



ΑΛΕΞΑΝΔΡΕΙΟ Τ.Ε.Ι. ΘΕΣΣΑΛΟΝΙΚΗΣ
ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ



ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

**Ανάπτυξη Ευέλικτων Διαδικτυακών Εφαρμογών για
Επιχειρήσεις και Οργανισμούς με εργαλεία ανοιχτού
κώδικα – Το εργαλείο Eclipse**



Της φοιτήτριας
Παπαδημητρίου Ιωάννας
Αρ. Μητρώου: 02/2007

Επιβλέπων καθηγητής
Σφέτσος Παναγιώτης

ΠΕΡΙΛΗΨΗ

Καθώς οι δικτυακές εφαρμογές λαμβάνουν το επίκεντρο των σημερινών επιχειρήσεων, για να αποκτήσουν ανταγωνιστικό πλεονέκτημα στη συνεχή μεταβαλλόμενη αγορά, βασικές επιχειρησιακές διαδικασίες είναι όλο και περισσότερο αυτοματοποιημένες με τη χρήση διαδικτυακών τεχνολογιών. Έτσι είναι απαραίτητη μια ευέλικτη προσέγγιση στο σχεδιασμό, την κατασκευή, και τη διαχείριση συστημάτων πληροφοριών για τις επιχειρήσεις, αφού η ευελιξία, έχει σαν βάση της την προσαρμογή και όχι την σταθερή τυποποίηση. Οι ευέλικτες διαδικτυακές εφαρμογές επιτρέπουν από τη φύση τους την επεκτασιμότητα, αλλά και ανοχή στα λάθη, και ενθαρρύνουν τις απαιτήσεις των αλλαγών, κάτι το οποίο είναι επιθυμητό χαρακτηριστικό για την ανάπτυξη ευέλικτων διαδικτυακών εφαρμογών σε επιχειρήσεις και οργανισμούς.

Στα παρακάτω κεφάλαια ακολουθεί μία βιβλιογραφική ανασκόπηση ευέλικτων διαδικτυακών εφαρμογών, μέσω πειραμάτων και ερευνητικών μελετών, περιγραφή του κύκλου ζωής των ευέλικτων διαδικτυακών εφαρμογών, περιγραφή και ανάλυση του εργαλείου Eclipse και σημαντικών έργων που έχουν δημιουργηθεί μέσω αυτού, και τέλος ανάπτυξη μίας απλής διαδικτυακής εφαρμογής για την παρουσίαση του εργαλείου Eclipse.

ΠΕΡΙΕΧΟΜΕΝΑ

Περίληψη	3
Περιεχόμενα.....	5
ΚΕΦΑΛΑΙΟ 1: Εισαγωγή.....	9
ΚΕΦΑΛΑΙΟ 2: Σχετική Έρευνα (Related Work).....	11
2.1.: Παρουσίαση πίνακα.....	18
2.2: Επίλογος.....	21
ΚΕΦΑΛΑΙΟ 3: Ο κύκλος ζωής των Ευέλικτων Διαδικτυακών Εφαρμογών..	23
3.1: Εισαγωγή.....	23
3.2: Μεθοδολογία υλοποίησης Ευέλικτων Διαδικτυακών Εφαρμογών	24
3.2.1: Η φάση των Απαιτήσεων.....	26
3.2.2: Η φάση της Ανάλυσης.....	29
3.2.3: Η φάση του Σχεδιασμού.....	32
3.2.4: Η φάση της Κωδικοποίησης.....	34
3.2.5: Η φάση του Ελέγχου.....	35
3.2.6: Η φάση της Ανάπτυξης.....	37
3.3: Επίλογος.....	38
ΚΕΦΑΛΑΙΟ 4: Το Εργαλείο Eclipse.....	39
4.1: Εισαγωγή.....	39
4.1.1: Ιστορικά.....	39
4.1.2: Αρχιτεκτονική της Πλατφόρμας.....	40
4.1.3: Ανοιχτή Αρχιτεκτονική.....	41
4.1.4: Αρχική μορφή της πλατφόρμας.....	41

4.1.5: Η δομή της πλατφόρμας.....	42
4.1.5.1 Χώροι Εργασίας (Workspaces).....	43
4.1.5.2 Workbench και UI Toolkits (SWT και JFace).....	43
4.1.5.3 Υποστήριξη Ομαδικής Εργασίας (Team Support).....	45
4.1.5.4 Μηχανισμός Βοήθειας (Help).....	46
4.1.6: Επέκταση της πλατφόρμας με plug-ins.....	46
4.1.7:Επίλογος.....	47
4.2: Εργαλεία και πλαίσια OSGI.....	48
4.2.1: Desktop Εφαρμογές.....	48
4.2.2: Web Εφαρμογές.....	49
4.2.3: SOA Εφαρμογές.....	50
4.2.4: Enterprise Εφαρμογές.....	51
4.3: Pulsar Εργαλεία ανάπτυξης mobile εφαρμογών.....	53
4.4: Πλαίσια εφαρμογών και Εργαλεία Μοντελοποίησης.....	55
4.4.1 Eclipse Modeling Framework.....	55
4.4.2 Graphical Modeling Project.....	59
4.4.3 Model Development Tools.....	64
4.4.4 UML Plug-ins.....	65
4.5: Το έργο Mylyn.....	67
4.6:Το έργο Pulse.....	74
ΚΕΦΑΛΑΙΟ 5: Ανάπτυξη web Εφαρμογής.....	81

ΚΕΦΑΛΑΙΟ 6: ΒΙΒΛΙΟΓΡΑΦΙΑ.....107

ΚΕΦΑΛΑΙΟ 1

ΕΙΣΑΓΩΓΗ

Στην παρούσα Πτυχιακή Εργασία, περιγράφεται και μελετάται το εργαλείο Eclipse. Το Eclipse είναι μία κοινότητα ανοιχτού λογισμικού τα έργα της οποίας έχουν επικεντρωθεί στην οικοδόμηση μίας ανοιχτής πλατφόρμας ανάπτυξης, που αποτελείται από επεκτάσιμα πλαίσια και εργαλεία, για την ανάπτυξη και την διαχείριση του λογισμικού σε όλο τον κύκλο ζωής του.

Κύριος σκοπός της εργασίας είναι η πλήρης ανάλυση του εργαλείου και των σημαντικότερων έργων που έχουν αναπτυχθεί με αυτό, καθώς και η ανάπτυξη μίας Διαδικτυακής Εφαρμογής για την παρουσίαση του εργαλείου.

Στο Κεφάλαιο 2 ακολουθεί μία σχετική εργασία (*Related Work*) για τον εντοπισμό ερευνητικών μελετών, πειραμάτων και συλλογή εμπειρικών δεδομένων, όσον αφορά τις ευέλικτες διαδικτυακές εφαρμογές. Οι ευέλικτες διαδικτυακές εφαρμογές διακρίνονται από τις μέσες εφαρμογές, λόγω της ευκολίας τροποποίησης τους, καθώς επίσης και της επέκτασης του σχεδιασμού τους, για την αντιμετώπιση των μεταβαλλόμενων αναγκών των επιχειρήσεων και οργανισμών. Η έρευνα πραγματοποιήθηκε μέσω ηλεκτρονικών βιβλιοθηκών ορισμένες από τις οποίες είναι οι εξής: IEEE Xplore, ACM digital Library, SpringerLink, Kluwer Online, CiteseerX, και άλλες, με τις λέξεις κλειδιά: Agile Web Applications.

Στο Κεφάλαιο 3 Αναλύονται τα πλαίσια ανάπτυξης και ο κύκλος ζωής των ευέλικτων διαδικτυακών εφαρμογών. Ο κύκλος ζωής περιλαμβάνει όλα τα στάδια, από την σύλληψη της ιδέας για την πραγματοποίηση μίας ευέλικτης διαδικτυακής εφαρμογής, μέχρι την ανάπτυξη και την παράδοσή της.

Στη συνέχεια, στο Κεφάλαιο 4 αναλύεται το εργαλείο Eclipse, που είναι μία πλατφόρμα ανάπτυξης λογισμικού ανοιχτού κώδικα, η οποία διατίθεται δωρεάν, και αποτελείται από ένα IDE και ένα plug-in σύστημα, ώστε να είναι εύκολη η επέκτασή της. Ακολουθούν ορισμένα από τα σημαντικότερα έργα που έχουν αναπτυχθεί με το εργαλείο Eclipse, εργαλεία και πλαίσια OSGI για την υλοποίηση desktop, web, SOA και enterprise εφαρμογών, τα σημαντικότερα πλαίσια εφαρμογών και εργαλεία μοντελοποίησης που έχουν αναπτυχθεί στην πλατφόρμα,

καθώς και δύο πολύ σημαντικά έργα της κοινότητας Eclipse, το έργο Mylyn και το έργο Pulse, όσον αφορά την ανάλυση και χρήση του πλαισίου ανάπτυξης διεργασιών.

Τέλος στο Κεφάλαιο 5 ακολουθεί η ανάπτυξη μίας μικρής Web Εφαρμογής με στόχο την παρουσίαση του εργαλείου που περιγράφεται στο Κεφάλαιο 4, ακολουθώντας τα βήματα της ανάπτυξης και του κύκλου ζωής των ευέλικτων Διαδικτυακών Εφαρμογών που παρουσιάσθηκαν στο Κεφάλαιο 3.

ΚΕΦΑΛΑΙΟ 2

Σχετική Έρευνα (Related Work)

Στο άρθρο του Steffen Bartsch, του 2010, ενσωματώθηκε ένα κομμάτι του RBAC (*role – based access control*) μέσα στην ανάπτυξη της ευέλικτης διαδικασίας, για την βελτίωση της πολιτικής ασφάλειας. Για να διευκολυνθεί το έργο διαχείρισης της ασφάλειας, προτάθηκε ένας αλγόριθμος και ένα πρωτότυπο εργαλείο. Ο AI - based αλγόριθμος αλλαγής υποστήριξης, βοηθά να βρεθεί η πιο κατάλληλη τροποποίηση ενεργειών, σύμφωνα με την επιθυμητή αλλαγή, η οποία δίνεται στην πολιτική της περίπτωσης δοκιμής. Επίσης παρουσιάζεται ένα πρωτότυπο GUI για ειδικούς του domain ώστε να χρησιμοποιήσουν τον αλγόριθμο και να δώσουν αναφορά σε προ αποτελέσματα από μη ειδικούς ασφαλείας, που χρησιμοποιούν το εργαλείο σε διαδικτυακή εφαρμογή, σε μία επιχείρηση στον πραγματικό κόσμο. Τα πρώτα αποτελέσματα είναι πολλά υποσχόμενα, καθώς η αλλαγή θα επιτρέψει την μεγαλύτερη ολοκλήρωση των ειδικών στο αναπτυξιακό τμήμα πολιτικής άδειας της ανάπτυξης λογισμικού.

Στο άρθρο των Wookjin Lee, et al, του 2005 προτείνετε μια μεθοδολογία για την ευέλικτη και συστηματική ανάπτυξη της διαδικτυακής εφαρμογής. Η μεθοδολογία περιλαμβάνει διαδικτυακά μοντέλα, εργαλεία υποστήριξης, και μια ευέλικτη διαδικασία. Τα διαδικτυακά αυτά μοντέλα επεκτείνονται από την UML εστίαση, στη συμπεριφορά της διαδικτυακής εφαρμογής και επικεντρώνονται σε μια συμπεριφοριστική άποψη η οποία εκπροσωπείται με navigation model (NM) και component communication model (CCM). Η ευέλικτη διαδικασία έχει ως στόχο την ταχεία προσαρμοστικότητα στις αλλαγές των απαιτήσεων, λαμβάνοντας παράλληλα τα πλεονεκτήματα των δραστηριοτήτων μοντελοποίησης. Εργαλεία συμπεριλαμβανομένων Process Manager και Web Modeler έχουν τεθεί σε εφαρμογή και υποστηρίζουν την δραστηριότητα μοντελοποίησης, ώστε να καθοδηγήσει την ανάπτυξη κατά την εφαρμογή της διαδικασίας.

Ένα ελεγκτικό μοντέλο είναι υπό εκτέλεση. Στο μέλλον σχεδιάζετε μία εφαρμογή που θα υποστηρίζει μία συνολική αναπτυξιακή διαδικασία, και θα την επικυρώνει με την εφαρμογή σε πραγματικά μεγάλα έργα.

Στο άρθρο του Lee Keoo, του 2008, έγινε προσπάθεια κατασκευής αυτόματης δομής με βάση την μαθηματική εξίσωση, με διάφορα επίπεδα αφαίρεσης των απαιτήσεων. Ο προτεινόμενος μηχανισμός ασχολείται με την ευελιξία για τη διαχείριση των δυναμικών απαιτήσεων, και δείχνει πως μπορούν να αντιμετωπιστούν, όσον αφορά την ευελιξία και την αποδοτικότητα. Επίσης έχει προταθεί το μετά-μοντέλο διαδικασίας που ορίζει τη σύνδεση μεταξύ των μοντέλων συμπεριφοράς στην UML για την ευκολία διαχείρισης, επαναχρησιμοποίησης, καθώς και την ευελιξία επέκτασης. Όλα τα υλικά από το μετά-μοντέλο έχουν ρυθμιστεί και συμπεριλαμβάνουν προσαρμοσμένες και παραγόμενες συμπεριφορές δυναμικά κατά το χρόνο εκτέλεσης. Οι ροές των γεγονότων που εξάγονται από το μετά-μοντέλο, σημαίνουν την εμπειρία ή τις προτιμήσεις του χρήστη, και εκπροσωπούνται σε πολλά διαφορετικά επίπεδα αφαίρεσης ως ομαδοποιημένα, καθώς συγκεντρώνονται από τις απαιτήσεις των χρηστών. Επίσης, οι λειτουργικές απαιτήσεις, με τα διάφορα σενάρια των υπηρεσιών μπορούν να επεκταθούν σε πολύ συγκεκριμένες απαιτήσεις, ώστε να συνίστανται για καθοδήγηση, και για να επισημοποιηθούν με τη μορφή ενός προτύπου συμπεριφοράς στη UML.

Στο άρθρο των Hu Ran, et al, του 2008, βρέθηκε μια νέα διαδικτυακή μέθοδος ανάπτυξης για την ικανοποίηση της απαίτησης ότι το διαδικτυακό σύστημα εφαρμογής μπορεί άμεσα να λειτουργήσει, να έχει γρήγορη ανταπόκριση, και ταχεία προσαρμογή της αλλαγής του διαδικτυακού συστήματος εφαρμογής: την ευέλικτη διαδικτυακή ανάπτυξη με διαδικτυακό πλαίσιο (AWDWF). Το ISO/IEC9126 ποιοτικό μοντέλο χρησιμοποιείται για την ανάλυση της ποιότητας της AWDWF, και τα αποτελέσματα δείχνουν ότι αυξάνει την αποτελεσματικότητα της διαδικτυακής ανάπτυξης, και παράλληλα διασφαλίζει την ποιότητά της. Η ταχεία και υψηλής ποιότητας διαδικτυακή μέθοδος ανάπτυξης, συνδυάζει τα πλεονεκτήματα της ευέλικτης ανάπτυξης και της επαναχρησιμοποίησης του διαδικτυακού πλαισίου, και είναι μία εξαιρετική τεχνική για τα διαδικτυακά συστήματα εφαρμογών που απαιτούν τη γρήγορη εξυπηρέτηση, γρήγορη ανταπόκριση, και την προσαρμογή της αλλαγής. Η ποιοτική ανάλυση της SWDWF γίνεται βάση έξι παραγόντων ποιότητας του ISO/IEC9126, την λειτουργικότητα, την αξιοπιστία, την ευχρηστία, την αποδοτικότητα, την συντηρησιμότητα και τέλος την φορητότητα. Τα αποτελέσματα δείχνουν ότι η SWDWF βελτιώνει την

αποτελεσματικότητα της διαδικτυακής ανάπτυξης, ενώ την ίδια στιγμή διασφαλίζει και την ποιότητά της.

Στο άρθρο των Richard F., et al, του 2006 παρουσιάζεται μία ευέλικτη διαδικασία για την παροχή ασφαλών διαδικτυακών εφαρμογών. Αυτή η ευέλικτη προσέγγιση του πειραματικού συστήματος παρέχει ένα ελαφρύ περιβάλλον για την ανάπτυξη, με στοιχειώδη ανάλυση των κινδύνων, και με την παροχή σε κάθε αύξηση των κατάλληλων μηχανισμών ασφάλειας. Με την εκ νέου εκτίμηση κινδύνων, η προσέγγισή αποφεύγει την προσθήκη μηχανισμών που εμποδίζουν, ή ανατρέπουν τους υφιστάμενους μηχανισμούς ασφάλειας. Οι έλεγχοι αυτοί, όταν εφαρμοστούν, θα επηρεάσουν διαφορετικά επίπεδα μίας τυπικής διαδικτυακής εφαρμογής. Φυσικά όλα αυτά είναι βασισμένα στην προϋπόθεση ότι είναι μία σαφής εμπειριστατωμένη επιχειρηματική ανάλυση και σαφής περιγραφή του συνόλου της πολιτικής ασφαλείας. Εάν, κατά τη διάρκεια μιας επανάληψης, η πολιτική έχει αλλάξει, οι προγραμματιστές είναι πιθανό να χρειαστεί να αναλύσουν ένα σημαντικό μέρος του υφιστάμενου συστήματος, για να εξασφαλιστεί ότι εξακολουθεί να είναι ασφαλές από την τροποποιημένη πολιτική.

Ο Vidar Kongsli στο άρθρο του 2007, παρουσίασε το selenium το οποίο είναι ένα εργαλείο για την δημιουργία και την εκτέλεση αυτόματων διαδικτυακών ελέγχων και ταιριάζει σε ευέλικτα έργα τα οποία μπορούν να χρησιμοποιηθούν για την δημιουργία δοκιμών αποδοχής. Ο ερευνητής δείχνει πως το selenium μπορεί να λειτουργήσει ως εργαλείο για την δημιουργία ελέγχων ασφαλείας. Πρώτα μοντελοποιούνται οι απειλές ασφάλειας σαν ιστορίες κακής χρήσης, δεύτερον οι ιστορίες χρηστών, και στη συνέχεια δημιουργείται ένα τεστ ασφαλείας στο selenium το οποίο ανακοινώνει τις ιστορίες κακής χρήσης εκμεταλλευόμενο τα τρωτά σημεία της εφαρμογής.

Αυτή η προσέγγιση μπορεί να θεωρηθεί ως συνεισφορά στην ενίσχυση της ασφαλείας που επικεντρώνεται στα ευέλικτα έργα, προσπαθώντας να εφαρμόσει παρόμοιες ευέλικτες έννοιες, μεθόδους και εργαλεία στις πτυχές ασφαλείας της εφαρμογής. Έχει βρεθεί ότι αρκετές από τις πιο κοινές ευπάθειες ασφαλείας στις διαδικτυακές εφαρμογές μπορούν να αντιμετωπιστούν με αυτή τη προσέγγιση, όπως υποκλοπή ταυτότητας και διαχείρισης πρόσβασης, διαρροή πληροφοριών και ακατάλληλη χρήση σφαλμάτων.

Στο άρθρο των Xiong Guang-cai, et al, του 2010, χρησιμοποιήθηκαν διαδικτυακές τεχνολογικές υπηρεσίες για την κατασκευή της ευέλικτης εφοδιαστικής αλυσίδας, μέσω της χρήσης τεχνολογιών καίριας σημασίας των διαδικτυακών υπηρεσιών (XML), η οποία πρέπει να έχει γρήγορη ανταπόκριση στην αγορά, υψηλό βαθμό ανταλλαγής πληροφοριών, καθώς επίσης και υψηλό βαθμό ολοκλήρωσης των διεργασιών και της διαλειτουργικότητας. Στη συνέχεια μελετήθηκε ο πυρήνας της τεχνολογίας των διαδικτυακών υπηρεσιών, ερευνήθηκε η συνολική προσομοίωση των επιχειρηματικών πρακτικών και μια υπηρεσία δικτυακής ένθετης ομαδοποίησης. Το πλαίσιο των συστημάτων εφοδιαστικής αλυσίδας υλοποιήθηκε με NET πλατφόρμα. Με την χρήση της τεχνολογίας διαδικτυακών υπηρεσιών και το τυπικό πρωτόκολλο της που βασίζεται σε XML, παρουσιάστηκε η υλοποίηση της. Αυτό το είδος των διαδικτυακών υπηρεσιών με βάση το ευέλικτο σύστημα τους, έχουν το πλεονέκτημα της «ανεξάρτητης πλατφόρμας», της «έγκαιρης συναρμολόγησης» και της «ευέλικτης κλίμακας» . Έτσι η απρόσκοπτη ολοκλήρωση των επιχειρηματικών εφαρμογών σε ετερογενείς πλατφόρμες έγινε πραγματικότητα. Συνεπώς, η έξυπνη επιχειρηματική διαδικασία επιτεύχθηκε.

Στο άρθρο του Ioakim (Makis) Marmaridis, et al, του 2007, επιδεικνύεται μια ευέλικτη προσέγγιση στο σχεδιασμό, την κατασκευή, και τη διαχείριση συστημάτων πληροφοριών για τις επιχειρήσεις. Η υλοποίηση της προσέγγισης πραγματοποιήθηκε, μέσω της τεχνολογίας smart business object (SBO), και της μηχανής Business Process Flow (LBPF). Η συνεργασία μεταξύ των δύο επιτρέπει μια παράλληλη λειτουργία της ανάπτυξης, όπου οι προγραμματιστές μπορούν να δημιουργήσουν και να απελευθερώσουν την σκοπούμενη εφαρμογή, μέσα σε ένα εξαιρετικά συμπιεσμένο χρονοδιάγραμμα, ενώ οι τελικοί χρήστες των επιχειρήσεων μπορούν να διαχειριστούν στη συνέχεια την εξέλιξη της ροής εργασίας των web-based εφαρμογών, σε ελεγχόμενο περιβάλλον μέσω web - based εργαλείων. Χρησιμοποιώντας την προτεινόμενη προσέγγιση και το σύνολο των εργαλείων, κατασκευάστηκε με επιτυχία σε επίπεδο επιχείρησης, διαδικτυακή εφαρμογή για το Πανεπιστήμιο του Σύδνεϋ με σκοπό την διαχείριση των μαθημάτων και των ενοτήτων τους. Η εφαρμογή ονομάζεται Online Course Approval System (OCAS) και κάνει πολύ μεγάλη χρήση των ροών εργασίας με παράλληλη εκτέλεση και συγχρονισμό σε πολλαπλά σημεία.

Στο άρθρο του Siew Poh, et al, του 2006, διεξάχθηκε μία συγκριτική μελέτη μεταξύ της ανάπτυξης των διαδικτυακών υπηρεσιών και της ευέλικτης μεθοδολογίας, για τον προσδιορισμό των πρόσθετων μέτρων που απαιτούνται στην ανάπτυξη των διαδικτυακών υπηρεσιών. Η συγκεκριμένη προσέγγιση έχει ως στόχο να επεκτείνει την υπάρχουσα ευέλικτη μεθοδολογία με τη συμπερίληψη των πρακτικών των διαδικτυακών υπηρεσιών σε κάθε φάση του κύκλου ζωής του ευέλικτου λογισμικού. Η ερευνητική ομάδα έχει κάνει μια λεπτομερή περίπτωση μελέτης για την εφαρμογή της μεθοδολογίας που αναφέρεται στην IBM-Rational RUP και στον extreme προγραμματισμό (XP). Τα αποτελέσματα δείχνουν τις συγκεκριμένες δραστηριότητες και εργασίες των διαδικτυακών υπηρεσιών για RUP και XP φάσεις ανάπτυξης λογισμικού αντίστοιχα. Η ερευνητική ομάδα διεξήγαγε έρευνα με επιλεγμένες εταιρείες της Σιγκαπούρης για την επικύρωση της προσέγγισής τους.

Στο άρθρο του Bhuvan Urganekar, et al, του 2008, προτάθηκε μία νέα δυναμική τεχνική τροφοδότησης για multi-tier διαδικτυακές εφαρμογές, που πρώτον χρησιμοποιούν ένα ευέλικτο μοντέλο ουράς, για να καθορίσει το ποσοστό των πόρων που μπορούν να διατεθούν σε κάθε βαθμίδα της αίτησης, και δεύτερον ένας συνδυασμός μεθόδων πρόβλεψης και αντίδρασης που καθορίζει την κάλυψη αυτών των πόρων, τόσο σε μεγάλες, όσο και μικρές χρονικές κλίμακες. Οι ερευνητές προτείνουν μία αρχιτεκτονική που βασίζεται στην εικονική μηχανή οθονών για την μείωση της πρόβλεψης των γενικών εξόδων. Τα πειράματα σε σαράντα μηχανές Xen / Linux-based πλατφόρμες φιλοξενίας καταδεικνύουν τον βαθμό ανταπόκρισης της τεχνικής στον χειρισμό δυναμικού φόρτου εργασίας. Σε ένα σενάριο όπου είχε διπλασιαστεί ο φόρτος εργασίας σε τρία επίπεδα εφαρμογής, η τεχνική ήταν σε θέση να διπλασιάσει μέσα σε πέντε λεπτά την ικανότητα της εφαρμογής διατηρώντας παράλληλα τους στόχους του χρόνου απόκρισης. Η τεχνική μείωσε επίσης τα γενικά έξοδα της μεταγωγής των servers σε εφαρμογές, από αρκετά λεπτά ή περισσότερο, σε λιγότερο από ένα δευτερόλεπτο.

Ο Gary B. Wills, et al, στο άρθρο του 2007, παρουσίασε μία ευέλικτη προσέγγιση για την ανάπτυξη εφαρμογών υπερκείμενου, η οποία επικεντρώνεται στις απαιτήσεις και τα στάδια της ανάλυσης, κάτι που αγνοείται σε μεγάλο βαθμό στις

τρέχουσες μεθοδολογίες. Σε αυτή την έρευνα έχει προταθεί ένα ευέλικτο μοντέλο προσέγγισης, για τις απαιτήσεις και την ανάλυση, βασιζόμενο στα τωρινά μοντέλα σχεδιασμού και τις τωρινές μεθοδολογίες. Η έρευνα βασίζεται στην εμπειρία των ερευνητών που επί σειρά ετών, παράγουν υπερκείμενα διαδικτυακών εφαρμογών, με σύντομους χρόνους παράδοσης σε μεγάλες βιομηχανικές και έργα που σχετίζονται με υπερμέσα, συμπεριλαμβανομένης της mPLAT, FREMA και έργα του Σημασιολογικού Ιστού IPAS.

Στο άρθρο της Lili Qu, et al, του 2009, προτάθηκε μία προσέγγιση που καθιστά την εφοδιαστική αλυσίδα, εφαρμογή της πολύπλοκης δυναμικής σύνθεσης των επιχειρηματικών διαδικασιών, και την ετερογενή εφαρμογή των συστημάτων ολοκλήρωσης των επιχειρήσεων. Ο συντονισμός και η ολοκλήρωση του πλαισίου του συστήματος της ευέλικτης εφοδιαστικής αλυσίδας, έχει καθοριστεί με βάση τις προτεινόμενες σημασιολογικές διαδικτυακές υπηρεσίες και Service-Oriented Architecture (SOA). Αυτό το αναπτυγμένο ιεραρχικό πλαίσιο έχει τα πλεονεκτήματα της χαλαρής ζεύξης για τα ετερογενή συστήματα αλληλεπίδρασης και ευέλικτης εφοδιαστικής αλυσίδας για την ανασυγκρότηση του συστήματος, το οποίο είναι μια εφικτή και αποτελεσματική λύση για την υλοποίηση των χαρακτηριστικών της, συμπεριλαμβανομένης της ευελιξίας και της ανασυγκρότησης.

Έτσι λοιπόν οι τεχνολογίες των διαδικτυακών υπηρεσιών επιτρέπουν τις αόριστα συνδεδεμένες επιχειρησιακές διαδικασίες και εφαρμογές με πολύ λιγότερη προσπάθεια σε σχέση με παλαιότερες τεχνολογίες.

Στο άρθρο του A. Tappenden, et al, του 2005, διερευνήθηκαν τα κοινά τρωτά σημεία των διαδικτυακών εφαρμογών και προτάθηκαν δύο προσεγγίσεις για τον περιορισμό τους. Η πρώτη προσέγγιση είναι η χρήση μίας ιδιαίτερα ελέγξιμης αρχιτεκτονικής για την ανάπτυξη των web-based συστημάτων, και η δεύτερη είναι για την υποστήριξη της διαδικασίας ελέγχου ασφαλείας χρησιμοποιώντας open source πλαίσιο δοκιμών, που μπορεί να παρακάμψει τα επίπεδα, και να αλληλεπιδρά άμεσα με τον server της διαδικτυακής εφαρμογής μέσω HTTPUnit. Η συνολική στρατηγική δοκιμών εμπλέκεται με τις ευέλικτες αναπτυξιακές προσπάθειες και δίνει στην ομάδα ανάπτυξης τη δυνατότητα να παράγει εφαρμογές που έχουν το δικαίωμα "λειτουργικότητας" και το δικαίωμα του "επιπέδου ασφαλείας". Έτσι η ομάδα ανάπτυξης αποκτά την ικανότητά του να

εκτελεί τις δοκιμές ασφαλείας για κρίσιμης σημασίας τρωτά σημεία, που είναι καλύτερο να μετριάστούν με ασφαλείς πρακτικές προγραμματισμού στον server της διαδικτυακής εφαρμογής.

Στο άρθρο του Vítor Silva, et al, του 2005, παρουσιάσθηκε μια ευέλικτη προσέγγιση για την ανάπτυξη των διαδικτυακών εφαρμογών που εφαρμόζει την έννοια της ευέλικτης μοντελοποίησης, υιοθετεί ένα πρότυπο αρχιτεκτονικής λογισμικού και βασίζεται σε μεγάλο βαθμό σε πλαίσια, επιταχύνοντας την ανάλυση του συστήματος, τον σχεδιασμό και την υλοποίησή του. Όπως και με την συμβατική τεχνολογία λογισμικού, η διαδικασία webE (*Web Engineering*) ξεκινάει με την εύρεση των αναγκών της επιχείρησης και συνεχίζει με τον σχεδιασμό του έργου. Στη συνέχεια γίνεται λεπτομερής ανάλυση απαιτήσεων και διαμορφώνεται λαμβάνοντας υπόψη την ανάλυση και την προοπτική του σχεδιασμού. Τότε η εφαρμογή δημιουργείται χρησιμοποιώντας εργαλεία και μεθόδους ειδικευμένα στο διαδίκτυο. Τέλος το σύστημα ελέγχεται και παραδίδεται στον τελικό χρήστη. Από τις περιπτώσιολογικές μελέτες που διεξάχθηκαν χρησιμοποιώντας την προτεινόμενη προσέγγιση, προκύπτει το συμπέρασμα ότι, η δημιουργία ευέλικτων διαδικασιών λογισμικού για την ανάπτυξη διαδικτυακών εφαρμογών χρησιμοποιώντας πλαίσια, είναι πολλά υποσχόμενη.

Συγγραφείς, Τίτλος, Έτος	Ευρήματα	Ασφάλεια	UML	Επέκταση των ευέλικτων διαδικτυακών εφαρμογών	Service Oriented Architecture (SOA)
Bartsch, S. Supporting Authorization Policy Modification in Agile Development of Web Applications 2010	Δημιουργία αλγορίθμου και εργαλείου για την βελτίωση της ασφάλειας	✓			
Wookjin Lee, Sanghyun Park, Keeyoull Lee, Chunwoo Lee, Byungjeong Lee, Woosung Jung, Taeksu Kim, Heechern Kim, and Chisu Wu Agile Development of Web Application by Supporting Process Execution and Extended UML Model, 2005	Δημιουργία διαδικτυακών μοντέλων, εργαλείων και διαδικασιών για την ανάπτυξη ευέλικτης διαδικτυακής εφαρμογής		✓		
Keoo Lee An Agile Method for Web Applications in Dynamic Requirements 2008	Δημιουργία μηχανισμού και μοντέλου για την διαχείριση των απαιτήσεων		✓		
Ran Hu, Zhuo Wang, Jun Hu, Jianfeng Xu, and Jun Xie Agile Web Development with Web Framework 2008	Δημιουργία ευέλικτης διαδικτυακής μεθόδου ανάπτυξης με διαδικτυακό πλαίσιο			✓	
Richard F. Paige, Xiaocheng Ge, Fiona A.C. Polack, Howard Chivers, and Phillip J. Brooke Agile Development of Secure Web Applications 2006	Ανάπτυξη προσέγγισης για την αντιμετώπιση κινδύνων ασφαλείας κατά την ανάπτυξη ευέλικτων διαδικτυακών εφαρμογών	✓			

Vidar Kongsli Security Testing with Selenium 2007	Ανάπτυξη εργαλείου για δημιουργία ελέγχων και δοκιμών σε ευέλικτες διαδικτυακές εφαρμογές	✓		✓	
Xiong Guang-cai, Wang Zhi-feng, Zhang Xin-jia, and Ji Guo-jun Research of Agile Supply Chain Management System Based on Web Services 2010	Κατασκευή πλαισίου ευέλικτης εφοδιαστικής αλυσίδας για διαδικτυακές υπηρεσίες σε ετερογενείς πλατφόρμες			✓	
Marmaridis I., Xufeng Liang, and Ginige A., Facilitating Agile Model Driven Development and End-User Development for Evolving Web-based Workflow Applications 2007	Υλοποίηση προσέγγισης και μηχανισμού για την ανάπτυξη ευέλικτων διαδικτυακών εφαρμογών			✓	
Siew Poh Lee, Lai Peng Chan, and Eng Wah Lee Web Services Implementation Methodology for SOA Application, 2006	Ανάπτυξη μεθοδολογίας για την προσθήκη διαδικτυακών υπηρεσιών στην ευέλικτη μεθοδολογία			✓	✓
Bhuvan Urgaonkar, Prashant Shenoy, Abhishek Chandra, Pawan Goyal, and Timothy Wood Agile dynamic provisioning of multi-tier Internet applications 2008	Ανάπτυξης δυναμικής τεχνικής τροφοδότησης για χειρισμό δυναμικού φόρτου εργασίας σε διαδικτυακές εφαρμογές			✓	

Gary B. Wills, Noura Abbas, Rakhi Chandrasekharan, Richard M. Crowder, Lester Gilbert, Yvonne M. Howard, David E. Millard, Sylvia C. Wong, and Robert J. Walters An Agile Hypertext Design Methodology 2007	Δημιουργία ευέλικτου μοντέλου για την ανάπτυξη εφαρμογών υπερκειμένου			✓	✓
Lili Qu, Yan Chen, and Ming Yang, The Coordination and Integration of Agile Supply Chain based on Service-oriented Technology 2009	Ανάπτυξη προσέγγισης που εφαρμόζει την εφοδιαστική αλυσίδα στην σύνθεση επιχειρηματικών διαδικασιών			✓	✓
Tappenden A., Beatty P., Miller J., Geras A., and Smith M. Agile Security Testing of Web-Based Systems via HTTPUnit 2005	Ανάπτυξη προσεγγίσεων για την ασφάλεια των ευέλικτων διαδικτυακών εφαρμογών	✓			
Vitor Silva, Ricardo Almeida, An Agile Approach for Web Systems Engineering 2005	Δημιουργία προσέγγισης για την ανάπτυξη ευέλικτης μοντελοποίησης διαδικτυακών εφαρμογών			✓	

Πίνακας 2.1: Τα ευρήματα των συγγραφέων, ο τίτλος, το έτος έκδοσης, καθώς και το αντικείμενο του κάθε ευρήματος

2.2 Επίλογος

Από ότι φαίνεται μία πλειοψηφία επιστημόνων ασχολείται με την ασφάλεια των ευέλικτων διαδικτυακών εφαρμογών, μία άλλη πλειοψηφία με την δυναμική εξέλιξη και επέκτασή τους, δημιουργώντας νέα βελτιστοποιημένα μοντέλα, και αναπτύσσοντας νέες τεχνικές που βελτιώνουν την ανάπτυξη των ευέλικτων διαδικτυακών εφαρμογών.

Στην παρούσα Πτυχιακή Εργασία θα αναπτύξουμε μία μικρή διαδικτυακή εφαρμογή με το εργαλείο Eclipse. Η εφαρμογή μας έχει σαν θέμα την δημιουργία ενός Εβδομαδιαίου Προγράμματος Μαθημάτων. Ο χρήστης μπορεί να εισάγει το όνομα του μαθήματος που θέλει, τις ημέρες και τις ώρες, και έτσι το πρόγραμμα του εμφανίζει το εβδομαδιαίο πρόγραμμα των μαθημάτων του.

ΚΕΦΑΛΑΙΟ 3

Ο κύκλος ζωής των Ευέλικτων Διαδικτυακών Εφαρμογών

3.1 Εισαγωγή

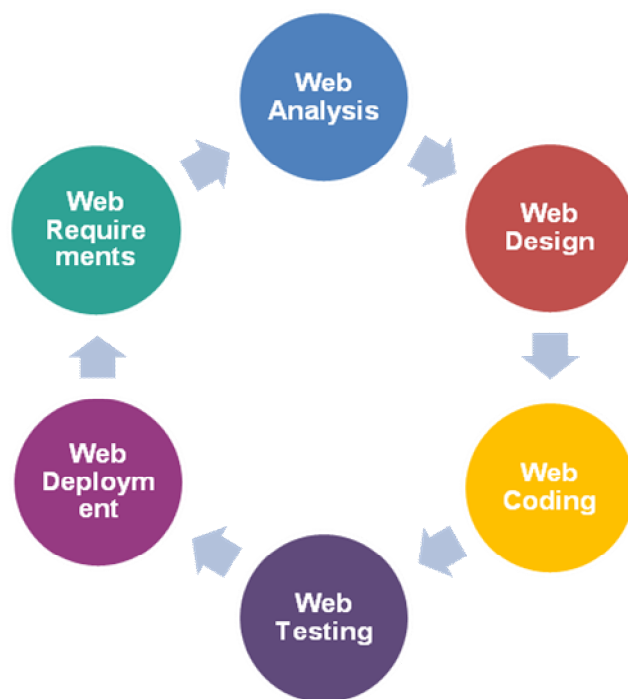
Ο σχεδιασμός για την ανάπτυξη μίας ευέλικτης διαδικτυακής εφαρμογής είναι πιο εύκολος στα λόγια παρά στην πράξη. Το πρώτο που είναι απαραίτητο για την ανάπτυξη μίας οποιαδήποτε ευέλικτης διαδικτυακής εφαρμογής, είναι μία ιδέα, καθώς επίσης και η γνώση του να αντιλαμβάνεται κανείς αυτή την ιδέα ώστε να ξεκινήσει. Ευέλικτο σημαίνει γρήγορο και οι οργανισμοί συνειδητοποιούν τα σημαντικά οφέλη της ευέλικτης διαδικτυακής μεθοδολογίας ανάπτυξης, αφού είναι πιο αποτελεσματική από τις παραδοσιακές προσεγγίσεις, και ο στόχος είναι η ικανοποίηση του πελάτη.

Στην ευέλικτη διαδικτυακή μεθοδολογία ανάπτυξης, τα έργα χωρίζονται σε διαφορετικές φάσεις οι οποίες περιγράφουν και τον κύκλο ζωής των ευέλικτων διαδικτυακών εφαρμογών. Οι φάσεις αυτές είναι οι εξής:

- Η φάση των Απαιτήσεων
- Η φάση της Ανάλυσης
- Η φάση του Σχεδιασμού
- Η φάση της Κωδικοποίησης
- Η φάση του Ελέγχου
- Και η φάση της Ανάπτυξης

Η ευέλικτη διαδικτυακή μεθοδολογία δίνει έμφαση στον έλεγχο και την προσαρμογή. Η μετάβαση από φάση σε φάση δεν χρειάζεται να είναι μία διαδοχική διαδικασία. Αντίθετα, η διαδικασία τείνει να είναι επαναληπτική και σταδιακή και φυσικά αρκετά ευέλικτη, ώστε να μπορεί να φιλοξενήσει τις συνεχείς αναθεωρήσεις.

Μία φάση, όταν χρησιμοποιείται στο πλαίσιο εφαρμογής του κύκλου ζωής μίας ευέλικτης διαδικτυακής εφαρμογής, αναφέρεται στην περίοδο του χρόνου που πραγματοποιούνται οι δραστηριότητες που σχετίζονται με την εφαρμογή.



Εικόνα 3.1: Ο κύκλος ζωής της Ευέλικτης Διαδικτυακής Εφαρμογής

3.2 Μεθοδολογία υλοποίησης Ευέλικτων Διαδικτυακών Εφαρμογών

Ο όρος ευέλικτες διαδικτυακές εφαρμογές περιγράφει έναν εξειδικευμένο τύπο λογισμικού, ο οποίος έχει σχεδιαστεί για να υποστηρίξει ένα τυποποιημένο τρόπο για την παροχή υπηρεσιών μέσω του διαδικτύου, και μέσω της συμμόρφωσης με τα ανοιχτά πρότυπα, όπως η γλώσσα XML και άλλα. Οι ευέλικτες εφαρμογές σε αντίθεση με τις παραδοσιακές, δεν προορίζονται για άμεση κατανάλωση των τελικών χρηστών. Αντίθετα, αποτελούν ένα κομμάτι της επιχειρησιακής λογικής, έχουν προγραμματιστικές διασυνδέσεις και μέσω αυτών των διασυνδέσεων οι προγραμματιστές μπορούν να επεκτείνουν ή να δημιουργήσουν νέα διαδικτυακά συστήματα εφαρμογών. Το κίνητρο πίσω από τις ευέλικτες διαδικτυακές εφαρμογές είναι να διευκολύνει τις επιχειρήσεις και τους οργανισμούς να αλληλεπιδρούν με άλλες επιχειρήσεις και πελάτες, χωρίς να χρειάζεται να

περάσουν από μακρά σχεδιαστική ολοκλήρωση, ή ακόμη και να αποκαλύπτουν εμπιστευτικές εσωτερικές λεπτομέρειες της εφαρμογής χωρίς λόγο.

Μία διαδικτυακή εφαρμογή, φυσικά είναι ένα στοιχείο λογισμικού, αλλά λόγω του εξειδικευμένου interface και του μηχανισμού της να λειτουργεί σε συνεργασία με άλλους, όλη η επικρατούσα γενική μεθοδολογία ανάπτυξης λογισμικού θα πρέπει να προσαρμοστεί για να χειριστεί τα μοναδικά χαρακτηριστικά της ευέλικτης διαδικτυακής εφαρμογής. Αυτό θα μπορούσε να μεταφραστεί σε προσδιορισμό ειδικών απαιτήσεων της διαδικτυακής εφαρμογής (π.χ. συμμόρφωση με τα πρότυπα της διαδικτυακής εφαρμογής), την ανάλυση των συγκεκριμένων συνεπειών της για το συνολικό σύστημα, τη σχεδίαση διασύνδεσης της διαδικτυακής εφαρμογής, την κωδικοποίηση, τον έλεγχο, την ανάπτυξη και την εκτέλεσή της.

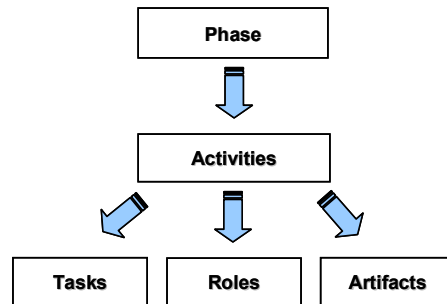
Οι διαδικτυακές εφαρμογές πρέπει να είναι ευέλικτες ώστε να μπορούν να φιλοξενήσουν βελτίωση καθ' όλη τη διάρκεια του κύκλου ανάπτυξης σε μία επαναληπτική και αυξητική προσέγγιση. Η μεθοδολογία ανάπτυξής τους πρέπει να αποτελείται από τις φάσεις που καλύπτουν τις περιοχές από τη σύλληψη της ανάγκης της διαδικτυακής εφαρμογής για την κατασκευή της, και τον προορισμό της για χρήση της τελικής εφαρμογής από τον πελάτη.

Η ανάπτυξη είναι επαναληπτική και αυξητική. Σε κάθε επανάληψη, η διαδικτυακή εφαρμογή θα πρέπει να περνάει από όλα τα στάδια (δηλαδή απαιτήσεις, ανάλυση, σχεδιασμό, κωδικοποίηση, έλεγχο και ανάπτυξη) με αποτέλεσμα την ανάπτυξη και τελειοποίησης της σε όλο τον κύκλο ζωής του έργου. Επιπλέον, όσον αφορά τον έλεγχο, ένα πλήθος δοκιμών πρέπει να διεξάγονται για να διασφαλιστεί ότι έχει αναπτυχθεί σύμφωνα με τις λειτουργικές απαιτήσεις.

Οι προδιαγραφές που παράγονται σε κάθε μία από τις φάσεις ανάπτυξης, αποτελούν πηγές εισροών ώστε να εξαντληθούν τα σενάρια δοκιμών και οι περιπτώσεις δοκιμής. Από αυτές τις περιπτώσεις δοκιμής, συγκεντρώνονται τα χειρόγραφα και τα δεδομένα δοκιμών, τα οποία θα χρησιμοποιηθούν στη μονάδα δοκιμής, στη λειτουργική δοκιμή, στη δοκιμή ενσωμάτωσης και λειτουργικότητας, στη δοκιμή της απόδοσης, και την τελική δοκιμή αποδοχής από τους χρήστες.

Η εφαρμογή του κύκλου ζωής της ευέλικτης διαδικτυακής εφαρμογής, περιγράφει ουσιαστικά τις φάσεις μίας τυπικής διαδικτυακής υπηρεσίας, από την αναγνώριση

της ανάγκης της, μέχρι την τελική της εγκατάσταση και χρήση από τους τελικούς χρήστες. Οι φάσεις προσδιορίζονται να είναι σχετικές με τον κύκλο ζωής της ευέλικτης διαδικτυακής εφαρμογής. Σε κάθε μία από τις φάσεις ανάπτυξης, εκτελούνται συγκεκριμένες δραστηριότητες.



Εικόνα 3.2: Απεικόνιση της ανωτέρω σχέσης μεταξύ της φάσης, των δραστηριοτήτων, των αντίστοιχων καθηκόντων, των ρόλων και των αντικειμένων.

3.2.1 Η φάση των Απαιτήσεων

Ο στόχος της φάσης των απαιτήσεων είναι να κατανοήσουμε τις απαιτήσεις (π.χ. των επιχειρήσεων, αυτό που θέλει ο πελάτης), και την μετατροπή τους όσο αφορά τα χαρακτηριστικά, σε απαιτήσεις διαδικτυακών υπηρεσιών, σε διαχωρισμό των λειτουργικών και μη-λειτουργικών απαιτήσεων, αλλά και τους περιορισμούς εντός των οποίων η διαδικτυακή εφαρμογή πρέπει να εκπληρώσει. Οι λειτουργικές απαιτήσεις περιγράφουν την διαδικασία προσδιορισμού των απαραίτητων εργασιών, ενεργειών, ή δραστηριοτήτων που πρέπει να επιτευχθούν, δηλαδή η ανάλυση όλων των λειτουργιών για την λειτουργική ανάλυση της εφαρμογής, ενώ οι μη-λειτουργικές απαιτήσεις, είναι αυτές που καθορίζουν τα κριτήρια που μπορούν να χρησιμοποιηθούν για να κρίνουν τη λειτουργία της εφαρμογής και όχι συγκεκριμένες συμπεριφορές. Οι απαιτήσεις πρέπει να είναι τεκμηριωμένες, δραστικές, μετρήσιμες και ελέγξιμες σε σχέση με τις προσδιορισθείσες ανάγκες των επιχειρήσεων, να ορίζονται σε επίπεδο λεπτομέρειας που επαρκεί για το σχεδιασμό της εφαρμογής, και να μην είναι ασαφείς, ελλιπείς ή αντιφατικές. Η εκμείωση των απαιτήσεων πρέπει να γίνεται από τον αναλυτή των απαιτήσεων, όμως, είναι καθήκον του η επικοινωνία με τα ενδιαφερόμενα μέλη της διαδικτυακής

εφαρμογής, των οποίων η συμμετοχή κρίνεται απαραίτητη για την καταγραφή, ώστε να καθορίσουν ποιες είναι οι δικές τους απαιτήσεις, όπως για παράδειγμα ο πελάτης, οι τελικοί χρήστες κ.α. Μετά από αυτό, ο αναλυτής πρέπει να ερμηνεύσει, να παγιώσει και να κοινοποιήσει αυτές τις απαιτήσεις στην υπόλοιπη ομάδα ανάπτυξης. Σε περίπτωση που είναι δυνατόν, το έγγραφο των απαιτήσεων θα ήταν καλύτερο να βρίσκεται σε ένα κεντρικό χώρο αποθήκευσης, από όπου θα μπορεί να προβληθεί οποιαδήποτε στιγμή, και να εξορυχτεί για την αναθεώρησή του. Σε όλες τις περιπτώσεις, που επιτρέπουν στην ομάδα ανάπτυξης να συλλάβει απαιτήσεις εύκολα, η αναζήτηση, και η επεξεργασία τους είναι η κύρια λειτουργία του κεντρικού αυτού χώρου αποθήκευσης.

Στη φάση αυτή πραγματοποιούνται διάφορες δραστηριότητες που είναι οι ακόλουθες:

- Καθορισμός της ανάγκης της ευέλικτης διαδικτυακής εφαρμογής
 - Προσδιορισμός των ενδιαφερόμενων μερών. Στους ενδιαφερόμενους συνήθως περιλαμβάνονται οι τελικοί χρήστες, ο διαχειριστής του έργου κ.α.
 - Κατανόηση των ελλείψεων και των προβλημάτων για την αντιμετώπισή τους.
 - Προσδιορισμός των απαιτήσεων για την τεχνολογία της ευέλικτης διαδικτυακής εφαρμογής. Με βάση την τρέχουσα διαθέσιμη τεχνολογία, τον προσδιορισμό των αναγκών για διαδικτυακές εφαρμογές
 - Καθορισμός της θέσης της διαδικτυακής εφαρμογής εντός των ορίων του προβλήματος
 - Ορισμός των χαρακτηριστικών της εφαρμογής βάσει του καταλόγου των αναγκών
 - Προσδιορισμός των περιορισμών που θα επιβληθούν στην εφαρμογή

- Συγκέντρωση των απαιτήσεων της ευέλικτης διαδικτυακής εφαρμογής
 - Προσδιορισμός των πηγών για τη συλλογή των απαιτήσεων με βάση τη λίστα των λειτουργιών. Προσδιορισμός των τμημάτων, των τελικών χρηστών κτλ., οι οποίοι θα επηρεαστούν από την εφαρμογή
 - Προσδιορισμός των λειτουργικών απαιτήσεων για την εφαρμογή και κατηγοριοποίησή τους
 - Εντοπισμός των μη-λειτουργικών απαιτήσεων για την ευέλικτη διαδικτυακή εφαρμογή. Οι μη-λειτουργικές απαιτήσεις είναι οι απαιτήσεις που αφορούν την ευχρηστία, την αξιοπιστία, την απόδοση, την επεκτασιμότητα και άλλους παράγοντες σχεδιασμού

- Διαχείριση των απαιτήσεων της ευέλικτης διαδικτυακής εφαρμογής
 - Προσδιορισμός της διαδικτυακής εφαρμογής με βάση τις λειτουργικές απαιτήσεις και δημιουργία εξαρτήσεων και προτεραιοτήτων
 - Δημιουργία πινάκων ιχνηλασιμότητας από τις απαιτήσεις για την προσδιοριζόμενη διαδικτυακή εφαρμογή
 - Διαχείριση των αλλαγών στις απαιτήσεις

- Μοντελοποίηση του σεναρίου χρήσης
 - Μετατροπή των λειτουργικών απαιτήσεων σε θεωρητικά μοντέλα χρήσης που χρησιμοποιούν κάποιες μορφές ανάλυσης τεχνικών μοντελοποίησης
 - Καθορισμός των σεναρίων αλληλεπίδρασης με τους πελάτες της εφαρμογής

- Προετοιμασία των περιπτώσεων ελέγχου για τη δοκιμή αποδοχής από τους χρήστες και τον έλεγχο του συστήματος
 - Συγγραφή του σεναρίου της επιχειρηματικής δοκιμής με βάση τις απαιτήσεις που συγκεντρώθηκαν, ώστε να χρησιμοποιηθεί για τη δοκιμή αποδοχής από τους χρήστες και τον έλεγχο του συστήματος. Αυτός είναι και ένας τρόπος για την επαλήθευση των απαιτήσεων κατά την εφαρμογή τους
 - Κατασκευή πίνακα επαλήθευσης απαιτήσεων. Ο πίνακας αυτός πρέπει να περιλαμβάνει τις απαιτήσεις και μία αναφορά των περιπτώσεων ελέγχου που θα επικυρώνει τις απαιτήσεις
 - Διαχείριση αλλαγών στις περιπτώσεις δοκιμής, όταν υπάρχει τροποποίηση των απαιτήσεων

3.2.2 Η φάση της Ανάλυσης

Στη φάση της ανάλυσης, οι απαιτήσεις πρέπει να βελτιωθούν περαιτέρω, και να μεταφραστούν σε εννοιολογικά μοντέλα τα οποία μπορούν να κατανοηθούν από την τεχνική ομάδα ανάπτυξης. Επίσης, σε αυτή τη φάση γίνεται η αρχιτεκτονική ανάλυση, για να καθορισθεί η δομή υψηλού επιπέδου, και ο προσδιορισμός των συμβάσεων διασύνδεσης της διαδικτυακής εφαρμογής. Αυτή η διαδικασία πραγματοποιείται από τον αναλυτή των απαιτήσεων και κοινοποιείται στο σχεδιαστή και την υπόλοιπη ομάδα ανάπτυξης.

Οι δραστηριότητες που πραγματοποιούνται σε αυτή τη φάση είναι οι ακόλουθες:

- Επιλογή της τεχνολογικής πλατφόρμας που θα χρησιμοποιηθεί όπως και το πλαίσιο υλοποίησης
 - Καθορισμός των προτύπων της διαδικτυακής εφαρμογής που η υλοποίησή της πρέπει να τηρεί. Εδώ γίνεται ο προσδιορισμός των προτύπων με βάση τις απαιτήσεις και τους περιορισμούς. Εξετάζονται θέματα όπως η συμβατότητα των προτύπων, η έκδοση των προτύπων, η υιοθέτηση προτύπων σε τομείς όπως η βιομηχανία, και η έγκρισή τους

- Λήψη απόφασης για την τεχνολογική πλατφόρμα που θα χρησιμοποιηθεί στην υλοποίηση της εφαρμογής, με βάση τους περιορισμούς και τις εκτιμήσεις για την υποστήριξη των προτύπων, αλλά και τις απαιτήσεις λειτουργικότητας. π.χ. πλατφόρμα Java
 - Λήψη απόφασης για τα IDE εργαλεία που θα χρησιμοποιηθούν. Πολλές φορές η επιλογή των IDE εργαλείων συνδέεται με την πλατφόρμα υλοποίησης
- Ορισμός υποψήφιας αρχιτεκτονικής για την διαδικτυακή εφαρμογή
- Καθορισμός υψηλού επιπέδου αρχιτεκτονικής
 - Προσδιορισμός των αρχιτεκτονικών συστατικών που εκθέτουν τη λειτουργικότητα της εφαρμογής
 - Καθορισμός της ανταλλαγής πληροφοριών με τους πελάτες. Προσδιορισμός και καθορισμός των πρώτων μηνυμάτων που ανταλλάσσονται με τους πελάτες της εφαρμογής. Ο προσδιορισμός αυτός περιλαμβάνει τα στοιχεία των δεδομένων, τον τύπο και τη μορφή των δεδομένων
- Προσδιορισμός της αναλυτικότητας της ευέλικτης διαδικτυακής εφαρμογής
- Λήψη απόφασης σχετικά με τις λειτουργίες της εφαρμογής. Ο ορισμός αυτών των λειτουργιών εξαρτάται από τα σενάρια χρήσης και τις απαιτήσεις
 - Προσδιορισμός της λειτουργικότητας της ομάδας. Δηλαδή με βάση τις απαιτήσεις και τα κριτήρια, γίνεται ο εντοπισμός των λειτουργιών οι οποίες χρειάζονται για την ομάδα
 - Απόφαση σχετικά με τους μηχανισμούς για τη σύνθεση της συνολικής λειτουργικότητας.

- Προσδιορισμός επαναχρησιμοποιήσιμων διαδικτυακών εφαρμογών
 - Προσδιορισμός των αρχιτεκτονικών συστατικών που μπορούν να χρησιμοποιηθούν από τις υπάρχουσες διαδικτυακές εφαρμογές. Εάν η λειτουργικότητα των συστατικών της αρχιτεκτονικής μπορεί να εκπληρωθεί με υπάρχουσες διαδικτυακές εφαρμογές (ή μέρος αυτών), πρέπει να προσδιορίζονται αυτά τα αρχιτεκτονικά συστατικά για να κάνουν χρήση των υπαρχόντων εφαρμογών
 - Εντοπισμός και συλλογή πληροφοριών σχετικά με το φορέα παροχής των υπαρχόντων διαδικτυακών εφαρμογών
 - Προσδιορισμός των κυριότερων σεναρίων επαναχρησιμοποίησης. Δηλαδή των λειτουργιών που πρόκειται να χρησιμοποιηθούν

- Ορισμός της διασύνδεσης των υπηρεσιών για τη νέα εφαρμογή
 - Καθορισμός της υπογραφής των λειτουργιών της νέας διαδικτυακής εφαρμογής, με βάση τα μοντέλα χρήσης και τα μοντέλα ανάλυσης προσδιορίζονται οι λειτουργίες και οι υπογραφές τους.
 - Ορισμός XML σχήματος για την ανταλλαγή μηνυμάτων. Εάν εμπλέκονται ανταλλαγές μηνυμάτων, πρέπει να καθοριστεί το XML σχήμα που καθοδηγεί τη δομή του μηνύματος

- Προετοιμασία των περιπτώσεων ελέγχου για την δοκιμή των επιδόσεων
 - Συγγραφή των περιπτώσεων ελέγχου που θα χρησιμοποιηθούν στη δοκιμή των επιδόσεων, που μπορεί να προέρχονται από τις προδιαγραφές του αρχιτεκτονικού σχεδιασμού
 - Οι περιπτώσεις ελέγχου πρέπει να καλύπτουν όλα τα σενάρια δοκιμών, έτσι ώστε να υπάρχουν πληροφορίες για το πως το σύστημα θα λειτουργεί όταν υπάρχει μεγάλο φόρτο. Για παράδειγμα ταυτόχρονων συναλλαγών και αιτήσεων από πολλούς χρήστες

- Προετοιμασία των περιπτώσεων ελέγχου για τις δοκιμές ενσωμάτωσης και διαλειτουργικότητας
 - Συγγραφή των περιπτώσεων ελέγχου για δοκιμές ενσωμάτωσης και διαλειτουργικότητας που προέρχονται από τις προδιαγραφές του αρχιτεκτονικού σχεδιασμού

- Προετοιμασία των περιπτώσεων ελέγχου για λειτουργική δοκιμή
 - Συγγραφή των λειτουργικών περιπτώσεων ελέγχου που θα χρησιμοποιηθούν για λειτουργική δοκιμή

- Προετοιμασία πλατφόρμας δοκιμών
 - Εγκατάσταση του περιβάλλοντος ελέγχου που περιλαμβάνει υλικό και λογισμικό

3.2.3 Η φάση του σχεδιασμού

Σε αυτή τη φάση γίνεται ο λεπτομερής σχεδιασμός των διαδικτυακών υπηρεσιών της εφαρμογής. Οι σχεδιαστές καθορίζουν τη σύμβαση διασυνδέσεων των διαδικτυακών υπηρεσιών, που έχουν προσδιορισθεί κατά τη φάση της ανάλυσης. Ο ορισμός της σύμβασης διασυνδέσεων εντοπίζει τα στοιχεία και τους τύπους δεδομένων (για παράδειγμα ενός XML σχήματος), καθώς και τον τρόπο αλληλεπίδρασης της εφαρμογής με τον πελάτη.

Οι δραστηριότητες που πραγματοποιούνται σε αυτή τη φάση είναι οι εξής:

- Μετασχηματισμός των υπογραφών από επαναχρησιμοποιήσιμες διαδικτυακές εφαρμογές
 - Εφόσον απαιτείται, προσδιορισμός χαρτογράφησης των τύπων δεδομένων. Αν ο τύπος μίας παραμέτρου των επαναχρησιμοποιήσιμων εφαρμογών δεν υποστηρίζεται άμεσα από την προσδιοριζόμενη πλατφόρμα, θα πρέπει να εκτελείται η χαρτογράφηση του τύπου δεδομένων

- Προσδιορισμός των σχεδιαστικών προτύπων (*design patterns*) για τη χαρτογράφηση της διασύνδεσης της επαναχρησιμοποιούμενης εφαρμογής. Ορισμένα σχεδιαστικά πρότυπα μπορούν να χρησιμοποιηθούν για την επαναχρησιμοποίηση υπάρχοντων διαδικτυακών εφαρμογών, όπως για παράδειγμα το πρότυπο Adapter θα μπορούσε να χρησιμοποιηθεί για τη δημιουργία νέας διασύνδεσης σε μία υπάρχουσα διαδικτυακή εφαρμογή
- Βελτίωση των υπηρεσιών διασύνδεσης της διαδικτυακής εφαρμογής
 - Βελτίωση των υπογραφών διασύνδεσης. Στο στάδιο του λεπτομερούς σχεδιασμού, η υπογραφή μπορεί να βελτιωθεί περαιτέρω. Πρέπει να ληφθεί μέριμνα για να διασφαλιστεί ότι η απόφαση του σχεδιασμού δεν πρέπει να επηρεάζει την διαλειτουργικότητα των υπηρεσιών
 - Βελτίωση του XML σχήματος για την ανταλλαγή μηνυμάτων. Το XML σχήμα πρέπει να βελτιώνεται για να επεκτείνει περαιτέρω τη δομή και τους τύπους των δεδομένων, τα πεδία ονομάτων και άλλα
- Σχεδιασμός της ευέλικτης διαδικτυακής εφαρμογής
 - Χρήση κάποιας μορφής τεχνικής μοντελοποίησης για την περιγραφή της εσωτερικής δομής της εφαρμογής. Ο σχεδιασμός της εσωτερικής δομής πρέπει να εξετάζει τη λήψη και την προ-επεξεργασία, τη μεταβίβαση, και την επεξεργασία της αίτησης, καθώς και την αποστολή της απάντησης. Τα σχεδιαστικά πρότυπα παραδοσιακών τεχνικών μοντελοποίησης όπως η UML, θα μπορούσαν να χρησιμοποιηθούν στο σχεδιασμό
 - Πρέπει να ληφθούν υπόψη οι μη-λειτουργικές απαιτήσεις (π.χ. χρηστικότητα, αξιοπιστία, απόδοση, επεκτασιμότητα, κτλ.), και οι περιορισμοί του σχεδιασμού (π.χ. διαλειτουργικότητα, κτλ.)

- Βελτίωση των περιπτώσεων ελέγχου για λειτουργική δοκιμή
 - Οι περιπτώσεις ελέγχου μπορούν να βελτιωθούν με τις προδιαγραφές του σχεδιασμού

3.2.4 Η φάση της κωδικοποίησης

Η φάση της κωδικοποίησης, αλλά και του εντοπισμού σφαλμάτων (*debugging*) για τις ευέλικτες διαδικτυακές εφαρμογές, ουσιαστικά έχει αρκετές ομοιότητες με την φάση της κωδικοποίησης του λογισμικού γενικότερα. Μετά την ολοκλήρωση της φάσης του σχεδιασμού, έχουν ληφθεί οι περισσότερες από τις πιο σημαντικές αποφάσεις σχετικά με τη διαδικτυακή εφαρμογή. Ο στόχος της φάσης κωδικοποίησης, είναι να μεταφράσει το σχεδιασμό της εφαρμογής σε κώδικα σε μία συγκεκριμένη γλώσσα προγραμματισμού. Για ένα συγκεκριμένο σχέδιο, ο στόχος αυτής της φάσης είναι η εφαρμογή του σχεδιασμού με τον καλύτερο δυνατό τρόπο. Η φάση κωδικοποίησης επηρεάζει τόσο τον έλεγχο όσο και τη συντήρηση. Ένας καλά γραμμένος κώδικας μειώνει τον έλεγχο και την προσπάθεια συντήρησης. Η απλότητα και η σαφήνεια είναι δύο πολύ σημαντικοί παράγοντες που πρέπει να επιχειρούνται κατά την φάση της κωδικοποίησης.

Παρακάτω περιγράφονται οι δραστηριότητες που πραγματοποιούνται σε αυτή τη φάση

- Κατασκευή κώδικα για την ευέλικτη διαδικτυακή εφαρμογή
 - Με βάση την επιλογή της γλώσσας προγραμματισμού, υλοποιείται ο κώδικας της διαδικτυακής εφαρμογής σύμφωνα με το σχεδιασμό. Για την επιλογή της γλώσσας προγραμματισμού πρέπει να ληφθούν υπόψη και άλλοι περιορισμοί που επιβάλλονται από την ίδια τη γλώσσα.
 - Έκθεση δημόσιων APIs όπως η διασύνδεση της εφαρμογής.

3.2.5 Η φάση του ελέγχου

Για τις δοκιμές των ευέλικτων διαδικτυακών εφαρμογών, εκτός από τον έλεγχο για τη λειτουργική ορθότητα και πληρότητα, πρέπει να διεξαχθούν δοκιμές λειτουργικότητας μεταξύ των διαφόρων πλατφόρμων και των προγραμμάτων των πελατών. Επιπλέον, πρέπει να διενεργείται και ο έλεγχος των επιδόσεων για να διασφαλιστεί ότι η διαδικτυακή εφαρμογή είναι σε θέση να αντέξει το φορτίο, όπως αυτό ορίστηκε στο πλαίσιο των μη-λειτουργικών προδιαγραφών απαιτήσεων. Άλλες διεργασίες, όπως για παράδειγμα το προφίλ της διαδικτυακής εφαρμογής, πρέπει να εκτελούνται σε αυτή τη φάση.

Στις ευέλικτες διαδικτυακές εφαρμογές, μπορούν να διεξάγονται πρόσθετοι έλεγχοι για την διασφάλιση της διαλειτουργικότητας, της ασφάλειας και της επεκτασιμότητας. Είναι αναγκαία μία δοκιμή επιδόσεων για τη λειτουργία πριν χρησιμοποιηθεί, ώστε να διαπιστωθεί αν μπορεί να ανταποκριθεί στις συμφωνίες σε επίπεδο υπηρεσιών.

Οι δραστηριότητες που πραγματοποιούνται σε αυτή τη φάση είναι:

- Έλεγχος λειτουργικότητας της δομής
 - Έλεγχος βασικής λειτουργικότητας. Η εφαρμογή πρέπει να ανταποκρίνεται σωστά στα αιτήματα των πελατών. Σε αυτή τη φάση πολλές φορές πραγματοποιείται λάθος έλεγχος ώστε να διαπιστωθεί πως η εφαρμογή χειρίζεται απρόβλεπτες εισόδους. Τα σενάρια ελέγχου και τα δεδομένα που προετοιμάζονται στις προηγούμενες φάσεις, εκτελούνται σε αυτή τη δραστηριότητα. Τα αποτελέσματα των δοκιμών πρέπει να καταγράφονται, καθώς επίσης να γίνεται αναφορά των σφαλμάτων ώστε να διορθωθούν από τον προγραμματιστή
 - Έλεγχος ασφάλειας. Τυχόν ανεπάρκειες που μπορεί να οδηγήσουν σε πιθανές παραβιάσεις της ασφάλειας πρέπει να αναφέρονται και να επιλύονται

- Έλεγχος ενσωμάτωσης στη διαδικτυακή εφαρμογή
 - Εκτέλεση του ελέγχου διαλειτουργικότητας που βασίζεται σε διάφορα σενάρια

- Εκτέλεση του ελέγχου ολοκλήρωσης. Με βάση τις περιπτώσεις ελέγχου που προετοιμάζονται στη φάση της ανάλυσης, τα σενάρια και τα δεδομένα ελέγχου τα οποία κατασκευάζονται, εκτελούνται και αναλύονται σε αυτή τη δραστηριότητα
- Έλεγχος του συστήματος
- Έλεγχος της λειτουργικότητας του συστήματος και του χρόνου απόκρισης. Οι περιπτώσεις ελέγχου που έχουν συνταχθεί στις προηγούμενες φάσεις εκτελούνται σε αυτή τη δραστηριότητα. Τα αποτελέσματα των ελέγχων πρέπει να αναλύονται ώστε να καθορίζεται τα πιθανά σημεία συμφόρησης και αν το σύστημα είναι επεκτάσιμο
 - Έλεγχος της λειτουργικότητας και του χρόνου απόκρισης κάτω από διαφορετικούς συνδυασμούς έγκυρων και μη αιτήσεων. Τα αποτελέσματα από την εκτέλεση των ελέγχων πρέπει να αναλύονται για να καθορίσουν εάν το σύστημα μπορεί να καταστήσει την αναμενόμενη ποιότητα των υπηρεσιών, όπως αυτή ορίζεται στις προδιαγραφές των μη-λειτουργικών απαιτήσεων
- Έλεγχος αποδοχής της ευέλικτης διαδικτυακής εφαρμογής από τους χρήστες
- Εδώ γίνεται η εκτέλεση των περιπτώσεων ελέγχου για την αποδοχή της εφαρμογής από τους χρήστες. Οι περιπτώσεις ελέγχου που έχουν προετοιμαστεί κατά τη φάση των απαιτήσεων, χρησιμοποιούνται σε αυτή τη δραστηριότητα για να επικυρώσουν την ορθότητα και τη πληρότητα της εφαρμογής. Τυχόν σφάλματα πρέπει να αναφέρονται και να διορθώνονται άμεσα

3.2.6 Η φάση της ανάπτυξης

Ο σκοπός αυτής της φάσης είναι η διασφάλιση της σωστής ανάπτυξης της διαδικτυακής εφαρμογής. Η φάση θα εκτελέσει την δοκιμή της εφαρμογής. Τα τελικά σημεία εξυπηρέτησης της εφαρμογής, καθορίζονται όταν αυτή έχει αναπτυχθεί και χρειάζεται να προσδιορισθούν και να διαμορφωθούν ανάλογα. Το κύριο καθήκον της φάσης είναι να διασφαλιστεί ότι η εφαρμογή είναι κατάλληλα διαμορφωμένη, διαχειρίζεται και τρέχει μετά από την εγκατάσταση των δοκιμών, ώστε να είναι έτοιμη για χρήση και να παραδοθεί στον πελάτη.

- Προετοιμασία του περιβάλλοντος εγκατάστασης
 - Εγκατάσταση και διαμόρφωση του υλικού (*hardware*) για την ανάπτυξη της εφαρμογής
 - Εγκατάσταση και διαμόρφωση του λογισμικού (*software*) για την ανάπτυξη. Το λογισμικό μπορεί να περιλαμβάνει τον server, βάσεις δεδομένων και άλλα.

- Ανάπτυξη της εφαρμογής
 - Καθορισμός του URL της εφαρμογής. Η URL διεύθυνση είναι μοναδική και καθορίζει την τοποθεσία
 - Προετοιμασία του σεναρίου εγκατάστασης
 - Εκτέλεση της εφαρμογής

- Δημιουργία υλικού οδηγιών χρήσης για τους τελικούς χρήστες

- Δημοσίευση της ευέλικτης διαδικτυακής εφαρμογής

- Συντήρηση της ευέλικτης διαδικτυακής εφαρμογής

3.3 Επίλογος

Σύμφωνα με τα παραπάνω καταλαβαίνουμε πως η ανάπτυξη μίας ευέλικτης διαδικτυακής εφαρμογής δεν είναι μία απλή υπόθεση. Ιδίως όταν ασκείται μεγάλη πίεση από την άποψη του χρόνου, τις συνεχείς μεταβολές των διαδικτυακών τεχνολογιών, αλλά και τις συνεχώς μεταβαλλόμενες απαιτήσεις των πελατών.

Ευέλικτη διαδικτυακή εφαρμογή δεν σημαίνει μόνο κατασκευή ιστοσελίδας. Μπορεί να αποτελεί ένα πολύπλοκο σύστημα που βασίζεται σε μία ποικιλία υλικού και λογισμικού, πρωτόκολλα, γλώσσες προγραμματισμού και πρότυπα. Διαδικτυακή εφαρμογή χαρακτηρίζεται μία οποιαδήποτε εφαρμογή λογισμικού η οποία εξαρτάται από το διαδίκτυο για την ορθή εκτέλεσή της. Επίσης, ως διαδικτυακές εφαρμογές ορίζονται διάφορα συστήματα πληροφοριών που χρησιμοποιούν το διαδίκτυο για την ενσωμάτωση και τον ορθολογισμό των επιχειρηματικών διαδικασιών σε ολόκληρη την οργάνωσή τους (πελάτες, αντιπροσώπους, προμηθευτές και άλλους) αλλά και σε εκτενή γεωγραφικά σύνορα.

Παρόλα αυτά όταν ο κύκλος ζωής μίας ευέλικτης διαδικτυακής εφαρμογής είναι καλά ορισμένος, και η αναπτυξιακή ομάδα συνεργάζεται άμεσα και συνεχώς με τον πελάτη και τους τελικούς χρήστες, το έργο της ανάπτυξης της γίνεται ευκολότερο.

ΚΕΦΑΛΑΙΟ 4

Το εργαλείο Eclipse

4.1 Εισαγωγή

Το Eclipse είναι ένα πολύ-γλωσσικό (*multi-language*) περιβάλλον ανάπτυξης λογισμικού, που αποτελείται από ένα ολοκληρωμένο περιβάλλον ανάπτυξης (IDE) και ένα επεκτάσιμο plug-in σύστημα. Είναι γραμμένο κυρίως σε Java και μπορεί να χρησιμοποιηθεί για την ανάπτυξη εφαρμογών σε Java και, μέσω των διαφόρων plug-ins άλλες γλώσσες προγραμματισμού, συμπεριλαμβανομένων Ada, C, C + +, COBOL, Perl, PHP, Python, R, Ruby, Scala, Clojure, Groovy και Scheme. Το IDE συχνά αποκαλείται Eclipse ADT (Ada Εργαλειοθήκη Ανάπτυξης) για Ada, Eclipse CDT για C / C + +, Eclipse JDT για Java και Eclipse PDT για την PHP. Η αρχική βάση κώδικα (*codebase*) προήλθε από την VisualAge. Στην προεπιλεγμένη φόρμα του, προορίζεται για προγραμματιστές Java, και αποτελείται από Java εργαλεία ανάπτυξης (*JDT*). Οι χρήστες μπορούν να επεκτείνουν τις ικανότητές του με την εγκατάσταση plug-ins που γράφτηκαν για το πλαίσιο λογισμικού Eclipse, όπως εργαλειοθήκες ανάπτυξης, για τις άλλες γλώσσες προγραμματισμού, και μπορεί να συμβάλλει στις δικές τους plug-in ενότητες. Κυκλοφόρησε κάτω από τους όρους της Δημόσιας Άδειας Eclipse. Το Eclipse είναι δωρεάν λογισμικό ανοιχτού κώδικα.

4.1.1 Ιστορικά

Το Eclipse ξεκίνησε ως ένα έργο της IBM το 2001 στον Καναδά. Αναπτύχθηκε από την Object Technology International (OTI) ως μία Java-based εφαρμογή αντικατάστασης του Smalltalk based VisualAge family για IDE προϊόντα. Τον Νοέμβριο του 2001, είχε συσταθεί μια κοινοπραξία για να προωθήσει την ανάπτυξη του Eclipse ως εργαλείο ανοιχτού κώδικα (*open source*). Έτσι τον Ιανουάριο του 2004, δημιουργήθηκε το Eclipse. Στο Eclipse 3.0 (που κυκλοφόρησε στις 21 Ιουνίου 2004) επιλέγονται OSGI προδιαγραφές πλατφόρμας υποστήριξης, για runtime αρχιτεκτονική. Το Eclipse αρχικά διατέθηκε βάση της Κοινής Δημόσιας Χρήσης, αλλά αργότερα επανεκδόθηκε με την EPL (*Eclipse Public License*) Άδεια Δημόσιας Χρήσης Eclipse. Η κοινότητα Eclipse χρηματοδοτείται με ετήσια τέλη από τα μέλη της, και διέπεται από ένα Διοικητικό

Συμβούλιο. Χρησιμοποιεί πλήρους απασχόλησης επαγγελματικό προσωπικό για την παροχή υπηρεσιών στο κοινωνικό σύνολο, αλλά δεν απασχολεί προσωπικό για την ανάπτυξη έργων ανοιχτού κώδικα. Οι `commiters`, είναι αυτοί οι οποίοι αναπτύσσουν έργα Eclipse και συνήθως χρησιμοποιούνται από τις οργανώσεις, ή είναι ανεξάρτητοι προγραμματιστές που προσφέρουν εθελοντικά το χρόνο τους για να εργαστούν σε ένα έργο ανοικτού πηγαίου κώδικα.

4.1.2 Αρχιτεκτονική της πλατφόρμας Eclipse

Το Eclipse είναι μια πλατφόρμα που έχει σχεδιαστεί από την αρχή για να χτίζονται πάνω της ολοκληρωμένα εργαλεία ανάπτυξης εφαρμογών διαδικτύου, και εφαρμογών λογισμικού. Από το σχεδιασμό της, η πλατφόρμα δεν παρέχει μεγάλο επίπεδο χρηστικότητας από μόνη της. Η αξία της βρίσκεται στο γεγονός ότι ενθαρρύνει την ανάπτυξη επιπλέον δυνατοτήτων βασισμένων σε ένα μοντέλο από `plug-ins`. Το Eclipse προσφέρει μια κοινή διαπροσωπεία προς το χρήστη (*User Interface*) για να δουλεύει με εργαλεία. Είναι σχεδιασμένο να λειτουργεί σε πολλά διαφορετικά λειτουργικά συστήματα, ενώ ταυτόχρονα παρέχει στιβαρή ολοκλήρωση με το κάθε λειτουργικό σύστημα που βρίσκεται κάτω από αυτό. Τα `plug-ins` μπορούν να στηρίζονται στα φορητά APIs του Eclipse και έτσι να λειτουργούν αμετάβλητα σε οποιοδήποτε από τα υποστηριζόμενα λειτουργικά συστήματα. Στον πυρήνα του Eclipse βρίσκεται μια αρχιτεκτονική για δυναμική αναζήτηση, φόρτωση και εκτέλεση των `plug-ins`. Η πλατφόρμα χειρίζεται τις λειτουργίες της εύρεσης και εκτέλεσης του σωστού κώδικα, και διαπροσωπεία χρήστη της πλατφόρμας παρέχει ένα τυποποιημένο μοντέλο πλοήγησης. Κάθε `plug-in` μπορεί στη συνέχεια να εστιάσει στην σωστή υλοποίηση ενός μικρού αριθμού εργασιών. Παραδείγματα τέτοιων εργασιών είναι η κωδικοποίηση, ο έλεγχος, η μεταγλώττιση, η αποσφαλμάτωση, η δημιουργία διαγραμμάτων, κτλ.

4.1.3 Ανοιχτή αρχιτεκτονική

Η πλατφόρμα Eclipse ορίζει μια ανοιχτή αρχιτεκτονική έτσι ώστε η κάθε ομάδα ανάπτυξης plug-in να μπορεί να εστιάζει στη δική της περιοχή εξειδίκευσης. Η ιδέα είναι ότι οι ειδικοί στον τομέα τους χτίζουν το οπίσθιο μέρος, και οι ειδικοί στην χρηστικότητα χτίζουν τα εργαλεία που θα χρησιμοποιήσει ο τελικός χρήστης (τα *plug-ins*). Η πλατφόρμα Eclipse χρησιμοποιεί το μοντέλο του πάγκου εργασίας (*workbench*) για να ενσωματώνει τα εργαλεία από τη σκοπιά του τελικού χρήστη. Εργαλεία που αναπτύσσει ο καθένας μπορούν να ενσωματωθούν στον πάγκο εργασίας με τη χρήση καλά ορισμένων διεπαφών που ονομάζονται σημεία επέκτασης (*extension points*).

Και η ίδια η πλατφόρμα είναι χτισμένη σε στρώματα από plug-ins, καθένα από τα οποία ορίζει επεκτάσεις πάνω σε σημεία επεκτάσεων των plug-ins του χαμηλότερου στρώματος. Με τη σειρά τους και αυτά τα plug-ins ορίζουν τα δικά τους σημεία επέκτασης για να παρέχουν τη δυνατότητα περαιτέρω προσαρμογής από plug-ins ανώτερων στρωμάτων. Αυτό το μοντέλο επέκτασης επιτρέπει σε ομάδες ανάπτυξης plug-ins να προσθέτουν πολλές και διάφορες λειτουργίες στη βασική πλατφόρμα εργαλείων. Τα διάφορα τεχνήματα για κάθε εργαλείο, όπως αρχεία και άλλα δεδομένα, συντονίζονται από ένα κοινό για όλη την πλατφόρμα μοντέλο διαχείρισής τους.

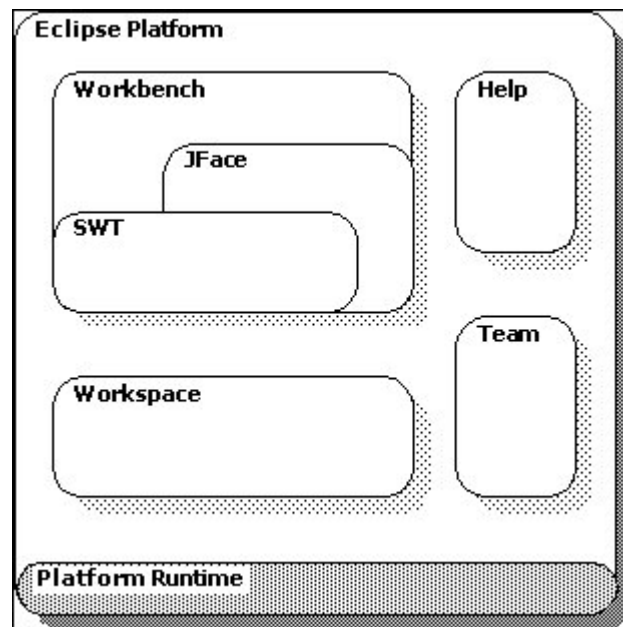
Η πλατφόρμα δίνει στους χρήστες έναν κοινό τρόπο να δουλεύουν με τα εργαλεία και παρέχει διαχείριση των πόρων (*resources*) που αυτοί δημιουργούν με τα plug-ins. Οι ομάδες ανάπτυξης των plug-ins επωφελούνται επίσης από αυτή την αρχιτεκτονική, καθώς η πλατφόρμα διαχειρίζεται την περιπλοκότητα των διαφορετικών περιβαλλόντων εκτέλεσης, για παράδειγμα των διαφορετικών λειτουργικών συστημάτων.

4.1.4 Αρχική μορφή της πλατφόρμας

Η βασική πλατφόρμα που παρέχεται στο χρήστη αρχικά είναι ένα ολοκληρωμένο περιβάλλον ανάπτυξης (Integrated Development Environment). Τα plug-ins είναι αυτά που καθορίζουν την τελική λειτουργικότητα της πλατφόρμας. Τα νέα plug-ins μπορούν να παρέχουν υποστήριξη για επεξεργασία και διαχείριση επιπρόσθετων τύπων πόρων, όπως αρχεία Java, προγράμματα C, σελίδες HTML και αρχεία JSP.

4.1.5 Η δομή της πλατφόρμας

Η πλατφόρμα είναι δομημένη σαν υποσυστήματα που υλοποιούνται από ένα ή περισσότερα plug-ins το καθένα. Τα υποσυστήματα είναι χτισμένα πάνω σε ένα μικρό πυρήνα χρόνου εκτέλεσης (Platform runtime). Το σχήμα 4.1 δείχνει μια απλοποιημένη όψη.



Σχήμα 4.1: Δομή της πλατφόρμας Eclipse

Τα plug-ins που απαρτίζουν ένα υποσύστημα ορίζουν σημεία επέκτασης για πρόσθεση νέας συμπεριφοράς στο σύστημα.

Ακολουθεί ανάλυση των σημαντικότερων ψηφίδων της πλατφόρμας

4.1.5.1 Χώροι Εργασίας (Workspaces)

Τα διάφορα εργαλεία (plug-ins) που λειτουργούν στην πλατφόρμα Eclipse ενεργούν πάνω σε αρχεία που βρίσκονται στο χώρο εργασίας (*workspace*) του χρήστη. Ένας χώρος εργασίας αποτελείται από ένα ή περισσότερα ανώτατα έργα (*top-level projects*), όπου κάθε έργο (*project*) αντιστοιχεί σε ένα υποκατάλογο (*folder*) στο τοπικό σύστημα αρχείων ο οποίος έχει όνομα ορισμένο από το χρήστη (*user-defined*). Οι όροι που χρησιμοποιούνται στους χώρους εργασίας είναι πόροι όπως έργα αρχεία και υποκατάλογοι, μηχανισμοί καταγραφής αλλαγών (*history mechanisms*), σημειώματα (*markers*) όπως bookmarks, debugger breakpoints, to-do list items κτλ.

4.1.5.2 Workbench και UI Toolkits (SWT και JFace)

Το περιβάλλον εργασίας της πλατφόρμας Eclipse έχει δημιουργηθεί γύρω από έναν πάγκο εργασίας (*workbench*) ο οποίος παρέχει την γενική δομή και παρουσιάζει ένα επεκτάσιμο περιβάλλον στο χρήστη. Ο πάγκος εργασίας (*workbench*) έχει δημιουργηθεί μέσω της χρήσης δύο εργαλείων, του Standard Widget Toolkit (SWT) και του JFace. Το SWT παρέχει ένα κοινό προς τους χρήστες και ανεξάρτητο από το λειτουργικό σύστημα API για widgets και γραφικά δομικά στοιχεία (*components*) υλοποιημένα με τέτοιο τρόπο ώστε να επιτρέπεται η στενή ενσωμάτωσή τους με το τοπικό λειτουργικό σύστημα. Η στενή ενσωμάτωση με το τοπικό λειτουργικό σύστημα πάνω στο οποίο λειτουργεί η πλατφόρμα Eclipse δεν είναι μόνο ζήτημα αισθητικής (π.χ. χρήση του γηγενούς συστήματος γραφικών look and feel). Το SWT αλληλεπιδρά με λειτουργίες του τοπικού συστήματος όπως είναι το Drag and Drop (DND), και μπορεί να χρησιμοποιήσει δομικά στοιχεία (*components*) που έχουν αναπτυχθεί. Ολόκληρο το περιβάλλον εργασίας της πλατφόρμας Eclipse και των εργαλείων που ενσωματώνονται σε αυτήν χρησιμοποιούν το SWT για την παρουσίαση πληροφοριών στον τελικό χρήστη.

Το πλαίσιο εργασίας JFace είναι ένα εργαλείο για την υποστήριξη των διεπαφών χρήστη (*User Interface*) που αποτελείται από κλάσεις για την διεκπεραίωση κοινών προγραμματιστικών λειτουργιών σε σχέση με αυτό. Το JFace είναι

ανεξάρτητο από το τοπικό λειτουργικό σύστημα τόσο στο API που ορίζει όσο και στην υλοποίησή του, και είναι σχεδιασμένο να συνεργάζεται με το SWT χωρίς να το καλύπτει. Περιλαμβάνει συνηθισμένα στοιχεία διεπαφών όπως είναι τα μητρώα εικόνων και γραμματοσειρών (*image and font registries*), τα παράθυρα διαλόγου (*dialogs*), οι προτιμήσεις εμφάνισης (*preferences*), τα wizard frameworks, και οι αναφορές προόδου για μακροχρόνιες λειτουργίες.

Τα πιο ενδιαφέροντα χαρακτηριστικά του JFace είναι οι μηχανισμοί που παρέχει για τον καθορισμό ενεργειών (*actions*) και μηχανισμών προβολής (*viewers*). Ο μηχανισμός ενεργειών επιτρέπει το ορισμό εντολών από το χρήστη ανεξάρτητα από την τοποθεσία στην οποία αυτή έχει ισχύ μέσα στο περιβάλλον εργασίας. Μία ενέργεια (*action*) αναπαριστά μία εντολή η οποία μπορεί να ενεργοποιηθεί από το χρήστη μέσω ενός κουμπιού (*button*), μιας επιλογής κάποιου μενού (*menu item*), ή ενός στοιχείου σε κάποια μπάρα εργαλείων (*tool bar*).

Οι μηχανισμοί προβολής (*viewer*) αποτελούν προσαρμογείς (*adapters*), βασισμένους σε μοντέλα (*model-based*), κάποιων συγκεκριμένων SWT widgets. Οι μηχανισμοί αυτοί διεκπεραιώνουν την κοινή (γνώριμη) συμπεριφορά των στοιχείων και παρέχουν υψηλότερου επιπέδου έννοιες από αυτές που είναι διαθέσιμες από τα SWT widgets. Παραδείγματα τέτοιων μηχανισμών είναι οι μηχανισμοί προβολής λιστών (*list viewers*), οι μηχανισμοί προβολής δένδρων (*tree viewers*), οι μηχανισμοί προβολής πινάκων (*table viewers*) κτλ.

Η επιφάνεια εργασίας (*workbench*), σε αντίθεση με τα πλαίσια εργασίας SWT και JFace τα οποία είναι γενικής χρήσης εργαλεία διεπαφών, διαμορφώνει την προσωπικότητα του περιβάλλοντος εργασίας της πλατφόρμας Eclipse, και παρέχει τις δομές με τις οποίες τα εργαλεία (*plug-ins*) αλληλεπιδρούν με τον χρήστη. Για τον λόγο αυτό το workbench έχει ταυτιστεί σαν έννοια με το περιβάλλον εργασίας της πλατφόρμας Eclipse. Από πλευράς χρήστη ένα παράθυρο του workbench αποτελείται οπτικά από περιοχές προβολής (*views*) και περιοχές επεξεργασίας (*editors*). Με τον όρο perspective (όψη) εννοείται μία προκαθορισμένη διάταξη από τέτοιες περιοχές οι οποίες είναι τοποθετημένες σε προκαθορισμένες θέσεις. Στην πλατφόρμα Eclipse μπορούν να οριστούν πολλά διαφορετικά perspectives και είναι δυνατό την ίδια χρονική στιγμή να είναι

ανοιγμένα περισσότερα του ενός. Ωστόσο μόνο ένα perspective μπορεί να είναι ενεργό σε κάθε χρονική στιγμή. Στο perspective αυτό ορίζονται περιοχές προβολής (*views*) και επεξεργασίας (*editors*) που χρησιμοποιούνται από το εργαλείο καθώς και οι θέσεις τους στο περιβάλλον εργασίας, αποκρύπτοντας την ίδια στιγμή περιοχές προβολής (*views*) ή επεξεργασίας (*editors*) που δεν θα ήταν χρήσιμες για το εργαλείο. Οι περιοχές προβολής (*views*) και οι περιοχές επεξεργασίας (*editors*) έχουν τη μορφή παραθύρων (*windows*).

Οι περιοχές επεξεργασίας (*editors*) επιτρέπουν στο χρήστη το άνοιγμα, την επεξεργασία και το σώσιμο αντικειμένων. Όταν μια τέτοια περιοχή είναι ενεργή, μπορεί να συνεισφέρει διάφορες επιλογές ενεργειών (*actions*) στα μενού (*menus*) και στη μπάρα εργαλείων (*tool bar*) της επιφάνειας εργασίας. Οι περιοχές επισκόπησης (*views*) παρέχουν πληροφορίες για το αντικείμενο με το οποίο ασχολείται ο χρήστης στην επιφάνεια εργασίας και παρέχουν στον χρήστη τη δυνατότητα τροποποίησης του αντικειμένου αυτού. Τροποποιήσεις που γίνονται σε μία περιοχή επισκόπησης (όπως αλλάζοντας την τιμή ενός χαρακτηριστικού του υπό-επεξεργασία αντικειμένου) γενικά σώζονται άμεσα, και οι αλλαγές αντανακλώνται αμέσως στα τμήματα του περιβάλλοντος εργασίας τα οποία επηρεάζονται από την αλλαγή αυτή.

4.1.5.3 Υποστήριξη Ομαδικής Εργασίας (Team Support)

Η πλατφόρμα Eclipse επιτρέπει σε έργα (*projects*) που βρίσκονται μέσα στο χώρο εργασίας (*workspace*) να έχουν συγκεκριμένες εκδόσεις (*versions*) και παρέχει τη δυνατότητα διαχείρισης των ιδιοτήτων τους μέσω της χρήσης κατάλληλων μηχανισμών αποθήκευσης διαμοιραζόμενων πόρων (*team repositories*). Η πλατφόρμα διαθέτει σημεία επέκτασης και ένα API ορισμού repository, τα οποία επιτρέπουν τον ορισμό νέων team repositories. Γενικά, είναι δυνατή η συνύπαρξη πολλαπλών τέτοιων μηχανισμών (από διαφορετικούς προμηθευτές πιθανόν) στην πλατφόρμα Eclipse. Επίσης, η πλατφόρμα περιλαμβάνει την υποστήριξη μηχανισμών ομαδικής ανάπτυξης κώδικα (Concurrent Versions System / CVS)

που είναι προσβάσιμοι μέσω πρωτοκόλλων είτε `rserver` είτε `ssh`. Με τη χρήση αυτών των μηχανισμών ένα έργο (*project*) μπορεί να τροποποιηθεί ή να σχολιαστεί από τα μέλη της ομάδας του συγκεκριμένου αποθηκευτικού χώρου τα οποία μπορεί να βρίσκονται ακόμα και σε διαφορετικές χώρες.

4.1.5.4 Μηχανισμός Βοήθειας (Help)

Ο μηχανισμός βοήθειας της πλατφόρμας Eclipse επιτρέπει στα εργαλεία να ορίζουν και να συνεισφέρουν τεκμηρίωση (*documentation*) σε ένα ή περισσότερα online εγχειρίδια. Για παράδειγμα, ένα εργαλείο συνήθως συνεισφέρει τεκμηρίωση με την μορφή ενός οδηγού για το χρήστη και τεκμηρίωση των διεπαφών προγραμματισμού (API) (αν περιλαμβάνει κάτι τέτοιο) σε ένα ξεχωριστό οδηγό για τον προγραμματιστή. Το PDE (*Plug-in Development Environment / PDE*) αποτελεί το ενυπάρχων (*built-in*) *perspective* της πλατφόρμας που χρησιμοποιείται για την υλοποίηση *plug-ins*.

4.1.6 Επέκταση της πλατφόρμας με *plug-ins*

Όπως αναφέρθηκε και παραπάνω, η πλατφόρμα Eclipse είναι δομημένη ως εξής: στον πυρήνα υπάρχει μια μηχανή χρόνου εκτέλεσης (*core runtime engine*) και πάνω της υπάρχει ένα σύνολο επιπρόσθετων λειτουργιών που εγκαθίστανται σαν *plug-ins*. Τα *plug-ins* συνεισφέρουν λειτουργικότητα στην πλατφόρμα σε προκαθορισμένα σημεία επέκτασης. Ένα παράδειγμα ενός τέτοιου *plug-in* είναι και ο πάγκος εργασίας. Η γλώσσα στην οποία είναι υλοποιημένο το Eclipse είναι η Java. Όταν όμως εκκινείτε ο πάγκος εργασίας, δεν εκκινείτε ένα απλό πρόγραμμα Java. Ενεργοποιείται η μηχανή εκτέλεσης, η οποία μπορεί δυναμικά να ανακαλύπτει εγκατεστημένα *plug-ins* και να τα εκκινεί όποτε χρειάζεται. Από τα παραπάνω προκύπτει ότι όταν κάποιος θέλει να γράψει κώδικα ο οποίος επεκτείνει την πλατφόρμα, η διαδικασία που πρέπει να ακολουθήσει είναι διαφορετική από τη διαδικασία ανάπτυξης μιας απλής εφαρμογής Java. Συγκεκριμένα, θα πρέπει να ορίσει επεκτάσεις μέσα στο *plug-in* του, οι οποίες να συνδέονται σε κάποια από τα σημεία επεκτάσεων που έχει ορίσει η πλατφόρμα. Από τη σκοπιά της πλατφόρμας, το *plug-in* που γράφει κάποιος τρίτος δε διαφέρει

από τα βασικά plug-ins, σαν αυτό που χειρίζεται τους πόρους (resources) ή τον ίδιο τον πάγκο εργασίας.

Τα βήματα που πρέπει να ακολουθήσουμε είναι:

- Απόφαση για τον τρόπο με τον οποίο το plug-in θα ενσωματωθεί στην πλατφόρμα.
- Εύρεση των κατάλληλων σημείων επέκτασης στα οποία χρειάζεται να συνεισφέρουμε ώστε να ενσωματωθεί το plug-in.
- Υλοποίηση αυτών των επεκτάσεων σύμφωνα με την προδιαγραφή για τα σημεία επέκτασης.
- Δημιουργία ενός αρχείου manifest (MANIFEST.MF) το οποίο περιγράφει το πακετάρισμα και τα προαπαιτούμενα για τον κώδικά μας, και ενός plug-ins manifest (plug-in.xml) το οποίο περιγράφει τις επεκτάσεις που ορίζουμε.

4.1.7 Επίλογος

Η πλατφόρμα Eclipse παρέχει έναν πυρήνα από δομικά στοιχεία γενικού περιεχομένου και ένα σύνολο διεπαφών προγραμματισμού (APIs) όπως είναι ο χώρος εργασίας (*workspace*) και η επιφάνεια εργασίας (*workbench*), και διάφορα σημεία επέκτασης μέσω των οποίων είναι δυνατή η ενσωμάτωση νέων λειτουργιών. Μέσω αυτών των σημείων επέκτασης, εργαλεία υλοποιημένα σαν ξεχωριστά επιπρόσθετα (plug-ins) μπορούν να επεκτείνουν τη βασική πλατφόρμα Eclipse. Στον τελικό χρήστη παρουσιάζεται ένα περιβάλλον ανάπτυξης εφαρμογών (IDE) εξειδικευμένο με βάση το σύνολο των διαθέσιμων επιπρόσθετων (plug-in) εργαλείων. Ωστόσο, οι δυνατότητες επέκτασης δεν σταματούν εδώ. Τα εργαλεία μπορούν να ορίσουν νέα δικά τους σημεία επέκτασης και δικές τους διεπαφές προγραμματισμού (APIs) και με τον τρόπο αυτό να αποκτούν τη μορφή δομικών υλικών και να αποτελούν σημεία ενσωμάτωσης άλλων εργαλείων.

4.2 Εργαλεία και πλαίσια OSGI για την υλοποίηση εφαρμογών desktop - web - SOA- enterprise με συστατικά

Το Eclipse είναι γνωστό ως μία πηγή από αξιόπιστα εργαλεία ανάπτυξης, αλλά παρέχει επίσης μία ευρεία επιλογή run time, του ενδιάμεσο λογισμικού (*middleware*), και του πλαισίου των επιχειρήσεων. Το EclipseRT (*Run Time*) είναι το όνομα που δόθηκε σε αυτό το portfolio των Eclipse runtime έργων. Οι προγραμματιστές μπορούν να χρησιμοποιήσουν την τεχνολογία EclipseRT για να κατασκευάσουν πλούσια desktop, web, επιχειρήσεις, και ενσωματωμένες εφαρμογές, καθώς και αρχιτεκτονικές SOA (Service - oriented architectures).

Το EclipseRT βασίζεται στην έννοια του στοιχείου με προσανατολισμό την ανάπτυξη και τη συναρμολόγηση, ή CODA. Μία βασική πτυχή της CODA και του EclipseRT, είναι η δυνατότητα να συγκεντρώσουν και να επεκτείνουν διαφορετικά συστατικά για να καλυφθούν οι άμεσες ανάγκες ενός συγκεκριμένου τύπου εφαρμογής. Παρακάτω δείχνουμε πως το EclipseRT μπορεί να βοηθήσει στην εφαρμογή των CODA αρχών σε μερικές από τις πιο δημοφιλείς αρχιτεκτονικές εφαρμογών, όπως desktop εφαρμογές, web εφαρμογές, SOA και επιχειρηματικές εφαρμογές.

4.2.1 Desktop Εφαρμογές

Οι desktop εφαρμογές παρέχουν ένα πλούσιο, ταχείας ανταπόκρισης γραφικό περιβάλλον για τους τελικούς χρήστες. Αυτό το είδος της εφαρμογής επιτρέπει την ευκολότερη πρόσβαση στους desktop πόρους και την offline λειτουργία. Η Eclipse Rich Client πλατφόρμα (RCP) παρέχει ένα σύγχρονο desktop πλαίσιο, με πλούσια και ισχυρά χαρακτηριστικά που απαιτούνται για εφαρμογές υψηλής παραγωγικότητας. Επεκτείνοντας τις δυνατότητες του RCP, το έργο Rienna παρέχει ένα πλήρες πλαίσιο για τη δημιουργία νέας γενιάς client / server εφαρμογών. Εκτός από το ότι η ανάπτυξη GUI με RCP είναι ευκολότερη, το Rienna παρέχει επικοινωνίες back - end για την δημιουργία και λειτουργία συνδεδεμένων desktop εφαρμογών. Το eclipse BIRT (*Business Intelligence and Reporting Tools*) πλαίσιο αναφοράς χρησιμοποιείται συνήθως σε desktop εφαρμογές για το βαθμό αναφοράς της επιχείρησης. Για τη δημιουργία εφαρμογών με γραφικά

επεξεργασίας, το γραφικό πλαίσιο επεξεργασίας (GEF) και το Eclipse πλαίσιο μοντελοποίησης (EMF), είναι πολύτιμα για την επιτάχυνση της ανάπτυξης αυτών των τύπων εφαρμογών.

Η p2 πλατφόρμα προβλέψεων στο Equinox προσφέρει runtime πλαίσιο ενημέρωσης για την υποστήριξη συχνών χαρακτηριστικών διανομής. Δεν παρέχει απλώς μία θέση για να κατεβάσει κανείς ένα πρόγραμμα εγκατάστασης ή μία νέα έκδοση της εφαρμογής, η Equinox - driven desktop εφαρμογή μπορεί να κάνει ενημέρωση επιλογής συστατικών με p2 μέσω του δικτύου με αποτέλεσμα την ευκολότερη διαχείριση των κύκλων ενημέρωσης.

4.2.2 Web εφαρμογές

Οι web εφαρμογές είναι σαφώς ένα από τα αγαπημένα μοντέλα παράδοσης προσανατολισμένα προς τις εφαρμογές καταναλωτών. Ενώ είναι γρήγορη και εύκολη η δημιουργία τους, καθώς αυξάνονται τα αιτήματα για επέκταση και διατήρηση της εφαρμογής, αυτά μπορεί να ξεπεράσουν γρήγορα την ομάδα. Το EclipseRT παρέχει αξιόπιστο και ευέλικτο πλαίσιο για το διαδικτυακό περιβάλλον εργασίας, την back-end συνδεσιμότητα, καθώς και τα εργαλεία που υποστηρίζουν μία υγιή διαδικασία ανάπτυξης λογισμικού.

Το Eclipse παρέχει έναν ελαφρύ, υψηλών επιδόσεων web server που ονομάζεται Jetty. Χρησιμοποιείται ευρέως σε πολλά προϊόντα και υπηρεσίες, όπως Google, HP, και Cisco, όπως ένας διαδικτυακός περιέκτης στην παραγωγή και ανάπτυξη. Κατά τη διάρκεια της ανάπτυξης, η εύκολη διαχείριση της Jetty διάρθρωσης, και ο γρήγορος χρόνος εκκίνησης δίνει τη δυνατότητα γρήγορης ανατροφοδότησης του βρόγχου κώδικας - κατασκευή - ανάπτυξη. Στην πραγματικότητα είναι τόσο γρήγορα που εξαλείφουν τα τμήματα κατασκευής και ανάπτυξης και απλά τρέχουν έξω από το χώρο εργασίας για τους κύκλους δοκιμών, πράγμα που επιτρέπει στους προγραμματιστές να εστιάσουν στην άμεση διεργασία. Στην παραγωγή, το Jetty παρέχει ένα ευρύ φάσμα του προφίλ απόδοσης για διάφορα σενάρια, παρέχοντας ένα πλήρες runtime για διαδικτυακές εφαρμογές. Το Jetty ενσωματώνει δημοφιλή πλαίσια όπως το Spring, JBoss, Glassfish, ActiveMQ, και

Java διαδικτυακές τεχνολογίες. Και φυσικά το Jetty μπορεί να χρησιμοποιηθεί με την υπόλοιπη EclipseRT, όπως την Eclipse Rich Ajax πλατφόρμα και το Equinox.

Η Eclipse Rich Ajax πλατφόρμα (RAP) είναι ένα runtime πλαίσιο για εκτέλεση διαδικτυακών εφαρμογών, ένα πλήρες πλαίσιο, και το εργαλείο για την ανάπτυξη Ajax εφαρμογών. Η RAP είναι συντονισμένη για να συνεργάζεται με Jetty, με Tomcat, ή οποιοδήποτε διαδικτυακό πρότυπο. Εκτός από τις διαδικτυακές εφαρμογές, το RAP βασίζεται στο Eclipse UI πλαίσιο, έτσι μπορεί κανείς να γράψει μία εφαρμογή UI μία φορά και να καθιερωθεί ως ένα διαδικτυακό ή ένα desktop UI.

Ένα πλούσιο UI είναι ζωτικής σημασίας για κάθε σύγχρονη διαδικτυακή εφαρμογή, αλλά είναι επίσης απαραίτητη η πρόσβαση σε όλους τους τύπους δεδομένων και των back-end διαδικασιών. Το EclipseLink μπορεί να χρησιμοποιηθεί για την ενσωμάτωση δεδομένων χαρτογράφησης σε διαδικτυακές εφαρμογές, και το Eclipse πλαίσιο επικοινωνίας (ECF) μπορεί να χρησιμοποιηθεί με βάση την πρόσβαση σε back-end υπηρεσίες. Τέλος το Swordfish μπορεί να χρησιμοποιηθεί για την ένταξη σε SOA υποδομές.

4.2.3 SOA εφαρμογές

Το EclipseRT περιλαμβάνει ένα SOA πλαίσιο που ονομάζεται Swordfish. Χτισμένο σε αποδεδειγμένα ανοιχτού κώδικα συστατικά, όπως Apache ServiceMix, το Swordfish παρέχει ένα επεκτάσιμο πλαίσιο που επιτρέπει σε προγραμματιστές εφαρμογών και ολοκληρωμένων συστημάτων να δημιουργήσουν το δικό τους ESB (*enterprise service bus*) το οποίο μπορεί να είναι προσαρμοσμένο στις δικές τους ανάγκες. Άλλα συστατικά περιλαμβάνουν μία μηχανή διαδικασίας που παρέχεται από το Apache ODE για την υποστήριξη BPEL, την παρακολούθηση, καθώς και ένα μητρώο υπηρεσιών που απαιτούνται για να τρέξει ένα ESB-driven με SOA. Τα συστατικά στο Swordfish παρέχουν μία εκτενή πλατφόρμα για τη δημιουργία και την εκτέλεση service oriented αρχιτεκτονικών.

Το Eclipse πλαίσιο επικοινωνίας (ECF) παρέχεται για την ανακάλυψη, την πρόσβαση, την εφαρμογή rest-based υπηρεσιών, καθώς και πλήρη υποστήριξη του OSGI 4.2. Το ECF's μεταφορά - ανεξάρτητης αρχιτεκτονικής, παρέχει μία ποικιλία από πρωτόκολλα που πρέπει να χρησιμοποιηθούν (π.χ. http, JMS,

XMPP, multicast IP, Zeroconf, SLP, κ.α.) χωρίς τροποποίηση του κώδικα εφαρμογής που χρησιμοποιεί το πλαίσιο.

Το EclipseLink SDO προσφέρει την εφαρμογή αναφοράς της Service Data Objects, επιτρέποντας στις υπηρεσίες να περάσουν εύκολα δομημένα δεδομένα μεταξύ τους, διασταυρώνοντας τις υπηρεσίες και τα όρια της γλώσσας προγραμματισμού. Η υποδομή αυτή είναι συνδεδεμένη με το έργο Swordfish, αλλά θα μπορούσε να χρησιμοποιηθεί σε κάθε SOA αναπτυξιακή προσπάθεια.

Υπάρχουν δύο βασικά πλεονεκτήματα της χρήσης EclipseRT για SOA εφαρμογές. Το πρώτο είναι πως η στενή ενοποίηση με Eclipse εργαλεία καθιστά ευκολότερο το χτίσιμο, τη δοκιμή και την αποσφαλμάτωση νέων υπηρεσιών, και το δεύτερο είναι πως τα φυσικά συστατικά του EclipseRT παρέχουν την ευκολία ενσωμάτωσης και χρήσης άλλων EclipseRT συστατικών όπως BIRT, και EclipseLink για την ανάπτυξη νέων υπηρεσιών.

4.2.4 Enterprise εφαρμογές

Τα EclipseRT's driven φυσικά συστατικά είναι εξαιρετικά καλά για την παράδοση και την διατήρηση μεγάλων και σύγχρονων εφαρμογών των επιχειρήσεων. Τα τμήματα IT αναμένεται να παραδίδουν τις εφαρμογές πιο συχνά, παράδοση στην υπόσχεση της ευέλικτης ανάπτυξης λογισμικού. Όπως οι επιχειρηματικές απαιτήσεις εξελίσσονται με ταχείς ρυθμούς, το λογισμικό πρέπει να αλλάξει και να προχωρήσει με τον ίδιο ρυθμό. Την ίδια στιγμή, οι ομάδες ανάπτυξης πρέπει να παρέχουν σταθερότητα, ελέγχους, και προνοητικό σχεδιασμό που προέρχεται από την καλή αρχιτεκτονική των επιχειρήσεων.

Ξεκινώντας από το Equinox, ένα OSGI runtime πλαίσιο παρέχει τη βάση για μία αρχιτεκτονική των επιχειρήσεων, η οποία επιτρέπει την ειδίκευση της στοίβας των επιχειρήσεων, σύμφωνα με τις ανάγκες της εφαρμογής. Το EclipseRT portfolio δυνατοτήτων επιτρέπει την εύκολη επέκταση των προγραμματιστών, και τη συναρμολόγηση διαφορετικών πλαισίων για την ικανοποίηση των αναγκών σε υποδομές των επιχειρηματικών εφαρμογών. Το EclipseRT περιλαμβάνει τις εξής δυνατότητες:

- UI στρώματα με RCP, RAP, ή Riena
- Runtime Jetty και Equinox
- Υπηρεσίες με EclipseLink
- Υποβολή δεδομένων για υποστήριξη από BIRT
- Μοντελοποίηση δεδομένων με το πλαίσιο Eclipse Modeling framework(EMF)
- Απομακρυσμένη επικοινωνία OSGI με ECF
- SOA ενσωμάτωση με Swordfish
- Η πρόσβαση σε αδόμητη πληροφορία στην επιχείρηση μέσω λύσεων αναζήτησης με Smila

Ο συνδυασμός των EclipseRT έργων και εργαλείων, καθίσταται η ιδανική επιλογή για την ανάπτυξη επιχειρηματικών εφαρμογών.

Συμπέρασμα

Το EclipseRT παρέχει το portfolio της τεχνολογίας που επιτρέπει στους προγραμματιστές να αναπτύξουν εφαρμογές, με την επέκταση και συναρμολόγηση συστατικών. Πρόκειται για ένα νέο τρόπο ανάπτυξης εφαρμογών που βασίζονται στις αρχές της CODA, το OSGI πρότυπο παρέχει την ευελιξία στην ανάπτυξη εφαρμογών σε διαφορετικές αρχιτεκτονικές εφαρμογής όπως: desktop, web, enterprise, και SOA.

Το Eclipse έχει διαπρέψει στην παροχή cross-platform εργαλείων για πολλά χρόνια. Τώρα το EclipseRT παρέχει θεμέλια υψηλής ποιότητας για τη λειτουργία και την ανάπτυξη εφαρμογών. Η στενή ενοποίηση των Eclipse εργαλείων και του EclipseRT προσφέρει μεγάλη εμπειρία ανάπτυξης, γεγονός που καθιστά ακόμη πιο εύκολη την ανάπτυξη και τη λειτουργία κρίσιμων εφαρμογών.

4.3 Pulsar Εργαλεία ανάπτυξης mobile εφαρμογών

Pulsar είναι ένα εργαλείο ενοποιημένης πλατφόρμας για mobile προγραμματιστές. Είναι ευκολότερο να υπάρχουν τα εργαλεία και τα SDKs που χρειάζονται για την ανάπτυξη εφαρμογών κινητού. Περιλαμβάνει την πλατφόρμα eclipse, Java εργαλεία ανάπτυξης(JDT), mobile εργαλεία για Java (MJT), Mylyn και plug-in περιβάλλον ανάπτυξης (PDE). Το Pulsar διευκολύνει επίσης το κατέβασμα SDK από διαφορετικούς κατασκευαστές.

Ο σκοπός του pulsar είναι να καθοριστούν η εφαρμογή και η προώθηση μία κοινής ανάπτυξης mobile εφαρμογής. Διευκολύνει τη συνεργασία μεταξύ υπαρχόντων Eclipse έργων που ισχύουν για mobile εφαρμογές και την ανάπτυξη νέων έργων που βασίζονται σε προσδιορισμένες ανεκπλήρωτες απαιτήσεις.

Η Eclipse κοινότητα ανακοίνωσε το Pulsar σαν μία πλατφόρμα εργαλείων ανάπτυξης κινητών. Μία νέα πρωτοβουλία για τον ορισμό και τη δημιουργία ανάπτυξης πλατφόρμας εργαλείων για mobile εφαρμογές. Η πρωτοβουλία δημιουργήθηκε από την Motorola, τη Nokia, και την Genuitec. Επίσης συμμετέχοντα μέλη είναι η IBM, η RIM και η Sony Ericsson.

Το pulsar επικεντρώνεται σε τέσσερις περιοχές:

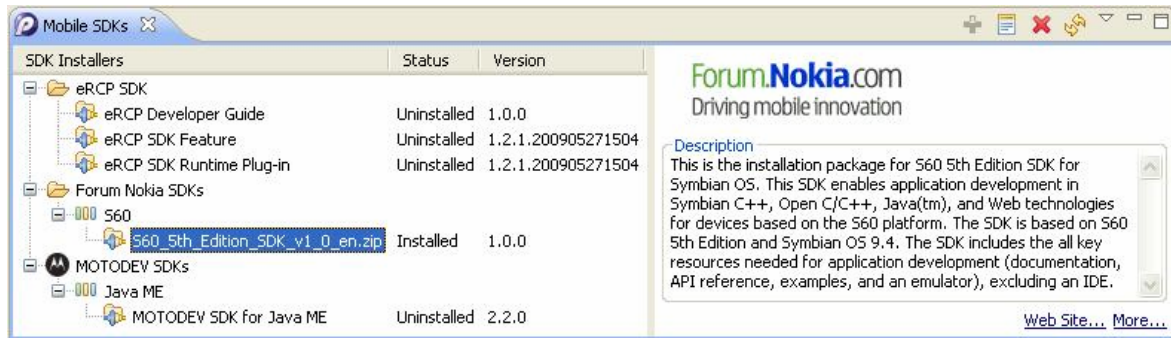
- Στη δημιουργία πακέτου διανομής που ονομάζεται Eclipse Pulsar Platform
- Σε έναν τεχνικό οδηγό για την ανάπτυξη των δυνατοτήτων της πλατφόρμας
- Σε μία συλλογή με τις καλύτερες οδηγίες
- Στην εκμάθηση της υιοθέτησης του Pulsar από προγραμματιστές εφαρμογών

Σαν κομμάτι των οδηγιών, το Pulsar υποστηρίζει ένα μεγάλο mobile περιβάλλον ανάπτυξης όπως η JavaME και διαδικτυακές mobile τεχνολογίες.

Πριν από το pulsar οι προγραμματιστές mobile απαιτούσαν τη χρήση μίας ποικιλίας από SDKs (software development kits) για να αναπτύξουν εφαρμογές από διαφορετικούς κατασκευαστές. Αυτό αποτελούσε ένα περιπτό βάρος για τους προγραμματιστές οι οποίοι θέλουν να δημιουργούν εφαρμογές για ένα εύρος συσκευών διαφορετικών κατασκευαστών. Το Pulsar αρχικοποιεί ένα κοινό σύνολο εργαλείων βασισμένων στο Eclipse, μέσα σε ένα πακέτο διανομής τα οποία λειτουργούν με αρκετά SDKs. Αυτό δίνει τη δυνατότητα στους προγραμματιστές να

μένουν σε ένα οικείο περιβάλλον ανάπτυξης καθώς δημιουργούν mobile εφαρμογές οι οποίες έχουν ως στόχο πολλαπλές οικογένειες συσκευών.

Κάθε SDK περιλαμβάνει πληροφορίες σχετικές με το πακέτο εγκατάστασης, όπως φαίνεται στην εικόνα 4.2, το τι περιλαμβάνεται σε ένα SDK, εξαρτήσεις και κατά περίπτωση πληροφορίες για την πρόσβαση.



Εικόνα 4.2: Περιγραφή των SDKs

Εν κατακλείδι το Pulsar διευκολύνει τους προγραμματιστές κινητών, όσον αφορά το κατέβασμα και την εγκατάσταση των διαφόρων εργαλείων που είναι απαραίτητα για την mobile ανάπτυξη.

4.4 Πλαίσια εφαρμογών και Εργαλεία Μοντελοποίησης

Υπάρχουν πολλά Eclipse έργα που παρέχουν πλαίσια τα οποία χρησιμοποιούνται λειτουργικά για την επιτάχυνση της διαδικασίας ανάπτυξης λογισμικού. Σε αντίθεση με προγραμματιστικά εργαλεία, τα πλαίσια εφαρμογών έχουν αναπτυχθεί με πραγματικές εφαρμογές.

Η μοντελοποίηση επικεντρώνεται στην εξέλιξη και την προώθηση του μοντέλου με βάση τις τεχνολογίες ανάπτυξης μέσα στην κοινότητα Eclipse, παρέχοντας ένα ενοποιημένο σύνολο πλαισίων μοντελοποίησης, εργαλείων και πρότυπα εφαρμογών.

Υπάρχει ένας πολύ μεγάλος αριθμός πλαισίων και εργαλείων μοντελοποίησης που έχουν αναπτυχθεί μέσα από Eclipse έργα. Ορισμένα, που είναι και τα σημαντικότερα αναλύονται παρακάτω:

4.4.1 Eclipse Modeling Framework

Το Eclipse Modeling Framework (EMF) είναι ένα πλαίσιο μοντελοποίησης και παραγωγής κώδικα για τη δημιουργία εργαλείων και άλλων εφαρμογών, βασισμένο σε δομή δεδομένων μοντέλου. Από ένα μοντέλο προδιαγραφών σε XMI, το EMF παρέχει εργαλεία και run-time υποστήριξη για την παραγωγή ενός συνόλου Java κλάσεων για το μοντέλο, μαζί με ένα σύνολο μετατροπών κλάσεων, που επιτρέπουν τη προβολή και τη σύνταξη εντολών του μοντέλου, και ένα βασικό πρόγραμμα επεξεργασίας. Οι δημιουργίες του EMF περιλαμβάνουν XML Schema Definition (XSD), ένα συστατικό από το Model Development Tools (MDT), και ένα EMF βασισμένο στην εφαρμογή του Service Data Objects (SDO). Το XSD παρέχει ένα μοντέλο και API, για τον χειρισμό των συστατικών ενός XML σχήματος.

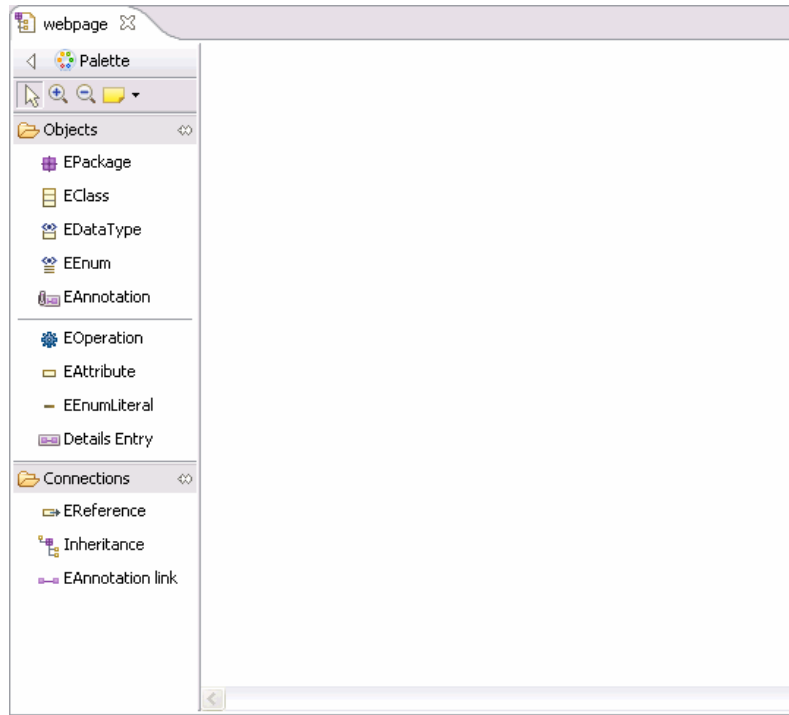
Το Eclipse Modeling Framework (EMF) μπορεί να χρησιμοποιηθεί για τη μοντελοποίηση του εννοιολογικού μοντέλου. Παρέχει μία διάκριση μεταξύ του μετά-μοντέλου και του πραγματικού μοντέλου. Το μετά - μοντέλο περιγράφει τη δομή του μοντέλου, και στη συνέχεια το μοντέλο είναι η παρουσίαση του μετά - μοντέλου.

Το EMF επιτρέπει τη δημιουργία μετά-μοντέλων με διάφορα μέσα, για παράδειγμα XMI, Java σχόλια, UML ή XML σχήμα. Μόλις προσδιορισθεί το EMF μετά-μοντέλο μπορούν να δημιουργηθούν οι αντίστοιχες Java υλοποιήσεις κλάσεων από αυτό.

Επίσης με το EMF μπορεί να δημιουργηθεί ρητά το εννοιολογικό μοντέλο που βοηθά στην παροχή σαφής εικόνας του μοντέλου. Παρέχει επίσης αλλαγή της λειτουργικότητας του μοντέλου σε περίπτωση αλλαγών. Δημιουργεί διασυνδέσεις για τη δημιουργία των αντικειμένων και έτσι διατηρεί την εφαρμογή καθαρή από επιμέρους τμήματα κλάσεων. Επίσης ένα πολύ σημαντικό πλεονέκτημα είναι ότι μπορεί να αναγεννά τον κώδικα Java από το μοντέλο σε οποιοδήποτε χρονική στιγμή.

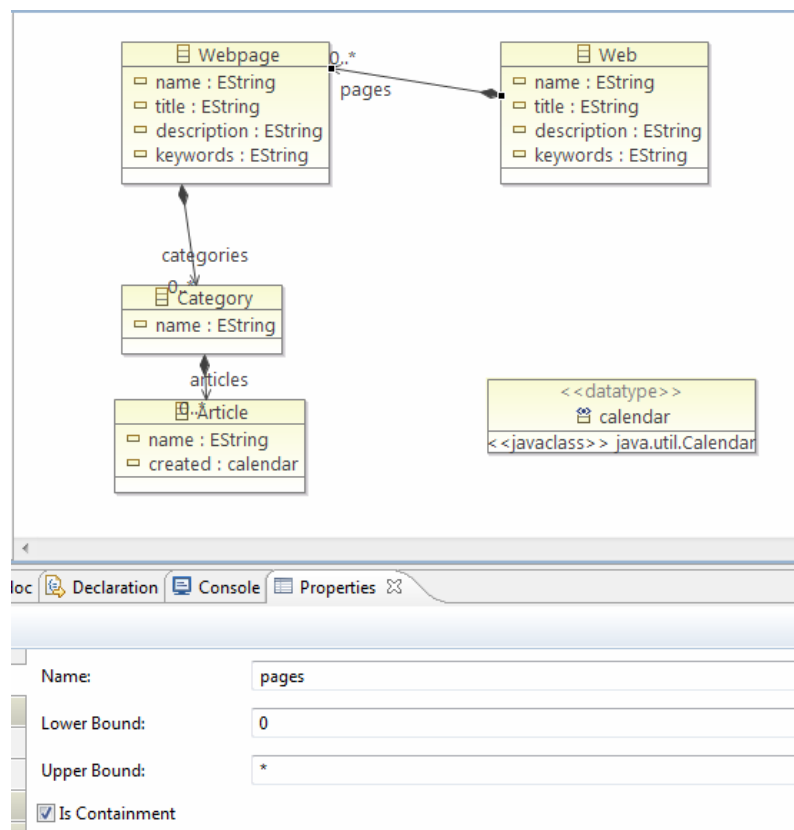
Στην πραγματικότητα το EMF βασίζεται σε δύο μετά - μοντέλα. Το Ecore και το Genmodel μοντέλο. Το Ecore περιέχει τις πληροφορίες για τις καθορισμένες κλάσεις, ενώ το Genmodel περιέχει επιπλέον πληροφορίες για την δημιουργία κώδικα, για παράδειγμα την διαδρομή και άλλες πληροφορίες του αρχείου. Επίσης περιέχει την παράμετρο ελέγχου για τον τρόπο με τον οποίο θα πρέπει να παραχθεί η κωδικοποίηση.

Εγκαθιστώντας το EMF στη συνέχεια δημιουργούμε ένα κενό EMF έργο επιλέγοντας Ecore διάγραμμα. Έτσι ανοίγει ο επεξεργαστής για τη δημιουργία EMF μοντέλων. Εικόνα: 4.3



Εικόνα: 4.3: Ο επεξεργαστής του EMF

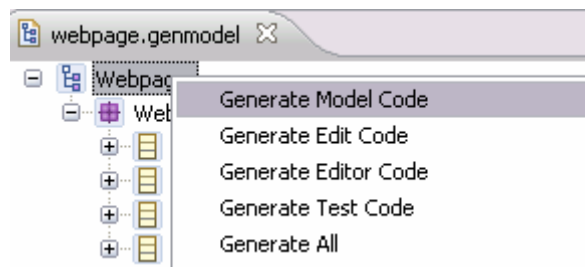
Για παράδειγμα:



Εικόνα: 4.4: Δημιουργία Ecore διαγράμματος

Αφού δημιουργήσουμε και το Genmodel διάγραμμα μπορούμε στη συνέχεια να παράγουμε τον κώδικα Java.

Επιλέγοντας Generate Model Code όπως φαίνεται στην εικόνα 4.5 θα δημιουργηθεί η Java εφαρμογή στο τρέχον έργο του EMF μοντέλου. Ο παραγόμενος κώδικας θα περιέχει διασυνδέσεις και κλάσεις Java.



Εικόνα: 4.5: Δημιουργία κώδικα Java από το EMF μοντέλο

Κάποια σημαντικά πλαίσια και εργαλεία του EMF είναι τα ακόλουθα:

Το CDO (Connected Data Objects) είναι ένα κοινό καταναμημένο πλαίσιο μοντέλου για τα EMF μοντέλα και μετά-μοντέλα. Το CDO επικεντρώνεται στην επεκτασιμότητα, τη φορητότητα, τη διατήρηση, τη διανομή και άλλα.

Το EMF Compare, φέρνει σε σύγκριση το μοντέλο με το πλαίσιο EMF, αυτό το εργαλείο παρέχει γενική υποστήριξη για κάθε είδους μετά-μοντέλου, προκειμένου να συγκρίνει και να συγχωνεύει μοντέλα. Οι στόχοι αυτού του συστατικού είναι να παρέχει μία σταθερή και αποτελεσματική εφαρμογή από τη σύγκριση των μοντέλων και να παρέχει ένα επεκτάσιμο πλαίσιο για συγκεκριμένες ανάγκες.

Το Service Data Objects(SDO) είναι ένα πλαίσιο που απλοποιεί και ενοποιεί τα δεδομένα ανάπτυξης εφαρμογών σε μία Service Oriented Αρχιτεκτονική (SOA). Υποστηρίζει και ενσωματώνει XML και J2EE πρότυπα και βέλτιστες πρακτικές. Το EMF περιλαμβάνει ένα EMF πλαίσιο, βασισμένο σε SDO.

4.4.2 Graphical Modeling Project

Το Graphical Modeling Project (GMP), παρέχει ένα σύνολο παραγωγικών συστατικών και runtime υποδομών για την ανάπτυξη γραφικών επεξεργαστών που βασίζονται σε EMF και σε GEF.

Το πλαίσιο εργασίας GEF (Graphical Editing Framework) επιτρέπει στους δημιουργούς εφαρμογών να δημιουργούν έναν πλούσιο γραφικό επεξεργαστή (*graphical editor*) βασισμένο στο μοντέλο της εφαρμογής αυτής καθαυτής και προβολών για το Eclipse Workbench UI. Το πλαίσιο εργασίας GEF αποτελείται από 2 επιπρόσθετα (plug-ins), το `org.eclipse.draw2d` και το `org.eclipse.gef`. Τα επιπρόσθετα αυτά παρέχουν όλη τη λειτουργικότητα που περιλαμβάνει το πλαίσιο εργασίας GEF. Ο δημιουργός μέσω της χρήσης των βιβλιοθηκών που παρέχονται από τα 2 προαναφερθέντα επιπρόσθετα (*plug-ins*) μπορεί να εκμεταλλευτεί πολλές κοινές λειτουργίες που παρέχονται στο GEF και / ή να τις επεκτείνει για ένα συγκεκριμένο τομέα εφαρμογών (*application domain*).



Εικόνα 4.6: Το πλαίσιο GEF

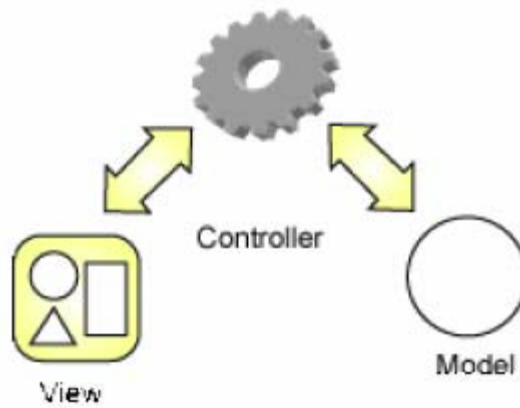
Το GEF υποθέτει πως υπάρχει ένα μοντέλο για το οποίο ο δημιουργός της εφαρμογής επιθυμεί να κάνει δυνατή την αναπαράστασή του και την επεξεργασία του με γραφικό τρόπο. Για το σκοπό αυτό το GEF παρέχει μηχανισμούς επισκόπησης (*viewers*) οι οποίοι μπορούν να χρησιμοποιηθούν οπουδήποτε μέσα στην επιφάνεια εργασίας (*workbench*) της πλατφόρμας Eclipse. Όπως και στην περίπτωση του πλαισίου εργασίας JFace έτσι και στην περίπτωση του πλαισίου εργασίας GEF οι μηχανισμοί επισκόπησης (*viewers*) αποτελούν προσαρμογείς (*adapters*) κάποιων συγκεκριμένων SWT γραφικών δομών (*widgets*) (SWT, ένα ανοιχτού κώδικα σύνολο εργαλείων για Java σχεδιασμό, ώστε να παρέχει αποτελεσματική πρόσβαση στη διεπαφή χρήστη για τις εγκαταστάσεις των λειτουργικών συστημάτων στα οποία εφαρμόζεται). Αλλά η ομοιότητά τους φτάνει ως εδώ. Οι μηχανισμοί επισκόπησης του GEF βασίζονται στην αρχιτεκτονική Μοντέλου – Προβολής – Ελεγκτών (Model-View-Controller / MVC αρχιτεκτονική) η οποία και θα επεξηγηθεί στη συνέχεια.

Η κεντρική ιδέα της MVC αρχιτεκτονικής είναι πως κάθε στοιχείο του Μοντέλου μίας εφαρμογής (Model), το οποίο ο χρήστης επιθυμεί να έχει τη δυνατότητα να τροποποιεί με γραφικό τρόπο, μπορεί να αναπαρασταθεί με μία γραφική αναπαράσταση (View). Την επικοινωνία και την σύνδεση μεταξύ των δύο δομικών στοιχείων (Model και View) αναλαμβάνουν οι Ελεγκτές (Controllers), οι οποίοι φροντίζουν να είναι διαρκώς συγχρονισμένα τα στοιχεία του Μοντέλου με τις αντίστοιχες γραφικές τους αναπαραστάσεις. Οι ελεγκτές στην MVC αρχιτεκτονική γεφυρώνουν το μοντέλο με την γραφική του αναπαράσταση (εικόνα 4.5). Κάθε ελεγκτής (controller, ή Edit Part όπως ονομάζονται), είναι υπεύθυνος για την αντιστοίχιση (mapping) του μοντέλου με την γραφική του αναπαράσταση καθώς και για την πραγματοποίηση αλλαγών στο μοντέλο. Ο ελεγκτής επίσης «παρακολουθεί» το μοντέλο, και ενημερώνει τη γραφική του αναπαράσταση ώστε να αντανakλά τις αλλαγές που πραγματοποιούνται στην κατάσταση του μοντέλου. Τα αντικείμενα με τα οποία ο χρήστης αλληλεπιδρά είναι οι ελεγκτές.

Ας θεωρήσουμε για παράδειγμα την έννοια «Class» την οποία επιθυμούμε να είμαστε σε θέση τόσο να την αναπαραστήσουμε όσο και να την τροποποιήσουμε γραφικά. Για το σκοπό αυτό πραγματοποιούμε τις ακόλουθες ενέργειες. Σε πρώτο στάδιο υλοποιούμε την έννοια με τη χρήση Java κλάσεων. Το «Model» τμήμα της

εικόνας 4.7 περιλαμβάνει την υλοποίηση αυτή (Java κλάσεις) της έννοιας. Επιλέγουμε για γραφική αναπαράσταση της έννοιας ένα ορθογώνιο το οποίο θα περιλαμβάνει στην κορυφή του το όνομα της κλάσης. Η γραφική αυτή αναπαράσταση ονομάζεται «figure», υλοποιείται σε κατάλληλη Java κλάση, και περιλαμβάνεται στο τμήμα «View» της εικόνας 4.7. Για τη σύνδεση της έννοιας «Class» με τη γραφική της αναπαράσταση ορίζεται ένας «Controller» (ή EditPart) ο οποίος αναλαμβάνει την αντιστοίχιση της έννοιας «Class» με τη γραφική της αναπαράσταση και επιπλέον ορίζει πολιτικές (policies) για τη διαχείριση ενεργειών που πραγματοποιούνται γραφικά, ενώ παράλληλα λειτουργεί ως «listener» των ενεργειών που πραγματοποιεί ο χρήστης γραφικά. Έτσι εάν οριστεί πως είναι επιθυμητή η δυνατότητα της άμεσης γραφικής επεξεργασίας του ονόματος της έννοιας (μέσω του ορισμού μίας Direct Edit Policy), τότε μέσω κατάλληλης γραφικής επιλογής από το χρήστη (μονό αριστερό κλικ πάνω στο όνομα της γραφικής αναπαράστασης) ενεργοποιείται η λειτουργία και ο «Controller» αναλαμβάνει την διεκπεραίωσή της (ενημέρωση τόσο του μοντέλου όσο και της γραφικής αναπαράστασης για την αλλαγή). Ο «Controller» υλοποιείται και αυτός σε μία Java κλάση.

Η αναπαράσταση των μοντέλων των εφαρμογών με τη χρήση του GEF γίνεται μέσω μηχανισμών που ονομάζονται “viewers”. Το GEF παρέχει δύο τύπους μηχανισμών επισκόπησης (viewers): τους γραφικούς και αυτούς που έχουν δενδρική μορφή. Ο καθένας παρέχει ένα διαφορετικό τύπο γραφικής αναπαράστασης (view). Ο γραφικός viewer χρησιμοποιεί σχήματα (figures) τα οποία εμφανίζονται (ζωγραφίζονται) πάνω στον καμβά του SWT. Τα σχήματα (figures) ορίζονται με τη χρήση του Draw2D plug-in, το οποίο περιλαμβάνεται στο GEF. Ο μηχανισμός δενδροειδούς προβολής (TreeViewer) χρησιμοποιεί μία δομή δένδρου που παρέχεται από το SWT καθώς και άλλα γραφικά στοιχεία (TreeItems) για την γραφική του αναπαράσταση.



Εικόνα 4.7: Model-View-Controller

Το GEF είναι εντελώς ανεξάρτητο από εφαρμογές και παρέχει βασικές υποδομές για τη δημιουργία σχεδόν οποιασδήποτε εφαρμογής, συμπεριλαμβανομένων των: επεξεργαστών διαγραμμάτων δραστηριοτήτων (activity diagram editors), εφαρμογές κατασκευής γραφικών διεπαφών (GUI builders), επεξεργαστές διαγραμμάτων κλάσεων (class diagram editors), και επεξεργαστές μηχανών καταστάσεων (state machines editors).

Τα χαρακτηριστικά του org.eclipse.draw2d plug-in και κατά συνέπεια και του GEF (του οποίου το plug-in αποτελεί τμήμα) είναι τα ακόλουθα:

- Υποστήριξη της έγκυρης αναπαράστασης των γραφικών τμημάτων της εφαρμογής και της ομαλής γραφικής τους διάταξης. Ουσιαστικά το χαρακτηριστικό αναφέρεται στην ικανότητα του plug-in που αφορά την υψηλή ποιότητα των γραφικών στοιχείων που ορίζονται μέσα από αυτό
- Ποικίλες υλοποιήσεις των γραφικών αναπαραστάσεων (figures) και των διατάξεων (layout) τους. Τα σχήματα που ορίζονται με τη χρήση του plug-in μπορούν να έχουν οποιαδήποτε μορφή και σχήμα (π.χ. ορθογώνια, τετράγωνα κτλ)
- Υποστήριξη ορίων (borders) στα σχήματα
- Υποστήριξη δεικτών ποντικιού (cursors) και επεξηγηματικών μηνυμάτων
- Υποστήριξη γραφικών συνδέσεων (connections) με τα εξής χαρακτηριστικά:
 - Δυνατότητα προσάρτησης της σύνδεσης (των δύο άκρων της σύνδεσης) σε καθορισμένα από το χρήστη σημεία τα οποία βρίσκονται πάνω στα γραφικά στοιχεία που συνδέονται με τη σύνδεση

- Δυνατότητα ορισμού της διαδρομής που ακολουθεί μία σύνδεση όταν συνδέει δύο γραφικά στοιχεία (*Routing*)
- Δυνατότητα διακόσμησης των συνδέσεων
- Πολλαπλά, διαφανή στρώματα (*layers*)
- Ευμετάβλητα συστήματα συντεταγμένων
- Δυνατότητα ορισμού μίας περιοχής επισκόπησης (*view*) η οποία παρουσιάζει τα διαγράμματα που δημιουργούνται από την εκάστοτε εφαρμογή σε κλίμακα (σμίκρυνση)
- Δυνατότητα εκτύπωσης των διαγραμμάτων που δημιουργούνται από την εκάστοτε εφαρμογή

Τα χαρακτηριστικά του org.eclipse.gef plug-in και κατά συνέπεια και του GEF (του οποίου το plug-in αποτελεί τμήμα) είναι τα ακόλουθα:

- Εργαλεία που παρέχουν τις δυνατότητες επιλογής) και δημιουργίας γραφικών στοιχείων. Επιπλέον εργαλεία που παρέχουν τη δυνατότητα δημιουργίας συνδέσεων και εργαλεία για την ταυτόχρονη επιλογή δύο ή περισσότερων γραφικών στοιχείων
- Μία παλέτα για την επίδειξη των παραπάνω εργαλείων
- Λαβές (*handles*) για την αλλαγή του μεγέθους (*resizing*) των γραφικών αντικειμένων και για την κάμψη (*bending*) των συνδέσεων
- Δύο είδη μηχανισμών επισκόπησης όπως περιγράφηκαν πιο πάνω (Γραφικοί και Δενδρικοί)
- Ένα πλαίσιο ελεγκτών για την αντιστοίχιση του μοντέλου της εφαρμογής με την γραφική του αναπαράσταση. Το πλαίσιο αυτό περιλαμβάνει:
 - Ορισμός πολιτικών των plug-ins με σκοπό την αντιστοίχιση την αλληλεπιδράσεων με την γραφική αναπαράσταση του μοντέλου στο ίδιο το μοντέλο
 - Ποικίλες υλοποιήσεις για την επίδειξη ανάδρασης (*feedback*) και την προσθήκη λαβών επιλογής γραφικών στοιχείων
 - Ποικίλοι τύποι αιτημάτων και εργαλεία ή ενέργειες που στέλνουν αυτά τα αιτήματα στους ελεγκτές
- Υποστήριξη των λειτουργιών Ακύρωσης / Επανάληψης (*Undo / Redo*)

4.4.3 Model Development Tools

Το μοντέλο ανάπτυξης εργαλείων MDT (Model Development Tools) επικεντρώνεται σε ένα μεγάλο "M" στο πλαίσιο του έργου μοντελοποίησης. Ο σκοπός είναι διπλός: Να παρέχει υλοποίηση πρότυπων μετά-μοντέλων στη βιομηχανία, και να παρέχει υποδειγματικά εργαλεία για την ανάπτυξη μοντέλων που βασίζονται σε αυτά τα μετά-μοντέλα.

Ορισμένα από τα πλαίσια του είναι τα ακόλουθα:

Το UML2 είναι ένα πλαίσιο ανάπτυξης με βάση την εφαρμογή ενιαίας γλώσσας μοντελοποίησης (Unified Modeling Language) (UML) μετά-μοντέλου για την πλατφόρμα Eclipse. Οι στόχοι του UML2 είναι να παρέχει, μία εύχρηστη εφαρμογή του UML μετά-μοντέλου για την στήριξη της ανάπτυξης των εργαλείων μοντελοποίησης, ένα κοινό XMI σχήμα ώστε να διευκολυνθεί η ανταλλαγή των σημασιολογικών μοντέλων, περιπτώσεις δοκιμής ως μέσο για την επικύρωση των προδιαγραφών, και κανόνες επικύρωσης ως ένα μέσο για τον καθορισμό και την επιβολή επιπέδων συμμόρφωσης.

Τα εργαλεία UML2 (UML2Tools) είναι μία γραφική μοντελοποίηση πλαισίου επεξεργαστή για το χειρισμό UML μοντέλων. Τα διαγράμματα που παρέχει είναι κλάσεων, περιπτώσεων χρήσης, δραστηριοτήτων, ακολουθίας και άλλα.

Το MoDisco είναι ένα πλαίσιο για την ανάπτυξη μοντέλου με γνώμονα τα εργαλεία για τον εκσυγχρονισμό του λογισμικού. Στόχος του είναι να υποστηρίζει τις περιπτώσεις χρήσης όπως, η διασφάλιση της ποιότητας, η τεκμηρίωση και η βελτίωση.

4.4.4 UML Plug-ins

ObjectAid UML Explorer

Το ObjectAid UML Explorer έχει αναπτυχθεί και βελτιστοποιηθεί για την γρήγορη και εύκολη δημιουργία διαγραμμάτων κλάσεων και διαγραμμάτων ακολουθίας από τον υπάρχοντα πηγαίο κώδικα της Java. Χρησιμοποιεί την UML για να δείξει μία γραφική αναπαράσταση του υπάρχοντα κώδικα με ακρίβεια.

ModelGoon UML4Java

Το ModelGoon φέρνει νέες απόψεις ενός Java έργου, χάρη στη στενή σύνδεση και αλληλεπίδραση με το Eclipse Java Development Tools (JDT). Παρέχει τέσσερα διαγράμματα τα οποία είναι χτισμένα και συγχρονίζονται άμεσα από τη Java.

Το διάγραμμα εξαρτήσεων πακέτων, όπου δείχνει τις εξαρτήσεις μεταξύ των πακέτων και υπογραμμίζει κυκλικές εξαρτήσεις. Κάθε γραμμή εξάρτησης μπορεί επίσης να εμφανίζει τις εμπλεκόμενες τάξεις.

Το διάγραμμα κλάσεων, το οποίο δείχνει μία διαρθρωτική προβολή των επιλεγμένων από το χρήστη κλάσεων, και επιτρέπει στο χρήστη να πλοηγηθεί από τα στοιχεία του διαγράμματος, με το αντίστοιχο στοιχείο Java στον πηγαίο κώδικα. Σημασιολογία όπως η σύνθεση (*composition*) ή συνάθροιση (*aggregation*) μπορεί να προστεθεί σε ενώσεις.

Το διάγραμμα αλληλεπίδρασης, το οποίο με βάση το διάγραμμα επικοινωνίας της UML (*communication diagram*), δείχνει τη συνολική αλληλεπίδραση μεταξύ των αντικειμένων, προκειμένου να ελέγξει τις ευθύνες επηρεασμών.

Τέλος το διάγραμμα ακολουθίας, που δημιουργείται από μία επιλεγμένη από το χρήστη μέθοδο.

Javadoc As UML View 0.2.2

Το Javadoc σαν UML plug-in παρέχει μία νέα Eclipse προβολή, η οποία αναλύει τον ενεργό πηγαίο κώδικα του προγράμματος επεξεργασίας προς τις υπάρχουσες αναφορές κλάσεων μέσα σε Javadoc σχόλια ή αναφορές μέσα στο σώμα μίας τάξης. Όλες αυτές οι αναφορές έχουν διερευνηθεί περαιτέρω για τη δημιουργία UML διαγράμματος σε πραγματικό χρόνο. Αυτό το διάγραμμα είναι χρήσιμο για την ανάλυση ενός συγκεκριμένου κομματιού κώδικα και για σκοπούς τεκμηρίωσης. Το μεγάλο πλεονέκτημα αυτού του plug-in είναι τα σχόλια Javadoc στον κώδικα του UML μοντέλου. Έτσι είναι πολύ άμεση και πολύ γρήγορη η αλληλεπίδραση μεταξύ των javadoc αναφορών και η τεκμηρίωση των UML διαγραμμάτων.

AmaterasUML 1.3.2

Το AmaterasUML είναι ένα Eclipse plug-in για το σχεδιασμό διαγραμμάτων κλάσεων και ακολουθίας της UML. Παρέχει την εξαγωγή διαγράμματος κλάσεων από πηγαίο κώδικα Java.

Green UML

Το GreenUML υποστηρίζει και την τεχνολογία λογισμικού και την αντίστροφη μηχανική του υπάρχοντος κώδικα. Χρησιμοποιείται για τη δημιουργία ενός διαγράμματος κλάσεων από τον υπάρχοντα κώδικα, ή για τη δημιουργία νέου κώδικα από ένα υπάρχον διάγραμμα κλάσεων. Αναπτύχθηκε με παιδαγωγικό στόχο την εστίαση των μαθητών στη σχεδίαση, ωστόσο έχει αποδειχθεί ότι είναι ένα ισχυρό και ευέλικτο εργαλείο που μπορεί να παρέχει UML υποστήριξη εύκολα σε οποιοδήποτε περιβάλλον.

eUML2

Το eUML2 για Java είναι μία ισχυρή UML μοντελοποίηση για Java προγραμματιστές στο Eclipse για την υποστήριξη τεράστιων μεγέθους έργων, και τη συνεργασία μεγάλων ομάδων. Πρόκειται για μία εγγενή ενοποίηση Eclipse με βάση όλα τα πλαίσια του Eclipse όπως το EMF, GEF, GMFUML2.

Slime UML

Το Slime UML είναι ένα λεπτό, γρήγορο και ισχυρό UML plug-in της πλατφόρμας Eclipse. Καθιστά εύκολη την τεκμηρίωση του Java κώδικα και την απεικόνιση σύνθετων βιβλιοθηκών της Java καθώς και εκτέλεση μοντελοποίησης με περιπτώσεις χρήσης, πακέτα και διαγράμματα κλάσεων. Διαθέτει αυτόματη διάταξη των διαγραμμάτων, πολλαπλή κληρονομικότητα, ενημερώσεις των διαγραμμάτων όταν γίνονται αλλαγές στον Java κώδικα, εξαγωγή σε pdf και πολλά άλλα.

4.5 Το έργο Mylyn

Εισαγωγή

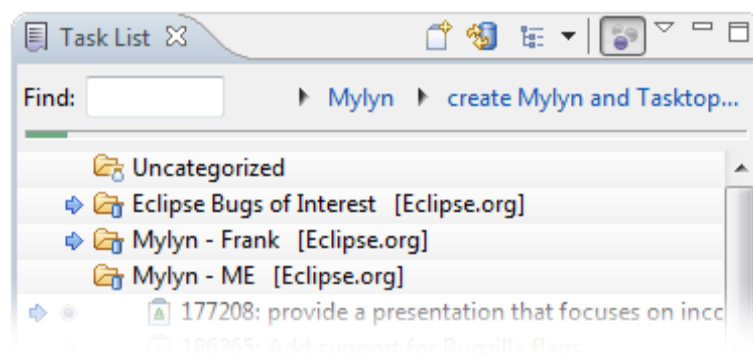
Mylyn είναι ένα υποσύστημα του Eclipse που επεκτείνει το Eclipse SDK με ένα σοφιστικό μηχανισμό για την παρακολούθηση και τη διαχείριση των διεργασιών. Μία εφαρμογή ανοιχτού κώδικα με επίκεντρο τη διεργασία, δηλαδή παρέχει μια εφαρμογή προγραμματισμού διεργασία για την ενσωμάτωση εργαλείων. Βοηθά τους προγραμματιστές λογισμικού, να εργάζονται αποτελεσματικά με πολλές διαφορετικές διεργασίες (όπως σφάλματα, αναφορές προβλημάτων ή νέα χαρακτηριστικά). Μία διεργασία ορίζεται ως κάθε μονάδα εργασίας που θέλει κανείς να ανακαλέσει ή να μοιραστεί με άλλους, όπως ένα σφάλμα αναφοράς από ένα χρήστη, ή μία σημείωση για την βελτίωση ενός χαρακτηριστικού. Για όλες τις διεργασίες που έχουν ενσωματωθεί, το Mylyn παρακολουθεί τις δραστηριότητες του χρήστη και προσπαθεί να εντοπίσει πληροφορίες σχετικές με αυτές τις δραστηριότητες. Χρησιμοποιεί αυτό το πλαίσιο διεργασιών για να επικεντρωθεί το Eclipse UI (*User Interface*) στις σχετικές πληροφορίες. Μπορεί να ενσωματώσει πηγές λογισμικού όπως Bugzilla, Trac, codeBeamer και άλλα, και επικεντρώνεται στη βελτίωση της παραγωγικότητας μειώνοντας την αναζήτηση και την πλοήγηση. Κάνοντας ρητό το πλαίσιο των διεργασιών, έχει ως στόχο να διευκολύνει το multitasking, το σχεδιασμό, την επαναχρησιμοποίηση των προηγούμενων

προσπαθειών, και την ανταλλαγή τεχνογνωσίας. Το αρχικό του όνομα ήταν Mylar και η Eclipse κοινότητα άλλαξε το όνομα σε Mylyn.

Οι διεργασίες μπορούν να αποθηκευτούν στο χώρο εργασίας σε τοπικό επίπεδο, ή μπορούν να προέρχονται από μία ή περισσότερες διεργασίες πηγών λογισμικού. Για να πραγματοποιηθεί η σύνδεση με μία συγκεκριμένη διεργασία χώρου αποθήκευσης πρέπει να υπάρχει εγκατεστημένο το Mylyn connector για την αποθήκευση.

Το Mylyn Eclipse παρακολουθεί και καταγράφει την αλληλεπίδραση του χρήστη σε ένα πλαίσιο διεργασίας. Στα αντικείμενα συστήματος, όπως αρχεία, τύποι, μέθοδοι και πεδία, αποδίδεται ένας βαθμός ενδιαφέροντος με βάση τη συχνότητα αλλά και το πόσο πρόσφατα έρχεται σε επαφή ο χρήστης μαζί τους. Το αποτέλεσμα είναι να φιλτράρονται τα στοιχεία χωρίς ενδιαφέρον μέσα από την Eclipse προβολή, επιτρέποντας στο χρήστη να επικεντρωθεί σχετικά με το τι είναι σημαντικό.

Η κύρια προβολή του Mylyn είναι η λίστα διεργασιών.



Εικόνα 4.8: Η λίστα διεργασιών του Mylyn

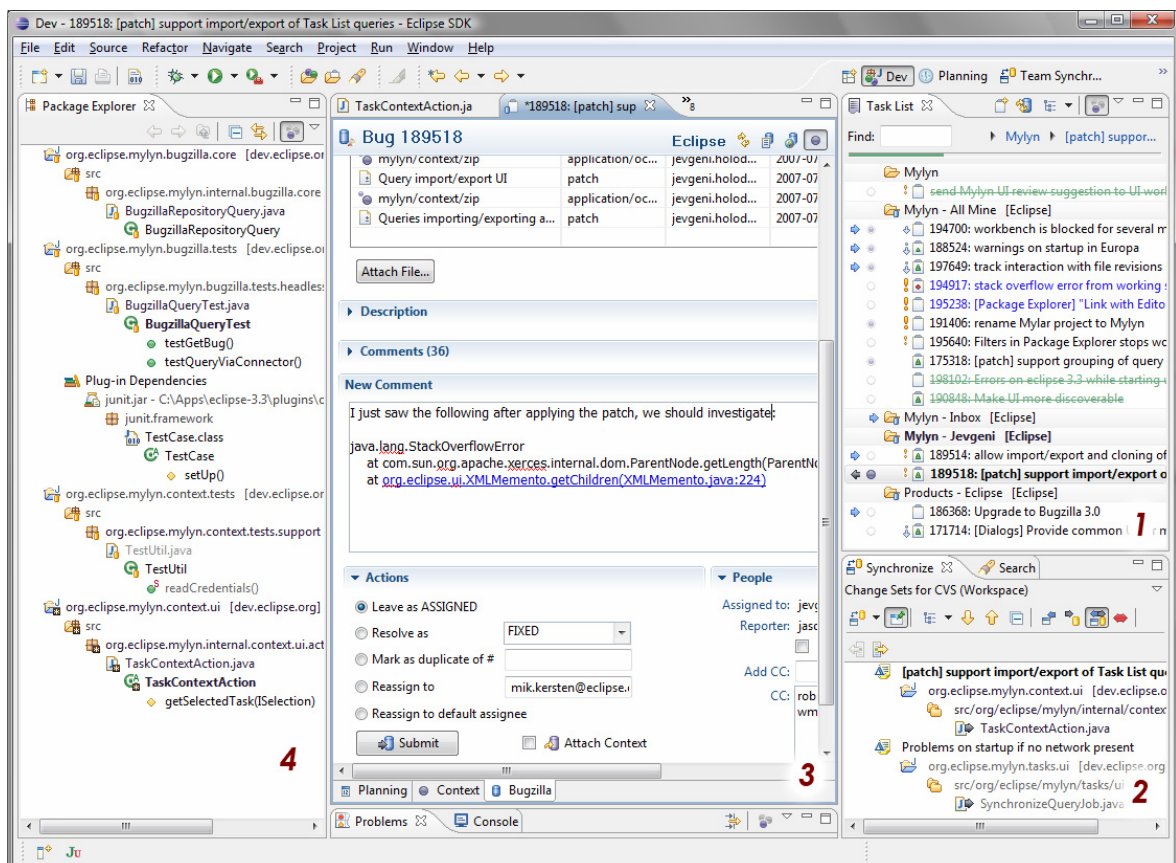
Αυτή η λίστα είναι το κεντρικό σημείο εισόδου εργασίας στο Mylyn, όπου γίνεται η διαχείριση των τοπικών και αποθηκευμένων διεργασιών, οι οποίες μπορούν να οργανωθούν ανάλογα με την κατηγορία στην οποία ανήκουν, ή το λειτουργικό σύνολο. Οι διεργασίες που δεν μοιράζονται με άλλους μέσω του δικτύου και παραμένουν τοπικά αποθηκευμένες στον υπολογιστή, είναι οι τοπικές διεργασίες, ενώ οι αποθηκευμένες διεργασίες που συχνά φιλοξενούνται σε δημόσιο χώρο αποθήκευσης, είναι ορατές σε αυτούς που έχουν δικαιώματα πρόσβασης.

Με τον καιρό, το αποτέλεσμα της χρήσης του Mylyn είναι μία λεπτή, αλλά ριζική αλλαγή στον τρόπο εργασίας. Μόλις αποκτηθεί η συνήθεια εργασίας με τρόπο

εστιασμένο στην διεργασία, παρατηρείται ικανοποιητική αύξηση της παραγωγικότητας. Το να υπάρχει οργάνωση, είναι το ζητούμενο της συνεργατικής διεργασίας, και ο εντοπισμός της προόδου του έργου γίνεται αποδοτικότερος.

Οι αριθμημένες περιοχές της εικόνας 4.9, δείχνουν κάποιες διεργασίες του Mylyn και πλαίσια διαχείρισης ιδιοτήτων του έργου.

1. Λίστα διεργασιών με μία ενεργή διεργασία και Bugzilla αναφορά
2. Μία σειρά αλλαγών διαχείρισης με πλαίσιο διεργασίας
3. Ένας ικανός επεξεργαστής διεργασιών με υποστήριξη χωρίς σύνδεση
4. Κατάσταση εστιασμένης διεργασίας στο πακέτο περιήγησης του Eclipse



Εικόνα 4.9: Mylyn's Task - focused UI

Αποθήκευση συνδεδεμένων διεργασιών

Σε αυτό το σημείο, χρησιμοποιείται ο σύνδεσμος Bugzilla του Mylyn για να υπογραμμιστούν οι βασικές έννοιες με τις αποθηκευμένες διεργασίες ενοποίησης. Οι έννοιες είναι πανομοιότυπες και για άλλες υποστηριζόμενες διεργασίες αποθήκευσης, αλλά το επίπεδο ενοποίησης των συνδέσμων ποικίλει. Ο Bugzilla και οι Trac connectors που είναι πακέτο με το Mylyn εξετάζουν τις εφαρμογές αναφορών.

Ένας πλήρως ενοποιημένος σύνδεσμος παρέχει τα ακόλουθα χαρακτηριστικά:

- Ερωτήματα: Το ερώτημα είναι ο μηχανισμός για την ανάκτηση του συνόλου των διεργασιών στη λίστα διεργασιών του Mylyn.
- Πλούσια Επεξεργασία: Οι διεργασίες και οι ερωτήσεις μπορούν να επεξεργαστούν από τον ενοποιημένο επεξεργαστή. Αυτή η διευκόλυνση παρέχει την υπερσύνδεση των διεργασιών και άλλων δομημένων τμημάτων καθώς και την ένταξη του Eclipse και την επιφάνεια εργασίας με τη μέθοδο drag and drop. Οι σύνδεσμοι οι οποίοι παρέχουν πλούσια επεξεργασία μπορούν ακόμη να παρέχουν ευκολίες για το άνοιγμα διεργασιών και για ερωτήσεις στο ενσωματωμένο πρόγραμμα περιήγησης του Eclipse.
- Συννημένα: Μπορεί να γίνει επισύναψη και ανάκτηση αρχείων από το χώρο αποθήκευσης. Αυτά τα χαρακτηριστικά επιτρέπουν την επισύναψη αρχείων μέσα από το λειτουργικό σύστημα ή από το χώρο εργασίας του Eclipse, και κάνουν δυνατές τις διευκολύνσεις όπως την ανταλλαγή πλαισίου.
- Υποστήριξη χωρίς σύνδεση: Υπάρχει η δυνατότητα εργασίας χωρίς σύνδεση, πρόσβαση διεργασιών, και απευθείας ερωτήματα χωρίς να χρειάζεται αναμονή ενός server. Αυτή η ευκολία παρέχει επίσης ενημέρωση αλλαγών, η οποία δίνει τη δυνατότητα χρήσης της λίστας διεργασιών σαν εισερχόμενα για την διεργασία.

Για να επιτραπεί η πλήρης ενοποιημένη εμπειρία που παρέχει το Mylyn, και να σταματήσει η στήριξη σε εξωτερικό ή διαδικτυακό βασισμένο UI, ο σύνδεσμος πρέπει να υποστηρίζει όλες τις τέσσερις παραπάνω διευκολύνσεις. Αλλά υπάρχει και η ωφέλεια εάν μια αποθηκευμένη διεργασία δεν είναι διαθέσιμη για τον σύνδεσμο να γίνουν τα εξής:

- Χρήση της Mylyn υποστήριξης για τις τοπικές διεργασίες. Αν η μη υποστηριζόμενη αποθήκευση έχει UI με βάση το διαδίκτυο, μπορούν να τοποθετηθούν τα URLs από τον πλοηγό στο Mylyn για να δημιουργηθούν οι διεργασίες.
- Χρήση του γενικού διαδικτυακού συνδέσμου αποθήκευσης, ο οποίος παρέχει βασική υποστήριξη ερωτημάτων για πολλούς χώρους αποθήκευσης βασισμένους στο διαδίκτυο: Google Code Hosting (code.google.com), IssueZilla (java.net, dev2dev, tigris.org), GForge (objectweb.org), SourceForge (sf.net), JIRA, Trac, phpBB, and vBulletin
- Ερώτηση στην διεργασία που βρίσκεται στο χώρο αποθήκευσης για να δημιουργήσει ένα σύνδεσμο για το Mylyn
- Δημιουργία ενός νέου συνδέσμου

Διεργασίες και Ερωτήματα

Το Mylyn παρέχει υποστήριξη για εργασία με δύο ειδών διεργασίες:

- **Τοπικές διεργασίες:** Αποθηκεύονται στο χώρο εργασίας Eclipse και παρέχουν βασικές διεργασίες διαχείρισης χαρακτηριστικών, όπως ο προγραμματισμός. Είναι ιδιωτικές και ορατές μόνο στο χρήστη.
- **Αποθηκευμένες διεργασίες:** Αποθηκεύονται σε μία αποθήκη διεργασιών που αντιστοιχεί σε μία εξωτερική εφαρμογή ή σε ένα server. Αυτές είναι συχνά κοινές μεταξύ των χρηστών, αλλά εξακολουθούν να παρέχουν όλες τις παροχές των τοπικών διεργασιών όπως ο προσωπικός προγραμματισμός και οι σημειώσεις.

Από τη στιγμή που θα δημιουργηθούν κάποιες τοπικές διεργασίες, το Mylyn παρέχει τη δυνατότητα οργάνωσή τους με τη δημιουργία κατηγοριών.

Προγραμματισμός εστιασμένος στη διεργασία

Υπάρχει περίπτωση καθώς ολοκληρώνονται οι αλλαγές σε ένα νέο χαρακτηριστικό να υπάρξει μία κατάρρευση ή ένα κόλλημα. Κατά τη διάρκεια της εργασίας, χτίζεται μία ποικιλία γνώσης από όλες τις κλάσεις και μεθόδους που έχουν διαμορφωθεί

και τα APIs τα οποία έχουν πρόσβαση στη δημιουργία των χαρακτηριστικών. Με τη χρήση του Mylyn αυτή η γνώση αποτυπώνεται αυτόματα στο πλαίσιο διεργασίας. Όσο πλησιάζει η ολοκλήρωση της διεργασίας ένα σημαντικό σφάλμα απαιτεί άμεση προσοχή.

Με ένα απλό κλικ, ενεργοποιείται η αναφορά σφάλματος και αρχίζει η έρευνα του προβλήματος. Καθώς γίνεται η πλοήγηση στην υπερσύνδεση της στοίβας του λάθους, χρησιμοποιώντας τον επεξεργαστή διεργασίας του Mylyn, δημιουργείται ένα νέο πλαίσιο διεργασίας με τον περιηγητή και τα διαγνωστικά. Όταν γίνεται ενεργοποίηση στον αυτόματο έλεγχο πλαισίου του Mylyn, ο οποίος τρέχει δομημένες μονάδες ελέγχου σχετικές με το πλαίσιο το οποίο έχει δημιουργηθεί, εξερευνείται ότι το λάθος είναι σε έναν συναδελφικό κώδικα. Χάρη στην διασύνδεση των ιστοριών αναθεώρησης της διεργασίας του Mylyn, είναι αμέσως ορατές οι συναδελφικές διεργασίες οι οποίες ευθύνονται για την προβληματική αλλαγή. Με μία άλλη ενεργοποίηση, γίνεται επανεκχώρηση του σφάλματος στην συνεργασία και μοιράζει το πλαίσιο διεργασίας του αποτελέσματος για τη διάγνωση. Καθώς γίνονται όλα αυτά χωρίς να εγκαταλειφθεί το Eclipse, έτσι ώστε με ένα κλικ στη λίστα εργασιών με το κουμπί Back αμέσως επαναφέρεται το πλαίσιο της προηγούμενης διεργασίας. Η συνεργασία εν τω μεταξύ, έχει πάρει την αναφορά σφάλματος ακριβώς από εκεί που είχε αφεθεί. Αυτό είναι μόνο ένα σενάριο από τις ευκολίες τις οποίες έχουν οι προγραμματιστές χρησιμοποιώντας ένα UI εστιασμένης διεργασίας.

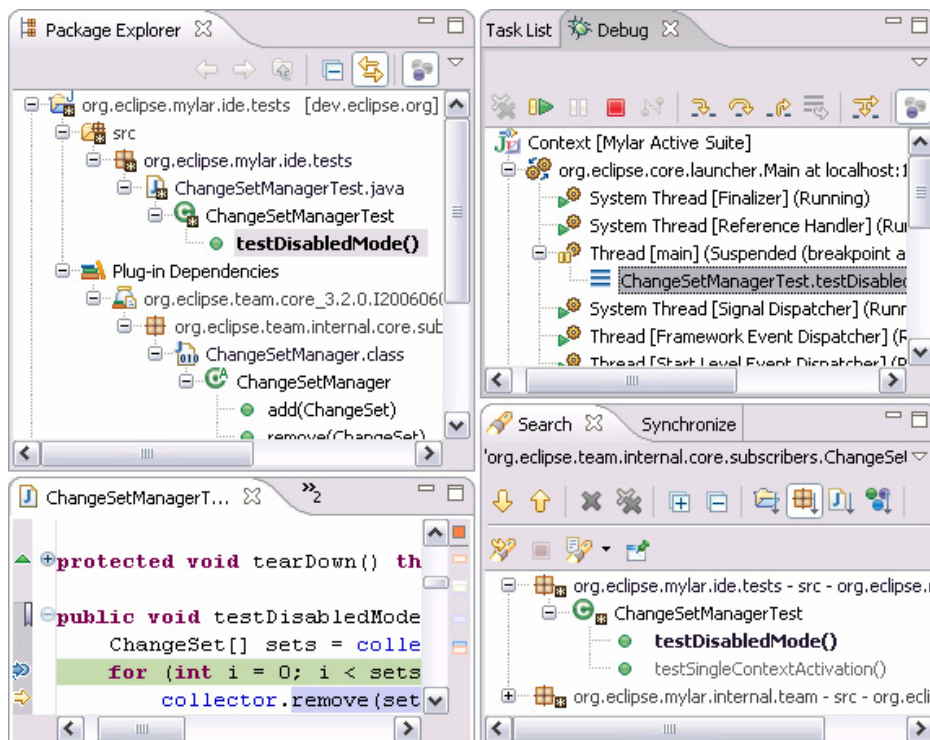
Ο βαθμός ενδιαφέροντος του μοντέλου

Ο έλεγχος του πλαισίου διεργασίας του Mylyn είναι βασισμένος σε μία ιδέα στην οποία η αλληλεπίδραση με το σύστημα μπορεί να μεταμορφωθεί μέσα σε ένα βαθμό ενδιαφέροντος του μοντέλου, στο οποίο κάθε κομμάτι μέσα στο σύστημα ζυγίζεται σύμφωνα με τη διεργασία που διαχειρίζεται. Ο βαθμός ενδιαφέροντος του μοντέλου ζυγίζει τις φόρμες του πλαισίου από όλα τα σχετικά στοιχεία της διεργασίας. Το πλαίσιο διεργασίας μπορεί να χρησιμοποιηθεί στο UI, σαν ένδειξη των σημαντικότερων στοιχείων, φιλτράροντας τα μη σημαντικά, και αφήνοντας να λειτουργήσουν τα στοιχεία ενδιαφέροντος. Το πλαίσιο διεργασίας χτίζεται σε έναν δρόμο πρόβλεψης καθώς γίνεται το έργο: Καθώς ενεργοποιείται μία διεργασία,

κάθε στοιχείο το οποίο επιλέγεται ή επεξεργάζεται (όπως ένα αρχείο ή μία Java μέθοδος) γίνεται ένα κομμάτι του πλαισίου διεργασίας. Όσο πιο πολύ αλληλεπιδρά το στοιχείο, η διεργασία γίνεται υψηλότερου βαθμού ενδιαφέροντος. Εάν το ενδιαφέρον ενός στοιχείου είναι αρκετά υψηλό γίνεται ορόσημο, το οποίο δημιουργεί έναν σελιδοδείκτη.

Η χρήση του Mylyn στην Java ανάπτυξη

Οι Προγραμματιστές οι οποίοι χρησιμοποιούν το Mylyn για κώδικα Java, τυπικά κρατάνε το Package Explorer σε εστιασμένη κατάσταση. Έτσι από προεπιλογή το κουμπί εστίασης στην ενεργή διαδικασία γυρνάει αυτόματα όταν ενεργοποιείται η διεργασία και ξαναγυρνάει όταν απενεργοποιείται. Όταν ενεργοποιείται είναι ορατά μόνο τα στοιχεία της Java μέσα στο πλαίσιο. Καθώς επιλέγεται μία Java κλάση, είτε με πλοήγηση μέσα σε αυτήν είτε με την απλή χρήση της, ο τύπος γίνεται ένα κομμάτι του πλαισίου και εμφανίζεται στο φιλτραρισμένο Package Explorer. Κάθε μέθοδος που επιλέγεται και επεξεργάζεται προσθέτετε στο πλαίσιο διεργασίας και για αυτό εμφανίζεται στο Package Explorer. Στην εικόνα 4.10 που ακολουθεί το Package Explorer είναι εστιασμένης κατάστασης.



Εικόνα: 4.10: Package Explorer

Επομένως το Mylyn επεκτείνει το Eclipse με μηχανισμούς για την παρακολούθηση των διεργασιών, πράγμα που έχει ως αποτέλεσμα τη διαθεσιμότητα των πληροφοριών που χρειάζεται ο χρήστης, διευκολύνοντας την ανταλλαγή εμπειρογνωμοσύνης.

4.6 Το έργο Pulse

Εισαγωγή

Pulse είναι μία εφαρμογή διαχείρισης του κύκλου ζωής τεχνολογίας (ALM) (*Application lifecycle management technology*) που αναπτύχθηκε και συντηρείται από την Genuitec, ιδρυτικό και στρατηγικό μέλος της Κοινότητας Eclipse. Το Pulse είναι χτισμένο στην κορυφή της Eclipse Equinox(OSGI)/p2 πλατφόρμας, και ενσωματώνει και τα ιδιόκτητα αλλά και τα ανοιχτού κώδικα λογισμικά για την παράδοση λογισμικού, την απελευθέρωση της διαχείρισης, και του περιβάλλοντος συνεργασίας. Η τρέχουσα έκδοσή του είναι η 4.0.

Το Pulse παρέχει μία οικογένεια προϊόντων που βοηθά επιχειρήσεις και ιδιώτες να διαχειριστούν ή να αποκτήσουν λογισμικό. Διαθέτει τρεις εκδόσεις.

Η έκδοση της Κοινότητας (*Community Edition*), η οποία είναι δωρεάν για τους μεμονωμένους προγραμματιστές και επιτρέπει να διαχειρίζονται τις τοπικές βασισμένες σε Eclipse επιλογές εργαλείων.

Η έκδοση για τη διαχείριση της ομάδας (*Managed Team Edition*) που έχει ως στόχο τη διαχείριση των αναγκών της ομάδας ανάπτυξης. Είναι μία προσφορά που χτίστηκε πάνω στην έκδοση της κοινότητας που επικεντρώνεται στην παροχή πλήρους προσαρμοσμένων εργαλείων και συγχρονισμό για μικρές και μεσαίες ομάδες ανάπτυξης, και τις επιτρέπει να παραμένουν συγχρονισμένες σε κοινό χώρο εργασίας με συνεργατικά χαρακτηριστικά.

Τέλος υπάρχει η ιδιωτική έκδοση παράδοσης λογισμικού (*Private label*) που έχει σχεδιαστεί για εταιρική χρήση, και είναι ένα προϊόν λευκής ετικέτας για την δημιουργία προσαρμοσμένης εγκατάστασης (προγράμματα υπολογιστών) λογισμικού, που χρησιμοποιούν το διαδίκτυο ως κανάλι.

Η χρησιμότητα του Pulse

Το Pulse είναι μια λύση η οποία συνεργάζεται με τον πελάτη (και την υπάρχουσα υποδομή του), ώστε να εφαρμόσουν μια πλήρη διαχείριση του κύκλου ζωής του λογισμικού, καθώς και λύση παράδοσης. Με τη χρήση της ίδιας πλατφόρμας για τη συνεργασία, την ανάπτυξη και την παράδοση, ο πελάτης εξοικονομεί χρόνο αλλά και χρήμα. Ο χρήστης μπορεί να δημιουργήσει οποιοδήποτε αριθμό των προσαρμοσμένων στοιβών εργαλείων, για παράδειγμα, κάποιος που έχει αναπτύξει μία ιστοσελίδα σε Java, μπορεί εύκολα να αναπτύξει ένα προφίλ προσαρμοσμένο ακριβώς σε αυτό που χρειάζεται, αφού μπορεί να εκτελέσει μόνο τα εργαλεία που του είναι αναγκαία. Αυτό παρέχει τεράστια βελτίωση αφού δεν είναι απαραίτητο να τρέχουν πολλά εργαλεία χωρίς να χρειάζονται.

Η πλατφόρμα του Pulse

Η Pulse πλατφόρμα της Genuitec είναι ένας ενιαίος μηχανισμός παροχής λογισμικού που επιτρέπει σε επιχειρήσεις την παράδοση του λογισμικού. Παρέχει ενοποιημένη εγκατάσταση για όλα χαρακτηριστικά λογισμικού του τελικού χρήστη, και διασφαλίζει την πλήρη συμβατότητα και συμμόρφωση στην επιφάνεια εργασίας του. Επιτρέπει ενημέρωση σε πραγματικό χρόνο και καθορισμό σφαλμάτων διαχείρισης. Το Pulse παρέχει περαιτέρω τη συνεργασία για την τυποποίηση των εσωτερικών ομάδων προγραμματιστών και ουσιαστικά εγκαθιστά όλα τα είδη των λειτουργικών συστημάτων των επιχειρήσεων (Windows, Mac, Linux). Αυτό σημαίνει ότι το Pulse παρέχει IT διαχειριστές με μέγιστη ευελιξία και ορατότητα όσον αφορά το πως το λογισμικό τους παραδίδεται στους τελικούς χρήστες. Το Pulse έχει σχεδιαστεί ώστε να είναι εύκολο στη χρήση και ως εκ τούτου χρειάζεται ελάχιστος χρόνος για τους προγραμματιστές να προσανατολιστούν στην τεχνολογία.

Μπορεί να χτίσει τη διαχείριση εγκαταστατών. Αυτό είναι ένα νεότερο είδος εγκατάστασης που προέρχεται από τη σύγχρονη αγορά. Δηλαδή ένα έξυπνο πρόγραμμα εγκατάστασης που υπάρχει στο διαδίκτυο και έχει τη δυνατότητα να διαβάζει το λειτουργικό και υλικό σύστημα του χρήστη. Αυτοί οι διαχειριστές εγκατάστασης απαιτούν την λιγότερη αλληλεπίδραση του χρήστη κατά τη διάρκεια της διαδικασίας, έτσι ώστε να έχει την υψηλότερη πιθανότητα να λάβει τη σωστή εγκατάσταση την πρώτη φορά.

Διανομή και έλεγχος της εικόνας του Λογισμικού

Διανομή Λογισμικού: Το Λογισμικό μπορεί να χειριστεί από μία γενική βάση χρηστών με πλήρεις δυνατότητες διαχείρισης. Από μικρές ομάδες μέχρι χιλιάδες χρήστες το εύρος του Pulse μπορεί να καλύψει τις ανάγκες.

Προστασία από Παραποιώσεις: Το Pulse κλειδώνει το σύστημα και δεν μπορεί να γίνει καμία παραποίηση στο IPs.

Εργαλεία/Αρχειοθέτηση Λογισμικού: Πολλά terabytes δεδομένων μπορούν να συμπιεστούν και να τα αποθηκευτούν σε ένα DVD στη σελίδα ανάπτυξης του Pulse, κερδίζοντας χρήματα με τη διατήρηση πρόσβασης των δεδομένων εργαλείων.

Έλεγχος και Αναφορά: Το Pulse αναδημιουργεί τόσο το χώρο εργασίας των προγραμματιστών όσο και τις στοίβες εργαλείων πολλά χρόνια. Μπορεί να βλέπει κανείς τις στοίβες των εργαλείων ανάπτυξης που χρησιμοποιήθηκαν για την κατασκευή των προϊόντων.

Είναι κατανοητό από το Pulse πόσο πολύπλοκη είναι η διανομή λύσεων λογισμικού σε μία πληθώρα τύπων από τελικούς χρήστες. Εσωτερικά, οι μηχανικοί μπορεί να χρειαστούν διαφορετικά επίπεδα πρόσβασης και ένα μείγμα από ιδιοκτήτες και ανοιχτού κώδικα πηγές λογισμικού. Εξωτερικοί πελάτες μπορεί να χρειαστούν ορισμένα δικαιώματα αδείας που διαφέρουν ανάλογα με τα διαπιστευτήριά τους. Μπορεί να υπάρχουν νομικές ανησυχίες σχετικά με τη διανομή προϊόντων με άδεια ανοιχτού κώδικα μαζί με μία ιδιόκτητη λύση. Το Pulse καλύπτει όλα αυτά τα σενάρια.

Ακολουθεί μία Περίπτωση Χρήσης για τον τρόπο με τον οποίο το Pulse βοηθά τις μεγάλες επιχειρήσεις να διαχειριστούν τη διανομή του λογισμικού τους. Δέκα χιλιάδες Επιχειρήσεις Προγραμματισμού τυποποιήθηκαν στη χρήση ανοιχτού Eclipse Λογισμικού με Linux.

Το πρόβλημα

Στις Ηνωμένες πολιτείες οι πωλητές υλικού και λογισμικού αντιμετωπίζουν αποθαρρυντική διεργασία. Χρειάζονται δέκα χιλιάδες προγραμματιστές για να τυποποιήσουν εργαλεία βασισμένα σε Eclipse ανοιχτού κώδικα χρησιμοποιώντας Linux μηχανές. Κάτω από μία καινούργια ηγεσία και καθοδήγηση, η Fortune 50 εταιρεία έχτισε τη δική της μηχανισμό διανομής λογισμικού αλλά δεν επιθυμούσε να κατέχει περαιτέρω τη διαχείρισή του. Ο οργανισμός επίσης απαιτούσε διαφορετικές ομάδες να έχουν μοναδικές στοίβες εργαλείων Eclipse, επομένως μία τεχνολογία "ένα μέγεθος που ταιριάζει σε όλους" δεν θα δούλευε για αυτή την εταιρεία.

Αυτή η επιχείρηση επιπλέον απαιτούσε ότι ένας μηχανισμός διανομής ενοποιείται με τις υπάρχουσες τεχνολογίες, όπως όλοι οι προγραμματιστές εκδίδουν ταυτότητα χρήστη και κωδικό της εταιρείας. Έτσι, η νέα τεχνολογία θα χρειάζεται για την επικύρωση και την εργασία με όλα τα άλλα συστήματα μέσα στο δίκτυο.

Μία επιπλέον επιπλοκή ήταν ότι οι εσωτερικοί προγραμματιστές αντισταθήκαν στις απαιτήσεις της ανώτερης διαχείρισης και δεν ήθελαν να εγκαταστήσουν το νεότερο IDE Eclipse για πολλούς λόγους. Παρ' όλα αυτά η ηγεσία έκρινε ότι για να παραμείνει ανταγωνιστικός ο οργανισμός για κάθε πελάτη και μέτοχο, πρέπει να υιοθετήσει τις νεότερες Eclipse τεχνολογίες. Η μετάβαση κατά συνέπεια έπρεπε να είναι απρόσκοπτη και εύκολη για το προσωπικό ανάπτυξης.

Μετά από εκτενή έρευνα στην αγορά για εφαρμογές διαχείρισης κύκλου ζωής ALM (Application Lifecycle Management) και τεχνολογίες διαχείρισης λογισμικού, η εταιρεία Fortune 50 αποφάσισε ότι χρειάζεται ένα εργαλείο που θα είναι εύκολο για κάθε προγραμματιστή να κατανοήσει και να χρησιμοποιήσει, προσαρμόσιμο σε κάθε επίπεδο, καθώς επίσης και ιδιαίτερα επεκτάσιμο και διαισθητικό για μελλοντικές ανάγκες. Η εταιρεία αποφάσισε να χρησιμοποιήσει την Pulse πλατφόρμα της Genuitec ως τον μηχανισμό για να πετύχει όλους τους στόχους της.

Η ανάλυση

Ξεκινώντας το 2008, ο οργανισμός ξεκίνησε την διαδικασία επιλογής ALM, ελέγχοντας διάφορες τεχνολογίες από δείγματα υψηλών προδιαγραφών εμπλεκόμενων ομάδων προγραμματιστών ώστε να βρει ποιο λογισμικό ταιριάζει καλύτερα στις ανάγκες και στους στόχους του. Από τις πολυάριθμες ομάδες ελέγχου λογισμικού, αποφασίστηκε από τις εταιρικές ομάδες ανάπτυξης, ότι το Pulse της Genuitec προσφέρει τις πιο ολοκληρωμένες λύσεις για την επίτευξη των στόχων.

Το Pulse δοκιμάστηκε για πρώτη φορά σε μία εικονική μηχανή για να φανεί αν είναι εύκολο στην κατανόηση και τη χρήση. Με τη βοήθεια της ομάδας Pulse, η ομάδα προγραμματιστών της Fortune 50 εξοικειώθηκε εύκολα με το λογισμικό της Pulse και με το κέντρο του χώρου εργασίας. Διαφορετικές ομάδες ελέγχου έλαβαν διαφορετικές "συνταγές" του Eclipse, όπως θα απαιτείτο εάν αυτό ήταν πραγματική εγκατάσταση και όχι δοκιμή.

Στη συνέχεια η ομάδα προγραμματιστών της Fortune 50 χρειαζόταν να γνωρίζει εάν το Pulse μπορούσε να διαχειριστεί ένα μεγάλο εύρος διανομής. Το Pulse επιτρέπει την άμεση δημιουργία εγκαταστατών για το Web (ή ένα κλικ), έτσι σε λίγες ώρες, η ομάδα ελέγχου δημιούργησε εγκαταστάτες και τις έστειλε σε άλλους προγραμματιστές στον οργανισμό ώστε να ελέγξουν εάν το Pulse μπορεί να εγκατασταθεί επιτυχώς στους υπολογιστές τους. Χωρίς θέμα, το Pulse ολοκλήρωσε αυτή τη διεργασία και οι προγραμματιστές ελέγχου είχαν ελέγξει ακόμη ένα αντικείμενο της λίστας τους.

Δεδομένου ότι η εν λόγω εταιρεία είναι προμηθευτής λογισμικού, χρειαζόταν να ελέγξει εάν το Pulse της Genuitec θα μπορούσε να μεταφέρει το ιδιόκτητο της λογισμικό εσωτερικά. Επίσης, δεδομένου ότι, κάθε προγραμματιστής ελέγχου της εταιρείας είχε το Pulse εγκατεστημένο στον υπολογιστή του, η ομάδα στη συνέχεια έστειλε σε κάθε μέλος μία έκδοση του λογισμικού της εταιρείας για να διαπιστώσει εάν αυτό λειτουργεί παράλληλα με τις προβλεπόμενες ανοιχτές τεχνολογίες. Και πάλι χωρίς προβλήματα η Pulse πλατφόρμα χειρίστηκε τη διανομή άψογα.

Τέλος, η Fortune 50 εταιρεία έπρεπε να γνωρίζει εάν το ανεπτυγμένο λογισμικό θα μπορούσε να δεχθεί διορθώσεις σφαλμάτων, νέα χαρακτηριστικά και βελτιώσεις ενώ έχει ήδη αναπτυχθεί σε εσωτερικό υπολογιστή. Όπως είχε προγραμματιστεί,

το Pulse ήταν σε θέση να κάνει αλλαγές στους υπολογιστές έτσι ώστε η εταιρεία να μπορέσει να διαχειριστεί τον κύκλο ενημερώσεων.

Η εταιρεία ήταν ευχαριστημένη με την ενοποίηση του Pulse με τα υπάρχοντα εργαλεία και τεχνολογίες τους, και ότι θα επικυρώνονται ονόματα χρηστών και κωδικοί πρόσβασης με τον εσωτερικό Server. Αυτό ήταν σημαντικός παράγοντας προς την κατεύθυνση μείωσης κόστους, καθώς η εταιρεία δεν ενδιαφερόταν για την αγορά μίας λύσης παράδοσης η οποία απαιτούσε την αγορά και εφαρμογή πολλαπλών νέων τεχνολογιών.

Το Συμπέρασμα

Στο τέλος, αυτή η δημόσια εταιρεία τεχνολογίας επέλεξε τη χρήση του Pulse της Genuitec για διάφορους λόγους. Πρώτον και σημαντικότερον, διανέμει κάθε τύπο λογισμικού στη μεγάλη βάση προγραμματιστών και διαχωρίζει συγκεκριμένα έργα λογισμικού μεταξύ ομάδων με άδειες και δικαιώματα πρόσβασης καθορισμένα από το Pulse λογισμικό και το κέντρο του χώρου εργασίας.

Η πλατφόρμα του Pulse χρησιμοποιεί άμεσα διαθέσιμες τεχνολογίες, έτσι ώστε αυτή η οργάνωση δεν χρειαζόταν να αγοράσει ακριβά και βαριά λογισμικά εγκατάστασης για να έχουν μία δυνατή ALM λύση. Και, αφού το Pulse είχε εγκατασταθεί σε όλη την επιχείρηση, επιτεύχθηκε ο στόχος της μετακίνησης των προγραμματιστών, είτε το επιθυμούσαν είτε όχι, πάνω στην Eclipse πλατφόρμα ανοιχτού κώδικα, ώστε σήμερα να παραμένει ανταγωνιστική στην ανάπτυξη λογισμικού και την κατασκευή υλικού.

Τεχνικές Λεπτομέρειες

Η επιχείρηση χρειαζόταν να αναπτύξει ένα IDE βασισμένο στο Eclipse για να προσομοιώσει μία Linux ανάπτυξη cluster. Στην εικονική εγκατάσταση, το Pulse επιτρέπει σε αυτή την εταιρεία η κάθε εικονική μηχανή να παρέχει μία προσαρμοσμένη εμπειρία ανάπτυξης, βασισμένη στους συνδεδεμένους προγραμματιστές χωρίς καμία επιπλέον λήψη λογισμικού και από ένα μόνο σημείο έναρξης για την ευκολία του τελικού χρήστη.

Η εταιρεία είχε μία εφαρμογή διαθέσιμη στο διαδίκτυο αλλά και μία πλούσια πλατφόρμα πελάτη (RCP) βασισμένη στο Eclipse. Η ομάδα ανάπτυξης εφαρμογών ήθελε να μεταβούν οι χρήστες στην RCP έκδοση, δεδομένου ότι είχε καλύτερες δυνατότητες για πλοήγηση και αναφορές δεδομένων. Οι χρήστες παλαιότερα είχαν πρόσβαση στην έκδοση του Web αφού δεν υπήρχε κανένας τρόπος να αποκτήσουν πρόσβαση στο RCP εύκολα, και ένα εβδομαδιαίο e-mail αναφοράς χρησιμοποιούσε μία υπερσύνδεση σε μία δικτυακή πύλη με πρόσβαση στα εβδομαδιαία δεδομένα. Το Pulse επέτρεψε στην εταιρεία να έχει ένα URL στο e-mail που τους οδηγεί απ' ευθείας σε μία τρέχουσα περίπτωση του RCP με αυτόματη απεικόνιση δεδομένων. Για τους χρήστες οι οποίοι δεν είχαν εγκαταστήσει το RCP, το Pulse εγκαταστέι αυτόματα το RCP και τότε θα ανακτούνται τα δεδομένα.

Αυτός είναι πολύ μεγάλος οργανισμός και ήθελε να παρέχει διαφορετικούς καταλόγους λογισμικού και τα προφίλ των χρηστών σε διάφορες περιοχές, καθώς και ανάθεση διαχειριστών για αυτές τις περιοχές. Το Pulse επέτρεψε στην εταιρεία να ορίσει διαφορετικά τμήματα της επιχειρηματικής μονάδας, όλες φιλοξενούμενες από ένα Server, με ανάθεση διαφορετικών διαχειριστών σε αυτές τις περιοχές. Όχι μόνο δίνει τη δυνατότητα οι περιοχές να παίρνουν προσαρμοσμένους καταλόγους, αλλά και να μπορούν να διαχειριστούν και να επεκτείνουν το λογισμικό τους και τον κατάλογο προφίλ όπως απαιτείται.

Εν κατακλείδι η Pulse πλατφόρμα της Genuitec δημιουργεί μία μυριάδα από εγκαταστάτες με βάση τις ιδιαίτερες ανάγκες διανομής της επιχείρησης. Παρέχει μία εξαιρετικά επεκτάσιμη εμπειρία του τελικού χρήστη με την κατασκευή από αυτόν της παροχής λογισμικού μέσω του διαδικτύου, ή πίσω από ένα τείχος προστασίας σε μία ποικιλία από γνωστά κανάλια. Το Pulse επιτρέπει σε παλιές και νέες τεχνολογίες να συνυπάρξουν και δεν κάνει διακρίσεις στο τι μία επιχείρηση επιλέγει να παραδώσει. Είναι γρήγορο και αξιόπιστο και έχει αποδειχθεί αδιάσπαστη λύση σε πολλές εταιρείες.

ΚΕΦΑΛΑΙΟ 5

Ανάπτυξη Web Εφαρμογής για την παρουσίαση του Εργαλείου Eclipse

Η ανάγκη δημιουργίας της παρούσας εφαρμογής που είναι ένα Εβδομαδιαίο Πρόγραμμα Μαθημάτων, είναι για την παρουσίαση του εργαλείου Eclipse, που είναι και η κύρια απαίτηση δημιουργίας της. Τα ενδιαφερόμενα μέρη είναι οι τελικοί χρήστες που θα χρησιμοποιήσουν την εφαρμογή, ενώ η τεχνολογία που απαιτείται είναι η εγκατάσταση του εργαλείου Eclipse, του JDK, καθώς και του Tomcat 7.

Με τη βοήθεια της Ενοποιημένης Μεθοδολογίας (Unified Process) δημιουργούμε την εφαρμογή που λειτουργεί στο περιβάλλον Eclipse, χρησιμοποιώντας τη Java ως γλώσσα προγραμματισμού.

Αρχικά την πρώτη ημέρα ξεκινάμε με τη φάση των απαιτήσεων και τη φάση της ανάλυσης που έχουν περιγραφεί στο κεφάλαιο 3, καταγράφοντας τις Περιπτώσεις Χρήσης δηλαδή τις δυνατότητες και τις προϋποθέσεις με τις οποίες πρέπει να συμμορφώνεται η εφαρμογή. Μια από τις μεγαλύτερες προκλήσεις της διαδικασίας καθορισμού των απαιτήσεων, είναι η ανεύρεση και η καταγραφή όλων των απαιτούμενων χαρακτηριστικών με τρόπο σαφή και ευανάγνωστο και για τον χρήστη, αλλά και για την ομάδα ανάπτυξης.

Παρακάτω περιγράφονται οι προδιαγραφές των Περιπτώσεων Χρήσης (ΠΧ) με τον υψηλότερο βαθμό σημαντικότητας, ώστε να συνεχίσουμε με τα εναπομείναντα βήματα.

Περιπτώσεις Χρήσης - Απαιτήσεις

<u>Απαραίτητη κατάσταση:</u> Το σύστημα πρέπει να είναι σε λειτουργία και να είναι εύχρηστο και φιλικό προς το χρήστη.
<u>Βασικός όρος:</u> Ένας χρήστης πρέπει να θέλει να δημιουργήσει ένα προσωπικό εβδομαδιαίο πρόγραμμα.
<u>Λειτουργία του Εβδομαδιαίου Προγράμματος Μαθημάτων</u>
1) Μία περίπτωση χρήσης ξεκινά όταν υπάρχει τουλάχιστον ένας χρήστης που επιθυμεί να χρησιμοποιήσει την εφαρμογή
2) Ο χρήστης ανοίγει την εφαρμογή
3) Ο χρήστης έρχεται σε επαφή με τη διεπιφάνεια της εφαρμογής
4) Ο χρήστης εισάγει το όνομα του μαθήματος
5) Ο χρήστης εισάγει τις ημέρες που αναλογούν στο μάθημα που εισήγαγε σύμφωνα με το δικό του πρόγραμμα
6) Ο χρήστης εισάγει τις ώρες που αφορούν το μάθημα στην εφαρμογή
7) Ο χρήστης μπορεί να προσθέσει τουλάχιστον δύο μαθήματα την ίδια ώρα και ημέρα
8) Ο χρήστης επιλέγει εάν θέλει εκτύπωση του προγράμματος
9) Σε περίπτωση που ο χρήστης δεν επιθυμεί εκτύπωση του προγράμματος μπορεί να το αποθηκεύσει
<u>Εναλλακτικές Περιπτώσεις Χρήσης</u>
<u>1η Εναλλακτική Περίπτωση:</u> Ο χρήστης καταχωρεί το μάθημα και επιλέγει τις ώρες του μαθήματος αλλά δεν επιλέγει τις ημέρες. Ο χρήστης επιστρέφει στο βήμα 4
<u>2η Εναλλακτική Περίπτωση:</u> Ο χρήστης καταχωρεί το μάθημα και τις ημέρες αλλά δεν επιλέγει τις ώρες του μαθήματος Το σύστημα εισάγει αυτόματα τις αρχικές ώρες επιλογής που είναι από 8:00 π.μ. έως 10:00 π.μ.
<u>3η Εναλλακτική Περίπτωση:</u> Ο χρήστης εισάγει ημέρα και ώρα αλλά όχι το μάθημα Ο χρήστης επιστρέφει στο βήμα 4
<u>4η Εναλλακτική Περίπτωση:</u> Ο χρήστης δεν επιθυμεί τελικά να χρησιμοποιήσει το πρόγραμμα Εγκαταλείπει την εφαρμογή

Πίνακας Συμμετεχόντων - Στόχων

Ως πρώτο βήμα της ενοποιημένης προσέγγισης (έπειτα από την εξαγωγή των απαιτήσεων της εφαρμογής), μπορεί να θεωρηθεί η δημιουργία ενός πίνακα που θα περιέχει τους βασικούς χρήστες (συμμετέχοντες) της εφαρμογής, και τους στόχους που έχουν σχετικά με αυτή. Αυτός ο πίνακας θα αποδειχθεί ιδιαίτερα χρήσιμος για τη συνέχεια, διότι ορισμένοι από τους στόχους θα μετασχηματιστούν σε περιπτώσεις χρήσης (ΠΧ), οι οποίες αποτελούν το σημαντικότερο μηχανισμό για την κατανόηση και περιγραφή των απαιτήσεων ενός συστήματος.

Αξιολόγηση σημαντικότητας Περιπτώσεων Χρήσης

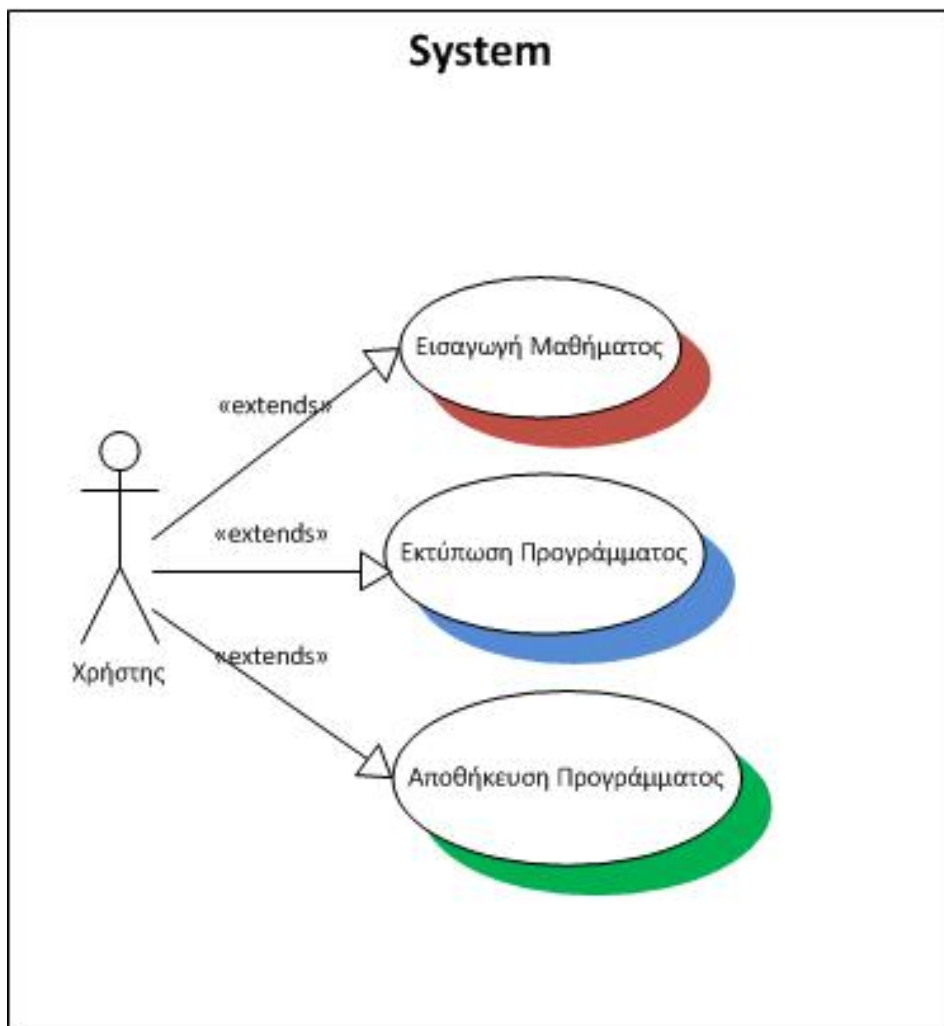
Σε αυτό το σημείο, πρέπει να αξιολογήσουμε με κάποιο βαθμό σημαντικότητας τις περιπτώσεις χρήσεις που έχουμε να προδιαγράψουμε, ώστε να δώσουμε το εναρκτήριο έναυσμα, για την περαιτέρω ανάλυση και περιγραφή των απαιτήσεων της εφαρμογής.

	ΣΥΣΤΗΜΑ ΔΗΜΙΟΥΡΓΙΑΣ ΕΒΔΟΜΑΔΙΑΙΟΥ ΠΡΟΓΡΑΜΜΑΤΟΣ ΜΑΘΗΜΑΤΩΝ	
	ΑΠΑΙΤΗΣΕΙΣ	
ΧΡΗΣΤΕΣ	ΣΤΟΧΟΙ	ΠΡΟΤΕΡΑΙΟΤΗΤΑ
ΧΡΗΣΤΗΣ	ΕΙΣΑΓΩΓΗ ΜΑΘΗΜΑΤΩΝ	1
	ΕΚΤΥΠΩΣΗ	2
	ΑΠΟΘΗΚΕΥΣΗ	1

Πίνακας 5.1: Πίνακας Περιπτώσεων χρήσης Συμμετεχόντων - Στόχων

Διάγραμμα Περιπτώσεων Χρήσης

Το διάγραμμα ΠΧ , αποτελεί μία γραφική μορφή αναπαράστασης των συμμετεχόντων και του τρόπου με τον οποίο χρησιμοποιούν την εφαρμογή. Ένα διάγραμμα ΠΧ μας βοηθάει στο να έχουμε μία καλύτερη οπτική άποψη της εφαρμογής και των ενεργειών που πρόκειται να προβεί ο κάθε συμμετέχοντας, ώστε να μπορέσουμε να αξιολογήσουμε τις ΠΧ ως προς τη σημαντικότητά τους με ευκολότερο τρόπο. (Εικόνα 5.2)



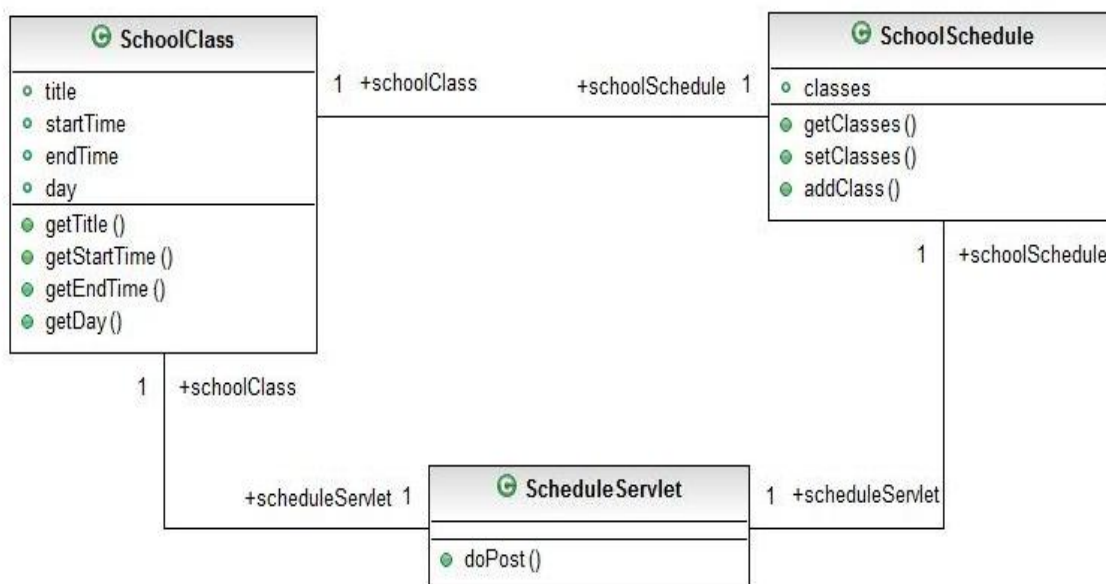
Εικόνα 5.2: Διάγραμμα Περιπτώσεων χρήσης

Για την ολοκλήρωση των δύο φάσεων που πραγματοποιήθηκαν παραπάνω χρειάστηκαν δέκα ημέρες από 29 Αυγούστου έως 7 Σεπτεμβρίου, έτσι ώστε να διευκρινιστούν πλήρως όλες οι απαιτήσεις, να γίνει καταγραφή των Περιπτώσεων Χρήσης και να τεθούν σε εφαρμογή.

Εννοιολογικό Μοντέλο

Το εννοιολογικό μοντέλο αποτελεί μία απεικόνιση των οντοτήτων που συμμετέχουν στην εφαρμογή, και του τρόπου με τον οποίο αλληλεπιδρούν μεταξύ τους. Θα μας βοηθήσει ιδιαίτερα στη σχεδίαση του αναλυτικού διαγράμματος ακολουθίας. Σημείωση: Τα χαρακτηριστικά των οντοτήτων είναι ενδεικτικά, στην πράξη θα πρέπει να επιλεχθούν με προσεκτικό τρόπο.

model

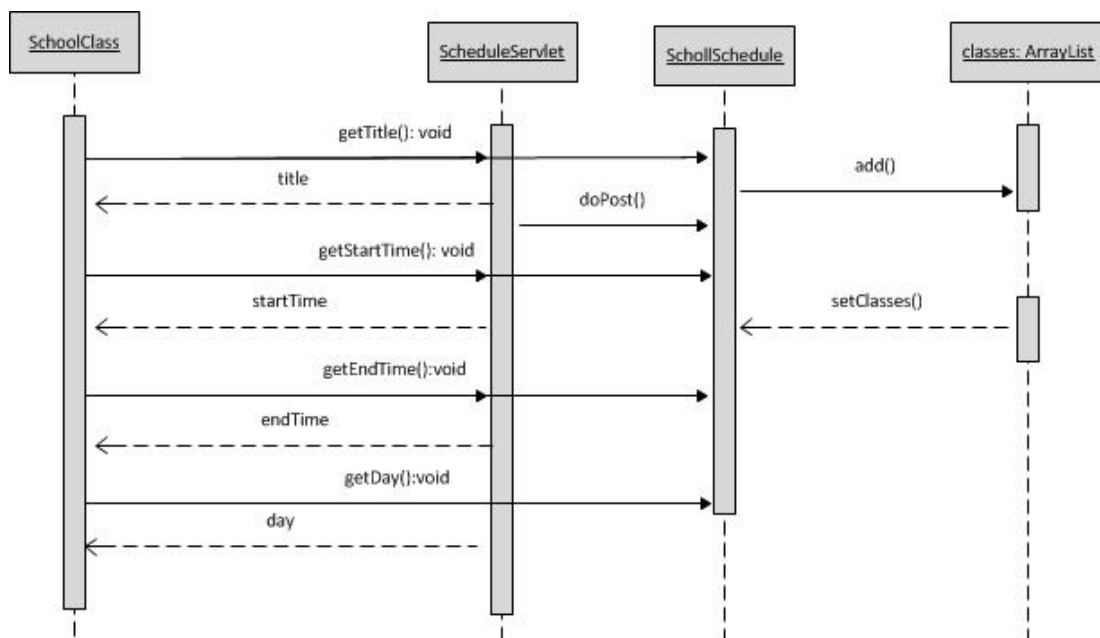


Εικόνα 5.3: Εννοιολογικό Μοντέλο

Διάγραμμα Ακολουθίας

Το διάγραμμα ακολουθίας μας βοηθάει να παρακολουθήσουμε τη ροή των μηνυμάτων που στέλνουν οι συμμετέχοντες στην εφαρμογή, και την απόκριση της για κάθε μήνυμα που δέχεται. Μία καλή τακτική είναι αρχικά να δημιουργήσουμε ένα διάγραμμα ακολουθίας που θα περιέχει ως μηνύματα τα βήματα που έχουν καταγραφεί στη βασική ροή της ΠΧ που προδιαγράψαμε προηγουμένως, και μετέπειτα να μετατρέψουμε το κείμενο των βημάτων σε ένα όνομα λειτουργίας που θα χρησιμοποιηθεί στην πράξη (κώδικας), ακολουθώντας κάποια συγκεκριμένη ονοματολογία για τις λειτουργίες που θα δηλώσουμε, δηλαδή έχοντας καταγράψει τις εννοιολογικές κλάσεις του πεδίου εφαρμογής και τις μεταξύ τους συσχετίσεις, που υποδηλώνουν την αλληλεπίδρασή τους, προχωρούμε στην σχεδίαση αυτών των αλληλεπιδράσεων για κάθε μία περίπτωση χρήσης ξεχωριστά.

Βρισκόμαστε πλέον στη διαδικασία σχεδιασμού και συνεπώς θα πρέπει να σκεφτόμαστε με προγραμματιστικούς όρους. Αυτό σημαίνει ότι δηλώνουμε και καλούμε μεθόδους που υλοποιούν λειτουργίες και εκτελούν εντολές, εξετάζοντας εναλλακτικές λύσεις αποσκοπώντας σε μία ποιοτική σχεδίαση της εφαρμογής μας.



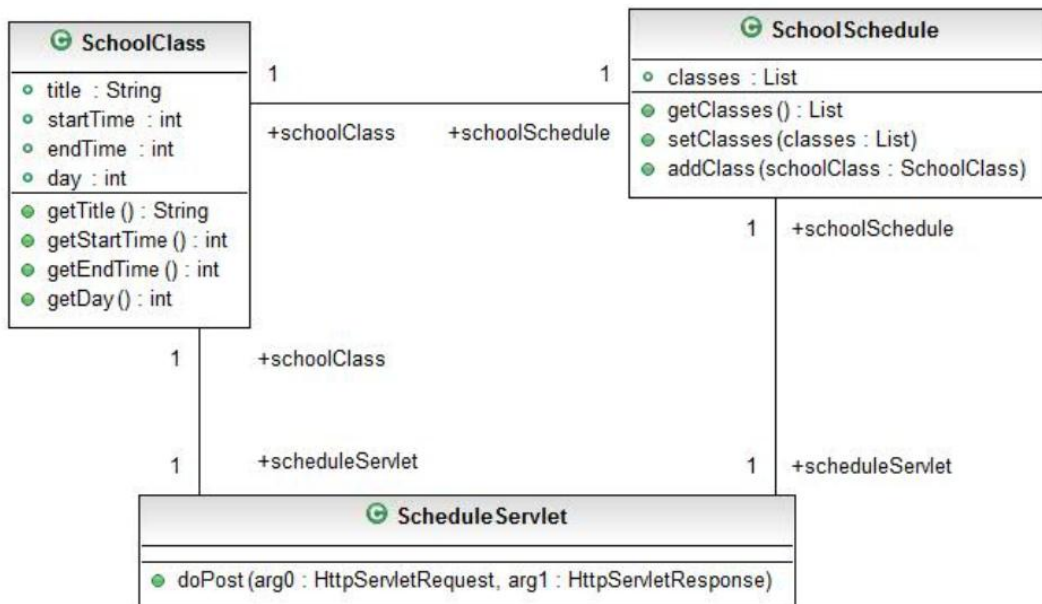
Εικόνα 5.4: Διάγραμμα Ακολουθίας

Όσον αφορά το εννοιολογικό μοντέλο και το διάγραμμα ακολουθίας, δαπανήθηκαν εννέα ημέρες για την εφαρμογή τους από 12 έως 21 Σεπτεμβρίου, έτσι ώστε να

διευκρινιστούν πλήρως οι κλάσεις μέσα από τις οποίες θα διέπεται η εφαρμογή, καθώς και οι μέθοδοι της κάθε κλάσης.

Διάγραμμα Κλάσεων

Στο λεπτομερές διάγραμμα ακολουθίας έχουμε αναπαράσχει τον βήμα προς βήμα σχεδιασμό των περιπτώσεων χρήσης, με τις ανταλλαγές μηνυμάτων μεταξύ των κλάσεων και των αντικειμένων που αυτές περιλαμβάνουν, για την υλοποίηση της λειτουργικότητας τους. Αντιλαμβανόμαστε τη σπουδαιότητα αυτού του διαγράμματος, λόγω του ότι αποτελεί πεδίο λήψης σχεδιαστικών αποφάσεων, ως προς την αποστολή και τη λήψη μηνυμάτων, πράγμα που σημαίνει ανάθεση αρμοδιοτήτων σε κάθε κλάση με την τοποθέτηση σε αυτή μίας ή περισσότερων μεθόδων. Σε ένα διάγραμμα ακολουθίας όμως μπορούμε να δούμε μόνο ένα μέρος της συνολικής λειτουργικότητας της εφαρμογής. Θα πρέπει επομένως οι μέθοδοι (με τις παραμέτρους τους) που έχουν προσδιορισθεί στα διαγράμματα ακολουθίας, να προστεθούν σταδιακά στο εννοιολογικό μοντέλο συγκεντρωτικά (δηλαδή το αρχικό διάγραμμα κλάσεων) ώστε να έχουμε μία πλήρη άποψη του σχεδιασμού ολόκληρης της εφαρμογής.



Εικόνα 5.5: Διάγραμμα Κλάσεων

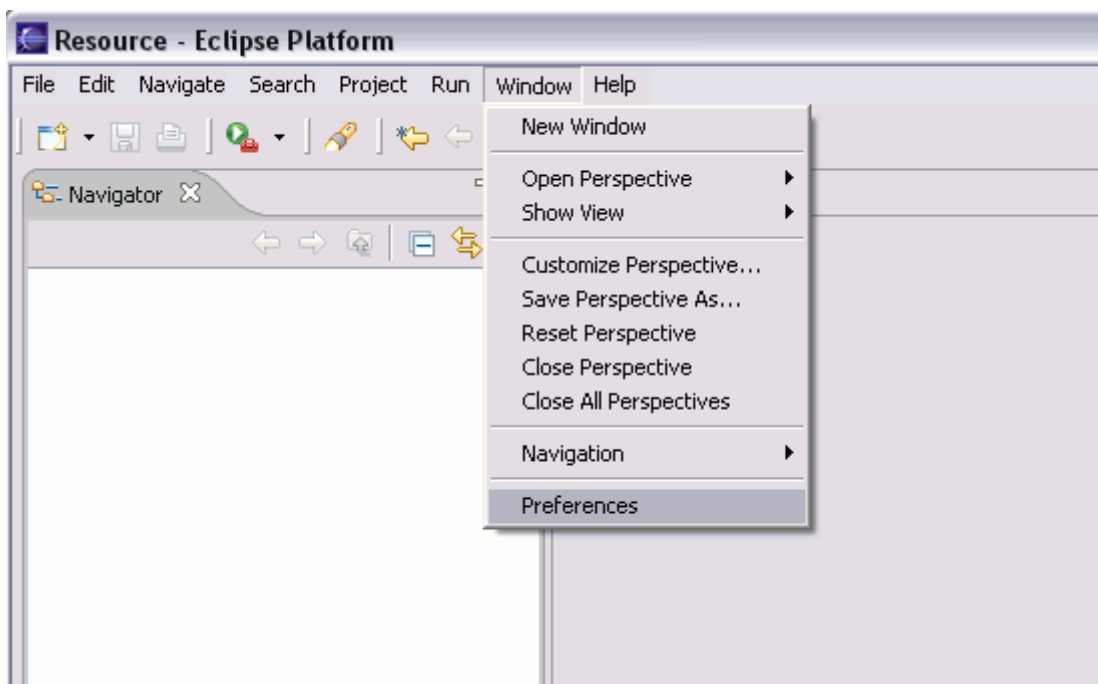
Το διάγραμμα κλάσεων ολοκληρώθηκε μέσα σε δύο ημέρες από 22 έως 23 Σεπτεμβρίου λόγω του ότι υπήρχε το εννοιολογικό μοντέλο και ήταν πλέον ξεκάθαρες οι απαιτήσεις.

Ο επόμενος στόχος μας είναι να μεταφράσουμε το σχεδιασμό της εφαρμογής σε κώδικα σε μία συγκεκριμένη γλώσσα προγραμματισμού. Αυτή είναι η φάση της κωδικοποίησης, όπου γίνεται η υλοποίηση του κώδικα της διαδικτυακής εφαρμογής σύμφωνα με το σχεδιασμό. Η επιλογή της γλώσσας προγραμματισμού είναι η Java, όπως προαναφέρθηκε στην αρχή του κεφαλαίου.

Αφού έχουμε δημιουργήσει τις κλάσεις μας και είμαστε σίγουροι για την ορθότητά τους σύμφωνα με τους ελέγχους που πραγματοποιήσαμε κατά την διάρκεια της φάσης του ελέγχου, προχωράμε στην επόμενη φάση, ώστε να αναπτύξουμε την εφαρμογή. Για να δημιουργήσουμε και να εκτελέσουμε τη διαδικτυακή εφαρμογή στο Eclipse (το οποίο μας επιτρέπει να ελέγξουμε και να διορθώσουμε την εφαρμογή μας) πρέπει να έχουμε εγκαταστήσει το JDK και τον Tomcat server. Ο server Tomcat απαιτεί το JDK ώστε να μπορεί να μεταγλωττίσει servlets και JSPs.

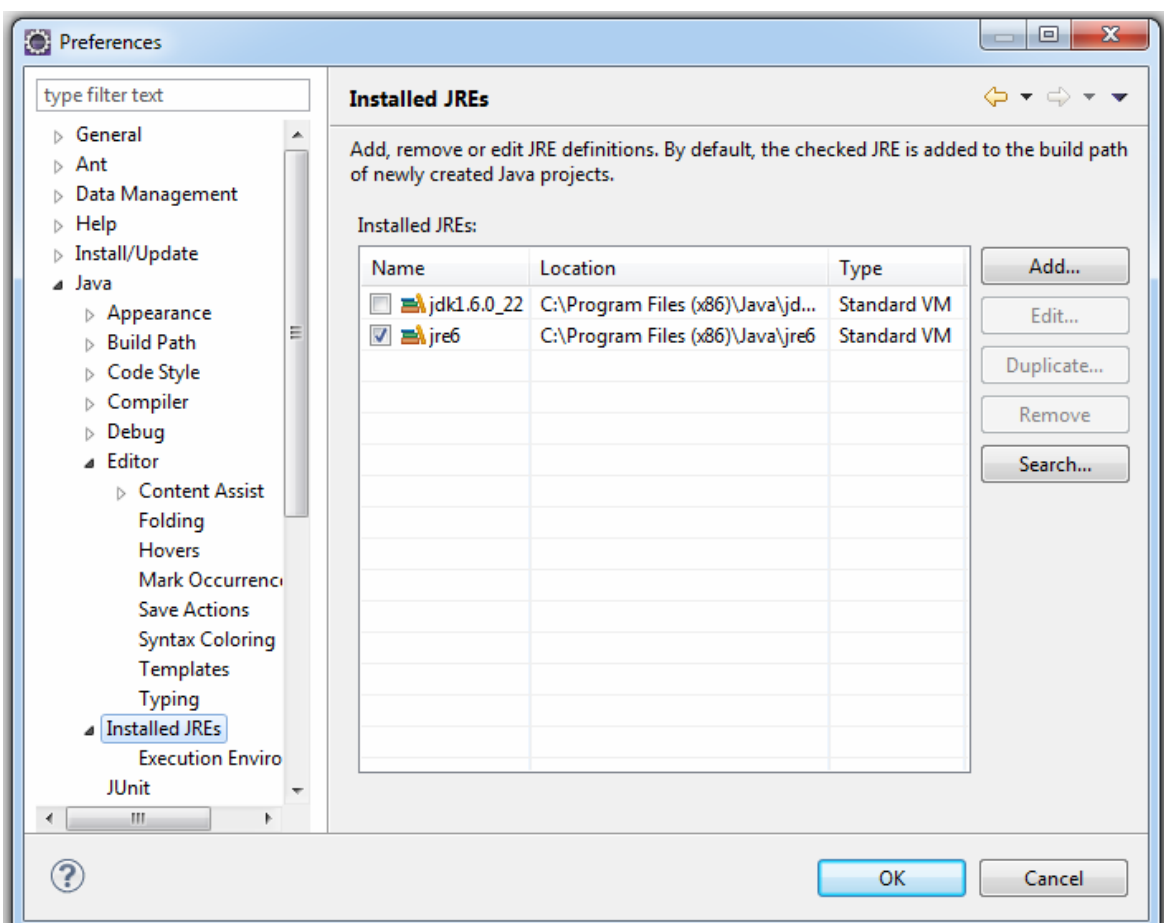
Για την εγκατάσταση των προγραμμάτων που απαιτούνται για την δημιουργία της εφαρμογής χρειάστηκε μία ολόκληρη ημέρα στις 30 Σεπτεμβρίου, ενώ για την ολοκλήρωσή της χρειάστηκαν περίπου δύο με τρεις εβδομάδες από 3 έως 22 Οκτωβρίου. Μετά την εγκατάσταση των προγραμμάτων ακολουθήσαμε τα βήματα όπως περιγράφονται παρακάτω.

Από το περιβάλλον του Eclipse επιλέγουμε Window->Preferences, (Εικόνα 5.6)



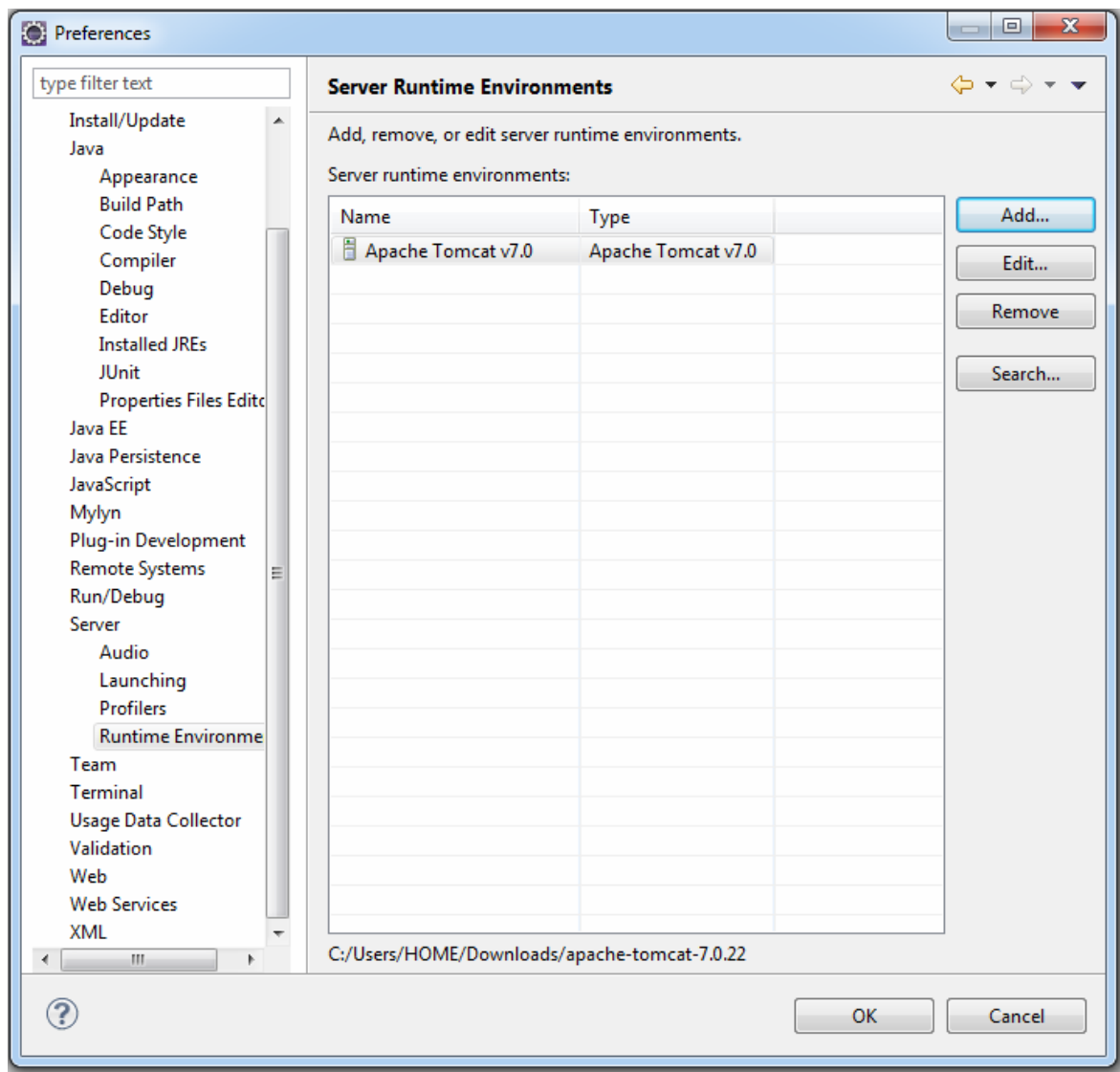
Εικόνα 5.6

και από το μενού αριστερά επιλέγουμε Java->Installed JREs. Στη συνέχεια επιλέγουμε προσθήκη (Add) και ανοίγει το παράθυρο διαλόγου JRE. Επιλέγοντας προσθήκη για τον κεντρικό κατάλογο JRE επιλέγουμε την τοποθεσία όπου θα εγκατασταθεί το JDK και το JDK εμφανίζεται στη λίστα. (Εικόνα 5.7)



Εικόνα 5.7

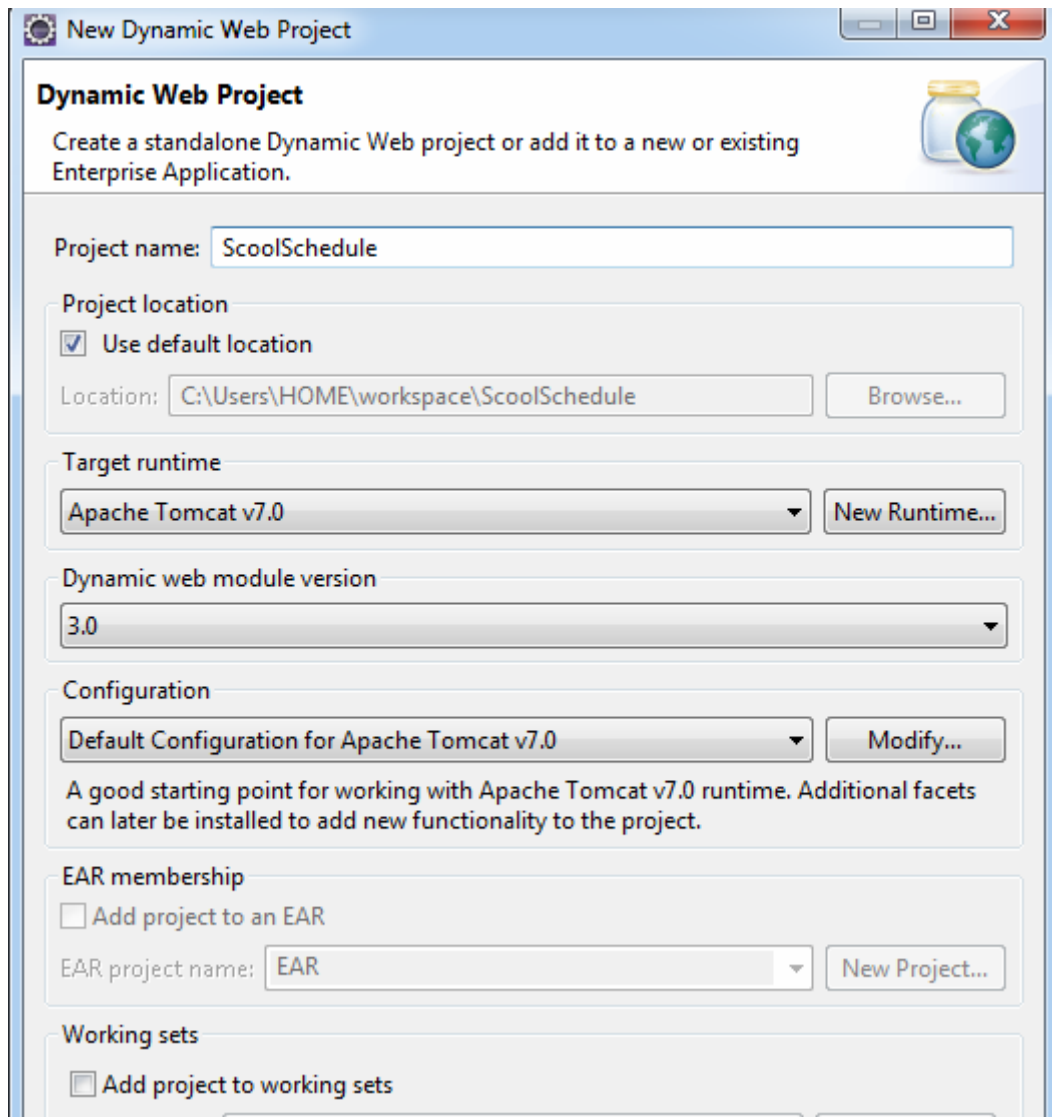
Στη συνέχεια επιλέγουμε και πάλι Window->Preferences->Server->Runtime Environments->Add και επιλέγουμε Apache Tomcat v7.0. Κάνοντας κλικ στην αναζήτηση επιλέγουμε την τοποθεσία όπου έχουμε εγκαταστήσει το Tomcat και το JDK που μόλις έχει εγκατασταθεί. (Εικόνα 5.8)



Εικόνα 5.8

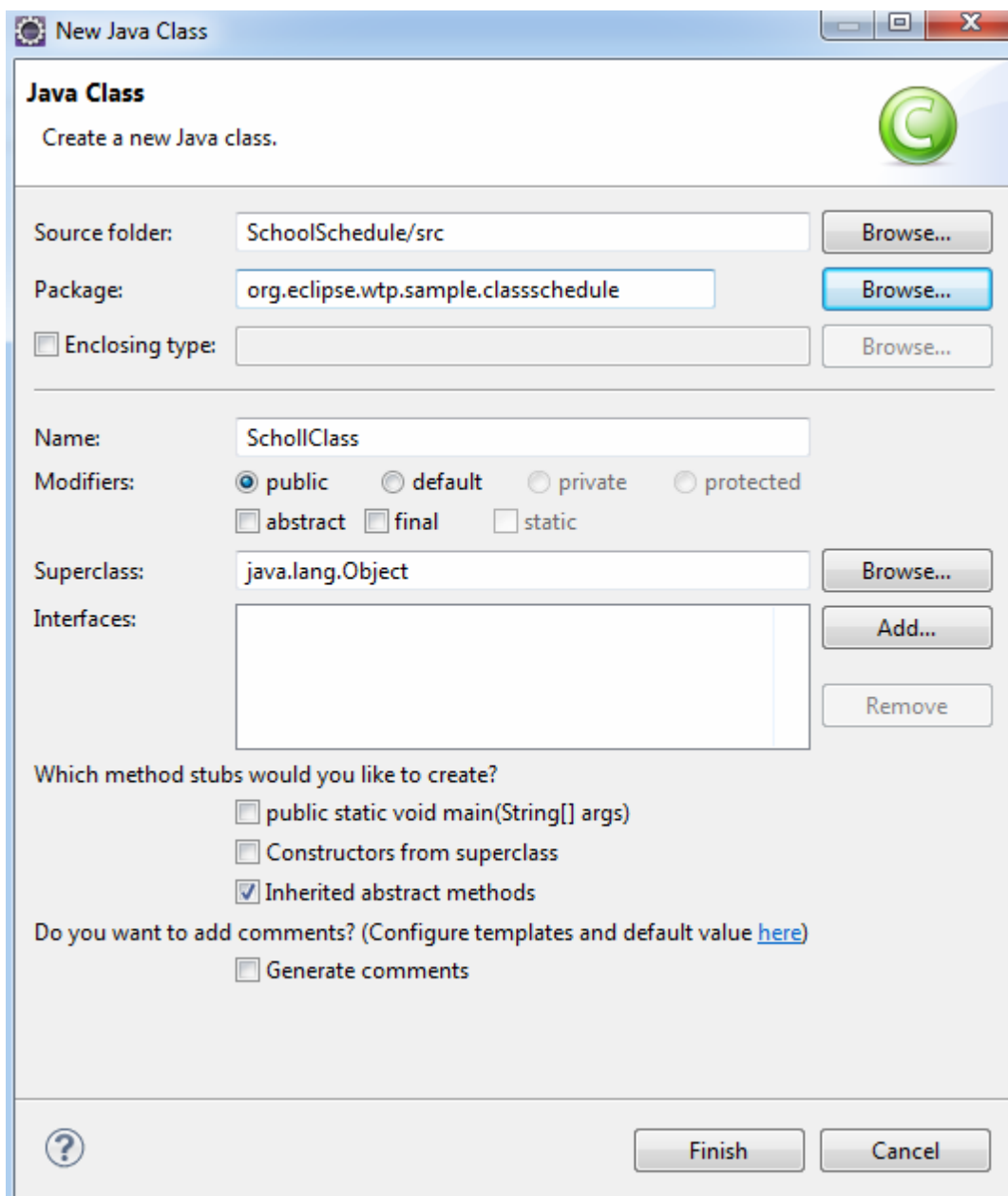
Πατώντας Ok, κλείνουμε το παράθυρο διαλόγου και έτσι το JDK και το Tomcat έχουν ρυθμιστεί στο Eclipse.

Τώρα μπορούμε να δημιουργήσουμε την εφαρμογή μας. Επιλέγουμε New Project ->Dynamic Web Project και ονομάζουμε το Project μας SchoolSchedule (Εικόνα 5.9)



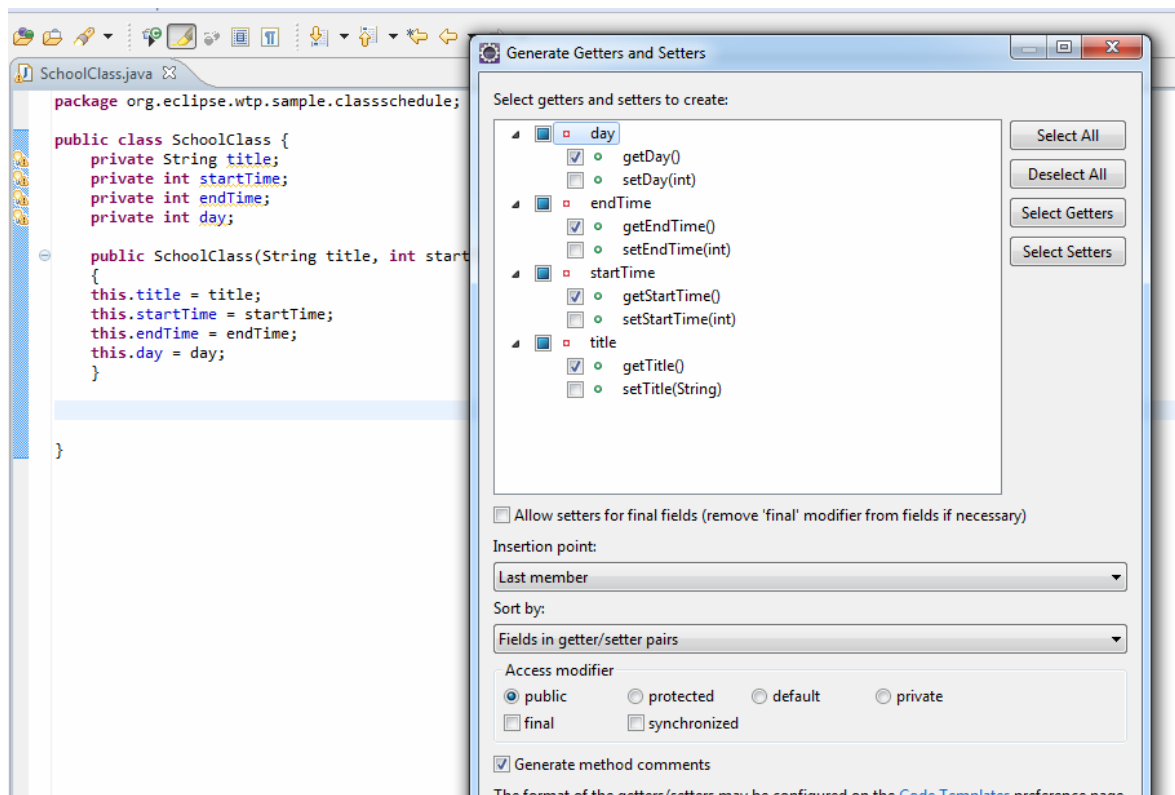
Εικόνα 5.9

Για να περάσει τις πληροφορίες της κλάσης μεταξύ της JSP σελίδας που εμφανίζει το χρονοδιάγραμμα και το servlet που καταγράφει τις κλάσεις με το χρονοδιάγραμμα (πρόγραμμα), θα χρειαστούμε την κλάση ScoolClass που έχουμε δημιουργήσει καθώς και την SchoolSchedule για το χρονοδιάγραμμα. Επιλέγουμε new class, δίνουμε το όνομα ScoolClass και ορίζουμε το πακέτο org.eclipse.wtp.sample.classSchedule. Η κλάση εμφανίζεται στον JavaSource φάκελο και ανοίγουμε το πρόγραμμα επεξεργασίας Java. (Εικόνα 5.10)



Εικόνα 5.10

Αυτή η κλάση θα πρέπει να περιλαμβάνει τα αντικείμενα: τίτλος, ημέρα, ώρα έναρξης και ώρα λήξης του μαθήματος, και οι μεταβλητές να είναι τύπου `private`, καθώς επίσης και ένα δομητή (Constructor) και μεθόδους `get` για καθένα από τα αντικείμενα. Παρατηρούμε πως στο εργαλείο Eclipse δεν χρειάζεται να γράφουμε εμείς τον κώδικα για τις μεθόδους `set` και `get` που ήδη έχουμε δημιουργήσει, γιατί αυτό μπορεί να γίνει αυτόματα. Κάνοντας δεξί κλικ στον επεξεργαστή, επιλέγουμε `source->Generate Getters and Setters` και μας δίνεται η δυνατότητα να επιλέξουμε ποια μέθοδο θέλουμε και έτσι αυτή προστίθεται στην κλάση (Εικόνα 5.11)



Εικόνα 5.11

Ο κώδικας της κλάσης είναι ο ακόλουθος :

```
package org.eclipse.wtp.sample.classschedule;
```

```
public class SchoolClass {
```

```
    private String title;
```

```
    private int startTime;
```

```
    private int endTime;
```

```
    private int day;
```

```
    public SchoolClass(String title, int startTime, int endTime, int day)
```

```
{
```

```
    this.title = title;
```

```
    this.startTime = startTime;
```

```
    this.endTime = endTime;
```

```
    this.day = day;
```

```
}
```

```
/**
```

```
 * @return the title
```

```
 */
```

```

    public String getTitle() {
        return title;
    }

    /**
     * @return the startTime
     */
    public int getStartTime() {
        return startTime;
    }

    /**
     * @return the endTime
     */
    public int getEndTime() {
        return endTime;
    }

    /**
     * @return the day
     */
    public int getDay() {
        return day;
    }
}

```

Στη συνέχεια δημιουργούμε την κλάση `SchoolSchedule` κάνοντας δεξί κλικ στο `org.eclipse.wtp.sample.classschedule` και επιλέγοντας `new Class`. Εδώ σημειώνουμε ότι το πεδίο πακέτο (*Package*) συμπληρώνεται αυτόματα. Στην κλάση `SchoolSchedule` προσθέτουμε μία `private` κλάση τύπου λίστας.

```
private List classes = new ArrayList();
```

Κάνοντας δεξί κλικ στον επεξεργαστή, επιλέγουμε `source->Organize imports`, και από εκεί `java.util.List`. Έτσι οι απαραίτητες δηλώσεις εισαγωγής προστίθενται στην κλάση. Και σε αυτή την κλάση κάνοντας και πάλι δεξί κλικ στον επεξεργαστή και επιλέγοντας `source->generate Getters and Setters` προστίθενται αυτόματα οι μέθοδοι `set` και `get`.

Στη συνέχεια πρέπει να δημιουργήσουμε μία μέθοδο για την προσθήκη νέων κλάσεων στη λίστα. Την μέθοδο την ονομάζουμε `addClass` και παίρνει μία παράμετρο της `SchoolClass` και την προσθέτει στη λίστα.

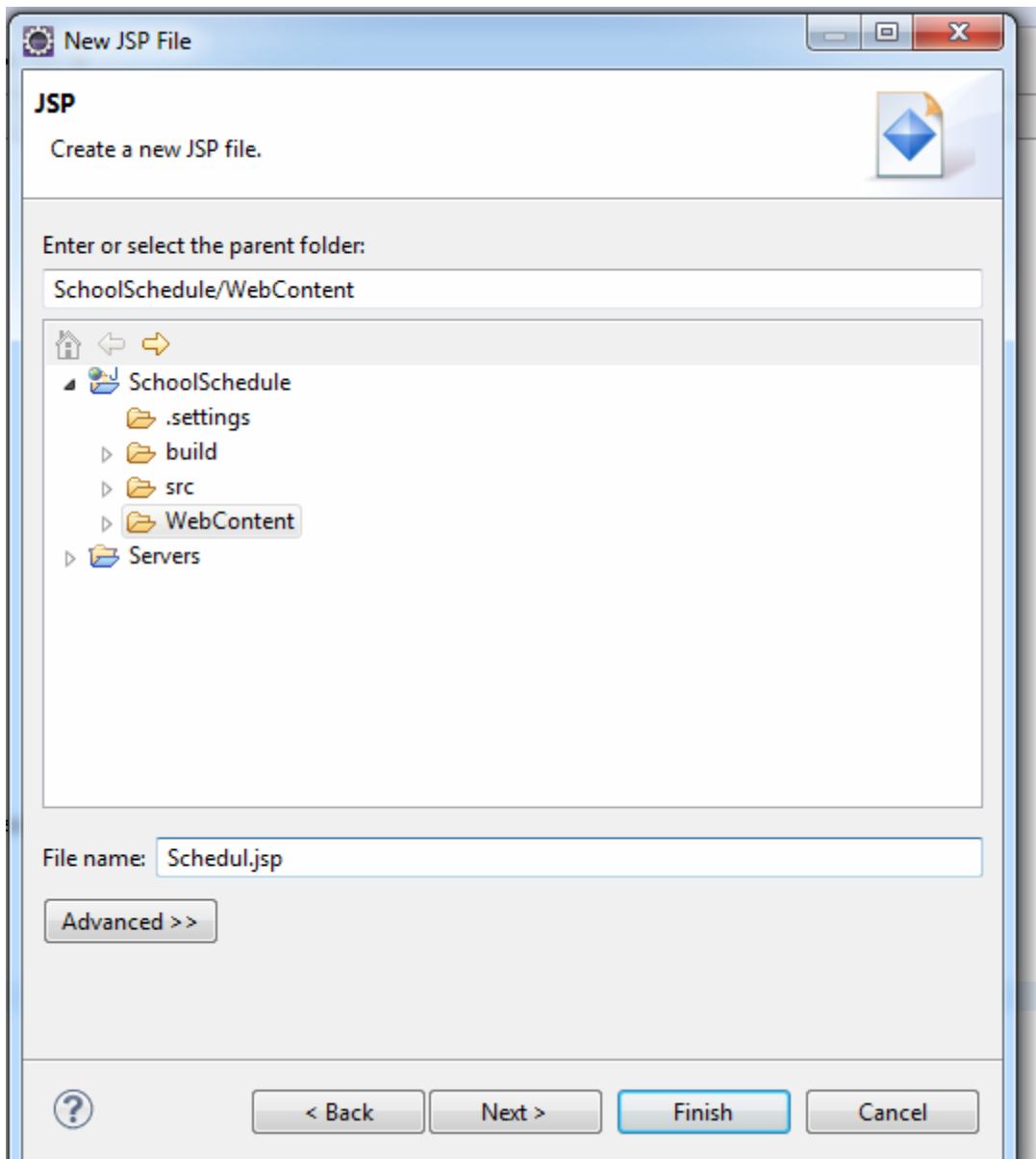
Ο κώδικας της κλάσης είναι ο ακόλουθος:

```
package org.eclipse.wtp.sample.classschedule;

import java.util.ArrayList;
import java.util.List;
public class SchoolSchedule {
    private List classes = new ArrayList();
    /**
     * @return the classes
     */
    public List getClasses() {
        return classes;
    }
    /**
     * @param classes the classes to set
     */
    public void setClasses(List classes) {
        //this.classes = classes;
    }
    public void addClass(SchoolClass schoolClass)
    {
        classes.add(schoolClass);    } }
```

Τώρα θα πρέπει να δημιουργήσουμε μία Java Server Page (JSP), που θα επιτρέπει στους χρήστες να προσθέτουν μία κατηγορία για τον προγραμματισμό τους. Οι JSPs παρέχουν ένα τρόπο πρόσβασης σε αποθηκευμένες πληροφορίες Java αντικειμένων, και εκτελούν Java λειτουργίες σε μία HTML σελίδα.

Κάνοντας δεξί κλικ στον φάκελο `WebContent` επιλέγουμε `new->JSP file` και το ονομάζουμε `Schedule.jsp` (Εικόνα 5.12)



Εικόνα 5.12

Έτσι τώρα μπορούμε να δημιουργήσουμε την φόρμα υποβολής των μαθημάτων. Η δομή μίας JSP σελίδας είναι παρόμοια με την δομή μίας HTML σελίδας. Μία JSP σελίδα μπορεί να χρησιμοποιήσει απλά περισσότερες ετικέτες και να προσφέρει περισσότερες λειτουργίες. Στην πραγματικότητα δημιουργούμε μία HTML σελίδα με μία φόρμα που έχει ένα πλαίσιο κειμένου για την υποβολή ενός τίτλου μαθήματος, πλαίσια επιλογής για τις ημέρες της εβδομάδος και ένα πτυσσόμενο μενού για την επιλογή της ώρας έναρξης και λήξης του μαθήματος.

Ο κώδικας είναι ο εξής:

```
// Όσον αφορά την εμφάνιση του προγράμματος πρέπει να χρησιμοποιήσουμε ορισμένες από τις προσαρμοσμένες βιβλιοθήκες ετικετών που είναι διαθέσιμες για εργασία με JSPs. Αυτές είναι διαθέσιμες στη σελίδα http://jakarta.apache.org/taglibs/.
```

```
//Δήλωση των βιβλιοθηκών ετικετών στο JSP αρχείο.
```

```
<%@taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>
```

```
// Δημιουργούμε δυναμικά έναν πίνακα που θα κρατήσει το πρόγραμμά μας. Ο πίνακας αυτός θα καταγράφει τις ημέρες της εβδομάδος από τις 8:00 π.μ. μέχρι τις 9:00 μ.μ. Για κάθε χρονική στιγμή θα ελέγχει αν υπάρχει μία κατηγορία και αν υπάρχει θα εμφανίζει τον τίτλο της κατηγορίας.
```

```
//Δημιουργία της κεφαλίδας του πίνακα
```

```
<TABLE border="1" cellspacing="0">
<TBODY>
<TR>
<TH align="center" valign="middle" width="80"></TH>
<TH align="center" valign="middle" width="100">Sunday</TH>
<TH align="center" valign="middle">Monday</TH>
<TH align="center" valign="middle">Tuesday</TH>
<TH align="center" valign="middle">Wednesday</TH>
<TH align="center" valign="middle">Thursday</TH>
<TH align="center" valign="middle">Friday</TH>
<TH align="center" valign="middle">Saturday</TH>
</TR>
```

```
//Δημιουργία του περιεχομένου του πίνακα. Για να δημιουργηθεί το περιεχόμενο θα κάνουμε επαναλήψεις μέσω των ωρών της ημέρας και των ημερών της εβδομάδας. Δεδομένου ότι το πρόγραμμα θα εμφανίζει από τις 8:00 π.μ. έως τις 9:00 μ.μ. οι επαναλήψεις θα είναι από 8 έως 21 για την ώρα, ενώ για τις ημέρες της εβδομάδας από 0 έως 6. Υπάρχει κάποια απλή λογική για την εμφάνιση για την εμφάνιση του χρόνου με ωραίο τρόπο. Όπως μπορούμε να δούμε στον κώδικα παρακάτω, οι ετικέτες με το πρόθεμα "c", που προέρχονται από τις βιβλιοθήκες ετικετών επιτρέπουν την επανάληψη και την λογική που χρησιμοποιήσαμε.
```

```
<c:forEach begin="8" end="22" step="2" var="time">
<TR>
<TD align="center" valign="middle" width="80">
<c:choose>
<c:when test="{time == 12}">
<c:out value="{time}" />:00pm
</c:when>
<c:when test="{time > 12}">
```

```

<c:out value="{time - 12}" />:00pm
</c:when>
<c:otherwise>
<c:out value="{time}" />:00am
</c:otherwise>
</c:choose></TD>
<c:forEach begin="0" end="6" step="1" var="day">
<TD align="center" valign="middle" width="100">//Εδώ βλέπουμε που ανακτάται η πληροφορία από το
αντικείμενο του school Schedule που ο servlet δημιουργεί και που //ανακτάται η μεμονωμένη πληροφορία από κάθε
SchoolClass αντικείμενο.
<c:forEach items="{schoolschedule.classes}" var="clazz">
<c:if test="{clazz.startTime <= time
&& clazz.endTime >= time
&& clazz.day == day}">
<c:out value="{clazz.title}" />
</c:if>
</c:forEach>
</TD>
</c:forEach>
</TR>
</c:forEach>
</TBODY>
</TABLE>

<HTML>
<HEAD>
<TITLE>Student Schedule</TITLE>
</HEAD>
<BODY>
//Ορισμός της τοποθεσίας του Servlet που δημιουργούμε αργότερα
<FORM action="/SchoolSchedule/Schedule" method="post">
Course Name: <INPUT type="text" name="title" size="35"><BR>
Course Time:
Mon<INPUT type="checkbox" name="day" value="mon">
Tue<INPUT type="checkbox" name="day" value="tue">
Wed<INPUT type="checkbox" name="day" value="wed">
Thu<INPUT type="checkbox" name="day" value="thu">
Fri<INPUT type="checkbox" name="day" value="fri">
Sat<INPUT type="checkbox" name="day" value="sat">
Sun<INPUT type="checkbox" name="day" value="sun">
<SELECT name="starttime">

```

```
<OPTION value="8">8:00am</OPTION>

<OPTION value="10">10:00am</OPTION>

<OPTION value="12">12:00pm</OPTION>

<OPTION value="14">14:00pm</OPTION>

<OPTION value="16">16:00pm</OPTION>

<OPTION value="18">18:00pm</OPTION>

<OPTION value="20">20:00pm</OPTION>

</SELECT>
to
<SELECT name="endtime">

<OPTION value="10">10:00am</OPTION>

<OPTION value="12">12:00pm</OPTION>

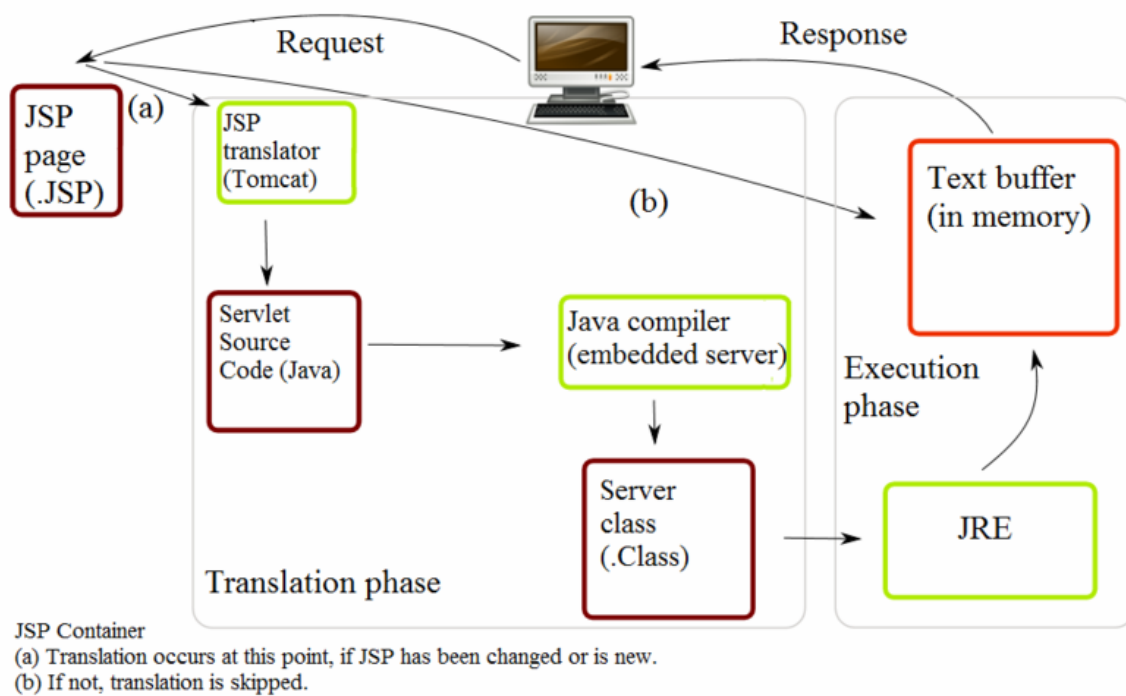
<OPTION value="14">14:00pm</OPTION>

<OPTION value="16">16:00pm</OPTION>

<OPTION value="18">18:00pm</OPTION>
<OPTION value="20">20:00pm</OPTION>

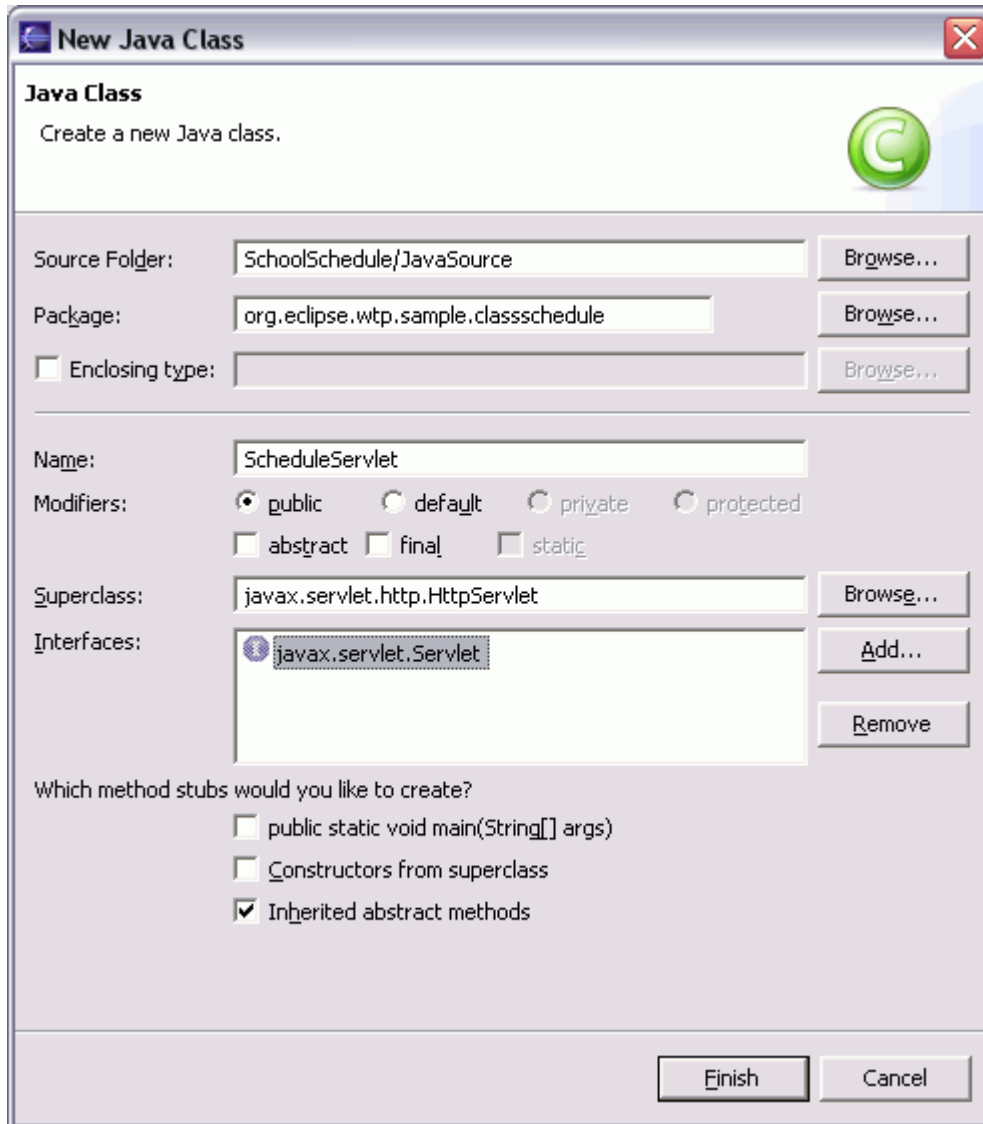
<OPTION value="22">22:00pm</OPTION>
</SELECT>
<BR>
<BR>
<INPUT type="submit" name="Submit" value="Add Course">
//Εκτύπωση του προγράμματος
<p class="dontprintme">
<button type=button onClick="window.print()"
style="width: 100px; height: 50px;">&nbsp;&nbsp; PRINT&nbsp;&nbsp; </button>
<p></FORM>
</BODY>
</HTML>
```

Όταν υποβάλλεται η φόρμα στο Schedule.jsp, πρέπει να εκτελείται μία δράση στα δεδομένα. Τώρα πρέπει να δημιουργήσουμε το servlet που επεξεργάζεται τα στοιχεία εισόδου από τη φόρμα, δημιουργεί τα απαραίτητα SchoolClass αντικείμενα και τα προσθέτει στο SchoolSchedule. Ένα servlet είναι μία Java κλάση που εκτελείται στο διακομιστή. Η λειτουργία του Servlet περιγράφεται αναλυτικά στην εικόνα 5.13.



Εικόνα 5.13

Για να δημιουργήσουμε το servlet κάνουμε δεξί κλικ στο πακέτο org.eclipse.wtp.sample.classschedule, επιλέγουμε new->class και το ονομάζουμε ScheduleServlet. Στο πεδίο SuperClass επιλέγουμε javax.servlet.http.HttpServlet και στο πεδίο Interfaces επιλέγουμε javax.servlet.Servlet. (Εικόνα 5.14)



Εικόνα 5.14

Στον επεξεργαστή με διπλό κλικ επιλέγουμε source->override/Implement Methods και επιλέγουμε τη doPost (HttpServletRequest, HttpServletResponse) μέθοδο. Η λειτουργία αυτής της μεθόδου είναι, να λαμβάνει τα στοιχεία εισόδου από τη φόρμα και να τα εκτελεί. Χρησιμοποιούμε τη doPost μέθοδο επειδή καθορίσαμε από τη φάση της κωδικοποίησης ότι τα δεδομένα θα υποβάλλονται στη φόρμα με τη μέθοδο Post.

```

package org.eclipse.wtp.sample.classschedule;

import java.io.IOException;

import javax.servlet.Servlet;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

public class ScheduleServlet extends HttpServlet implements Servlet {

    @Override
    protected void doPost(HttpServletRequest arg0, HttpServletResponse arg1)
        throws ServletException, IOException {
        // TODO Auto-generated method stub
        super.doPost(arg0, arg1);

        //Το πρώτο πράγμα που πρέπει να κάνουμε στη μέθοδο doPost είναι να διαβάζουμε τις πληροφορίες που υποβάλλονται
        //από τη φόρμα. Αυτές οι πληροφορίες μπορούν να ανακτηθούν μέσα από τη μέθοδο getParameter που δημιουργούμε, η
        //οποία χρησιμοποιείται για τις επιλεγόμενες ημέρες του προγράμματος.

        String title = arg0.getParameter("title");
        int starttime = Integer.parseInt(arg0.getParameter("starttime"));
        int endtime = Integer.parseInt(arg0.getParameter("endtime"));
        String[] days = arg0.getParameterValues("day");

        //Για να λάβουμε το υπάρχον SchoolSchedule (εάν υπάρχει) ή για να δημιουργήσουμε νέο στην περίπτωση που δεν έχει
        //δημιουργηθεί, εκτελούμε τον παρακάτω κώδικα

        SchoolSchedule schedule
        =(SchoolSchedule)arg0.getSession(true).getAttribute("schoolschedule");
        if(schedule == null)
        {
            schedule = new SchoolSchedule();
        }
    }
}

```

//Τώρα μπορούμε να δημιουργήσουμε νέα SchoolClass αντικείμενα. Επειδή μπορεί να υπάρχουν πολλές ημέρες για το μάθημα που θα προσθέτει ο χρήστης, δημιουργούμε ένα νέο SchoolClass αντικείμενο για την κάθε ημέρα που επιλέγεται.

```
if(days != null)
    {
    for(int i = 0; i < days.length; i++)
    {
    String dayString = days[i];
    int day;
    if(dayString.equalsIgnoreCase("SUN")) day = 0;
    else if(dayString.equalsIgnoreCase("MON")) day = 1;
    else if(dayString.equalsIgnoreCase("TUE")) day = 2;
    else if(dayString.equalsIgnoreCase("WED")) day = 3;
    else if(dayString.equalsIgnoreCase("THU")) day = 4;
    else if(dayString.equalsIgnoreCase("FRI")) day = 5;
        else day = 6;

    SchoolClass clazz = new SchoolClass(title, starttime, endtime,
day);
    schedule.addClass(clazz);
    }
    }

//Θα πρέπει να ορίσουμε το πρόγραμμα έτσι ώστε το Schedule.jsp να έχει πρόσβαση σε αυτό

arg0.getSession().setAttribute("schoolschedule", schedule);

getServletContext().getRequestDispatcher("/Schedule.jsp").forward(arg0, arg1);
    }

}
```

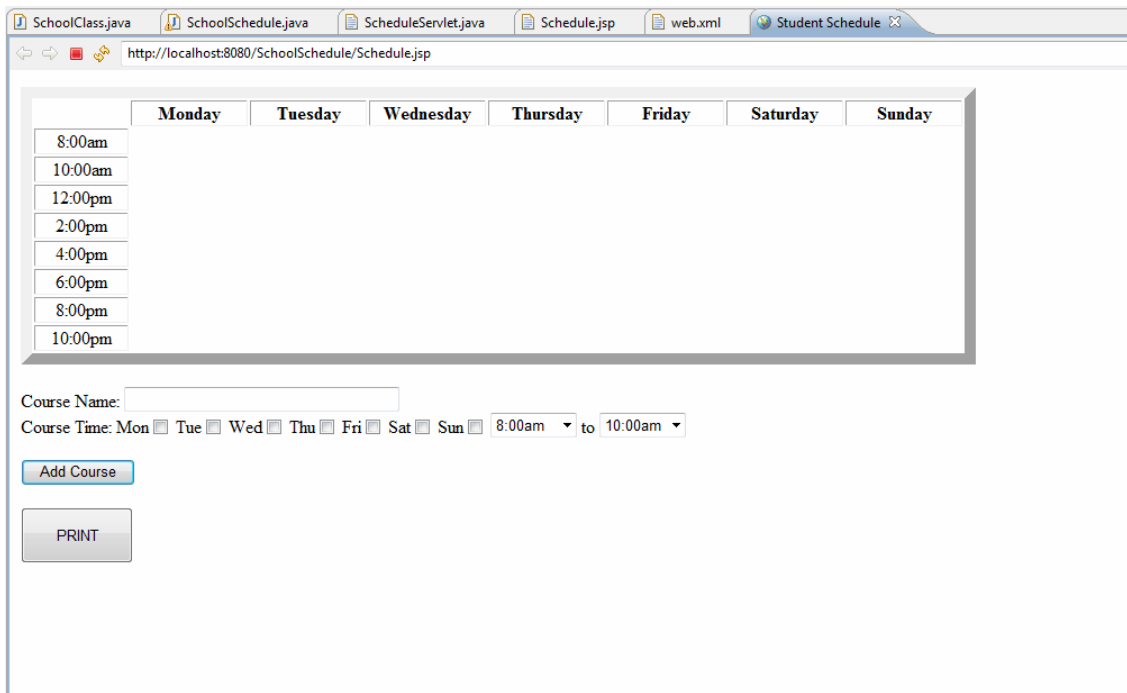
Μέχρι στιγμής έχουμε δημιουργήσει το Servlet μας αλλά η εφαρμογή Web που χρησιμοποιούμε δεν γνωρίζει γι 'αυτό. Έτσι πρέπει να το ορίσουμε στο Deployment Descriptor. Κάνοντας διπλό κλικ επάνω του δημιουργείται το web.xml όπου πληκτρολογούμε τον εξής κώδικα:

```
<servlet>
<description></description>
<display-name>Schedule</display-name>
<servlet-name>Schedule</servlet-name>
<servlet-class>
org.eclipse.wtp.sample.classsschedule.ScheduleServlet
</servlet-class>
</servlet>
<servlet-mapping>
<servlet-name>Schedule</servlet-name>
<url-pattern>/Schedule</url-pattern>
</servlet-mapping>
```

Τώρα που έχουμε ορίσει την τοποθεσία του servlet πρέπει να καθορίσουμε την τοποθεσία και στο Schedule.jsp

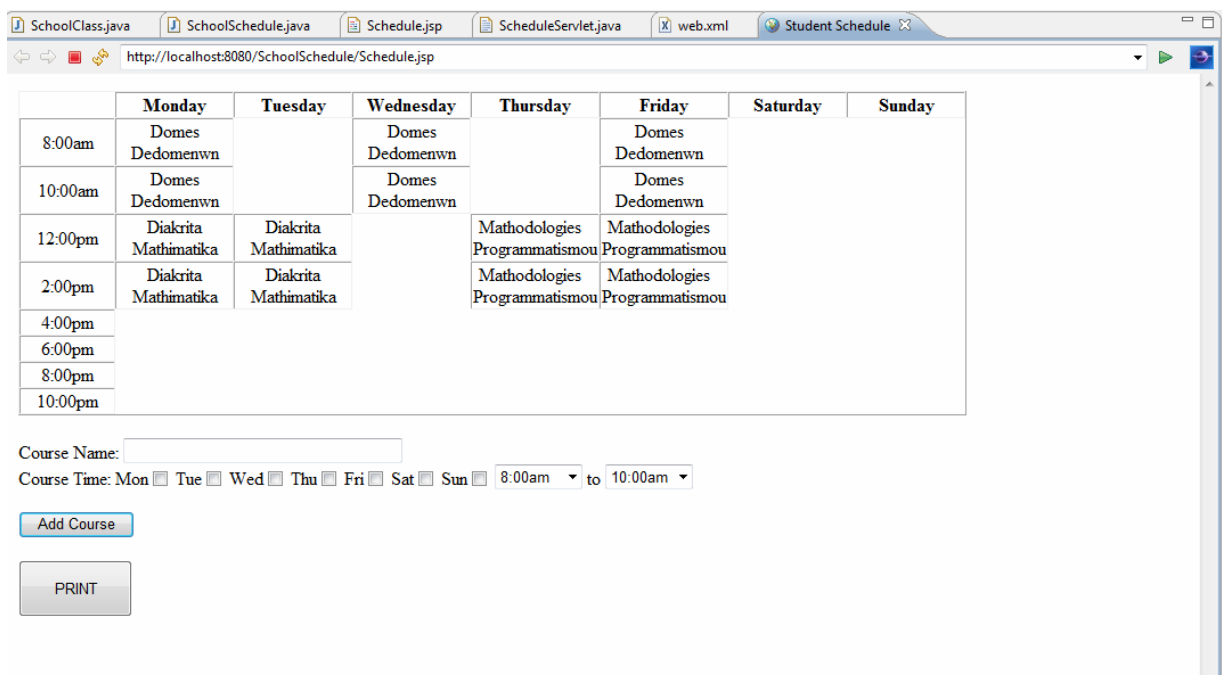
```
<FORM action="/SchoolSchedule/Schedule" method="post">
```

Το Servlet έχει πλέον ορισθεί με τη διαδικτυακή εφαρμογή και είναι έτοιμο να χρησιμοποιηθεί. Κάνοντας δεξί κλικ στο Schedule.jsp και επιλέγοντας Run As->Run on Server, θα ανοίξει ένα παράθυρο του προγράμματος περιήγησης με την εφαρμογή. Το Εβδομαδιαίο Πρόγραμμα Μαθημάτων αρχικά είναι κενό (Εικόνα 5.15).



Εικόνα 5.15

Προσθέτοντας όμως κάποιες κατηγορίες και επιλέγοντας το προσθήκη μαθήματος, το πρόγραμμα μας θα φαίνεται για παράδειγμα ως εξής: (Εικόνα 5.16)



Εικόνα 5.16

Η εφαρμογή είναι διαθέσιμη στον ιστότοπο:

<http://localhost:8080/SchoolSchedule/Schedule.jsp>

Συμπεράσματα

Εν κατακλείδι, σε αυτή την Πτυχιακή Εργασία, κάναμε μία σχετική έρευνα σε σχέση με τις ευέλικτες διαδικτυακές εφαρμογές, μέσα από ερευνητικές μελέτες, συλλογή εμπειρικών δεδομένων και πειραμάτων.

Στη συνέχεια αναπτύξαμε τον κύκλο ζωής των ευέλικτων διαδικτυακών εφαρμογών, όπως αυτός διέπεται σε όλες τις φάσεις ανάπτυξης, από την ανάγκη δημιουργίας της εφαρμογής, έως την παράδοσή της.

Εξετάσαμε το εργαλείο Eclipse, των σημαντικότερων έργων που έχουν αναπτυχθεί με αυτό όπως το έργο Pulsar, τα πλαίσια OSGI για web και άλλες εφαρμογές, αλλά και των πλαισίων μοντελοποίησης που έχουν αναπτυχθεί στην πλατφόρμα του.

Τέλος αναπτύξαμε μία απλή διαδικτυακή εφαρμογή με θέμα την ανάπτυξη ενός εβδομαδιαίου προγράμματος μαθημάτων για την παρουσίαση του εργαλείου Eclipse.

ΒΙΒΛΙΟΓΡΑΦΙΑ

Bartsch, S. (2010), Supporting Authorization Policy Modification in Agile Development of Web Applications, International Conference on Availability, Reliability, and Security, pp. 495 – 500.

Bhuvan Urgaonkar, Prashant Shenoy, Abhishek Chandra, Pawan Goyal, and Timothy Wood, (2008), Agile dynamic provisioning of multi-tier Internet applications, Transactions on Autonomous and Adaptive Systems (TAAS) , Vol 3 Issue 1, pp.1 – 39.

Gary B. Wills, Noura Abbas, Rakhi Chandrasekharan, Richard M. Crowder, Lester Gilbert, Yvonne M. Howard, David E. Millard, Sylvia C. Wong, and Robert J. Walters, (2007), An Agile Hypertext Design Methodology, Proceedings of the eighteenth conference on Hypertext and hypermedia, pp. 1 – 4.

Keo Lee, (2008), An Agile Method for Web Applications in Dynamic Requirements , Fourth International Conference on , Networked Computing and Advanced Information Management, Vol. 2, pp. 178 – 182.

Lili Qu, Yan Chen, and Ming Yang, (2009), The Coordination and Integration of Agile Supply Chain based on Service-oriented Technology, Third International Symposium on Intelligent Information Technology Application, Vol 1, pp. 351 – 354.

Ran Hu, Zhuo Wang, Jun Hu, Jianfeng Xu, and Jun Xie, (2008), Agile Web Development with Web Framework, 4th International Conference on Wireless Communications, Networking and Mobile Computing, pp. 1 – 4.

Siew Poh Lee, Lai Peng Chan, and Eng Wah Lee, (2006), Web Services Implementation Methodology for SOA Application, IEEE International Conference on Industrial Informatics, pp.335 – 340.

Tappenden A., Beatty P., Miller J., Geras A., and Smith M., (2005), Agile Security Testing of Web-Based Systems via HTTPUnit, Proceedings of Agile Conference, pp. 29 – 38.

Vidar Kongsli, (2007), Security Testing with Selenium, conference on Object-oriented programming systems and applications companion, pp. 860 – 864.

Vitor Silva, Ricardo Almeida, (2005), An Agile Approach for Web Systems Engineering, Proceedings of the 11th Symposium on Multimedia and the web, pp. 1 – 3.

Wookjin Lee, Sanghyun Park, Keeyoull Lee, Chunwoo Lee, Byungjeong Lee, Woosung Jung, Taeksu Kim, Heechern Kim, and Chisu Wu, (2005), Agile Development of Web Application by Supporting Process Execution and Extended UML Model, Software Engineering Conference, 2005. APSEC '05. 12th Asia-Pacific, pp. 1 – 8.

Xiaocheng Ge, Richard F. Paige, Fiona A.C. Polack, Howard Chivers, and Phillip J. Brooke, (2006), Agile Development of Secure Web Applications, Proceedings of the 6th international conference on Web engineering, publisher Acm, pp. 305 – 312.

Xiong Guang-cai, Wang Zhi-feng, Zhang Xin-jia, and Ji Guo-jun, (2010), Research of Agile Supply Chain Management System Based on Web Services, 7th International Conference on Service Systems and Service Management (ICSSSM), pp. 1 – 6.

Xufeng Liang, Marmaridis I., Ginige A., (2007), Facilitating Agile Model Driven Development and End-User Development for Evolving Web-based Workflow Applications, IEEE International Conference on e-Business Engineering, pp.231 – 238.

Ιστοσελίδες

<http://www.idt.mdh.se/kurser/dva411/files/Labb1.pdf>

http://www.spidertg.com/cmsv5/images/stories/eBook_Library/Developer/Java%200EE%20Applications%20Made%20Easy.pdf

http://wiki.eclipse.org/Development_Resources/HOWTO/Incubation_Phase

<http://www.vogella.de/articles/Eclipse/article.html>

http://en.wikipedia.org/wiki/Eclipse_%28software%29

http://www.eclipse.org/projects/dev_process/development_process_2011.php

<http://www.eclipse.org/eclipsert/whitepaper/eclipseRT.php>

<http://www.eclipse.org/pulsar/>

http://www.eclipse.org/org/press-release/20090310_pulsar.php

<http://wiki.eclipse.org/EMIWG>

<http://www.eclipse.org/org/industry-workgroups/mobilewg.php>

<http://ostatic.com/blog/eclipse-pulsar-platform-uniting-mobile-manufacturers-with-a-single-development-platform>

<http://www.eclipse.org/home/categories/?category=modeling&tab=start>

<http://www.eclipse.org/home/categories/?category=frameworks&tab=start>

<http://www.eclipse.org/modeling/emf/>

<http://www.eclipse.org/modeling/gmp/>

<http://www.eclipse.org/gef/>

<http://www.eclipse.org/modeling/mdt/>

<http://www.eclipse.org/mylyn/start/>

<http://wiki.eclipse.org/index.php/Mylyn/FAQ>

<http://en.wikipedia.org/wiki/Mylyn>

<http://www.vogella.de/articles/Mylyn/article.html>

<http://www.poweredbypulse.com/>

<http://www.genuitec.com/>