Άσκηση 2

Μιχαήλ Μυλωνάκης

23 Νοεμβρίου 2015

Εμπειρία χρήσης της AspectJ

1. *Πόσο εύκολη ήταν η δουλειά σας;*

Σε γενικές γραμμές, η δημιουργία Aspects μέσω της AspectJ ήταν αρκετά απλή. Φυσικά, βασικός παράγοντας της ευκολίας της συγγραφής των Aspects ήταν και το γεγονός ότι το cross-cutting functionality που θέλαμε να προσθέσουμε ήταν απλό (έπρεπε μόνο να συγχρονίσουμε τα reads και τα writes σε μία stack. Δαπανώντας λοιπόν τον απαιτούμενο χρόνο για να εξοικειωθούμε με αυτό το νέο μοντέλο προγραμματισμού, η δουλειά μας ήταν εύκολη.

2. *Πόσο ταίριαζε η AspectJ για το συγκεκριμένο σκοπό και πώς πιστεύετε ότι γενικεύεται η εμπειρία σας σε άλλες περιστάσεις;*

Η εισαγωγή ταυτοχρονισμού σε μία υλοποίηση στοίβας αποτελεί ένα καλό παράδειγμα use case της AspectJ. Ο συγχρονισμός (και γενικότερα το transaction management αποτελούν έννοιες που είναι υπεράνω αντικειμένων, έχουν δηλαδή cross-cutting εφαρμογή, και κατ΄ επέκταση είναι προτιμότερο να υπάρχει κάπου κεντρικά η λογική (ο κώδικας) που τα επιβάλλει, παρά να υπάρχει διάσπαρτη στο σώμα των μεθόδων των αντικειμένων. Η επανάληψη του ίδιου κώδικα σε πολλά μέρη είναι error-prone. Στο συγκεκριμένο use case για παράδειγμα, ο κώδικας που θα έπρεπε να προστεθεί στις συναρτήσεις pop και push θα ήταν πανομοιότυπος. Γενικότερα θα ταίριαζε η χρήση του AspectJ για λειτουργικότητες όπως το Logging - Tracing - όπου θα μπορούσαμε να εμφανίζουμε μηνύματα ανάλογα με τις κλήσεις μεθόδων που λαμβάνουν χώρα, για το Testing - όπου θα μπορούσαμε να αντικαταστήσουμε κάποιες κλήσεις της new με mocks, ή για κάποιο απλό Security Policy. Η ακόμη θα μπορούσαμε να αλλάξουμε ένα μέρος της λειτουργικότητας μιας μεθόδου βιβλιοθήκης στην οποία δεν θα είχαμε πρόσβαση στον πηγαίο κώδικα (γενικότερα όπου δεν μπορούμε ή δεν θέλουμε να πειράξουμε κάποιο API. Φυσικά η παραπάνω λίστα δεν είναι εξαντλητική.

3. *Θα χρησιμοποιούσατε την AspectJ αν ήταν διαθέσιμη ευρύτατα;*

Αν η λειτουργικότητα που θα θέλαμε να επιβάλλουμε ήταν διασκορπισμένη στο σύνολο των κλάσεων του προβλήματός μας (όπως για παράδειγμα τα use cases που αναφέρθηκαν παραπάνω), και δεδομένου ότι το overhead που προκύπτει από την χρήση της είναι μάλλον μικρό, τότε ναι η AspectJ θα ήταν μια καλή λύση. Θέλει ωστόσο προσοχή ως προς την υπερβολική χρήση της, καθώς όσο πιο πολύπλοκα γίνονται τα Aspects, τόσο πιο δυσανάγνωστο γίνεται το πρόγραμμά μας, και κατ΄ επέκταση αντί να καταφέρουμε μια πιο συμπαγή υλοποίηση, με όλη την λογική συγκεντρωμένη σε ένα ΄μέρος΄, και άρα ένα πιο ευανάγνωστο πρόγραμμα, το αποτέλεσμα μπορεί να είναι το αντίθετο. Όσο λοιπόν μας βοηθάει, και κάνει το πρόγραμμά μας πιο απλό δίνοντάς μας την δυνατότητα να κρατήσουμε κάποια cross-cutting λειτουργικότητα συγκεντρωμένη σε ένα Aspect, η χρήση της είναι θεμιτή. Στο σημείο που τα Aspect φορτώνονται με πολύ - δυσνόητο κώδικα, μάλλον πιο πολύ θα μας δυσκολέψει πάρα θα βοηθήσει.