατικών. Δεν πρέπει να υπάρχει χάσμα μεταξύ χρήσης και σύνθεσης. Η μετάβαση εξομαλίνεται
  - Χρησιμοποιώντας συναρμολόγηση συστατικών.
  - Με αργή συναρμολόγηση επιπλέον περιπτώσεων.
  - Με τη δημιουργία κώδικα επιπλέον συναρμολογήσεων.
- Όλα αυτά μπορούν να συνδυασθούν αναλόγως των αναγκών των εφαρμογών. Αυτή την προσέγγιση ακολουθεί και η BlackBox. Δεν υπάρχει οξεία διάκριση μεταξύ συναρμολόγησης και χρήσης, παρόλο που θα μπορούσε να υπάρξει με την απουσία συστατικών σύνθεσης.
- Η ασύνδετη ολοκλήρωση περιβάλλοντος σύνδεσης και χρήσης, ειδικά σε σύνθετα κείμενα, είναι αποτελεσματική στην *γρήγορη ανάπτυξη εφαρμογών* (rapid application development - RAD).

### 16.3 Συστατικά Πέρα από τις Γραφικές Διασυνδέσεις Χρηστών

- Η ανάπτυξη τεχνολογίας συστατικών για εξυπηρετητές και άλλα μη αλληλεπιδρώντα συστήματα, έχει μείνει πολύ πιο πίσω από αυτή των αλληλεπιδρόντων συστημάτων και ειδικά αυτής των σύνθετων κειμένων και παρατήρησης διαδουκτίου.
- Για συστατικά βασισμένα σε εξυπηρετητές, ο διαχωρισμός σε χρόνο σύνθεσης και χρήσης είναι φυσικός.
  - Στο σύστημα New World Infrastructure (NewI) υπάρχουν προτάσεις για 'components everywhere' .
  - Τα 'servlets' είναι συστατικά της Java που έχουν σχεδιασθεί για να λειτουργούν σε εξυπηρετητή και μπορούν να συναρμολογηθούν οπτικά. Ενώ εναρμονίζονται καλά με σύγχρονα πρότυπα όπως η CORBA , η συναρμολόγηση πριν τη χρήση απαιτεί προκαταρτικές αποφάσεις ως προς το ποιες περιπτώσεις συστατικών θα τοποθετηθούν που, στο κατανεμημένο σύστημα. Στα περισσότερα συστήματα όπως και στις εφαρμογές CORBA , υπάρχει λίγη ή καθόλου υποστήριξη για μετανάστευση αντικειμένων που να διατηρεί τις υπάρχουσες αναφορές τους.
- Άλλο παράδειγμα κριτηρίων σύνθεσης συστατικών που δεν στοχεύει σε χρήση γραφικού περιβάλλοντος είναι το OPC (OLE for Process Control), με κριτήρια διασύνδεσης για οδηγούς συσκευών στην αυτοματοποίηση εργοστασίου.

### 16.4 Διαχειρισταία και 'Αυτο-οδηγούμενη' Συναρμολόγηση συστατικών

- Η συναρμολόγηση των συστατικών αφορά τις περιπτώσεις τους. Σε αυτήν ο κώδικας και οι πηγές που εφαρμόζουν τα συστατικά μένουν χωριστά. Ελαφρής κώδικας συνδέει τις περιπτώσεις, ενώ υπάρχει περισσότερος προγραμματισμός των ίδιων των συστατικών.
- Καθώς η συναρμολόγηση εστιάζεται στις περιπτώσεις μπορεί να ασχοληθεί μόνο με περιπτώσεις προαποφασισμένες τη στιγμή της συναρμολόγησης. Έτσι αποβάλλει την ικανότητα του λογισμικού να δημιουργεί κατά βούληση οποιοδήποτε αριθμό περιπτώσεων. Στην επανάληψη χρησιμοποίησης των ίδιων σχηματισμών από περιπτώσεις, τότε γίνεται αντιγραφή προτύπων συναρμολογήσεως. Τέτοιες εφαρμογές υπάρχουν στα συστήματα σύνθετου κειμένου με τη χρησιμοποίηση εντύπων προτύπου.

## Διαχειρισταία και 'Αυτο-οδηγούμενη' Συναρμολόγηση συστατικών

- Αν πρέπει να δουλεύουμε με δυναμικές περιπτώσεις συστατικών με απρόβλεπτους σχηματισμούς, οι υπόλοιπες πιθανότητες είναι διαχειρισταίες και αυτο-οδηγούμενες συναρμολογήσεις. Οι πρώτες βασίζονται στην αυτόματη συναρμολόγηση συστατικών που εφαρμόζει την πολιτική της δυναμικής συναρμολόγησης περιπτώσεων. Στην αυτο-οδηγούμενη συναρμολόγηση οι κανόνες περιέχονται στα ίδια τα συστατικά.

### 16.5 Η Εξέλιξη των Συστατικών

- Η τεχνολογία συστατικών είναι αργή συναρμολόγηση με αναβολή περαιτέρω διασυνδέσεων. Ταυτόχρονα η ευπάθεια του όλου συστήματος μεγαλώνει. Η τάση για συγκρούσεις εκδόσεων μειώνεται με τη γρήγορη εφαρμογή των νέων συστατικών.
- Σε κατανεμημένα συστήματα δεν μπορούν να αποσύρονται οι τρέχουσες περιπτώσεις για εγκατάσταση μίας νέας έκδοσης συστατικού. Η δυαδική διαλειτουργικότητα πελατών διαφορετικών εκδόσεων και περιπτώσεων πρέπει να προσχεδιάζεται.
- Σε πραγματικές ρυθμίσεις αναμένεται ότι διάφορες εκδόσεις του ίδιου συστατικού θα συνυπάρχουν στο ίδιο σύστημα. Η μετάβαση από τη μία γενιά του συστήματος στην επόμενη είναι το σημαντικό παράδειγμα. Η προσαρμογές γίνονται με *συστατικά περιτύλιξης* (wrapper components).
- Την καλύτερη προσέγγιση στα πιο πάνω την έχει η COM. Οι διασυνδέσεις της COM είναι αμετάβλητες χωρίς προβλήματα έκδοσης. Κάθε νέα έκδοση συστατικού πρέπει να χρησιμοποιεί νέα διασύνδεση. Το κυρίως πλεονέκτημα έγκειται στην δυνατότητα ταυτόχρονης στήριξης της παλιάς διασύνδεσης με τα παλιά σημασιολογικά αντικείμενα και της νέας διασύνδεσης με τα νέα σημασιολογικά αντικείμενα.
- Η προσέγγιση της CORBA στο θέμα είναι κατώτερη στο ότι πρέπει να συγχωνεύει όλες τις διεργασίες όλων των εκδόσεων σε μία διασύνδεση. Έτσι για κάθε αλλαγή των σημασιολογικών μίας διεργασίας πρέπει να αλλάζει και ο ενδείκτης της.
- Αν ένα αντικείμενο δημιουργεί άλλο αντικείμενο η μεθοδολογία εκδόσεων της CORBA επιτρέπει την επιβεβαίωση ότι ο εξυπηρετητής που υποστηρίζει το νέο αντικείμενο είναι κατάλληλης έκδοσης. Όταν όμως δημιουργηθεί το αντικείμενο, οι αναφορές σε αυτό πρέπει να μεταδίδονται χωρίς επιπλέον έλεγχο έκδοσης. Αυτό όμως στην πραγματικότητα είναι ανεπαρκές. Στη COM το πρόβλημα λύνεται με διαφύλαξη διασυνδέσεων διαφορετικών εκδόσεων ξεχωριστά. Αναφορά στη διασύνδεση ενός συστατικού έχει

## Η Εξέλιξη των Συστατικών

αναλλοίωτα σημασιολογικά. Μία παρόμοια λύση από τη CORBA θα ήταν δυνατό στη βάση των CORBA 2.0 RepositoryID's που εξυπηρετούν σαν μοναδικοί ενδείκτες διασυνδέσεων. Αυτό δεν υπάρχει ακόμη στον κώδικα του πελάτη.

- Η Java δεν έχει ειδικό μηχανισμό για εκδόσεις. Αδυνατεί να χειριστεί προβλήματα συνύπαρξης πελατών και προμηθευτών στη βάση διαφορετικών εκδόσεων της ίδιας διασύνδεσης. Μία μερική λύση βρίσκεται στο Java Object Serialization Service.
- Εν τούτοις η Java ορίζει εξεζητημένους κανόνες δυαδικής συμβατότητας. Η αξιοπιστία όμως μερικών είναι αμφισβητήσιμη όπως στην περίπτωση σταθερών σε διασυνδέσεις που σκοπό έχουν την παραμετροποίηση ορισμένων μεθόδων της διασύνδεσης. Στη μεταγλώττιση τάξης για ένα πελάτη αυτές αντιγράφονται στο αρχείο τάξης χωρίς εγγραφή εξαρτήσεων έκδοσης. Αν η διασύνδεση ανεθεωρηθεί και οι σταθερές επαναορισθούν ή απαλειφθούν, οι προηγούμενα μεταγλωττισμένες τάξεις θα συνεχίσουν τη χρήση των παλαιών σταθερών και θα τις περάσουν στις μεθόδους της νέας διασύνδεσης. Η επαναμεταγλώττιση τέτοιας τάξης πελάτη οδηγεί στην δυναμική αλλαγή συμπεριφοράς.