ΑΝΩΤΑΤΟ ΤΕΧΝΟΛΟΓΙΚΟ ΕΚΠΑΙΔΕΥΤΙΚΟ ΙΔΡΥΜΑ ΚΡΗΤΗΣ

Σχολή Τεχνολογικών Εφαρμογών
Τμήμα Εφαρμοσμένης Πληροφορικής και Πολυμέσων

Πτυχιακή Εργασία

Προηγμένες Τεχνικές Διαχείρισης Διαδραστικών Αντικειμένων

Μελέτη Περίπτωσης Βιβλιοθήκης Swing και Εφαρμογή στο eKoNES

Μηλολιδάκης Ιωάννης

Επιβλέπων Καθηγητής : Ακουμιανάκης Δημοσθένης

Ηράκλειο Μάιος 2007

Πίνακας Περιεχομένων

Πίνακας Εικόνων ..... **iii**

Ευχαριστίες ..... **iv**

1 Εισαγωγή **1**

2 Γραφικές διεπαφές χρήστη-υπολογιστή **3**

2.1 Γραφικά στυλ αλληλεπίδρασης χρήστη-υπολογιστή *4*

2.2 Συστατικά σύγχρονων διεπαφών *8*

2.3 Εργαλεία ανάπτυξης διεπαφών *9*

2.3.1 Γλώσσες προγραμματισμού 10

2.3.2 Βιβλιοθήκες γραφικών διαδραστικών αντικειμένων 10

2.3.3 Γλώσσες σήμανσης (Markup languages) 13

2.4 Προβλήματα με σύγχρονες βιβλιοθήκες *14*

3 Υποστήριξη σύνθετων συστατικών αλληλεπίδρασης **16**

3.1 Σύνθετα διαδραστικά στυλ αλληλεπίδρασης *16*

3.2 Προβλήματα υλοποίησης σύνθετων διαδραστικών στυλ *18*

3.3 Γιατί χρειάζεται ένα ενιαίο πλαίσιο υποστήριξης *19*

4 Πλαίσιο αναφοράς & στρατηγικές υποστήριξης **21**

4.1 Επαύξηση (Augmentation) *21*

4.2 Επέκταση (Expansion) *23*

4.3 Ενσωμάτωση (Integration) *24*

4.4 Αφαίρεση (Abstraction) *26*

5 Εφαρμογή πλαισίου στο Swing **27**

5.1 Επισκόπηση του έργου eKoNEΣ *27*

5.2 Παραδείγματα επαύξησης (augmentation) *28*

5.2.1 RadioCheckBoxTree 28

5.2.2 ClosableTabbedPane 32

5.3 Παραδείγματα επέκτασης (expansion) *35*

5.3.1 ActivityPanel (πάνελ δραστηριοτήτων) 36

5.4 Παραδείγματα ενσωμάτωσης (integration) *40*

Σύνοψη και συμπεράσματα **43**

Βιβλιογραφία **44**

Πίνακας Εικόνων

Εικόνα 1: Γραμμή εντολών (command line) **4**
Εικόνα 2: Συμπλήρωση πεδίων φόρμας (form fill-in) **5**
Εικόνα 3: Μενού επιλογής (menu selection) **6**
Εικόνα 4: Άμεση διαχείριση (direct manipulation) **7**
Εικόνα 5: Γενιές γλωσσών προγραμματισμού συστημάτων λογισμικού **9**
Εικόνα 6: MVC vs. UI delegate **11**
Εικόνα 7: Ιεραρχία κλάσεων του Swing **12**
Εικόνα 8: Το αντικείμενο συμπερίληψης 'βιβλίο' στο Motif **16**
Εικόνα 9: Ιεραρχία αντικειμένων στην MFC **16**
Εικόνα 10: TabbedPane με ένδειξη πολλαπλής ταυτόχρονης χρήσης στο Swing **17**
Εικόνα 11: Παράδειγμα σύνθετου διαδραστικού στυλ αλληλεπίδρασης **17**
Εικόνα 12: Αλλαγή τεχνικών αλληλεπίδρασης εργαλειοθήκης **19**
Εικόνα 13: Στρατηγική ενσωμάτωσης εργαλειοθήκης **19**
Εικόνα 14: Αφηρημένα συστατικά και 'σύνδεση με' αντί 'κλήση' βιβλιοθήκης **20**
Εικόνα 15: Επαύξηση (augmentation) **21**
Εικόνα 16: Η χρήση διακοπτών για την εκτέλεση καθηκόντων επεξεργασίας κειμένου **22**
Εικόνα 17: Επέκταση (expansion) **24**
Εικόνα 18: Ενσωμάτωση (Integration) **25**
Εικόνα 19: Κατηγορίες κουμπιών στο Xforms **25**
Εικόνα 20: Μπάρα κύλισης στο Xforms **26**
Εικόνα 21: RadioCheckBoxTree **29**
Εικόνα 22: Διάγραμμα κλάσεων του RadioCheckBoxTree **30**
Εικόνα 23: Σύνοψη των κλάσεων και των μεθόδων που υλοποιούν το RadioCheckBoxTree **31**
Εικόνα 24: ClosableTabbedPane **33**
Εικόνα 25: Διάγραμμα κλάσεων του ClosableTabbedPane **34**
Εικόνα 26: Σύνοψη κλάσεων και μεθόδων που υλοποιούν το ClosableTabbedPane **35**
Εικόνα 27: ActivityPanel **36**
Εικόνα 28: ActivityPanel (TV metaphor) **37**
Εικόνα 29: Διάγραμμα κλάσεων του ActivityPanel **38**
Εικόνα 30: Σύνοψη των κλάσεων και των μεθόδων που υλοποιούν το ActivityPanel **39**
Εικόνα 31: Το message board του eKoNES **41**
Εικόνα 32: Ενσωμάτωση βιβλιοθήκης JGraph **42**

Ευχαριστίες

Πρώτη απ' όλους θα ήθελα να ευχαριστήσω την οικογένεια μου, που χωρίς την δική της υποστήριξη, οικονομική αλλά πάνω από όλα ηθική, δεν θα είχα καταφέρει να τελειώσω τις σπουδές μου στην τριτοβάθμια εκπαίδευση. Η συμπαράσταση της οικογένειας μου ήταν αυτή που με βοήθησε να επιτύχω τους στόχους που είχα θέσει και η συμπαράσταση της θα είναι αυτή που θα με βοηθήσει να πετύχω και τους μελλοντικούς μου στόχους. Επίσης, θα ήθελα να ευχαριστήσω την Ρένα για την συμπαράσταση της και πάνω από όλα για την υπομονή της.

Η συμμετοχή στο έργο eKoNEΣ αλλά και η εργασία μου στο εργαστήριο iSTLab, στα πλαίσια της πρακτικής άσκησης του τμήματος, βοήθησαν σε έναν σημαντικό βαθμό στην ολοκλήρωση αυτής της εργασίας. Θα ήθελα να ευχαριστήσω τον κύριο Δημοσθένη Ακουμιανάκη για την καθοδήγηση του, τόσο κατά την διάρκεια της πρακτικής μου άσκησης, όσο και κατά την διάρκεια της συγγραφής αυτής της πτυχιακής εργασίας. Επίσης, θα ήθελα να ευχαριστήσω, τον κύριο Νίκο Βιδάκη για την βοήθεια του στα πλαίσια της εργασίας μου στο έργο eKoNEΣ, τον Γιώργο, τον Μήτσο, αλλά και όλα τα παιδία του iSTLab για την βοήθεια και την υποστήριξη τους.

1 Εισαγωγή

Η δημιουργία γραφικών διεπαφών χρήστη υπολογιστή τα τελευταία χρόνια έχει απλοποιηθεί κατά πολύ και δεν είναι λίγα τα εργαλεία που έχουν παρουσιαστεί για την γρήγορη και εύκολη ανάπτυξη τέτοιων διεπαφών. Ο προγραμματιστής κάνοντας χρήση σύγχρονων εργαλείων μπορεί πολύ εύκολα να δημιουργήσει διεπαφές, πολλές φορές απλά σύροντας διαδραστικά αντικείμενα πάνω σε μία άδεια οθόνη. Η διεπαφές αυτές δημιουργούνται πολύ εύκολα εξοικονομώντας προγραμματιστικό χρόνο, αλλά τι γίνεται στις περιπτώσεις που τα προσφερόμενα διαδραστικά συστατικά δεν επαρκούν για την υλοποίηση πιο σύνθετων διεπαφών.

Σε αυτήν την πτυχιακή εργασία θα παρουσιάσουμε διάφορες τεχνικές για την διαχείριση διαδραστικών αντικειμένων. Πρώτα από όλα όμως θα προσπαθήσουμε να εξηγήσουμε στον αναγνώστη τι είναι γραφικές διεπαφές χρήστη υπολογιστή, από ποια συστατικά αποτελούνται και τι θα πρέπει να ξέρει κάποιος για να υλοποιήσει τέτοιες διεπαφές. Επίσης θα αναλύσουμε τι είναι στυλ αλληλεπίδρασης, τι είναι διαδραστικά αντικείμενα, τι είναι οι βιβλιοθήκες που τα περιέχουν και ποια είναι τα προβλήματα που συναντά ο προγραμματιστής όταν έχει να διαχειριστεί σύνθετα διαδραστικά αντικείμενα και σύγχρονες βιβλιοθήκες αντικειμένων.

Η πτυχιακή αυτή γράφτηκε στα πλαίσια του έργου eKoNΕΣ το οποίο ήταν και το πεδίο εφαρμογής των υλοποιήσεων που έγιναν, αλλά και η αφορμή για την ενασχόληση με τέτοιου είδους ζητήματα. Η ενασχόληση με θέματα της Διεπαφής Χρήστη – Υπολογιστή προέκυψε μέσα από την υλοποίηση ενός εργαλείου το οποίο έχει την δυνατότητα να υποστηρίξει κοινότητες και ομάδες ανθρώπων για την επίτευξη των στόχων που η ίδια η ομάδα ή κάποιος άλλος θέτει. Η ανάγκη για υλοποιήσεις σύνθετων αντικειμένων και αντικειμένων που επιτελούν συγκεκριμένες λειτουργίες προήλθε τόσο από την πολυπλοκότητα του εργαλείου αυτού, όσο και από το ίδιο το πεδίο εφαρμογής του έργου, το οποίο ήταν ο τουρισμός. Η αντιμετώπιση των διαφόρων προβλημάτων που παρουσιάστηκαν αναλύονται στα επόμενα κεφάλαια της πτυχιακής αυτής.

Η εργασίες μας πάνω σε βιβλιοθήκες διαδραστικών αντικειμένων μας όπλισε με τις απαραίτητες δεξιότητες και την ανάλογη εμπειρία έτσι ώστε να είμαστε σε θέση να προτείνουμε λύσεις στα προβλήματα που αντιμετωπίζει σήμερα ο προγραμματιστής που χρησιμοποιεί τέτοιες βιβλιοθήκες. Η σύγχρονες βιβλιοθήκες, οι οποίες παρά τα όποια οφέλη αποκομίζει κάποιος από την χρήση τους, παρουσιάζουν προβλήματα όταν το πεδίο εφαρμογής τους γίνει εξειδικευμένο και πολύπλοκο. Έτσι, οι επιθυμητές υλοποιήσεις γίνονται με μεγάλο κόστος σε χρόνο και κόπο. Με τις τεχνικές που θα παρουσιαστούν παρακάτω και με το ενιαίο πλαίσιο υποστήριξης που θα προτείνουμε, προσπαθούμε να δώσουμε λύσεις στα πολύπλοκα προβλήματα που καλείται να λύσει ένας προγραμματιστής.

Η τεχνικές της Επαύξησης (Augmentation), της Επέκτασης (Expansion), της Ενσωμάτωσης (Integration) και της Αφαίρεσης (Abstraction) θα αναλυθούν στα επόμενα κεφάλαια όπου και θα παρουσιαστούν οι υλοποιήσεις που έγιναν στα πλαίσια του έργου eKoNΕΣ. Οι τεχνικές αυτές χρησιμοποιήθηκαν ώστε να δοθούν λύσεις σε προβλήματα που προέκυψαν όταν παρουσιάστηκε η ανάγκη για αντικείμενα αυξημένης χρηστικότητας ή αντικείμενα περιορισμένα από το πεδίο εφαρμογής. Επίσης θα παρουσιαστούν και οι περιπτώσεις όπου υπήρξε η ανάγκη για ενσωμάτωση και διαλειτουργικότητα ανάμεσα σε δύο βιβλιοθήκες διαδραστικών αντικειμένων, τα προβλήματα που παρουσιάστηκαν και οι τρόποι με τους οποίους αντιμετωπίστηκαν τα προβλήματα αυτά.

Θα γίνει αναλυτική αναφορά στους τρόπους υλοποίησης των νέων κλάσεων και μεθόδων που παρουσιάστηκαν για την δημιουργία των νέων αντικειμένων ή των αντικειμένων αυξημένης λειτουργικότητας που χρησιμοποιούνται στο eKoNEΣ, όπως επίσης και η σημασία χρήσης τους στο έργο. Επίσης, θα δοθεί μία σύντομη αναφορά για το τι είναι το eKoNEΣ, ποιους αφορά και τι σκοπό έχει ώστε να γίνει πιο σαφής η όλη εικόνα που θα έχει ο αναγνώστης. Τέλος, θα καταλήξουμε με τα απαραίτητα συμπεράσματα μέσα από την ανάλυση των αποτελεσμάτων που προέκυψαν από την εφαρμογή των τεχνικών και του ενιαίου πλαισίου υποστήριξης που προτείνουμε.

2 Γραφικές διεπαφές χρήστη-υπολογιστή

Για να επιτευχθεί αλληλεπίδραση μεταξύ ενός χρήστη και ενός υπολογιστικού συστήματος θα πρέπει οι εταίροι – ο χρήστης και το σύστημα – να 'συμφωνήσουν' σε ένα κοινά αποδεκτό πρωτόκολλο ανταλλαγής μηνυμάτων και επίγνωσης της τρέχουσας κατάστασης των εταίρων. Αυτό προϋποθέτει συμβάσεις και εργαλεία μέσω των οποίων εισάγονται δεδομένα από τον χρήστη και εξάγονται αποτελέσματα από το σύστημα. Τόσο η εισαγωγή δεδομένων όσο και η εξαγωγή αποτελεσμάτων από το σύστημα πρέπει να υποστηρίζονται με τρόπο και μέσα που να τις καθιστούν αντιληπτές. Ωστόσο αυτό που αντιλαμβάνεται ο άνθρωπος διαφέρει από αυτό που αντιλαμβάνεται ένας ηλεκτρονικός υπολογιστής και αυτό γιατί τα αντίστοιχα αισθητήρια όργανα διαφέρουν. Για παράδειγμα ένα πιλοτήριο αεροσκάφους έχει όλα εκείνα τα εργαλεία που χρειάζεται ένας πιλότος για να 'πετάξει' το αεροσκάφος. Ο πιλότος μπορεί να χρησιμοποιήσει το πηδάλιο για να ελέγξει τη κλίση των φτερών ή να χρησιμοποιήσει κάποιον μοχλό για να σηκώσει τους τροχούς του αεροσκάφους. Οι χειρισμοί των εργαλείων αυτών μετατρέπονται σε εντολές που διερμηνεύονται από το σύστημα το οποίο στη συνέχεια αναλαμβάνει αφενός να εκτελέσει όλες εκείνες τις μηχανικές λειτουργίες που απαιτούνται και αφετέρου να ενημερώσει το χειριστή ότι οι λειτουργίες ολοκληρώθηκαν με επιτυχία (ή και το αντίθετο). Όλα αυτά τα εργαλεία που επιτρέπουν την αμφίδρομη επικοινωνία (δηλαδή την εισαγωγή δεδομένων και την εξαγωγή αποτελεσμάτων) μεταξύ των δύο εταίρων αποτελούν τα συστατικά της διεπαφής ενός πιλοτηρίου.

Η διεπαφή χρήστη-υπολογιστή σχεδιάζεται και υλοποιείται έτσι ώστε να επιτυγχάνεται η μέγιστη δυνατή χρηστικότητα. Ως χρηστικότητα ορίζουμε τον βαθμό στον οποίο η διεπαφή ενός συστήματος συμμορφώνεται με τους κανόνες και τις θεωρίες της ανθρώπινης ψυχολογίας και τη φυσιολογία των χρηστών για να γίνει η διαχείριση του συστήματος περισσότερο αποτελεσματική, αποδοτική και ικανοποιητική. Ο καλός σχεδιασμός της διεπαφής μειώνει την προσπάθεια που απαιτείται να καταβάλει ο χρήστης για να εισάγει δεδομένα στο σύστημα και να μεταφράσει τα εξαγόμενα αποτελέσματα. Επίσης σημαντικό είναι ο χρόνος που απαιτείται από τον χρήστη για την κατανόηση και την εξοικείωση με την λειτουργία του συστήματος, να είναι ο βέλτιστος κάτι που μία καλή σχεδίαση της διεπαφής μπορεί να επιτρέψει.

Στην Επιστήμη Υπολογιστών το σύστημα με το οποίο αλληλεπιδρά ο χρήστης είναι ένας ηλεκτρονικός υπολογιστής. Τα εργαλεία που παρέχει ο υπολογιστής στον χρήστη για την διαχείριση του, αποτελούν τα συστατικά της διεπαφής του υπολογιστή. Όπως και στο πιλοτήριο, έτσι και στον υπολογιστή, τα συστατικά της διεπαφής θα πρέπει να είναι αυτά που θα επιτρέπουν την μέγιστη δυνατή χρηστικότητα του συστήματος. Μία διεπαφή στον υπολογιστή θα πρέπει να σχεδιάζεται με γνώμονα την αποτελεσματικότητα, την αποδοτικότητα και την ικανοποίηση στο πλαίσιο χρήσης του υπολογιστή.

Στην καθημερινή τους ζωή οι άνθρωποι για την μεταξύ τους επικοινωνία χρησιμοποιούν διάφορα κανάλια επικοινωνίας που εκμεταλλεύονται όλες τους τις αισθήσεις. Έτσι οι άνθρωποι χρησιμοποιώντας την όραση, την ακοή και την αφή τους μπορούν να φέρουν σε πέρας όλα τα καθημερινά καθήκοντα επικοινωνίας και αλληλεπίδρασης. Στην σχεδίαση διεπαφών χρήστη-υπολογιστή η συνδυασμένη χρήση συμπληρωματικών μέσων επικοινωνίας έχει πολλά πλεονεκτήματα. Για παράδειγμα στην διεπαφή ενός κινητού τηλεφώνου ο ήχος που ειδοποιεί τον χρήστη για ένα εισερχόμενο μήνυμα αυξάνει την ευχρηστία και κάνει την αλληλεπίδραση με το σύστημα αποδοτικότερη.

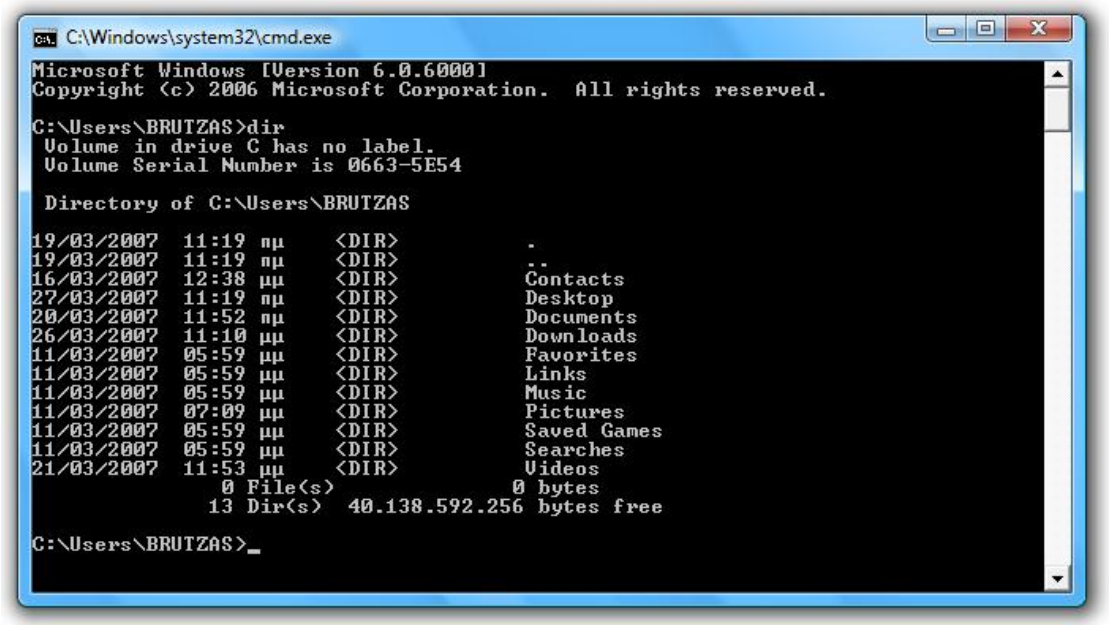
Παρακάτω θα εξετάσουμε τις γραφικές διεπαφές χρήστη υπολογιστή, τι ακριβώς είναι και ποια είναι τα δομικά συστατικά τους. Θα εξετάσουμε διάφορα οπτικά στυλ

αλληλεπίδρασης και τα διαδραστικά συστατικά των σύγχρονων διεπαφών όπως επίσης και διάφορα εργαλεία ανάπτυξης διεπαφών, γλώσσες προγραμματισμού και έτοιμες βιβλιοθήκες διαδραστικών αντικειμένων. Όσον αφορά τις βιβλιοθήκες θα εστιάσουμε περισσότερο στην βιβλιοθήκη Swing της JAVA. Επίσης θα σχολιάσουμε τα προβλήματα με τις υπάρχουσες βιβλιοθήκες στα οποία θα προσπαθήσουμε να προτείνουμε λύσεις στα επόμενα κεφάλαια.

2.1 Γραφικά στυλ αλληλεπίδρασης χρήστη-υπολογιστή

Ο τρόπος αλληλεπίδρασης μεταξύ ανθρώπων και υπολογιστών θεωρήθηκε εξαρχής ως μία οπτική διαδικασία και για αυτό το λόγο οι γραφικές διεπαφές χρήστη-υπολογιστή είναι το περισσότερο διαδεδομένο στυλ αλληλεπίδρασης στο χώρο της Επιστήμης Υπολογιστών. Πριν όμως εξετάσουμε τα συστατικά των γραφικών διεπαφών χρήσιμο θα ήταν να μελετήσουμε την έννοια του στυλ αλληλεπίδρασης για να κατανοήσουμε καλύτερα τα διαδραστικά φαινόμενα στα σύγχρονα συστήματα. Ένα στυλ αλληλεπίδρασης αποτελείται από τα δομικά στοιχεία μίας διεπαφής που συνδυάζονται για εκτελεστεί ένα διαδραστικό καθήκον. Θα εξετάσουμε τα στυλ αλληλεπίδρασης ως προς τον τρόπο που αντιλαμβάνεται ο χρήστης την κατάσταση και τις λειτουργίες του συστήματος.

Όπως εύκολα μπορούμε να καταλάβουμε τα στυλ αλληλεπίδρασης μπορούν να χωριστούν σε οπτικά και μη οπτικά. Όσον αφορά τα οπτικά στυλ αλληλεπίδρασης λίγο πολύ όλοι είμαστε εξοικειωμένοι με τις γραφικές διεπαφές των σύγχρονων υπολογιστών. Τα μη οπτικά στυλ αλληλεπίδρασης συνήθως χρησιμοποιούνται ως συμπληρωματικά των οπτικών. Πολλές φορές βέβαια υπάρχει η ανάγκη για μη οπτικά στυλ αλληλεπίδρασης και για αυτόν τον λόγο σχεδιάζονται μη οπτικά συστατικά διεπαφών τα οποία προσπαθούν να αντικαταστήσουν ή να συμπληρώσουν τα αντίστοιχα οπτικά.



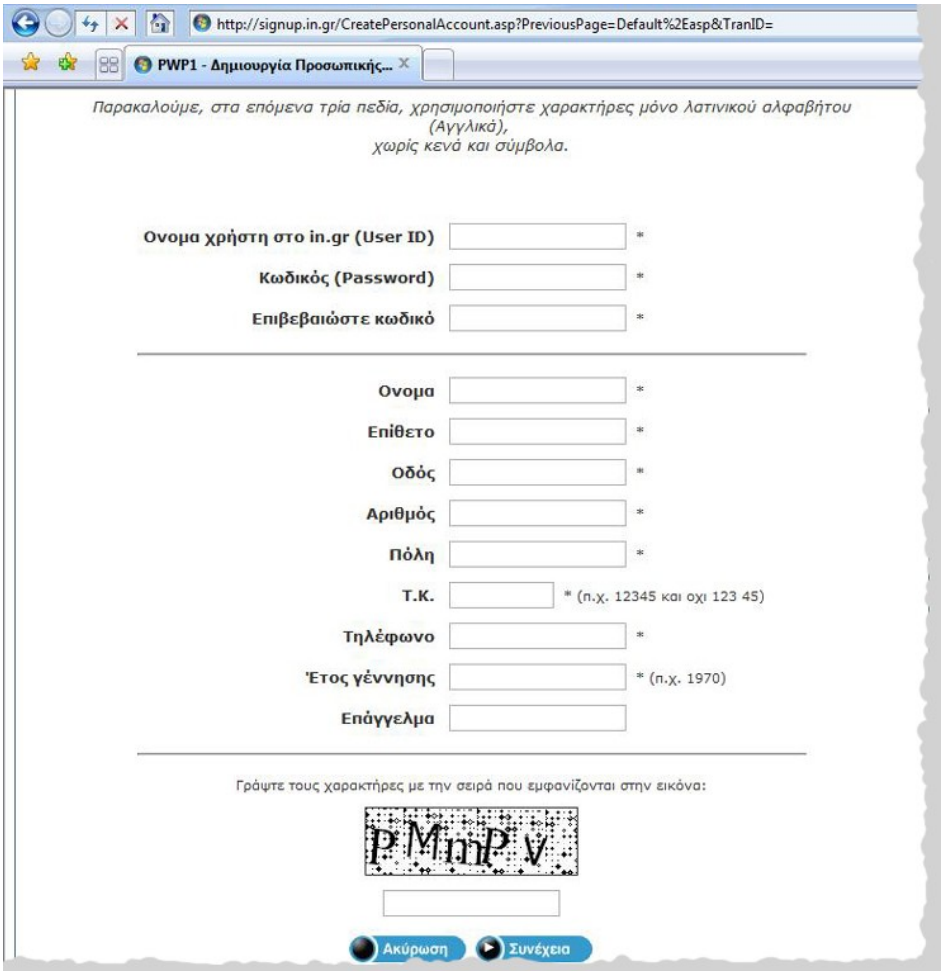
Εικόνα 1: Γραμμή εντολών (command line)

Ένα οπτικό στυλ αλληλεπίδρασης είναι η γραμμή εντολών (command line) η οποία υπήρξε από τα πρώτα και πιο συχνά χρησιμοποιούμενα στυλ αλληλεπίδρασης (βλέπε Εικόνα 1). Με την χρήση συγκεκριμένων εντολών και λέξεων-κλειδιών ο χρήστης μετέφερε εντολές προς το σύστημα και εισέπραττε την απόκριση του συστήματος. Οι χρήστες επίσης μπορούσαν να εισάγουν block εντολών για να εκτελέσουν σύνθετες

εργασίες. Το απλό στυλ αλληλεπίδρασης της γραμμής εντολών την έκανε κατάλληλη ακόμα και για συστήματα με χαμηλό bandwidth και συστήματα με περιορισμένη δυνατότητα γραφικών που ήταν και τα μόνα που υπήρχαν τότε. Η γραμμή εντολών παρείχε τέτοια ευελιξία που ακόμα και σήμερα χρησιμοποιείται σε πολλά συστήματα ως συμπληρωματικός τρόπος αλληλεπίδρασης.

Όμως παρά την ευελιξία που προσέφερε η γραμμή εντολών δεν ήταν κατάλληλη για μία μεγάλη ομάδα χρηστών, τους αρχαρίους χρηστές. Οι χρήστες έπρεπε να απομνημονεύσουν τις εντολές και η διαδικασία εκμάθησης δεν είναι ευχάριστη για όλους. Επίσης ακόμα και στους έμπειρους χρήστες παρατηρούνται λάθη κατά την πληκτρολόγηση των εντολών κάτι που κάνει την αλληλεπίδραση λιγότερη αποδοτική.

Ένα άλλο στυλ αλληλεπίδρασης που στόχευε αρχικά στους αρχάριους χρήστες ήταν η αλληλεπίδραση με συμπλήρωση πεδίων φόρμας (form fill-in). Σε αυτό το στυλ αλληλεπίδρασης (βλέπε Εικόνα 2) ο χρήστης συμπλήρωνε τα προκαθορισμένα πεδία με τα απαραίτητα δεδομένα. Σε αρχικές εκδόσεις του διαδραστικού στυλ όπου η μόνη συσκευή εισόδου ήταν το πληκτρολόγιο, το πλήκτρο `TAB` χρησίμευε για την εναλλαγή μεταξύ των πεδίων και πλήκτρο `ENTER` για την καταχώριση της φόρμας. Με αυτόν το τρόπο δεν ήταν αναγκαία η ύπαρξη μίας δεικτικής συσκευής όπως το ποντίκι.

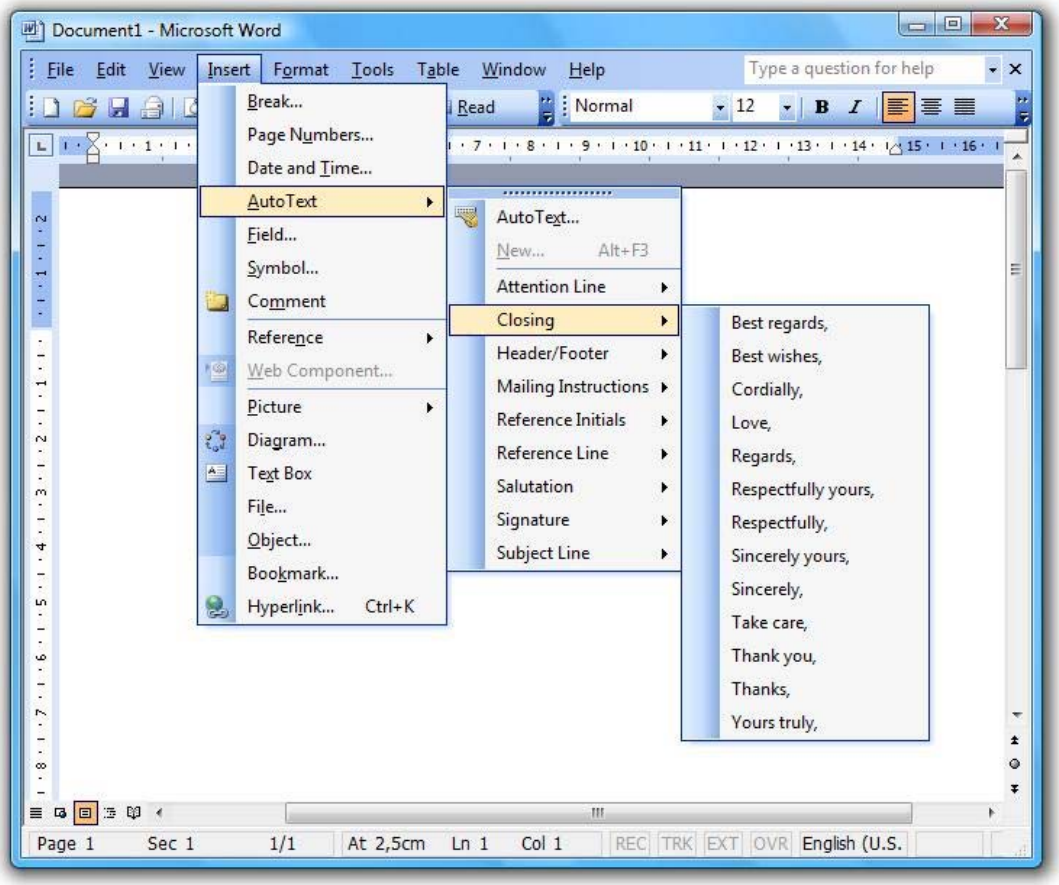


Εικόνα 2: Συμπλήρωση πεδίων φόρμας (form fill-in)

Οι φόρμες συμπλήρωσης πεδίων ήταν ιδανικές για εφαρμογές που απαιτούσαν καταχώριση μεγάλου όγκου δεδομένων αφού απλούστευαν την διαδικασία εισαγωγής δεδομένων. Επίσης μείωναν τον χρόνο εκμάθησης μιας και τα πεδία ήταν προκαθορισμένα άρα το μόνο που χρειαζόταν ήταν να αναγνωριστούν από τον χρηστή

και όχι να ανακληθούν από την μνήμη του. Βέβαια το μειονέκτημα των προκαθορισμένων πεδίων ήταν ότι περιόριζε την ευελιξία κάποιων εμπορικών εφαρμογών. Ακόμα όμως και σήμερα πολλές εφαρμογές όπως συστήματα πληρωμών, οικονομικά συστήματα, συστήματα υποστήριξης δανειστικών βιβλιοθηκών και πολλά άλλα, χρησιμοποιούν διεπαφές βασισμένες σε αυτό το στυλ αλληλεπίδρασης.

Τα μενού επιλογής (menu selection) είναι ένα ακόμη στυλ αλληλεπίδρασης που απευθύνεται σε αρχάριους χρήστες (βλέπε Εικόνα 3). Ένα μενού είναι μία συλλογή από εντολές στην οθόνη και η εκτέλεση μίας ή περισσότερων έχει ως αποτέλεσμα την αλλαγή της κατάστασης της διεπαφής (Paap and Roske-Hofstrand, 1989, as cited in Preece et al. 1994). Χρησιμοποιώντας ένα σύστημα με ένα τέτοιο στυλ αλληλεπίδρασης, ο χρήστης εκτελεί μία εντολή από το σύνολο των προκαθορισμένων εντολών που βρίσκονται ταξινομημένες σε μενού και παρατηρεί την απόκριση του συστήματος. Αν τα μενού και οι εντολές είναι κατανοητά διατυπωμένες και σωστά ταξινομημένες τότε οι χρήστες μπορούν να εκτελέσουν τα διάφορα καθήκοντα με ελάχιστη αποστήθιση των εντολών. Επίσης για την εξοικονόμηση χώρου στην οθόνη τα περιεχόμενα των μενού εμφανίζονται σε drop-down ή pop-up μενού.



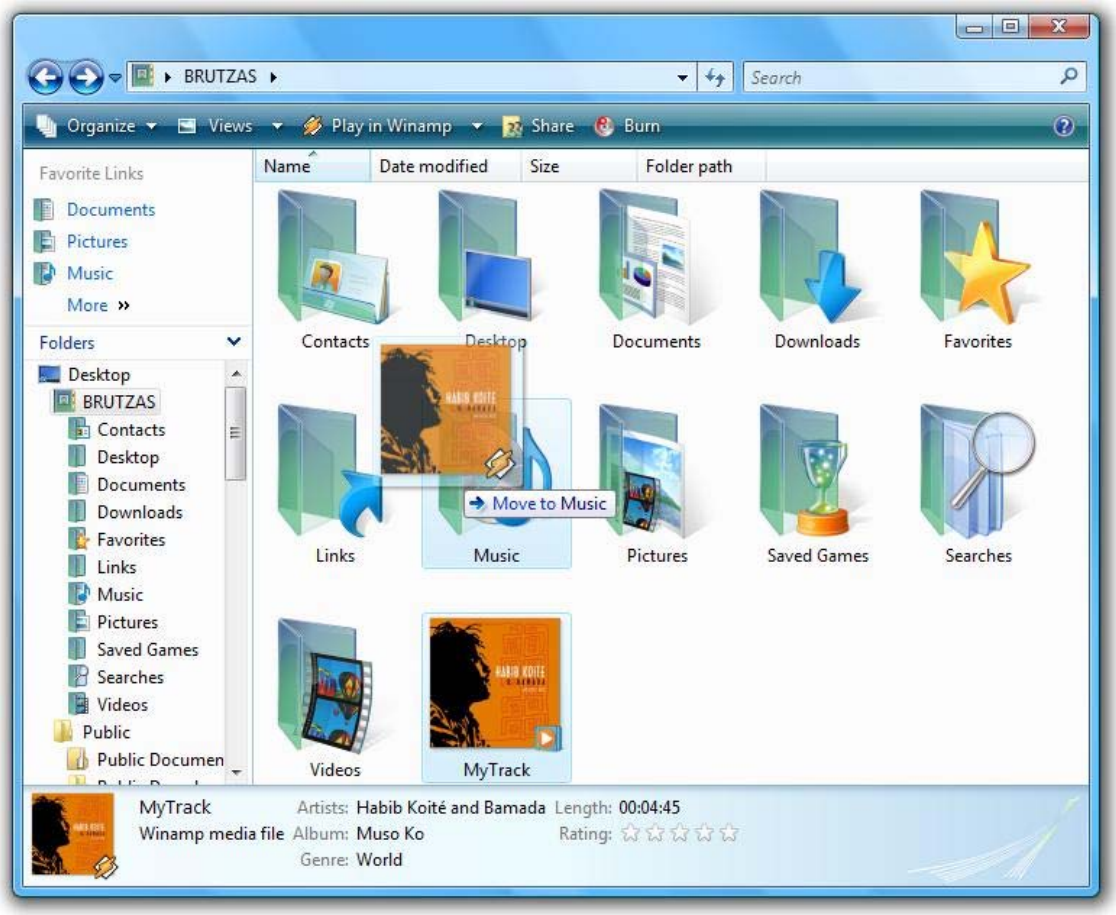
Εικόνα 3: Μενού επιλογής (menu selection)

Τα μενού επιλογής είναι ιδανικά για αρχάριους ή όχι τόσο συχνούς χρήστες της εφαρμογής και είναι ελκυστικά ακόμα και στους προχωρημένους χρήστες ιδιαίτερα όταν ενσωματώνουν και τις κατάλληλες συντομεύσεις. Επίσης επιτρέπουν την περιήγηση μέσα στα μενού για την επιλογή της κατάλληλης εντολής. Βέβαια τα πολλά μενού μπορεί να οδηγήσουν σε αύξηση της πολυπλοκότητας και του χρόνου που απαιτείται για την αλληλεπίδραση με το σύστημα κάτι το οποίο έχει σαν αποτέλεσμα την αποθάρρυνση των χρηστών και ειδικά των συχνών χρηστών του συστήματος. Επίσης τα μενού επιλογής

πολλές φορές είναι ακατάλληλα για συσκευές με μικρή οθόνη ή με υποτυπώδεις ικανότητες απεικόνισης γραφικών, όπως κινητά τηλέφωνα ή υπολογιστές παλάμης (palmtop).

Ένα άλλο πολύ σημαντικό στυλ αλληλεπίδρασης είναι η άμεση διαχείριση (direct manipulation). Ο όρος direct manipulation εισήχθη από τον Ben Shneiderman και σήμαινε ότι το αντικείμενο του ενδιαφέροντος παρουσιάζετε ως ένα διακριτό αντικείμενο στην διεπαφή του χρηστή (βλέπε Εικόνα 4) και η διαχείριση του πραγματοποιείται με έναν άμεσο τρόπο (Shneiderman 1983: p.57). Τα πλεονεκτήματα τέτοιων συστημάτων είναι η ικανότητα απεικόνισης του αντικειμένου, οι άμεσες, αναστρέψιμες και σταδιακές ενέργειες και η αντικατάσταση των σύνθετων εντολών με φυσικές ενέργειες πάνω στα εικονικά αντικείμενα, χαρακτηριστικά τα οποία τα καθιστούν εύκολα στην μάθηση και ανθεκτικά στα λάθη.

Έτσι, για την αλλαγή θέσης ενός εγγράφου ή για την διαγραφή του αρκεί η μετακίνηση του (drag and drop) σε έναν άλλον φάκελο ή στον κάδο αχρήστων. Αυτά τα στυλ αλληλεπίδρασης είναι πιο ελκυστικά στον χρήστη (αρχάριο ή μη) γιατί προσδίδουν μία αμεσότητα και μία φυσικότητα στην όλη αλληλεπίδραση του με τον ηλεκτρονικό υπολογιστή. Αυτά τα χαρακτηριστικά καθιστούν το συγκεκριμένο στυλ αλληλεπίδρασης από τα πιο δημοφιλή αλλά και το πιο προσιτό στον αρχάριο χρήστη που πλέον δε αποθαρρύνετε από την όποια δυσκολία χρήσης του υπολογιστή. Τέτοια στυλ αλληλεπίδρασης όμως, είναι δύσκολο να εφαρμοστούν σε συστήματα με λιγοστές γραφικές δυνατότητες, γιατί απαιτούν ισχυρή μηχανή γραφικών για την παρουσίαση των σύνθετων αναπαραστάσεων στην οθόνη.



Εικόνα 4: Άμεση διαχείριση (direct manipulation)

2.2 Συστατικά σύγχρονων διεπαφών

Σήμερα η ανάπτυξη των διεπαφών γίνεται εύκολα και γρήγορα στηριζόμενη σε ορισμένες βασικές τεχνολογίες και εργαλεία χαμηλού επιπέδου. Τέτοια εργαλεία, τα οποία σήμερα θεωρούνται δεδομένα, συνεισφέρουν στον σχεδιασμό των σύγχρονων γραφικών διεπαφών και αποτελούν τα βασικά συστατικά τους.

Η εξέλιξη των γραφικών διεπαφών έγινε παράλληλα με την εισαγωγή του παραθυρικού περιβάλλοντος χρήσης. Σε έναν υπολογιστή, οι συσκευές παρουσίασης και εισαγωγής της πληροφορίας είναι πεπερασμένου αριθμού (συνήθως μία οθόνη, ένα πληκτρολόγιο και ένα ποντίκι) και περιορισμένων διαστάσεων (η οθόνη έχει συγκεκριμένο μέγεθος και ανάλυση). Έτσι λοιπόν δημιουργείται η ανάγκη για διαχείριση των πόρων αυτών από τις διάφορες εφαρμογές. Η μέθοδος που επικράτησε ήταν αυτή της χρήσης διαφορετικών παραθύρων για κάθε εφαρμογή. Κάθε εφαρμογή ‘έτρεχε’ σε μία εικονική τερματική οθόνη (virtual display terminal) η οποία είχε το δικό της πρωτόκολλο χρήσης και διαχείρισης των γεγονότων που συνέβαιναν.

Ένα σύγχρονο παραθυρικό περιβάλλον εκτελεί δύο βασικές λειτουργίες. Πρώτον, την ανάθεση και διαχείριση των περιοχών της οθόνης σε ενεργές εφαρμογές εξασφαλίζοντας την ομαλή λειτουργία κάθε μίας ξεχωριστά καθώς και το συντονισμό των συσκευών εισόδου και δεύτερον, την διαχείριση των πόρων των συσκευών εξόδου έτσι ώστε διερμηνεύοντας τα διάφορα γεγονότα να εκτελούνται λειτουργίες όπως κλείσιμο παραθύρου, αλλαγή διαστάσεων παραθύρου, μετακίνηση παραθύρου κ.α. Με αυτόν τον τρόπο ένα παραθυρικό περιβάλλον μπορούσε μέσα σε πολλαπλά παράθυρα να παρουσιάσει και να επεξεργαστεί δεδομένα πολλών διαφορετικών εφαρμογών.

Ένα ζήτημα που προέκυψε με τα παραθυρικά περιβάλλοντα χρήσης ήταν η διαχείριση συμβάντων, δηλαδή ποια λειτουργικά καθήκοντα θα εκτελεστούν και υπό ποιες συνθήκες. Το κλασικό μοντέλο ανάπτυξης λογισμικού απαιτούσε ο έλεγχος να είναι ενσωματωμένος στο λειτουργικό μέρος της εφαρμογής. Η εναλλακτική προσέγγιση που παρουσιάστηκε ήταν η απομάκρυνση του ελέγχου ροής από την εφαρμογή και άντ’ αυτού η εφαρμογή να αντιδρά σε δεδομένα που προέρχονταν από μία συνιστώσα ειδοποιήσεων, η οποία επικοινωνεί με την εφαρμογή μέσω συμβάντων.

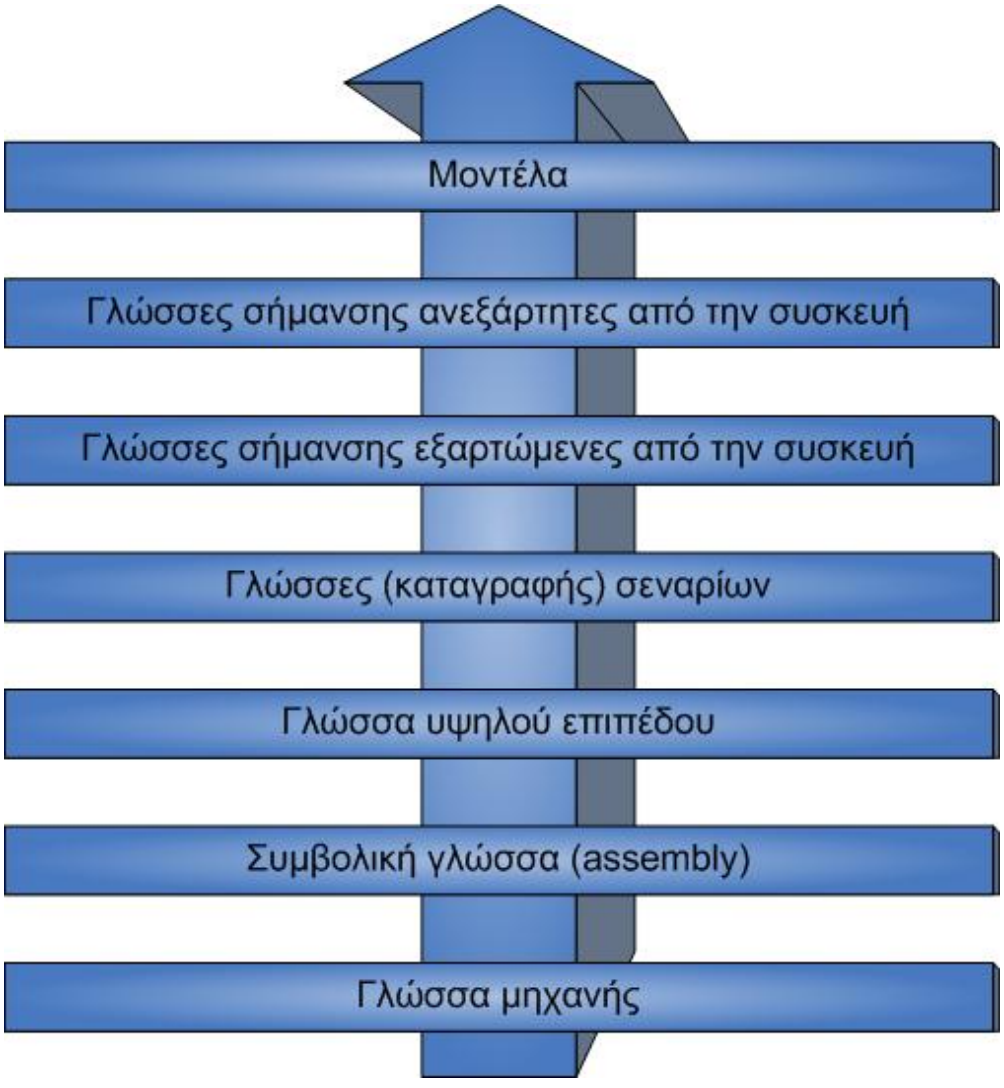
Η προσέγγιση αυτή σήμερα τυγχάνει καθολικής αποδοχής και οδήγησε στη σχεδίαση διαδραστικών συστημάτων λογισμικού που καθοδηγούνται από συμβάντα (event-driven software). Τα συστήματα αυτά σχεδιάζονται με τέτοιο τρόπο ώστε ο έλεγχος να βρίσκεται στο χρηστή με την έννοια ότι το λογισμικό αντιδρά σε χειρισμούς του χρήστη. Οι χειρισμοί αυτοί μεταφέρονται στο λειτουργικό τμήμα του συστήματος υπό την μορφή συμβάντων και ανάλογα εκτελείται το αντίστοιχο λειτουργικό καθήκον. Υπάρχουν δύο κατηγόριες συμβάντων, τα συμβάντα τα οποία αντιπροσωπεύουν χειρισμούς του χρήστη σε διαδραστικά αντικείμενα (πάτημα πλήκτρου ποντικιού, πληκτρολόγηση, κινήσεις ποντικιού) και αυτά που αντιπροσωπεύουν λειτουργίες του διακομιστή όπως η αλλαγή του καθεστώτος ενός αντικειμένου (η επανεμφάνιση ενός παραθύρου).

Η παραγωγή συμβάντων από τους χειρισμούς του χρήστη καθιέρωσε την ιδέα του διαδραστικού αντικειμένου ως την ίδια την πηγή των συμβάντων. Τα συμβάντα αφού μεταφερθούν στην εφαρμογή προσδιορίζουν την ροή ελέγχου εκτέλεσης των λειτουργικών καθηκόντων της εφαρμογής. Άρα λοιπόν ένα διαδραστικό αντικείμενο είναι το αντικείμενο το οποίο χειρίζεται ο χρήστης και οι χειρισμοί αυτοί γεννούν διάφορα γεγονότα / συμβάντα. Τα αντικείμενα αυτά έχουν διάφορα χαρακτηριστικά γνωρίσματα (μεταβλητές) και διάφορες μεθόδους. Μία συλλογή τέτοιων αντικειμένων ονομάζεται βιβλιοθήκη διαδραστικών αντικειμένων (user interface library ή toolkit) και

ένας προγραμματιστής μπορεί να χρησιμοποιήσει μία τέτοια έτοιμη βιβλιοθήκη για να δημιουργήσει γρήγορα και εύκολα μία διαδραστική εφαρμογή.

2.3 Εργαλεία ανάπτυξης διεπαφών

Μετά από την περιγραφή των βασικών συστατικών των σύγχρονων διεπαφών θα προσπαθήσουμε να αναλύσουμε διάφορα εργαλεία και προγραμματιστικές μεθόδους για την ανάπτυξη διεπαφών. Την δεκαετία του 1940, κατά την οποία ο πρώτος ηλεκτρονικός ψηφιακός υπολογιστής εμφανίστηκε, ο προγραμματισμός των εφαρμογών γινόταν με γλώσσα μηχανής (machine language programming). Πολύ σύντομα η γλώσσα μηχανής ξεπεράστηκε και την θέση της πήρε η συμβολική γλώσσα (assembly). Αργότερα εμφανίστηκαν γλώσσες υψηλού επιπέδου οι οποίες επηρέαζαν και τον τρόπο προγραμματισμού των διεπαφών ανάλογα με τους κατά καιρούς τεχνολογικούς στόχους που υποστήριζαν όπως γρήγορη ανάπτυξη, ανεξαρτησία πλατφόρμας, αυτοματοποίηση παραγωγής κώδικα κ.α. Στην Εικόνα 5 βλέπουμε την πορεία που ακολούθησαν οι γλώσσες προγραμματισμού.



Εικόνα 5: Γενιές γλωσσών προγραμματισμού συστημάτων λογισμικού

2.3.1 Γλώσσες προγραμματισμού

Ο προγραμματισμός χαμηλού επιπέδου ήταν η πρώτη μέθοδος προγραμματισμού που χρησιμοποίησαν οι προγραμματιστές για την υλοποίηση των διεπαφών. Ο περισσότερος κώδικας ήταν γραμμένος σε γλώσσα assembly και χρησιμοποιώντας εντολές μηχανής από τον ελεγκτή οθόνης ο προγραμματιστής πετύχαινε τον χειρισμό των εικονοστοιχείων της οθόνης. Τα προγράμματα σε assembly εκτελούνταν ταχύτατα και δεν απαιτούσαν μεταγλώττιση αλλά ο ίδιος κώδικας δε μπορούσε να μεταφερθεί σε άλλη πλατφόρμα αφού κάθε πλατφόρμα είχε το δικό της σύνολο εντολών μηχανής.

Σήμερα ο πιο δημοφιλής τρόπος προγραμματισμού διεπαφών είναι με την χρήση γλωσσών υψηλού επιπέδου. Τα πλεονεκτήματα που προσφέρουν οι γλώσσες υψηλού επιπέδου σε σχέση με άλλες είναι ο υψηλός βαθμός ελέγχου στον προγραμματιστή και οι αυξημένες δυνατότητες υλοποίησης πρότυπης και εξειδικευμένης διαδραστικής συμπεριφοράς. Βέβαια, από την άλλη, απαιτεί σημαντική εμπειρία και γνώση των ιδιαίτερων γνωρισμάτων της πλατφόρμας από τον προγραμματιστή. Από της πιο διαδεδομένες γλώσσες υψηλού επιπέδου που χρησιμοποιούνται σήμερα είναι οι C/C++, Java και Visual Basic. Χρησιμοποιώντας σήμερα τέτοιες γλώσσες, με την κατάλληλη επιλογή συμβατών βιβλιοθηκών διαδραστικών αντικειμένων, η ανάπτυξη γραφικών διεπαφών γίνεται αποδοτικότερη.

2.3.2 Βιβλιοθήκες γραφικών διαδραστικών αντικειμένων

Η πιο διαδεδομένη μέθοδος σήμερα για την παραγωγή γραφικών διεπαφών είναι με την χρήση μίας βιβλιοθήκης διαδραστικών αντικειμένων. Οι βιβλιοθήκες αυτές (ονομάζονται και widget toolkits ή GUI toolkits ή απλά toolkits) περιέχουν ένα σύνολο από έτοιμα, ολοκληρωμένα διαδραστικά συστατικά, τα οποία ένας προγραμματιστής μπορεί να χρησιμοποιήσει στην ανάπτυξη μίας διεπαφής. Τα διαδραστικά αντικείμενα μίας διεπαφής είναι υλοποιημένα σε χαμηλό επίπεδο προγραμματισμού και με τέτοιο τρόπο ώστε να είναι εύκολα στην χρήση από τον προγραμματιστή.

Με την χρήση έτοιμων βιβλιοθηκών διαδραστικών αντικειμένων αποφεύγεται ο χαμηλού επιπέδου προγραμματισμός όσον αφορά, είτε την δημιουργία και παρουσίαση των αντικειμένων στην οθόνη, είτε τον χειρισμό των συμβάντων χαμηλού επιπέδου των συσκευών εισόδου. Το μόνο που χρειάζεται να κάνει ο προγραμματιστής για να δημιουργήσει μία διεπαφή, είναι απλά να καλέσει τα συστατικά της επιλογής του και να τα τοποθετήσει κατάλληλα. Έτσι, γίνονται εύκολα και γρήγορα αλλαγές και διορθώσεις σε μία διεπαφή και επίσης επιτρέπετε η ταχύτερη ανάπτυξη πρωτοτύπων διεπαφής. Ο χρόνος ανάπτυξης της διεπαφής περιορίζεται σημαντικά και έτσι ο προγραμματιστής αφιερώνει περισσότερο χρόνο σε άλλα στάδια της ανάπτυξης.

Οι πιο διαδεδομένες βιβλιοθήκες είναι η Microsoft Foundation Classes (MFC) για την πλατφόρμα των Windows, Motif για την πλατφόρμα X-Windows και η Java Foundation Classes (JFC) της Sun Microsystems. Η σύνθεση της διεπαφής γίνεται αξιοποιώντας τα διαδραστικά αντικείμενα της βιβλιοθήκης και με την χρήση μίας κατάλληλης γλώσσας υψηλού επιπέδου υλοποιούνται οι απαραίτητες λειτουργίες. Μία διεπαφή που έχει δημιουργηθεί με αυτόν τον τρόπο μπορεί να μεταφερθεί σε οποιαδήποτε πλατφόρμα που είναι συμβατή με την βιβλιοθήκη, αρκεί βέβαια να μεταγλωττιστεί ξανά για αυτή την πλατφόρμα.

Οι βιβλιοθήκες διαδραστικών αντικειμένων χρησιμοποιούνται σήμερα σε μεγάλο βαθμό για την εύκολη και γρήγορη υλοποίηση εφαρμογών. Μία βιβλιοθήκη αξιολογείται από την ποικιλία και την ποιότητα των διαδραστικών συστατικών που προσφέρει, από

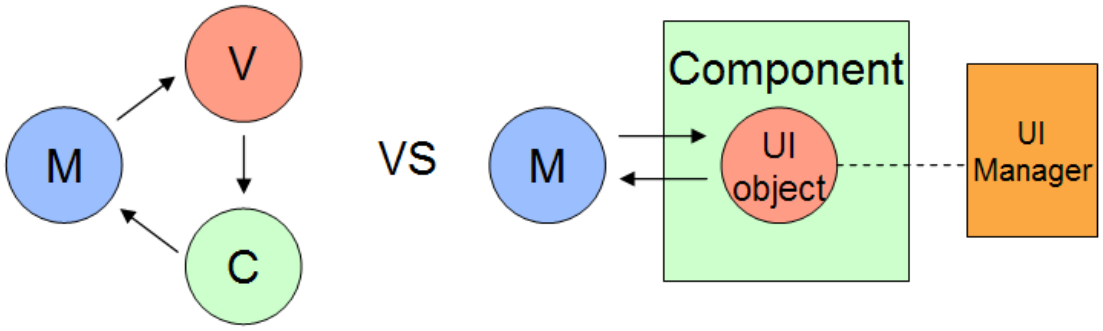
την σταθερότητα της, την καλή αρχιτεκτονική, από τις επιπλέον ευκολίες που προσφέρει (αν υποστηρίζουν τα συστατικά της διεθνοποίηση, ανεξαρτησία πλατφόρμας κ.α.) και από πολλές άλλες παραμέτρους. Ένα υποσύνολο της βιβλιοθήκης JFC και το οποίο θα αναλυθεί παρακάτω είναι το Swing. Παρακάτω θα εστιάσουμε στο Swing και στην αρχιτεκτονική του.

Το Swing

Το Swing είναι μία γραφική βιβλιοθήκη διαδραστικών αντικειμένων η οποία αποτελεί μέρος του Java Foundation Classes (JFC) και περιέχει γραφικά διαδραστικά συστατικά όπως Buttons, Panels, Split-Panes, Trees και πολλά άλλα. Το Swing προσφέρει πιο εξελιγμένα γραφικά συστατικά από την προγενέστερη βιβλιοθήκη Abstract Window Toolkit (AWT) και σε αντίθεση με το AWT μπορεί να 'τρέξει' σε οποιαδήποτε πλατφόρμα (platform independent). Επίσης υποστηρίζει διάφορα look and feels, όχι όμως χρησιμοποιώντας τους πόρους της πλατφόρμας αλλά προσπαθώντας να τους μιμηθεί.

Αρχικά το Swing προσπάθησε να ακολουθήσει την αρχιτεκτονική model-view-controller (MVC). Η αρχιτεκτονική MVC ορίζει ότι μία γραφική εφαρμογή πρέπει να 'σπάσει' σε τρία διακριτά κομμάτια. Ένα μοντέλο (model) το οποίο παριστά τα δεδομένα, μία όψη (view) η οποία είναι η γραφική παρουσίαση των δεδομένων και έναν ελεγκτή (controller) ο οποίος ερμηνεύοντας τις εντολές του χρήστη πάνω στο view τροποποιεί αναλόγως το model. Αργότερα διαπιστώθηκε ότι αυτός ο διαχωρισμός δε δούλευε σωστά στην πράξη γιατί το view και ο controller του αντικειμένου προϋποθέτουν μία στενή σχέση. Για παράδειγμα δεν μπορούσε να γραφτεί ένας γενικός controller που να μην γνώριζε τις ιδιαιτερότητες του view. Έτσι οι δύο αυτές οντότητες (ο controller και το view) συμπτύχθηκαν σε ένα μοναδικό UI (user-interface) object (βλέπε Εικόνα 6). Πολλές φορές το UI object ονομάζεται και UI delegate object ή UI delegate.

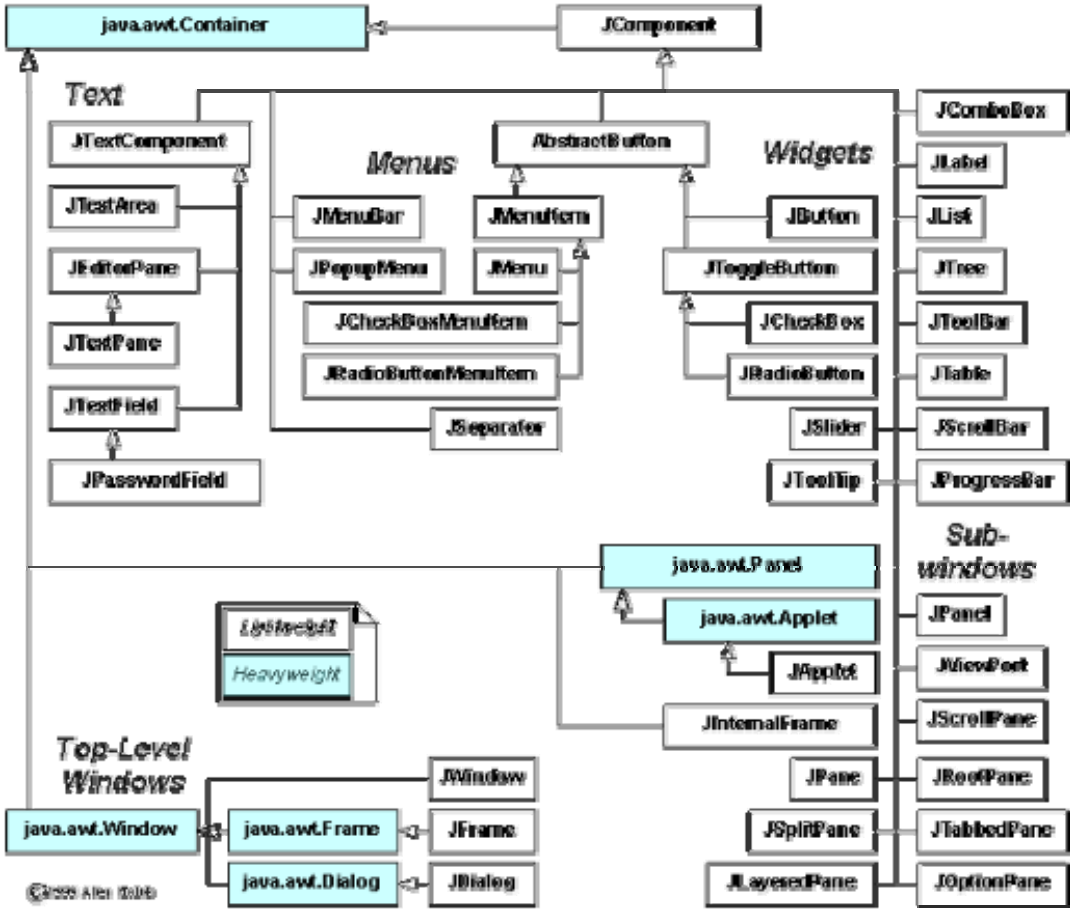
Όπως φαίνεται και στην Εικόνα 6, η αρχιτεκτονική του Swing γενικά (όχι απόλυτα) βασίζεται στην αρχιτεκτονική MVC. Πολλές φορές αυτή η εν μέρη MVC αρχιτεκτονική καλείται και separable model architecture. Έτσι λοιπόν η αρχιτεκτονική του Swing θεωρεί το model ως ένα ξεχωριστό στοιχείο αλλά 'βλέπει' ως μία οντότητα το controller και το view και καλεί την οντότητα αυτή UI delegate. Η ιεραρχία κλάσεων του Swing (swing component hierarchy) φαίνεται στην Εικόνα 7. Χαρακτηριστικά του Swing τα οποία είναι αξία να αναφερθούν είναι:



Εικόνα 6: MVC vs. UI delegate

- **Ανεξαρτησία πλατφόρμας**: Το Swing είναι ανεξάρτητο πλατφόρμας, και όσο αφορά την έκφανση του (JAVA) αλλά και όσο αφορά την υλοποίηση του (δε χρησιμοποιεί συστατικά της πλατφόρμας).
- **Επεκτασιμότητα**: Το Swing είναι δομημένο με τέτοιον τρόπο ώστε να επιτρέπει την διασύνδεση διάφορων υλοποιήσεων. Έτσι οι χρήστες μπορούν να παρακάμψουν της αρχικές υλοποιήσεις και να δημιουργήσουν δικές τους.
- **Προσαρμοστικότητα**: Η οπτική αναπαράσταση ενός συστατικού του Swing είναι μία σύνθεση από ένα καθορισμένο σετ στοιχείων όπως 'borders', 'insets', διακοσμητικά και άλλα. Έτσι οι χρήστες μπορούν προγραμματιστικά να αλλάξουν ένα συστατικό του Swing θέτοντας διαφορετικά 'borders', 'backgrounds' και άλλα, ως παραμέτρους του συστατικού.
- **Διαμορφώσιμο**: Η συσχέτιση του Swing με runtime μηχανισμούς και με έμμεσους τρόπους δημιουργίας, επιτρέπει θεμελιώδεις αλλαγές στις ρυθμίσεις του.
- **Lightweight UI**: Αυτή η προσαρμοστικότητα του Swing οφείλεται στο γεγονός ότι δεν χρησιμοποιεί το GUI του λειτουργικού που το φιλοξενεί αλλά 'ζωγραφίζει' τα συστατικά του με την χρήση του JAVA 2D API.

Το Swing είναι πολύ περισσότερο από μία απλή βιβλιοθήκη διαδραστικών αντικειμένων. Υποστηρίζει λειτουργίες αναιρέσεις (undo), ενσωματώνει διεθνοποίηση (internationalization), υποστηρίζει προσβασιμότητα (accessibility) και επίσης δε θα ήταν πλήρης βιβλιοθήκη αν δεν υποστήριζε βασικές λειτουργίες του user interface όπως drag and drop, διαχείριση συμβάντων (event handling) και window management.



Εικόνα 7: Ιεραρχία κλάσεων του Swing

2.3.3 Γλώσσες σήμανσης (Markup languages)

Ο όρος 'markup' προήλθε από τον παραδοσιακό τρόπο εκτύπωσης εντύπων, όπου στο περιθώριο του εγγράφου πρόσθεταν με μία συμβολική γλώσσα, διάφορες οδηγίες για τον εκτυπωτή. Αυτές οι οδηγίες περιείχαν πληροφορίες για το πώς θα πρέπει να εκτυπωθούν διάφορες περιοχές του κειμένου. Από αυτόν τον τρόπο σήμανσης ενός έγγραφου προήλθαν αργότερα οι γλώσσες σήμανσης. Έτσι λοιπόν μία γλώσσα σήμανσης εγγράφων περιέχει απλό κείμενο και περαιτέρω πληροφορίες για το κείμενο αυτό. Αυτές οι πληροφορίες αφορούσαν αρχικά την παρουσίαση του εγγράφου σε μία συσκευή ενώ σήμερα μπορούν να χαρακτηρίζουν και τη δομή του εγγράφου. Η πιο γνωστή γλώσσα σήμανσης είναι η HTML (Hyper Text Markup Language) και υπήρξε θεμελιώδες εργαλείο στην ανάπτυξη του Παγκοσμίου Ιστού (World Wide Web).

Η γλώσσες σήμανσης προσφέρουν ένα υψηλότερο επίπεδο αφαίρεσης από ότι οι βιβλιοθήκες διαδραστικών αντικειμένων. Πληροφορίες οι οποίες αφορούν το περιεχόμενο και κάποια ένδειξη για το τι ρόλο παίζει αυτό συμπεριλαμβάνονται σε ένα μόνο αρχείο κώδικα γραμμένο σε μία γλώσσα σήμανσης. Για παράδειγμα, σε ένα έγγραφο το κείμενο που αποτελεί τον τίτλο του εγγράφου, έχει διαφορετική σημασία και πρέπει να διαχειριστεί διαφορετικά από ότι το κύριο κείμενο του εγγράφου.

Πέρα όμως από την χρήση των γλωσσών αυτών στο Παγκόσμιο Ιστό χρησιμοποιήθηκαν και για την περιγραφή διεπαφών. Μία διεπαφή μπορεί να περιγραφεί με την καταγραφή των συστατικών που την αποτελούν με μία γλώσσα σήμανσης. Για παράδειγμα, ένα διαδραστικό συστατικό μπορεί να περιγραφεί αναφέροντας τον τύπο του, την πληροφορία που περιέχει, την λειτουργία που ακολουθεί η εκτέλεση του κ.α. Καταγράφοντας όλα τα συστατικά μίας διεπαφής και επισημαίνοντας την δομή τους μία γραφική διεπαφή μπορεί να περιγραφεί σχετικά απλά χρησιμοποιώντας μία κατάλληλη γλώσσα σήμανσης. Το συγκριτικό πλεονέκτημα είναι ότι προσφέρεται μεγάλο επίπεδο φορητότητας (portability), γεγονός που διευκολύνει την ανάπτυξη διεπαφών ανεξαρτήτως πλατφόρμας. Επίσης ένα άλλο πλεονέκτημα των γλωσσών αυτών είναι ότι δεν απαιτούνται εξειδικευμένες γλώσσες προγραμματισμού για την κατασκευή διεπαφών.

Τέτοιες γλώσσες βέβαια είναι εξαρτημένες από συγκεκριμένες συσκευές. Δεν μπορούν όλες οι συσκευές να μεταφράσουν σωστά την σήμανση του κειμένου και σε αυτές τις περιπτώσεις πληροφορίες που αφορούν την συσκευή αποθηκεύονται ξεχωριστά σε ένα άλλο αρχείο το οποίο ονομάζεται φύλλο στυλ (style sheets). Κάθε τέτοιο φύλλο προσδιορίζει την αντιστοίχιση μεταξύ των ετικετών (της σήμανσης) της γλώσσας με τους τύπους που μπορεί να υποστηρίξει μία συσκευή. Διάφορα πρότυπα περιγραφής πληροφοριών που αφορούν την συσκευή είναι τα XSL, CSS και DSSSL.

Μία γλώσσα σήμανσης προκείμενου να περιγράψει διαφόρων ειδών δεδομένων ή διεπαφών θα πρέπει να διαθέτει το κατάλληλο σύνολο εντολών. Η eXtensible Markup Language (XML) είναι μία γλώσσα σήμανσης γενικού σκοπού. Ο αρχικός της ρόλος ήταν να διευκολύνει την μεταφορά δεδομένων στο διαδίκτυο μεταξύ διαφορετικών πληροφοριακών συστημάτων. Η XML δεν διαθέτει προκαθορισμένες ετικέτες όπως η HTML αλλά ο χρηστής της θα πρέπει κάθε φορά να καθορίζει το σύνολο των ετικετών που θέλει, και αυτό το χαρακτηριστικό της την κάνει ιδανική για την περιγραφή διεπαφών. Η ελευθέρια που προσφέρει η XML, έχει επιτρέψει την δημιουργία πολλών εξειδικευμένων γλωσσών σήμανσης οι οποίες βασίζονται σε αυτήν για την προδιαγραφή διεπαφών. Κάποιες γλώσσες από αυτές είναι:

1. Alternative Abstract Interface Markup Language (AAIML)
2. Abstract User Interface Markup Language (AUIML)

3. Extensible Interface Markup Language (XIML)
4. Extensible User Markup Language (XUL)
5. Microsoft Extensible Application Markup Language (XAML)
6. OASIS User Interface Markup Language (UIML)
7. User Interface Extensible Markup Language (UsiXML)
8. User Interface Markup Language (UIML)
9. W3C Device Independence Working Group (DIWG)
10. W3C XForms Working Group (XForms)

Θα πρέπει να τονιστεί ότι η XML δεν είναι ένα πλήρες προγραμματιστικό εργαλείο αλλά μία περιγραφική γλώσσα. Υπολείπεται ορισμένων σημαντικών προγραμματιστικών ιδιοτήτων όπως συνθήκες επανάληψης, επιλογής και άλλα. Την συμπληρωματική λειτουργικότητα που χρειάζονται οι γλώσσες σήμανσης έρχονται να την προσθέσουν οι γλώσσες σεναρίων (scripting languages). Κάποιες από τις πιο γνωστές γλώσσες σεναρίων είναι οι JavaScript, Perl και Active Server Pages(ASP).

Μία ιδιαίτερα σημαντική εξέλιξη στις γλώσσες σήμανσης είναι η δημιουργία γλωσσών που είναι ανεξάρτητες από την συσκευή ή την πλατφόρμα παρουσίασης. Υπάρχουν διάφορες προσεγγίσεις σε αυτήν την κατεύθυνση και έχουν προταθεί αρκετές εξειδικευμένες διάλεκτοι με στόχο τον καθορισμό προδιαγραφών για την περιγραφή διεπαφών. Μία τέτοια γλώσσα είναι η User Interface Markup Language (UIML), που είναι βασισμένη στην XML, η οποία αναπτύχθηκε για να εξυπηρετήσει την ανάπτυξη διεπαφών πολλαπλής πλατφόρμας για ποικιλία συσκευών πρόσβασης. Για την ανάπτυξη μίας διεπαφής ο προγραμματιστής συντάσσει ένα έγγραφο UIML το οποίο περιλαμβάνει δομές παρουσίασης κατάλληλες για την τις συσκευές στις οποίες η διεπαφή θα γίνει διαθέσιμη. Το UIML έγγραφο μεταφράζεται αυτόματα στην γλώσσα που χρησιμοποιείται από την συγκεκριμένη συσκευή.

Θα πρέπει να σημειωθεί ότι η προσέγγιση των -ανεξάρτητων από την συσκευή- γλωσσών σύνταξης δεν συνιστούν απλά μία νέα κατηγορία εργαλείων για την δημιουργία διεπαφών αλλά εισάγουν ένα νέο μοντέλο ανάπτυξης για τις σύγχρονες διεπαφές όπου η διεπαφή και το λειτουργικό της κομμάτι προγραμματίζεται μία φορά και τρέχει σε όλες τις διαφορετικές συσκευές και πλατφόρμες.

2.4 Προβλήματα με σύγχρονες βιβλιοθήκες

Πάρα τα σημαντικά πλεονεκτήματα που παρέχουν οι σύγχρονες βιβλιοθήκες διαδραστικών αντικειμένων έχουν και αρκετούς περιορισμούς. Ο αριθμός των αντικειμένων που περιέχουν είναι πεπερασμένος και πολλές φορές υπάρχει η ανάγκη για μία μορφή αλληλεπίδρασης η οποία δεν υποστηρίζεται από τα έτοιμα διαδραστικά συστατικά της βιβλιοθήκης. Μία εφαρμογή που προσπαθεί να προσομοιώσει ένα αντικείμενο του φυσικού κόσμου, και του οποίου η λειτουργικότητα δεν προσφέρετε από κάποιο έτοιμο συστατικό της βιβλιοθήκης, δε μπορεί να υποστηριχθεί, τουλάχιστον όχι εύκολα. Ας πάρουμε για παράδειγμα την μεταφορά του δωματίου (room metaphor). Τα κατάλληλα δομικά συστατικά θα πρέπει να δημιουργηθούν που να προσομοιώνουν και να υλοποιούν την συμπεριφορά των φυσικών αντικειμένων ενός δωματίου. Διαδραστικά αντικείμενα όπως ‘πόρτα’ ή ‘πίνακας ανακοινώσεων’ με λειτουργίες όπως `ΑΝΟΙΞΕ`,

ΚΛΕΙΣΕ ή ΑΝΑΡΤΗΣΕ πρέπει να κατασκευαστούν ώστε να καταστεί δυνατόν η δημιουργία της διεπαφής.

Σε πολλές περιπτώσεις οι προγραμματιστές για να καλύψουν τις ανάγκες τους σε εξειδικευμένα διαδραστικά αντικείμενα, τα δημιουργούν οι ίδιοι είτε προσθέτοντας λειτουργικότητα σε ήδη υλοποιημένα συστατικά είτε δημιουργώντας νέα αντικείμενα από την αρχή. Έτσι οι προγραμματιστές πολλές φορές έρχονται στο σημείο να δημιουργούν αντικείμενα τα οποία θα χρησιμοποιήσουν μία μόνο φορά ή αντικείμενα που προσφέρουν διαδραστικότητα για ένα και μόνο πεδίο εφαρμογής. Βέβαια η δημιουργία νέων διαδραστικών αντικειμένων δεν είναι ένα απλό ζήτημα. Πρέπει να λαμβάνονται υπόψη και θέματα που αφορούν την μεταφερσιμότητα των αντικειμένων σε διαφορετικές πλατφόρμες, ποια και πόσα στοιχειώδη διαδραστικά αντικείμενα θα χρησιμοποιηθούν κ.α.

Την λύση σε τέτοια θέματα έρχονται να δώσουν βιβλιοθήκες με πρόσθετα διαδραστικά εργαλεία τα οποία εξυπηρετούν συγκεκριμένους σκοπούς είτε αλληλεπίδρασης, είτε παρουσίασης. Για παράδειγμα σε περιπτώσεις όπου υπάρχει η ανάγκη γραφικής παρουσίασης δεδομένων με ένα τρόπο διαφορετικό από αυτούς που προσφέρει η βιβλιοθήκη (εάν προσφέρει κάποιον). Για τέτοιες εφαρμογές όπου η διαφορετική απεικόνιση της πληροφορίας είναι αναγκαία πρέπει να βρεθεί η κατάλληλη βιβλιοθήκη που να υλοποιεί τέτοιου είδους αναπαραστάσεις. Έχουν αναπτυχθεί πολλές γραφικές βιβλιοθήκες οι οποίες προσφέρουν συστατικά τα οποία είναι περισσότερο εξειδικευμένα και άρα κατάλληλα για την χρήση σε συγκεκριμένου σκοπού διεπαφές. Βιβλιοθήκες που προσφέρουν δυνατότητα γραφικής αναπαράστασης δεδομένων, με πλήρη υλοποιημένη λειτουργικότητα, ή βιβλιοθήκες που παρέχουν ειδικού σκοπού συστατικά τα οποία είναι προσανατολισμένα σε ένα πεδίο εφαρμογής, χρησιμοποιούνται συχνά από τους προγραμματιστές σαν συμπληρωματικές της βασικής βιβλιοθήκης που χρησιμοποιούν.

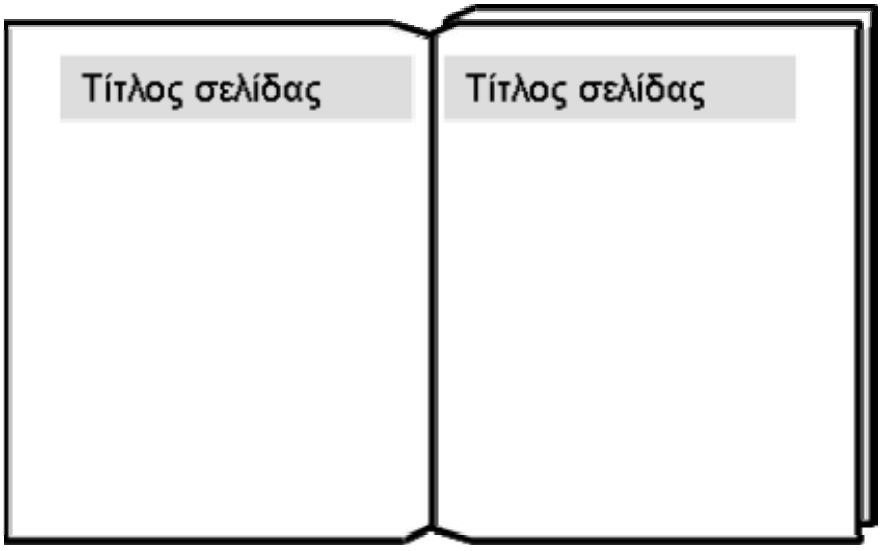
Προβλήματα βέβαια προκύπτουν από την χρήση μίας δεύτερης βιβλιοθήκης και έχουν να κάνουν με την μεταξύ τους συνεργασία και διαλειτουργικότητα. Δύο βιβλιοθήκες για να λειτουργήσουν μαζί θα πρέπει ο προγραμματιστής να είναι έμπειρος και να προετοιμάσει το έδαφος για την χρήση των δύο ή περισσοτέρων βιβλιοθηκών. Τα θέματα που προκύπτουν σε αυτές τις περιπτώσεις είναι το κατά πόσο οι βιβλιοθήκες αυτές μπορούν να ‘δέσουν’ μεταξύ τους και πως η έξοδος (output) της μίας να γίνεται είσοδος (input) στην άλλη και ο αντίστροφο.

Οι βιβλιοθήκες διαδραστικών αντικειμένων σήμερα χρησιμοποιούνται σε πολύ μεγάλο βαθμό και αποτελούν τον πιο δημοφιλή τρόπο κατασκευής διεπαφών. Οι περισσότεροι προγραμματιστές επωφελούνται από τις δυνατότητες για γρήγορη και εύκολη κατασκευή διεπαφών που προσφέρουν αλλά είναι και πολλές οι περιπτώσεις στις οποίες οι βιβλιοθήκες δεν επαρκούν να καλύψουν τις ανάγκες τους. Σε αυτές τις περιπτώσεις οι όποιες λύσεις υιοθετηθούν θα πρέπει να σχεδιαστούν με προσοχή και να ληφθούν υπόψη όλοι εκείνοι οι περιορισμοί που επέρχονται είτε με την δημιουργία εξειδικευμένων (custom) αντικειμένων, είτε με την χρήση επιπλέον βιβλιοθηκών.

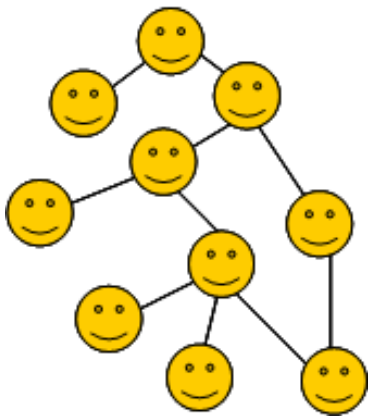
3 Υποστήριξη σύνθετων συστατικών αλληλεπίδρασης

3.1 Σύνθετα διαδραστικά στυλ αλληλεπίδρασης

Το προηγούμενο κεφάλαιο κατέληξε διατυπώνοντας την ανάγκη υποστήριξης προηγμένων σχεδιαστικών λύσεων που απαιτούν συστατικά αλληλεπίδρασης που δεν προσφέρονται έτοιμα από σύγχρονες εργαλειοθήκες όπως η MFC και το Swing. Η ανάγκη αυτή γίνεται ολοένα και περισσότερο άμεση αν αναλογιστεί κανείς την ευρύτητα των σχεδιαστικών λύσεων που επιδέχονται κάποια εξειδικευμένα προβλήματα. Ας θεωρήσουμε παραδείγματος χάριν το εξής σενάριο. Έστω ότι ένα προγραμματιστής έχει στη διάθεσή τους τρεις διαφορετικές βιβλιοθήκες διαδραστικών αντικειμένων. Η πρώτη είναι μία βελτιωμένη έκδοση του Motif για X-Windows που υλοποιεί ένα διαδραστικό αντικείμενο τύπου 'βιβλίο' που επιτρέπει επισκόπηση των περιεχομένων του βιβλίου, ξεφύλλισμα από σελίδα σε σελίδα και σήμανση επιλεγμένων περιοχών μίας σελίδας με τη χρήση μίας πένας αφής (βλέπε Εικόνα 8). Η δεύτερη βιβλιοθήκη είναι υλοποιημένη σε MFC και υποστηρίζει τη γραφική αναπαράσταση ιεραρχιών δεδομένων (βλέπε Εικόνα 9). Η τελευταία βιβλιοθήκη είναι το Swing που προσφέρει ένα συστατικό τύπου TabbedPane μόνο που το συστατικό αυτό υποστηρίζει ένδειξη πολλαπλής χρήσης (multiuser widget) όπως στην Εικόνα 10.



Εικόνα 8: Το αντικείμενο συμπερίληψης 'βιβλίο' στο Motif

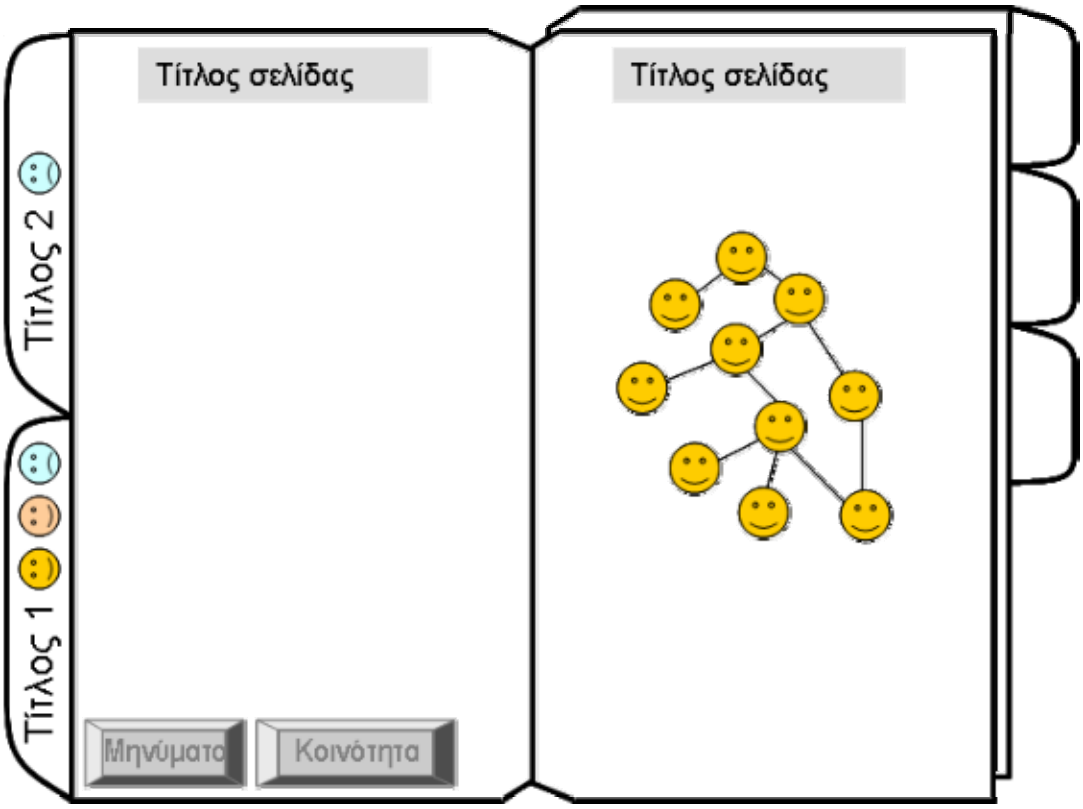


Εικόνα 9: Ιεραρχία αντικειμένων στην MFC

Εικόνα 10: TabbedPane με ένδειξη πολλαπλής ταυτόχρονης χρήσης στο Swing

Είναι προφανές ότι τα παραπάνω σενάρια εξυπηρετούν μόνο την ανάπτυξη ενός προβληματισμού που στοχεύει να καταδείξει το εύρος των σχεδιαστικών λύσεων που είναι διαθέσιμες σε ένα σχεδιαστή διεπαφών. Είναι επίσης προφανές ότι τα σενάρια αυτά θα μπορούσαν να εμπλουτιστούν περαιτέρω (π.χ. με βιβλιοθήκες animation για το ξεφύλλισμα του βιβλίου) ή να προσαρμοστούν σε συγκεκριμένα πεδία εφαρμογής (π.χ. ηλεκτρονικός φάκελος υγείας ασθενούς). Ωστόσο αυτό που ενδιαφέρει στην παρούσα ανάλυση δεν είναι η εξειδίκευση αλλά το κατά πόσο οι σχεδιαστές σύγχρονων διεπαφών μπορούν να βασιστούν και να αξιοποιήσουν την πληθώρα των εργαλειοθηκών που είναι σήμερα διαθέσιμες με δημιουργικό τρόπο – δηλαδή να συνθέσουν διεπαφές από σύνθετα διαδραστικά στυλ.

Με τον όρο ‘σύνθετο διαδραστικό στυλ’ εννοούμε τεχνικές αλληλεπίδρασης που αξιοποιούν συστατικά αλληλεπίδρασης υλοποιημένα από και διαθέσιμα μέσω διαφορετικών εργαλειοθηκών. Με άλλα λόγια θα ήταν χρήσιμο να αναρωτηθούμε προς στιγμή αν και κατά πόσο μπορεί να υλοποιηθεί μία διεπαφή σαν αυτή στην Εικόνα 11. Το ιδιαίτερο χαρακτηριστικό της συγκεκριμένης διεπαφής είναι ότι απαρτίζεται από συστατικά που ανήκουν (είναι υλοποιημένα) σε διαφορετικές εργαλειοθήκες. Παρατηρούμε ότι το αντικείμενο υποδοχέας (υψηλού επιπέδου αντικείμενο συμπερίληψης – top level container) είναι συνδυασμός του βιβλίου του Motif και του TabbedPane του Swing. Επιπρόσθετα, υπονοείται η διαλειτουργικότητα των δύο συστατικών αφού η αλληλεπίδραση με το TabbedPane επηρεάζει την κατάσταση του βιβλίου. Με άλλα λόγια, για να μπορεί ο συγκεκριμένος container να υλοποιηθεί θα πρέπει και το Swing και το Motif να είναι ταυτόχρονα ενεργά και να διαλειτουργούν.



Εικόνα 11: Παράδειγμα σύνθετου διαδραστικού στυλ αλληλεπίδρασης

Εξετάζοντας περαιτέρω τη διεπαφή παρατηρούμε ότι αν και η δομή των δύο σελίδων είναι παρόμοια τα περιεχόμενά τους διαφέρουν. Η αριστερή σελίδα περιέχει / φιλοξενεί συμβατικά κουμπιά εντολών του Swing ή της MFC ενώ στη δεξιά σελίδα υπάρχει μία ιεραρχία δεδομένων της MFC. Πρακτικά αυτό σημαίνει ότι και οι δύο σελίδες λειτουργούν ως υποδοχείς άλλων αντικειμένων που όμως ανήκουν σε διαφορετικές εργαλειοθήκες. Συνοψίζοντας λοιπόν πρέπει να τονίσουμε ότι η υλοποίηση της παραπάνω διεπαφής απαιτεί τρεις διαφορετικές βιβλιοθήκες που είναι ταυτόχρονα διαθέσιμες και διαλειτουργούν.

3.2 Προβλήματα υλοποίησης σύνθετων διαδραστικών στυλ

Από την παραπάνω ανάλυση και με βάση την υφιστάμενη κατάσταση των γραφικών εργαλειοθηκών θα μπορούσαμε εύκολα να εξάγουμε το συμπέρασμα ότι μία διεπαφή σαν αυτή στην Εικόνα 11 δεν είναι εφικτή με την υπάρχουσα τεχνολογία και γνώση, ή τουλάχιστον δεν είναι εύκολα υλοποιήσιμη. Αυτό βέβαια έχει επιπτώσεις και στο βαθμό ελευθερίας των σχεδιαστών που εμφανώς περιορίζεται από τις δυνατότητες των διαθέσιμων εργαλειοθηκών.

Αξίζει να διερευνήσουμε κάποιους από τους λόγους που δεν επιτρέπουν την (εύκολη) υλοποίηση τέτοιων διεπαφών και παράλληλα να εξάγουμε συμπεράσματα για το τι θα βελτίωνε την υφιστάμενη κατάσταση. Τα προβλήματα που περιορίζουν την ανάπτυξη σύνθετων διαδραστικών στυλ είναι κυρίως τρία και συνοψίζονται ως εξής:

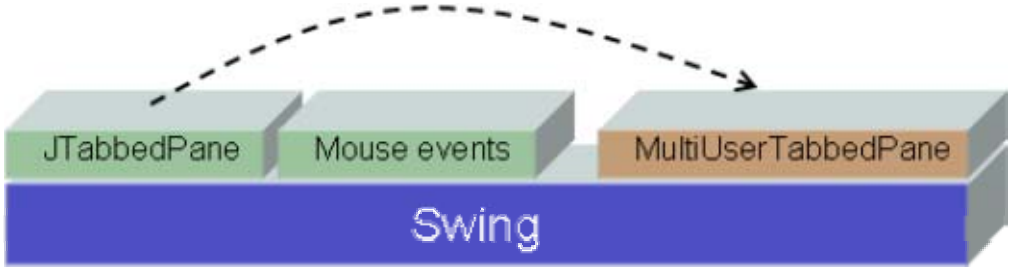
- Οι εργαλειοθήκες διαδραστικών αντικειμένων είναι υλοποιημένες σε διαφορετικές γλώσσες προγραμματισμού που δεν είναι πάντοτε συμβατές. Η μη συμβατότητα των εργαλειοθηκών έγκειται στο γεγονός ότι οι διαφορετικές γλώσσες προγραμματισμού που χρησιμοποιούνται για την υλοποίηση τους πολλές φορές διαφέρουν σε καίρια σημεία ενώ κάποιες επιτρέπουν λειτουργικότητες που άλλες δεν διαθέτουν.
- Οι εργαλειοθήκες αυτές δεν διαλειτουργούν ή δεν συγχρονίζονται στον απαιτούμενο βαθμό. Ανάλογα με τον τρόπο υλοποίησης των συστατικών μίας εργαλειοθήκης τα συστατικά μπορεί να διαφέρουν σε δυνατότητες (π.χ. παραμέτρους) ή λειτουργίες (π.χ. τεχνικές αλληλεπίδρασης) από συστατικά άλλων βιβλιοθηκών. Έτσι, η δυνατότητα συγχρονισμού των εργαλειοθηκών και η διαλειτουργικότητα των συστατικών τους πολλές φορές καθίσταται, αν όχι ανέφικτη, κατά πολύ περιορισμένη.
- Οι εργαλειοθήκες διαφέρουν όσο αφορά το προγραμματιστικό μοντέλο (toolkit programming model) που υιοθετούν και το API που υποστηρίζουν σε τέτοιο βαθμό που κάποιος θα πρέπει να είναι αρκετά έμπειρος χρήστης όλων των εργαλειοθηκών που θέλει να χρησιμοποιήσει ώστε να είναι σε θέση να ‘δέσει’ τις εργαλειοθήκες αυτές.

Θα μπορούσε κάποιος να αναφέρει επιπλέον προβλήματα όπως παραδείγματος χάριν διαφορές στη βασική αρχιτεκτονική δομή που υποστηρίζεται (MVC, M-UI delegate, PAC) από κάθε βιβλιοθήκη, ποσοστό χρήσης native συστατικών, δυνατότητα διαχείρισης πολλαπλών εισόδων δεδομένων (handling multiple input), κοκ, αλλά αυτός δεν είναι ο στόχος μας στην παρούσα φάση – αντίθετα ελπίζουμε να έχει γίνει σαφές ότι υπάρχουν πολλά επιμέρους ζητήματα που λειτουργούν περιοριστικά σε σχέση με το εύρος των διεπαφών που αναπτύσσονται με σύγχρονες και πλέον μοντέρνες εργαλειοθήκες διαδραστικών αντικειμένων.

3.3 Γιατί χρειάζεται ένα ενιαίο πλαίσιο υποστήριξης

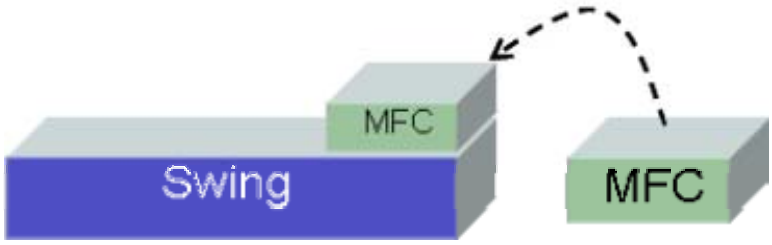
Έχοντας διαπιστώσει τα προβλήματα υλοποίησης σύνθετων διαδραστικών στυλ ας επιχειρήσουμε να διεισδύσουμε περαιτέρω στο πρόβλημα αναλύοντας κάποια από τα πράγματα που είναι υλοποιήσιμα σήμερα – αν και απαιτούν κόπο και ιδιαίτερη γνώση – και που θα συνεισέφεραν στον προβληματισμό που αναπτύχθηκε προηγούμενα.

Η πρώτη παρατήρηση είναι ότι δεδομένης μίας εργαλειοθήκης διαδραστικών αντικειμένων θα μπορούσαμε να αλλάξουμε κάποιες από τις υποστηριζόμενες τεχνικές αλληλεπίδρασης υποστηρίζοντας νέες (βλέπε Εικόνα 12). Η τεχνική αυτή είναι δημοφιλής και χρησιμοποιείται ευρέως για την υποστήριξη τεχνικών αλληλεπίδρασης βασισμένων σε εναλλακτικές συσκευές εισόδου δεδομένων (π.χ. χειρονομίες με πένα αφής ή παιχνιδολαβή). Με παρόμοιο τρόπο θα μπορούσαμε να μετατρέψουμε τις βιβλιοθήκες του Swing έτσι ώστε να υποστηρίζουν το TabbedPane που περιγράψαμε προηγουμένως.

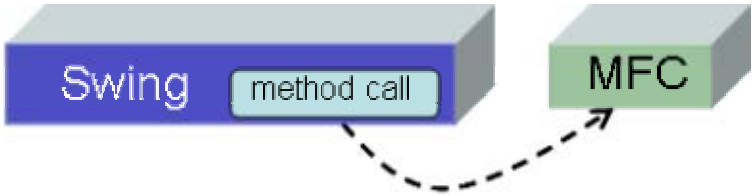


Εικόνα 12: Αλλαγή τεχνικών αλληλεπίδρασης εργαλειοθήκης

Επίσης, θα μπορούσαμε να εξασφαλίσουμε διαλειτουργικότητα δύο εργαλειοθηκών είτε μέσω της ενσωμάτωσης της μίας μέσα στην άλλη (βλέπε Εικόνα 13) είτε μέσω προσαρμογών σε επίπεδο API της μίας βιβλιοθήκης (βλέπε Εικόνα 15). Και οι δύο περιπτώσεις είναι δυνατές υπό συνθήκες ενώ είναι επίσης προφανές ότι η μία εργαλειοθήκη αναγκαστικά υιοθετεί ή πρέπει να είναι συμβατή με αυτή που την φιλοξενεί.



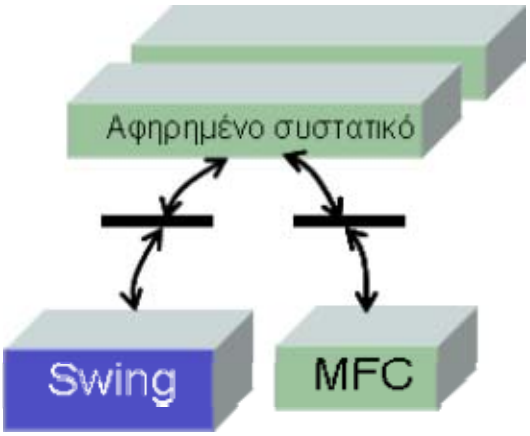
Εικόνα 13: Στρατηγική ενσωμάτωσης εργαλειοθήκης



Εικόνα 15: Προσαρμογή στο API μίας βιβλιοθήκης

Αναγκαία προϋπόθεση για την διαλειτουργικότητα δύο εργαλειοθηκών είναι να είναι υλοποιημένες, αν όχι με την ίδια γλώσσα προγραμματισμού, με συμβατές μεταξύ τους γλώσσες. Μία εργαλειοθήκη από αυτές (ή και οι δύο) είναι απαραίτητο να είναι κατασκευασμένη με τέτοιο τρόπο ώστε να επιτρέπει την εισαγωγή / ενσωμάτωση άλλων

εργαλειοθηκών. Επίσης, τα συστατικά της εργαλειοθήκης πρέπει να είναι υλοποιημένα έτσι ώστε να δίνουν την δυνατότητα από την εργαλειοθήκη που την καλεί / ενσωματώνει να μπορεί να τα διαχειριστεί. Σε όλες τις περιπτώσεις που προαναφέραμε απαιτούνται κλήσεις στις υλοποιημένες βιβλιοθήκες της εργαλειοθήκης γεγονός που δημιουργεί πρόβλημα.



Εικόνα 14: Αφηρημένα συστατικά και 'σύνδεση με' αντί 'κλήση' βιβλιοθήκης

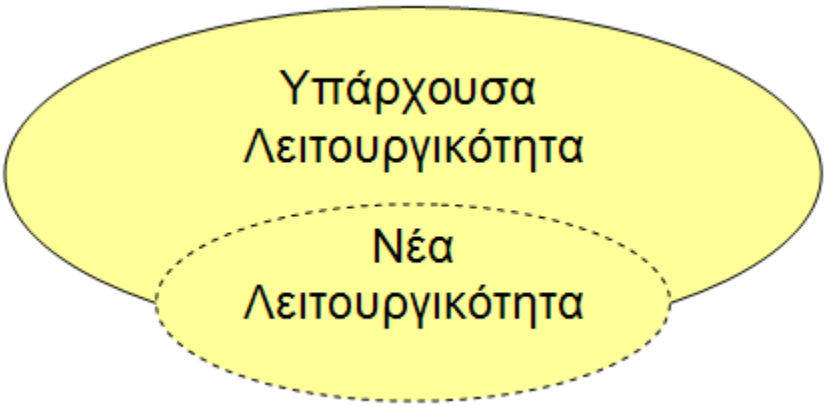
Μία άλλη προσέγγιση που περιορίζει το πρόβλημα αυτό, αν και δεν το εξαλείφει, συνοψίζεται στην Εικόνα 14. Εδώ ο τρόπος ενσωμάτωσης ή κλήσης της βιβλιοθήκης επιτυγχάνεται κάνοντας 'σύνδεση' αντί 'κλήση' της επιθυμητής βιβλιοθήκης. Την σύνδεση αυτήν αναλαμβάνει ένα επιπλέον επίπεδο το οποίο εκτελεί δύο βασικές λειτουργίες. Η πρώτη είναι ότι 'γνωρίζει' πώς να καλέσει και να διαχειριστεί βιβλιοθήκες μίας εργαλειοθήκης. Η δεύτερη λειτουργία είναι ότι προσφέρει ένα επίπεδο αφαίρεσης που κρύβει τις λεπτομέρειες κάθε βιβλιοθήκης με τρόπο που να είναι δυνατή μία γλώσσα επικοινωνίας με αυτές. Η γλώσσα αυτή συνήθως αποκαλείται 'γλώσσα προδιαγραφής διεπαφών' και προσφέρει δύο ειδών λειτουργίες: (α) ένα σύνολο αφηρημένων διαδραστικών αντικειμένων και (β) ένα εξειδικευμένο API για τη επικοινωνία με το ενδιάμεσο επίπεδο. Κατά το παρελθόν έχουν χρησιμοποιηθεί ποικίλες τεχνολογίες για την δημιουργία γλωσσών τέταρτης γενιάς που να υποστηρίζουν την παραπάνω αρχιτεκτονική. Πρέπει να τονίσουμε στα θετικά της προσέγγισης ότι με τον τρόπο αυτό προσφέρεται στον προγραμματιστή η δυνατότητα να χρησιμοποιεί ένα ενιαίο προγραμματιστικό μοντέλο για την διαχείριση των βιβλιοθηκών χωρίς να λαμβάνονται υπόψη τα ιδιαίτερα χαρακτηριστικά κάθε βιβλιοθήκης. Βεβαίως το μειονέκτημα είναι ότι η ανάπτυξη του ενδιάμεσου επιπέδου και η ενσωμάτωσή τους στην αφηρημένη γλώσσα προδιαγραφών είναι απαιτητικές προγραμματιστικές λειτουργίες.

4 Πλαίσιο αναφοράς & στρατηγικές υποστήριξης

Το προηγούμενο κεφάλαιο ανέδειξε τόσο την ανάγκη διαχείρισης διαφορετικών διαδραστικών αντικειμένων – που συχνά δεν προσφέρονται από μία και μόνο εργαλειοθήκη – αλλά και την πολυπλοκότητα που υπάρχει στην αξιοποίηση διαφορετικών εργαλειοθηκών. Στο κεφάλαιο αυτό θα επιχειρήσουμε να συνθέσουμε ένα ενιαίο πλαίσιο αναφοράς που να αντιμετωπίζει το πρόβλημα που περιγράψαμε στο προηγούμενο κεφάλαιο επαρκώς. Συγκεκριμένα θα περιγράψουμε ένα μικρό αριθμό στρατηγικών διαχείρισης μίας ή περισσοτέρων εργαλειοθηκών διαδραστικών αντικειμένων με τρόπο που να μπορούμε να υλοποιήσουμε σύνθετα στυλ αλληλεπίδρασης. Οι στρατηγικές αυτές περιγράφονται λεπτομερώς παρακάτω σε επίπεδο ορισμού και προσέγγισης ενώ στο επόμενο κεφάλαιο παρουσιάζονται παραδείγματα τρέχουσας υλοποίησης στα πλαίσια του έργου eKoNEΣ.

4.1 Επαύξηση (Augmentation)

Ο σχεδιασμός και η υλοποίηση διαδικασιών με τις οποίες νέα λειτουργικότητα προστίθεται σε ένα ήδη υπάρχον διαδραστικό αντικείμενο, το οποίο παρέχεται από μία βιβλιοθήκη διαδραστικών αντικειμένων, ονομάζεται επαύξηση (augmentation). Ο ρόλος της επαύξησης είναι να βελτιώσει την λειτουργικότητα ενός αντικειμένου ή να προσφέρει νέες τεχνικές αλληλεπίδρασης με τα αντικείμενα της βιβλιοθήκης. Επίσης η επαύξηση είναι χρήσιμη στις περιπτώσεις που η λειτουργικότητα ενός ή περισσοτέρων διαδραστικών αντικειμένων δεν επαρκεί να καλύψει τις ανάγκες αλληλεπίδρασης.

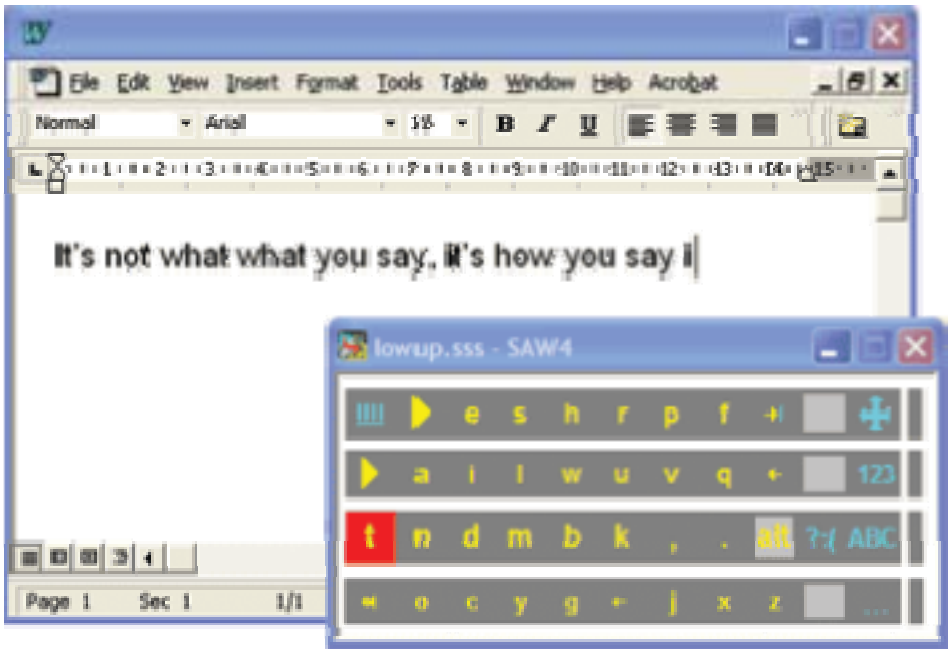


Εικόνα 15: Επαύξηση (augmentation)

Η επαύξηση έχει χρησιμοποιηθεί κατά το παρελθόν για να προσφέρει νέες τεχνικές αλληλεπίδρασης για χρήστες οι οποίοι αντιμετωπίζουν προβλήματα με την υπάρχουσα λειτουργικότητα των συστατικών της βιβλιοθήκης. Ενδεικτικά, ο έλεγχος μίας διεπαφής μέσω φωνής δεν υποστηρίζεται από πολλές βιβλιοθήκες διαδραστικών αντικειμένων – στην πραγματικότητα είναι ελάχιστες αυτές που υποστηρίζουν τέτοια λειτουργικότητα. Έτσι κάποιος ήταν αδύνατον να αλληλεπιδράσει με ένα σύστημα, για παράδειγμα, κατά την διάρκεια της οδήγησης ή όταν η συσκευή που χρησιμοποιεί είναι περιορισμένη (π.χ. οθόνη αφής). Επίσης χρήστες με κινητικά προβλήματα είναι αδύνατον να χρησιμοποιήσουν ένα σύστημα το οποίο δεν υποστηρίζει ευκολίες πρόσβασης. Προσθέτοντας όμως νέα λειτουργικότητα σε μία βιβλιοθήκη, χρηστές των παραπάνω

κατηγόριων, θα μπορούσαν κάνοντας χρήση των νέων αυτών τεχνικών να έχουν μία αποδοτική αλληλεπίδραση με την διεπαφή.

Μία υλοποίηση που σαν στόχο είχε να αντικαταστήσει την αλληλεπίδραση μέσω του ποντικού και του πληκτρολόγιου στο δημοφιλές λειτουργικό WINDOWS της Microsoft είναι το Special Access to Windows (SAW). Το SAW δίνει την δυνατότητα όλες οι λειτουργίες των WINDOWS να μεταφράζονται σε μία σειρά από σύνολα επιλογών (selection sets). Στην Εικόνα 16 εμφανίζεται η χρήση του SAW για την επεξεργασία κειμένου. Η βασική ιδέα είναι ότι ο χρήστης επιλέγει χαρακτήρες προς δακτυλογράφηση από ένα πάνελ επιλογής που προσομοιάζει το πληκτρολόγιο. Το πάνελ σαρώνεται από ένα highlighter που προσδιορίζει το τρέχον αντικείμενο εστίασης. Συνήθως ο έλεγχος του highlighter γίνεται με τη χρήση δύο διακοπτών όπου ο ένας μεταφέρει το highlighter στο επόμενο αντικείμενο ενώ ο άλλος κάνει επιλογή. Με τον τρόπο αυτό η εφαρμογή γίνεται πλήρως προσβάσιμη μέσω εναλλακτικών τεχνικών πρόσβασης.



Εικόνα 16: Η χρήση διακοπτών για την εκτέλεση καθηκόντων επεξεργασίας κειμένου

Τα πάνελ επιλογής του SAW μπορεί να περιέχουν γράμματα, λέξεις, σύμβολα, αριθμούς και σχήματα τα οποία μπορούν αυτόματα να σαρωθούν και να παρουσιαστούν ως προς επιλογή αντικείμενα, συχνά υποστηρίζοντας αυξημένη λειτουργικότητα στη διεπαφή. Έτσι ο χρήστης μπορεί να εισάγει ένα χαρακτήρα, να ζωγραφίσει ένα σχήμα ή να ενεργοποιήσει μία επιλογή, μόνο με την χρήση των συστατικών που περιέχονται στα σύνολα επιλογής. Επίσης, είναι δυνατό τα πάνελ επιλογής να υποστηρίζουν και επιλογές προσανατολισμού. Παραδείγματος χάριν, η κατεύθυνση κίνησης του highlighter (εμπρός ή πίσω, πάνω ή κάτω) μπορεί να αλλάξει επιλέγοντας το κατάλληλο σύμβολο (π.χ. soft κουμπί) από το αντίστοιχο πάνελ πλοήγησης / κατεύθυνσης. Ένα τέτοιο πάνελ θα μπορούσε επίσης να επιτρέπει επιλογή επόμενης γραμμής ή στήλης και έτσι να αυξάνεται η ταχύτητα εκτέλεσης των καθηκόντων του χρήστη. Βλέπουμε λοιπόν ότι όλες οι λειτουργίες του συστήματος – είτε πλοήγηση είτε λειτουργικό καθήκον όπως επεξεργασία κειμένου, ζωγραφική κλπ., μπορούν να εκτελεστούν με δύο εντολές όπως `ΕΠΟΜΕΝΗ ΕΠΙΛΟΓΗ` (next) και `ΕΠΕΛΕΞΕ` (select). Αυτό το χαρακτηριστικό καθιστά το SAW σε ένα πολύ χρήσιμο εργαλείο για τους χρήστες που χρησιμοποιούν εναλλακτικούς τρόπους αλληλεπίδρασης όπως voice commands ή κάποιον εκλεκτή περιορισμένων δυνατοτήτων.

Αυτή η τεχνική, βασισμένη σε επιλογές, είναι πολύ δημοφιλής στην αντιμετώπιση των προβλημάτων που προκύπτουν από τους σύνθετους τρόπους αλληλεπίδρασης. Μία λειτουργία σε ένα γραφικό περιβάλλον για να εκτελεστεί, θα πρέπει ο χρηστής να επιλέξει το ανάλογο διαδραστικό αντικείμενο και να 'πυροδοτήσει' το κατάλληλο γεγονός. Σύνθετα συστατικά που απαιτούν σύνθετους χειρισμούς, πρακτικά είναι ακατάλληλα για περιορισμένης διαδραστικότητας συσκευές. Υλοποιήσεις σαν το SAW μειώνουν την αυξημένη πολυπλοκότητα και παρουσιάζουν έναν πιο φιλικό τρόπο αλληλεπίδρασης.

Όμως, εκτός από το να υλοποιηθεί νέα λειτουργικότητα για να προσφερθούν εναλλακτικές τεχνικές αλληλεπίδρασης για διαφορετικές κατηγόριες χρηστών ή συσκευών, η επαύξηση της λειτουργικότητας του αντικειμένου μπορεί να επιφέρει σημαντικές βελτιώσεις στην χρηστικότητα του διαδραστικού αντικειμένου και κατ' επέκταση στην χρηστικότητα της διεπαφής. Έτσι, αντικείμενα που ήταν χρήσιμα για την εκτέλεση μίας λιγότερο σύνθετης λειτουργίας του συστήματος, μπορούν να τροποποιηθούν με τέτοιον τρόπο ώστε να είναι ιδανικά για την εκτέλεση πιο σύνθετων καθηκόντων. Επίσης αντικείμενα τα οποία ήταν ακατάλληλα για μία συγκεκριμένη λειτουργία, μπορούν με την κατάλληλη επαύξηση της λειτουργικότητας τους να γίνουν ιδανικά.

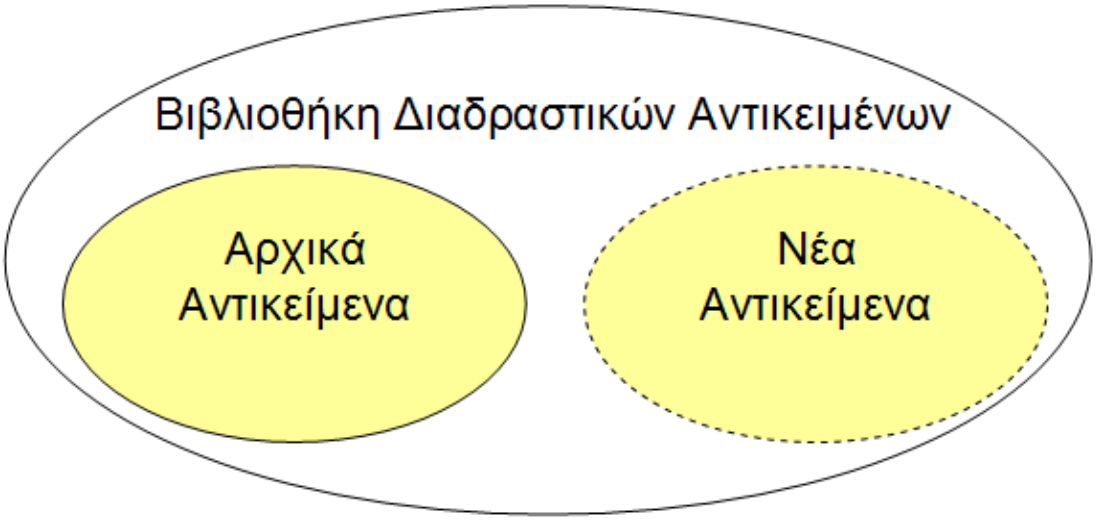
Η επαύξηση της λειτουργικότητας ενός ή περισσοτέρων διαδραστικών αντικειμένων της βιβλιοθήκης προϋποθέτει την υλοποίηση χαμηλών επιπέδων ελέγχων. Συγκεκριμένα σε αμιγώς αντικειμενοστραφή συστήματα (όπως η MFC και το Swing) θα πρέπει να υλοποιηθεί νέα λειτουργικότητα στους εκλεκτές συμβάντων ώστε οι ενέργειες του χρήστη μέσω των νέων συσκευών να μεταφράζονται κατάλληλα. Επίσης, θα πρέπει να παρέχονται νέες μέθοδοι για την διαχείριση του αντικειμένου εστίασης (focus object) ώστε να είναι δυνατή η επίγνωση της προέλευσης της ενέργειας του χρηστή αλλά και του τύπου του αποτελέσματος που η ενέργεια αυτή θα έχει. Η σχέση μεταξύ των αρχικών διαδραστικών αντικειμένων της βιβλιοθήκης με των νέων, επαυξημένης λειτουργικότητας, αντικειμένων είναι κληρονομική. Τα νέα αντικείμενα κληρονομούν όλα τα χαρακτηριστικά των αντίστοιχων που επαυξάνουν είτε κάνοντας sub classing σε όρους αντικειμενοστραφούς προγραμματισμού (object oriented programming), είτε κάνοντας επανασύνθεση του αρχικού αντικειμένου.

Η χρησιμότητα της επαύξησης έγκειται στο γεγονός ότι δεν χρειάζεται να δημιουργηθούν από την αρχή νέα αντικείμενα που να παρέχουν την επιπλέον λειτουργικότητα που απαιτείται. Αυτό είναι μεγάλο πλεονέκτημα στις περιπτώσεις όπου δεν θέλουμε να επανασχεδιάσουμε την διεπαφή (με νέα αντικείμενα που υλοποιούν τις ανάγκες μας) αλλά να χρησιμοποιήσουμε μία υπάρχουσα υλοποίηση. Έτσι μπορεί απλά να γίνει η σύνδεση της υλοποίησης με την νέα επαυξημένη βιβλιοθήκη και η διεπαφή να προσφέρει τον νέο ή τους νέους τρόπους αλληλεπίδρασης που επιθυμούμε. Βέβαια στις περιπτώσεις όπου η επαύξηση επιφέρει αύξηση της χρηστικότητας σε ένα ή περισσότερα διαδραστικά συστατικά, η διεπαφή είναι πολύ πιθανών να χρειάζεται επανασχεδιασμό ώστε να γίνει χρήση της αυξημένης χρηστικότητας των συστατικών.

4.2 Επέκταση (Expansion)

Επέκταση (expansion) μίας βιβλιοθήκης διαδραστικών αντικειμένων ορίζεται ως η διαδικασία κατά την οποία νέα διαδραστικά αντικείμενα προστίθενται στην ήδη υπάρχουσα συλλογή αντικειμένων τα οποία δεν παρέχονται από την συγκεκριμένη βιβλιοθήκη. Σκοπός της επέκτασης της βιβλιοθήκης είναι να εισάγει διαδραστικά αντικείμενα αυστηρά καθορισμένα από το σκοπό του συστήματος. Απαραίτητη προϋπόθεση για την επέκταση μίας βιβλιοθήκης είναι, τα νέα διαδραστικά αντικείμενα που την συμπληρώνουν να γίνονται διαθέσιμα με τον ίδιο (προγραμματιστικά) τρόπο με

τα αρχικά, έτσι ώστε ο χρήστης της βιβλιοθήκης να μη μπορεί να ξεχωρίσει ποια είναι τα νέα αντικείμενα από το σύνολο των αντικειμένων.



Εικόνα 17: Επέκταση (expansion)

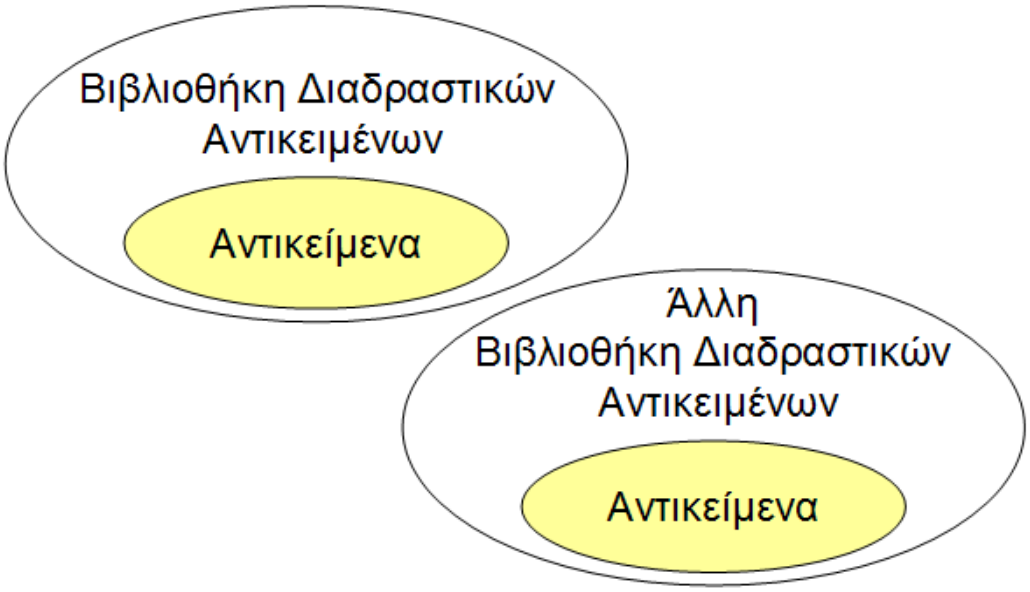
Κατά το παρελθόν, η επέκταση της βιβλιοθήκης έχει χρησιμοποιηθεί για να διευκολύνει την ανάπτυξη συστατικών για την παρουσίαση διαφορετικών μεταφορών αλληλεπίδρασης και επίσης για την ανάπτυξη καινοτόμων τεχνικών παρουσίασης πληροφορίας. Για παράδειγμα, σε κάποιες περιπτώσεις, μία κατάλληλη μεταφορά είναι αναγκαία για την παρουσίαση των εξειδικευμένων λειτουργιών ενός συστήματος. Αν τα διαθέσιμα διαδραστικά αντικείμενα δεν επαρκούν για την δημιουργία αυτής της μεταφοράς τότε νέα διαδραστικά αντικείμενα πρέπει να υλοποιηθούν.

Σε μία εικονική βιβλιοθήκη, για παράδειγμα, θα πρέπει να υπάρχει το κατάλληλο αντικείμενο που θα προσομοιώνει τη λειτουργία ενός βιβλίου ή ενός ραφιού. Σε ένα εκπαιδευτικό λογισμικό που απευθύνεται σε άτομα μικρής ηλικίας θα πρέπει να αναπτυχθούν τα διαδραστικά αντικείμενα που θα επιτρέπουν την υλοποίηση της κατάλληλης μεταφοράς. Αυτού του είδους τα συστήματα θα ‘τρέχουν’ σε μία γραφική βιβλιοθήκη και όπως γίνεται αντιληπτό οι προγραμματιστές θα πρέπει να την επεκτείνουν με τα κατάλληλα διαδραστικά συστατικά έτσι ώστε να μπορέσουν να τα υλοποιήσουν.

Η επέκταση μίας βιβλιοθήκης με νέα συστατικά συναντάτε πιο συχνά από ότι η επαύξηση της λειτουργικότητας των αρχικών συστατικών. Αυτό συμβαίνει γιατί η δημιουργία νέων συστατικών είναι ευκολότερη διαδικασία καθώς επίσης και γιατί είναι πιο συχνό το φαινόμενο να υπάρχει η ανάγκη για διαδραστικά αντικείμενα τα οποία έχουν συγκεκριμένο ρόλο (domain specific).

4.3 Ενσωμάτωση (Integration)

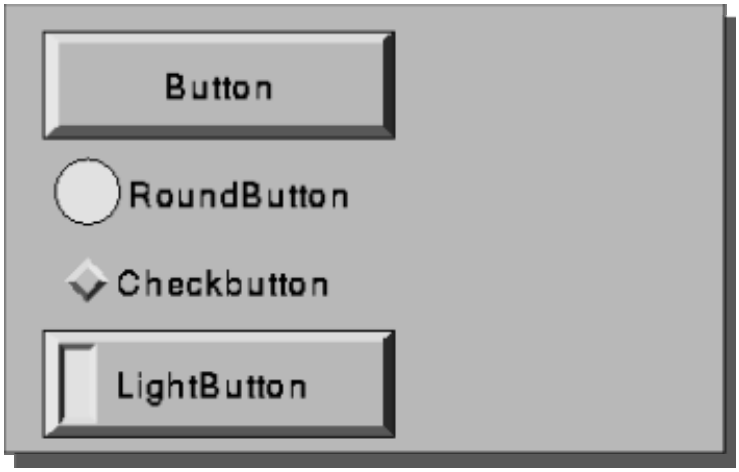
Μία πλατφόρμα ανάπτυξης διαδραστικών αντικειμένων όπως για παράδειγμα η βιβλιοθήκη αντικειμένων Swing της JAVA θεωρείται ότι υποστηρίζει ενσωμάτωση βιβλιοθήκης εάν επιτρέπει την εισαγωγή μίας δεύτερης βιβλιοθήκης με τρόπο κατά τον οποίο όλα τα συστατικά της δεύτερης γίνονται διαθέσιμα όπως τα συστατικά της αρχικής βιβλιοθήκης. Η ενσωμάτωση είναι αρκετά χρήσιμη στις περιπτώσεις όπου τα αρχικά συστατικά δεν επαρκούν να καλύψουν τις ανάγκες του συστήματος. Έτσι ενσωματώνοντας μία άλλη έτοιμη βιβλιοθήκη (third party library) μπορούμε να χρησιμοποιήσουμε τα συστατικά που υλοποιούνται σε αυτήν χωρίς να χρειαστεί να υλοποιήσουμε τα δικά μας διαδραστικά αντικείμενα (βλέπε επαύξηση).



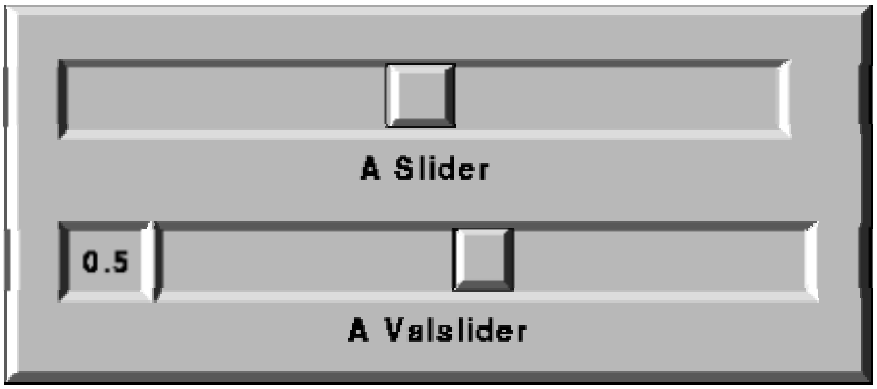
Εικόνα 18: Ενσωμάτωση (Integration)

Αρχικές εφαρμογές της στρατηγικής της ενσωμάτωσης υπήρξαν σε περιβάλλον X-Windows όπου οι διαθέσιμες βιβλιοθήκες ήταν πολλές (π.χ. Motif, XViews, XForms, Athena Widget Set) με καθεμία συχνά να υποστηρίζει διαφορετικούς τύπους διαδραστικών αντικειμένων. Παραδείγματος χάριν η εργαλειοθήκη XFORMS υποστήριζε διαφορετικές εκδόσεις κουμπιών εντολών (βλέπε Εικόνα 19), μπάρες κύλισης (βλέπε Εικόνα 20) καθώς και τα λεγόμενα αντικείμενα τύπου freeObject που καθιστούσαν την ανάπτυξη εξειδικευμένων αντικειμένων ευκολότερη σε σχέση με μία άλλη βιβλιοθήκη όπως το Athena Widget Set ή το Motif.

Έτσι ενώ ένας προγραμματιστής χρησιμοποιούσε το Motif ως βασική εργαλειοθήκη συχνά ήταν απαραίτητο να αξιοποιήσει και μία εναλλακτική βιβλιοθήκη που ήταν καταλληλότερη για τη σχεδίαση και χρήση νέων διαδραστικών αντικειμένων ή που προσέφερε βιβλιοθήκες των οποίων η επαύξηση ήταν ευκολότερη. Παραδείγματος χάριν μπορεί να ισχυριστεί κάποιος ότι αν ένα κουμπί εντολών επρόκειτο να υλοποιηθεί έτσι ώστε να δηλώνει από ποιους χρήστες χρησιμοποιείται ταυτόχρονα (multi-user widget) τότε το τελευταίο κουμπί στην Εικόνα 19 – όπου το κουμπί διαθέτει ένδειξη φωτός – θα ήταν ευκολότερο να επαυξηθεί από ότι ένα κλασικό κουμπί του Motif.



Εικόνα 19: Κατηγορίες κουμπιών στο Xforms



Εικόνα 20: Μπάρα κύλισης στο Xforms

Παρόλο που τέτοιου είδους εξειδικευμένα θέματα ήταν ευρέως διαδεδομένα στη κοινότητα του X-Windows το ίδιο δεν παρατηρείται για το περιβάλλον των Wndows σε προσωπικούς υπολογιστές όπου η κυριαρχία της Microsoft και το κλείδωμα βασικών τεχνολογιών περιόρισε την ανάπτυξη εναλλακτικών εργαλειοθηκών. Ωστόσο το θέμα της ενσωμάτωσης διαφορετικών εργαλειοθηκών εξακολουθεί και υφίσταται και ως ερευνητική κατεύθυνση αλλά και ως προσέγγιση ανάπτυξης μοντέρνων διαδραστικών συστημάτων.

4.4 Αφαίρεση (Abstraction)

Οι προηγούμενες στρατηγικές στις περιπτώσεις όπου εφαρμόζονται μπορεί να είναι ιδιαιτέρως απαιτητικές προγραμματιστικά, ειδικά όταν πρέπει ταυτόχρονα να συνυπάρξουν για την ανάπτυξη μίας εφαρμογής. Έτσι, συχνά προβάλλεται η ανάγκη υποστήριξης ενός επιπέδου υψηλότερου και ανεξάρτητου του επιπέδου των εργαλειοθηκών που να υποστηρίζει τον επιθυμητό βαθμό αφαίρεσης και διαφάνειας. Η δυνατότητα του εργαλείου ανάπτυξης διεπαφών να υποστηρίξει διαχείριση των διαδραστικών αντικείμενων με τρόπο πλήρως απαλλαγμένο από τις φυσικές διαδραστικές ιδιότητες τους, ορίζεται ως αφαίρεση (abstraction). Τα αφηρημένα διαδραστικά αντικείμενα είναι υψηλού επιπέδου διαδραστικές οντότητες οι οποίες απεικονίζουν γενικές ιδιότητες συμπεριφοράς. Αυτή η αφαιρετική προσέγγιση της βιβλιοθήκης επιτρέπει την υλοποίηση διεπαφών, όχι καλώντας απ' ευθείας τα διαδραστικά συστατικά άλλα δημιουργώντας μίας διασύνδεσης με αυτά. Ωστόσο, κατά την εκτέλεση, τα αφηρημένα διαδραστικά αντικείμενα μεταφράζονται στα ανάλογα συστατικά της βιβλιοθήκης. Η εφαρμογή μίας τέτοιας μεθόδου στην ανάπτυξη διεπαφών επιτρέπει στους προγραμματιστές να καθιερώσουν ένα ενιαίο μοντέλο προγραμματισμού για την χρήση διαφορετικών βιβλιοθηκών, ανεξάρτητα από τις ιδιότητες της κάθε βιβλιοθήκης.

5 Εφαρμογή πλαισίου στο Swing

Στο κεφάλαιο αυτό περιγράφουμε εφαρμογές των τεχνικών της επαύξησης, επέκτασης, ενσωμάτωσης και αφαίρεσης των διαδραστικών αντικειμένων μίας βιβλιοθήκης με αναφορά στο έργο eKoNΕΣ. Πληροφορίες για το έργο είναι διαθέσιμες από την ιστοσελίδα του έργου (www.e-kones.teiher.gr). Για την πληρέστερη παρουσία των αποτελεσμάτων της παρούσας εργασίας ακολουθεί μία σύντομη επισκόπηση του φυσικού αντικειμένου του έργου.

5.1 Επισκόπηση του έργου eKoNΕΣ

Το έργο eKoNΕΣ – eKονικές κοιΝότητες Επιχειρηματικότητας & καινοτομίαΣ στην περιφέρεια – στοχεύει στην εφαρμογή νέων τεχνολογιών για την ανάπτυξη νέων μεθόδων οργάνωσης της επιχειρηματικότητας με στόχο την καινοτομία και την ποιότητα. Στα πλαίσια του έργου αναπτύχθηκε ένα ηλεκτρονικό χωριό τοπικής κλίμακας με έμφαση το τοπικό τουριστικό προϊόν. Στόχος του ηλεκτρονικού χωριού ήταν η διεπιχειρησιακή εικονική δικτύωση εταίρων για την ενίσχυση της τεχνολογικής τους βάσης και την προαγωγή καινοτομιών (π.χ. νέα προϊόντα και υπηρεσίες) μέσω προηγμένων μορφών συνεργασίας.

Στα πλαίσια του eKoNΕΣ αναπτύχθηκε ένα εργαλείο το οποίο έδινε την δυνατότητα στα μέλη τη κοινότητας, μέσω τις ιδίας συνεισφοράς πόρων, να συμμετέχουν στην δημιουργία τουριστικών πακέτων. Οι χρήστες του εργαλείου αυτού διαχωρίζονται σε τρεις κατηγορίες με συγκεκριμένα καθήκοντα. Πρώτον, στους διαχειριστές του συστήματος (administrators) οι οποίοι είναι υπεύθυνοι για τον συντονισμό και την εποπτεία της διαδικασίας παραγωγής πακέτων, δεύτερον στους εταιρικούς συνεργάτες (business partners) οι οποίοι παρέχουν τους πόρους που διαθέτουν σε ένα πακέτο και τέλος, στους καταναλωτές του πακέτου, τους τελικούς χρήστες.

Με το συντονισμό του διαχειριστή, η κοινότητα διαμορφώνει μία ομάδα εργασίας (work group) η οποία έπειτα από την απαραίτητη ανταλλαγή απόψεων και θέσπιση κανόνων, είναι σε θέση να προσφέρει τα νέα τουριστικά πακέτα που παρήγαγε. Η δημιουργία των πακέτων αυτών πραγματοποιείται μέσω της συνεργασίας των εταιρικών συνεργατών που αποτελούν την ομάδα. Στο τέλος το παραγόμενο προϊόν είναι σε θέση να διαμορφωθεί κατάλληλα ανάλογα με τις προτιμήσεις του τελικού χρήστη.

Η εφαρμογή που υλοποιήθηκε είναι σε θέση να παρέχει όλη εκείνη την υποστήριξη που χρειάζεται η δημιουργία ενός τουριστικού πακέτου (π.χ. τον καθορισμό βασικών χαρακτηριστικών του πακέτου, την επεξεργασία των δραστηριοτήτων που αποτελούν ένα πακέτο κ.α.). Επίσης δόθηκε η δυνατότητα επισκόπησης της διαδικασίας δημιουργίας μέσα από δύο διαφορετικές σκοπιές, της ομάδας και του πακέτου. Από την σκοπιά του πακέτου μπορούμε να παρατηρήσουμε ή να τροποποιήσουμε χαρακτηριστικά που αφορούν το πακέτο και της δραστηριότητες που το αποτελούν, ενώ από την σκοπιά της ομάδας μπορούμε να διακρίνουμε τα στάδια από τα οποία περνά μία ομάδα κατά την διάρκεια της δημιουργίας του πακέτου.

Η διεπαφές του εργαλείου αυτού υλοποιήθηκαν σε JAVA και κάνοντας την χρήση της βιβλιοθήκης διαδραστικών αντικειμένων του Swing. Επίσης, για την δημιουργία κάποιων γραφικών αναπαραστάσεων χρησιμοποιήθηκε η βιβλιοθήκη JGraph. Τα σύνθετα διαδραστικά συστατικά που χρησιμοποιήθηκαν για την κατασκευή των διεπαφών του εργαλείου που αναπτύχθηκε, παρουσιάζουμε στις παρακάτω ενότητες.

5.2 Παραδείγματα επαύξησης (augmentation)

Παρακάτω θα αναλύσουμε την υλοποίηση δύο διαδραστικών συστατικών της βιβλιοθήκης Swing τα οποία εμπλουτίσαμε με αυξημένη (augmented) λειτουργικότητα. Στην πρώτη περίπτωση, ένα απλό `JTree` μετατράπηκε σε ένα συστατικό που μπορεί να προσφέρει δυνατότητα μονής ή / και πολλαπλής επιλογής, ενώ στην δεύτερη περίπτωση, σε ένα `JTabbedPane` (το οποίο προσφέρει δυνατότητα φιλοξενίας ενός ή περισσοτέρων `JPanel` αντικειμένων σε μορφή Tab) υλοποιήθηκε η λειτουργικότητα που υποστηρίζει την αφαίρεση των Tab με το κατάλληλο γεγονός που πυροδοτεί ο χρήστης. Το `JTabbedPane` με την νέα αυτή λειτουργικότητα δεν υποστηριζόταν από την βιβλιοθήκη μέχρι και την έκδοση 1.5 της JAVA. Στην τελευταία έκδοση 1.6 της JAVA υλοποιήθηκε η νέα αυτή λειτουργία του `JTabbedPane`. Όσον αφορά το `JTree` αντικείμενο που θα παρουσιάσουμε παρακάτω, η δυνατότητα που παρέχει δε προσφέρεται από κανένα άλλο αντικείμενο της βιβλιοθήκης.

5.2.1 RadioCheckBoxTree

Η νέα λειτουργικότητα που προσθέσαμε στο αντικείμενο `JTree`, προέκυψε από την ανάγκη μας να μπορούμε με ένα απλό τρόπο να επιλέγουμε μία κατηγορία από το σύνολο των διαθέσιμων κατηγοριών, με έναν όμως περιορισμό. Ο χρήστης να μπορεί να επιλέγει από μία δενδροειδή αναπαράσταση των κατηγοριών, μία ή περισσότερες κατηγορίες οι οποίες βρίσκονται στο τελευταίο επίπεδο (φύλλα) του δέντρου. Αυτό επιτρέπει μία επισκόπηση των γενικών κατηγόριων, επισκόπηση των όποιων υποκατηγοριών και επιλογή μόνο των κατηγοριών εκείνων που δεν είναι υπερκατηγορίες κάποιων άλλων.

Έστω ότι έχουμε την δενδροειδή μορφή που φαίνεται παρακάτω

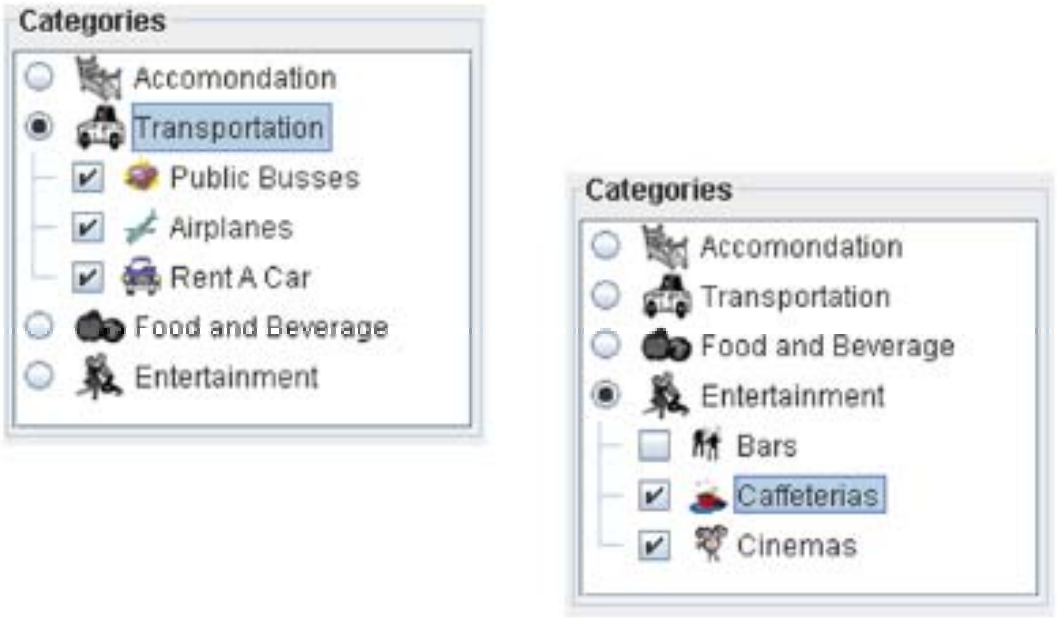
- *Μεταφορές*
  - *Μέσω αέρα*
    - *Olympic Airways*
    - *Easy Jet*
  - *Μέσω θαλάσσης*
    - *Minoan Lines*
    - *Flying Dolphins*

και ο χρηστής μπορεί να επιλέξει μόνο μία υποκατηγορία μεταφορών (μέσω αέρα ή μέσω θαλάσσης) και όσες υποκατηγορίες της κατηγόριας που επέλεξε είναι διαθέσιμες. Έτσι λοιπόν, αν επέλεξε μεταφορά μέσω θαλάσσης θα έχει δυνατότητα να επιλέξει μία ή περισσότερες από τις επιλογές που προσφέρει αυτή η κατηγόρια. Η διαδρομή (Tree path) που θα έχει επιλέξει, αν διάλεξε για παράδειγμα μεταφορές με Minoan Lines θα είναι 'Μεταφορές' – 'Μέσω θαλάσσης' – 'Minoan Lines'. Αυτό μεταφράζεται σε επιλογή της κατηγορίας 'Minoan Lines' μόνο. Αν ο χρήστης είχε επιλέξει και τις δύο διαθέσιμες υποκατηγορίες τότε αυτό θα μεταφραζόταν σε επιλογή δύο κατηγοριών, της 'Minoan Lines' και της 'Flying Dolphins'.

Όπως φαίνεται στο παραπάνω παράδειγμα υπάρχει η ανάγκη για μονή προεπιλογή η οποία ακολουθείται από πολλαπλή επιλογή. Μία τέτοια δυνατότητα δεν υποστηρίζετε από κάποιο αντικείμενο της βιβλιοθήκης με τρόπο όπου να μην είναι γνωστό το βάθος των κατηγοριών ή να μην ξέρουμε εξ αρχής ποιες κατηγόριες δεν έχουν υποκατηγορίες. Ένα δέντρο όμως μπορεί να υλοποιηθεί με τέτοιο τρόπο ώστε το βάθος του να είναι διαφορετικό για διαφορετικά μονοπάτια. Έτσι, αν ένας κόμβος του δέντρου έχει παιδιά άλλους κόμβους και όλοι αυτοί οι κόμβοι-παιδία του αρχικού εκτός από έναν, έχουν και αυτοί με την σειρά τους παιδιά, τότε αυτό το μονοπάτι του δέντρου που περιέχει τον

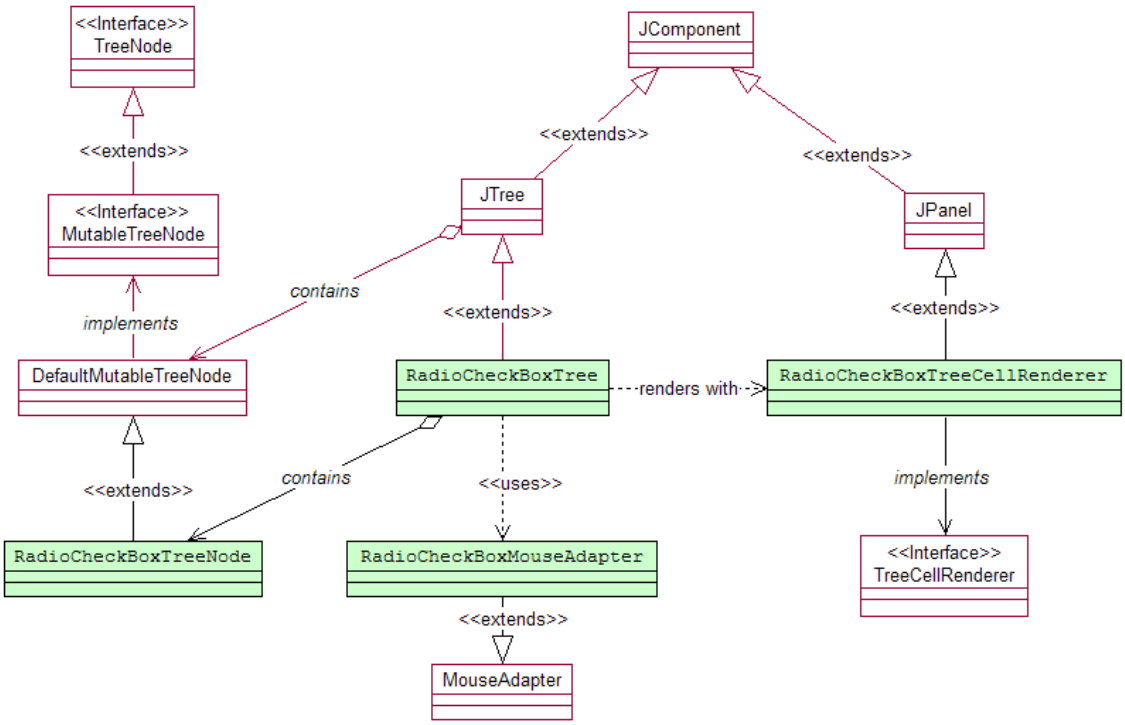
κόμβο χωρίς παιδία, θα έχει μικρότερο βάθος από τα υπόλοιπα μονοπάτια. Η υλοποίηση με την χρήση δενδροειδής μορφής μπορεί να αντιμετωπίσει τέτοιου είδους ζητήματα.

Στην περίπτωση του eKoNES, ο διαχειριστής του συστήματος για να εισάγει μία δραστηριότητα ενός πακέτου πρέπει, εκτός από την συμπλήρωση πεδίων όπως ώρα έναρξης, ώρα λήξης, περιγραφή κ.α., να επιλέξει και τις κατηγορίες των εταίρων που θα συμμετέχουν στην διαμόρφωση και υποστήριξη αυτής της δραστηριότητας. Ένας εταιρικός συνεργάτης μπορεί να ανήκει σε μία ή περισσότερες κατηγορίες – φύλλα του συνόλου των κατηγοριών και όχι σε κάποια γενική κατηγορία. Μία δραστηριότητα μπορεί να περιέχει κατηγορίες – φύλλα που ανήκουν σε μία μόνο γενική κατηγορία. Ο τύπος της δραστηριότητας ορίζεται από την κατηγορία που βρίσκεται πρώτη στο tree path της επιλεγμένης διαδρομής. Έτσι λοιπόν ο διαχειριστής του eKoNES, έχοντας να επιλέξει ανάμεσα από κατηγορίες παρόχων υπηρεσιών, διευκολύνεται από την χρήση του augmented `JTree` και της νέας διαδραστικότητας που προσφέρει. Στην Εικόνα 21 βλέπουμε το νέο αντικείμενο που υλοποιήθηκε.



Εικόνα 21: RadioCheckBoxTree

Για να εισάγουμε αυτή την νέα λειτουργικότητα στο `JTree` έπρεπε να δημιουργήσουμε κάποιες νέες κλάσεις, κληρονομώντας τις υπάρχουσες κλάσεις που υλοποιούν το αντικείμενο. Συνολικά δημιουργήσαμε τέσσερις νέες κλάσεις οι οποίες φαίνονται στην Εικόνα 22. Η βασική κλάση που υλοποιήσαμε είναι η `RadioCheckBoxTree` η οποία κάνει extends την κλάση `JTree` του Swing και είναι αυτή που κάνει instantiate το αντικείμενο. Σε αντιστοιχία με την κλάση `DefaultMutableTreeNode` δημιουργήσαμε την κλάση `RadioCheckBoxTreeNode` η οποία υλοποιεί τους κόμβους του δέντρου. Επίσης, αντί για την χρήση του `DefaultTreeCellRenderer` έγινε η υλοποίηση του `RadioCheckBoxTreeCellRenderer` που μας δίνει την δυνατότητα να κάνουμε render σε διαφορικό αντικείμενο τον κάθε κόμβο. Τέλος, υλοποιήσαμε την κλάση `RadioCheckBoxMouseAdapter` η οποία αναλαμβάνει να διαχειρίζεται τα γεγονότα που συμβαίνουν στο `RadioCheckBoxTree` και να δρα κατάλληλα στην κατάσταση του.



Εικόνα 22: Διάγραμμα κλάσεων του RadioCheckBoxTree

Η κλάση `RadioCheckBoxTreeNode` κληρονομεί την κλάση `DefaultMutableTreeNode` και επιπλέον εισάγει μία νέα μεταβλητή που δίνει την δυνατότητα να κρατάμε την πληροφορία αν ο συγκεκριμένος κόμβος είναι επιλεγμένος (selected) ή όχι. Φυσικά υλοποιήθηκαν και οι κατάλληλες μέθοδοι που σετάρουν την μεταβλητή και παίρνουν την τιμή της. Επίσης υπάρχουν οι μέθοδοι που θέτουν τον τύπο του κόμβου αυτού ως radio button κόμβο ή check box κόμβο. Δόθηκε και η δυνατότητα ένας κόμβος να περιέχει ένα `ImageIcon` για λόγους αισθητικούς.

Επίσης υλοποιήσαμε έναν νέο cell renderer. Αυτό που κάνει ένας cell renderer είναι να διαβάζει ένα κελί και να επιστρέφει τον εαυτό του. Έτσι αν θέλουμε ένα κελί να περιέχει ένα αντικείμενο της επιλογής μας, θα πρέπει ο cell renderer να κληρονομεί το αντικείμενο αυτό. Ο `RadioCheckBoxCellRenderer` κάνει implement το interface `TreeCellRenderer` και κληρονομεί την κλάση `JPanel`. Αυτό έγινε γιατί η κλάση `DefaultTreeCellRenderer` δεν επαρκούσε για την προσθήκη των radio button και check box συστατικών, μιας και κληρονομούσε το `JLabel`. Επίσης έγινε override η μέθοδος getTreeCellRendererComponent() η οποία επιστρέφει ένα `JPanel` στο οποίο έχει προστεθεί, ανάλογα με τον τύπο του κόμβου, ένα `JRadioButton` ή ένα `JCheckBox` αντικείμενο. Σε αυτό το `JPanel` προστίθενται εκτός από τα προηγούμενα αντικείμενα και ένα `JLabel` που περιέχει το αναγκαίο κείμενο. Φυσικά προκύπτει και η ανάγκη για την σωστή τοποθέτηση των αντικειμένων πάνω στο `JPanel` και για αυτόν τον λόγο γίνεται override η μέθοδος doLayout() ορίζοντας τον επιθυμητό τρόπο τοποθέτησης.

Η υλοποίηση ενός νέου mouse adapter έγινε με σκοπό τα όποια γεγονότα συμβαίνουν πάνω στο `RadioCheckBoxTree` να μεταφράζονται κατάλληλα, πρώτον, αλλάζοντας την κατάσταση του κόμβου σε επιλεγμένο ή όχι, και δεύτερον, να γίνεται το απαραίτητο expand ή collapse στο δέντρο. Η κλάση `RadioCheckBoxMouseAdapter` έγινε κληρονομώντας την κλάση `MouseAdapter` και ξαναγράφοντας την μέθοδο mousePressed() στην οποία υλοποιείται η όλη λειτουργικότητα. Έτσι λοιπόν, αν το

γεγονός του mouse προέρχεται από έναν κόμβο ο οποίος είναι radio button κόμβος, θα μεταφραστεί διαφορετικά από ένα γεγονός που συνέβη σε έναν check box κόμβο. Για παράδειγμα αν σε ένα επίπεδο έχουμε radio button κόμβους τότε η επιλογή του ενός σημαίνει αυτόματη αποεπιλογή των υπολοίπων. Αν αποεπιλεγεί ένας οποιοσδήποτε κόμβος, όλοι οι κόμβοι παιδία του θα αποεπιλεγούν και αυτά, όπως επίσης αν όλα τα παιδιά ενός κόμβου δεν είναι επιλεγμένα τότε και ο κόμβος πατέρας δεν μπορεί να είναι επιλεγμένος.

RadioCheckBoxTree	
addMouseListener(**RadioCheckBoxMouseAdapter**)	Μέθοδος που κληρονομείται από το JTree
setCellRenderer(**RadioCheckBoxTreeCellRenderer**)	Μέθοδος που κληρονομείται από το JTree
RadioCheckBoxTreeNode	
boolean isSelected	Νέα μεταβλητή για την υποστήριξη επιλογής
setAsRadioNode()	Νέα μέθοδος που χαρακτηρίζει ένα κόμβο σαν radio button κόμβο
setAsCheckBoxNode()	Νέα μέθοδος που χαρακτηρίζει ένα κόμβο σαν check box κόμβο
setRadioCheckBoxNodeIcon(**ImageIcon**)	Νέα μέθοδος για τον καθορισμό ενός ImageIcon
RadioCheckBoxTreeCellRenderer	
extends JPanel	Κάνει render κάθε κελί σε JPanel (έτσι ώστε να υποστηρίξει επιπλέον αντικείμενα)

Component getTreeCellRendererComponent(){
if (node.isRadioNode()){
leadPanel.add(**JRadioButton**)
}else if(node.isCheckBox()){
leadPanel.add(**JCheckBox**)
}
this.add(leadPanel);
return this;
}
	Παράκαμψη μεθόδου από το interface TreeCellRenderer η οποία επιστρέφει ένα JPanel για κάθε κόμβο

getPreferredSize()	Παράκαμψη μεθόδου του JPanel για υπολογισμό του νέου μεγέθους του Tree Cell
doLayout()	Παράκαμψη μεθόδου του JPanel για σωστή τοποθέτηση των υπολοίπων αντικειμένων
RadioCheckBoxTreeMouseAdapter	

mousePressed(MouseEvent){
if(mouseEvent.getButton()==mouseEvent.*LEFT*){
If(node.isRadioNode()){
//Χειρισμός του γεγονότος
}else if(node.isCheckBoxNode()){
//Χειρισμός του γεγονότος
}
}
}
	Παράκαμψη μεθόδου της MouseAdapter που κάνει διαχείριση των συμβάντων

Εικόνα 23: Σύνοψη των κλάσεων και των μεθόδων που υλοποιούν το RadioCheckBoxTree

Τέλος, η κλάση `RadioCheckBoxTree` κληρονομεί την κλάση `JTree` του Swing. Η κλάση αυτή δημιουργήθηκε για ευκολότερη δημιουργία του αντικειμένου. Έτσι ο

ορισμός του `RadioCheckBoxMouseAdapter` γίνεται εδώ, όπως επίσης και ο ορισμός του νέου cell renderer με τις μεθόδους addMouseListener() και setCellRenderer() αντίστοιχα. Στην Εικόνα 23 φαίνονται συνοπτικά ποιες νέες μέθοδοι υλοποιήθηκαν σε κάθε κλάση, ποιες μέθοδοι που υπήρχαν ήδη έγιναν override και ποιες νέες μεταβλητές προστέθηκαν.

Για να υλοποιηθεί ένα αντικείμενο τύπου `RadioCheckBoxTree` πρέπει καταρχήν να δημιουργήσουμε τον κεντρικό κόμβο (root) με τους επιθυμητούς κόμβους-παιδία του και να δώσουμε ως όρισμα στον κατασκευαστή του `RadioCheckBoxTree` αυτόν τον κεντρικό κόμβο. Έτσι λοιπόν για να δημιουργήσουμε ένα `RadioChechBoxTreeNode` γράφουμε:

```
RadioCheckBoxTreeNode node = new RadioCheckBoxTreeNode();
node.add(new RadioCheckBoxTreeNode());
```

κατόπιν για να υλοποιηθεί το δέντρο καλούμε τον κατασκευαστή (constructor) του `RadioCheckBoxTree` γράφοντας:

```
RadioCheckBoxTree tree = new RadioCheckBoxTree(node)
```

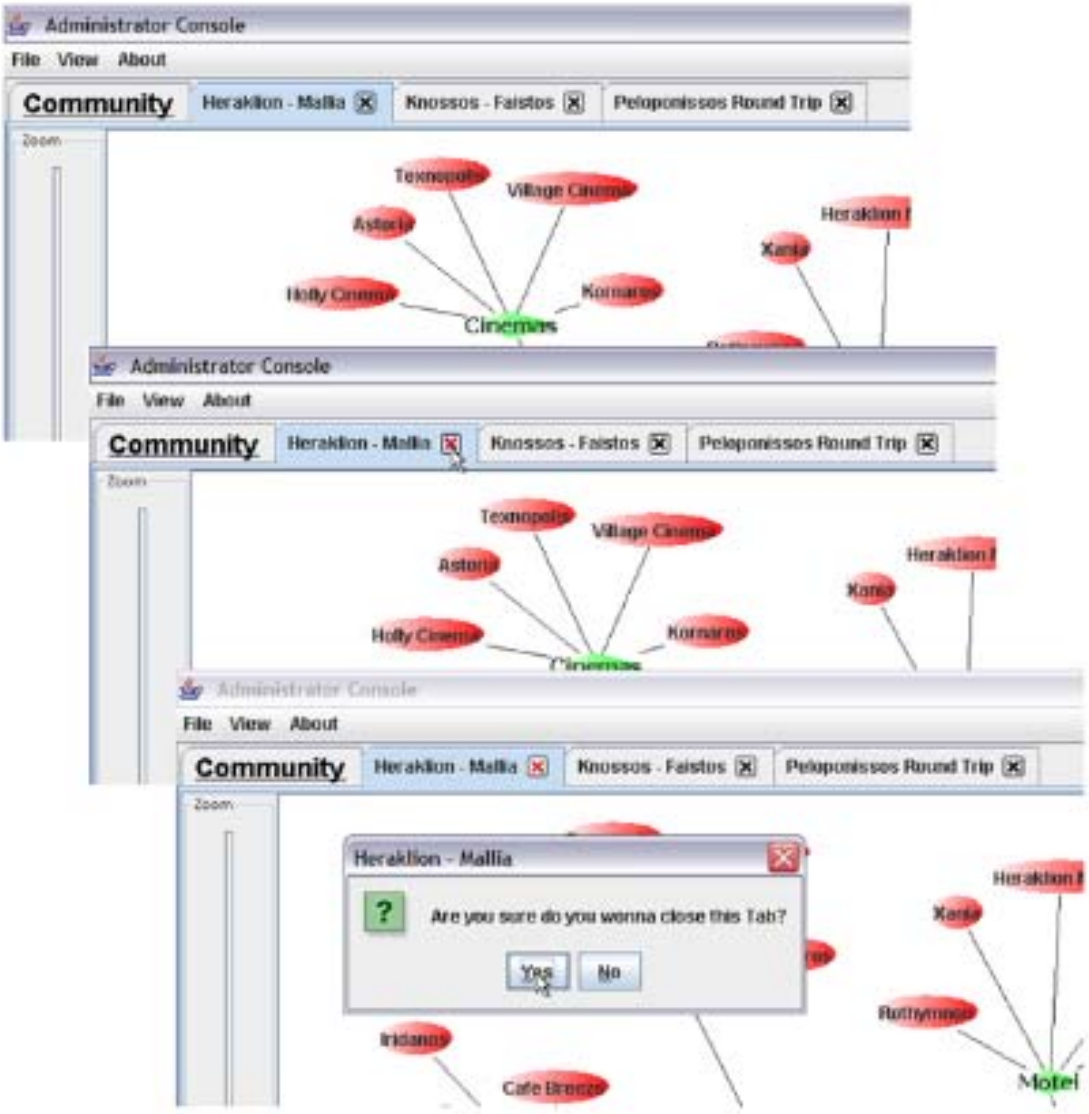
Αξίζει να σημειωθεί ότι την ίδια διαδικασία θα ακολουθούσαμε αν θέλαμε να δημιουργήσουμε ένα απλό `JTree`.

5.2.2 ClosableTabbedPane

Η νέα λειτουργικότητα που προσθέσαμε στο `JTabbedPane` μας επιτρέπει να έχουμε ένα χώρο ο οποίος μπορεί να περιέχει αρκετά πάνελ και δίνει την δυνατότητα στον χρήστη να αφαιρεί τα πάνελ αυτά. Στην περίπτωση του eKoNEΣ τόσο ο διαχειριστής όσο και ο εταιρικός συνεργάτης έχουν την δυνατότητα να κάνουν μία επισκόπηση του πακέτου μέσα από ένα πάνελ που περιέχεται στο χώρο αυτόν. Ένας χρήστης λοιπόν με την χρήση του αντικειμένου αυτού μπορεί να βλέπει περισσότερα του ενός πακέτα (δηλαδή, πάνελ που περιέχουν πακέτα) και έχει την δυνατότητα να αφαιρεί όποια πακέτα επιλέξει. Στην Εικόνα 24 φαίνεται το αντικείμενο με την νέα λειτουργικότητα που προστέθηκε.

Ο χρήστης μπορεί να επιλέγει τα πακέτα που θέλει να προσθέσει στο `ClosableTabbedPane` από μία λίστα που περιέχει όλα τα πακέτα, κάνοντας διπλό κλικ στα αντικείμενα της λίστας αυτής. Το πακέτο ανοίγει σε ένα πάνελ που προστίθεται στο `ClosableTabbedPane`. Ο χρήστης κάνοντας κλικ στο εικονίδιο που υπάρχει πάνω στο Tab μπορεί να αφαιρέσει το συγκεκριμένο πάνελ από το `ClosableTabbedPane`, δηλαδή να 'κλείσει' το πακέτο.

Η υλοποίηση του `ClosableTabbedPane` έγινε κληρονομώντας την κλάση `JTabbedPane` του Swing. Υλοποιήθηκε ένας mouse listener ο οποίος αναλαμβάνει να διαχειριστεί το γεγονός κλεισίματος ενός πάνελ και επίσης δημιουργήθηκαν κάποιες κλάσεις για την υποστήριξη των UIManager, basic και metal. Τέλος, δημιουργήθηκε και μία κλάση που υλοποιεί το εικονίδιο που εμφανίζεται πάνω στο Tab, το οποίο παίρνει τρεις μορφές `NORMAL`, `HOVER` και `PRESSED`. Στην Εικόνα 25 μπορούμε να δούμε τις κλάσεις που δημιουργήθηκαν και την ιεραρχία τους.



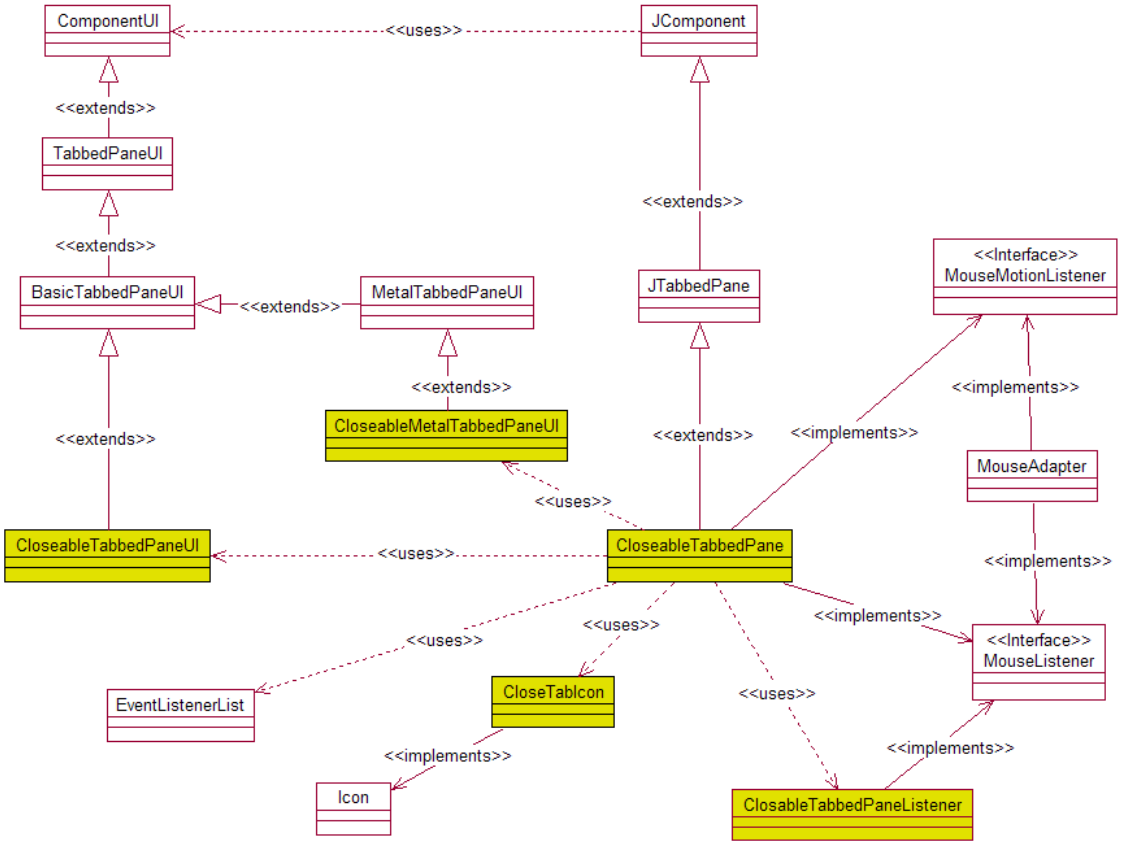
Εικόνα 24: ClosableTabbedPane

Η κλάση `ClosableTabbedPaneListener` υλοποιεί το interface `MouseListener` και σκοπός αυτής της κλάσης είναι να προσφέρει στον προγραμματιστή που θέλει να χρησιμοποιήσει το `ClosableTabbedPane`, την δυνατότητα να διαχειριστεί με τον τρόπο που θέλει την αφαίρεση ενός πάνελ. Εμείς κατά την αφαίρεση ενός πάνελ από το `ClosableTabbedPane` έχουμε επιλέξει να εμφανίζεται ένα `JOptionPane` για την επιβεβαίωση της ενέργειας του χρήστη. Το πάνελ αφαιρείται μόνο αν ο χρήστης επιλέξει την επιλογή `YES`.

Οι κλάσεις `ClosaableTabbedPaneUI` και `ClosableMetalTabbedPaneUI` έχουν υλοποιηθεί κληρονομώντας τις κλάσεις `TabbedPaneUI` και `MetalTabbedPaneUI` αντίστοιχα. Αυτές οι κλάσεις χρησιμοποιούνται από το `ClosableTabbedPane` ώστε να υποστηρίζονται δύο look and feel της πλατφόρμας, basic και metal. Στα πλαίσια του eKoNEΣ δεν υλοποιήσαμε τις αντίστοιχες κλάσεις για την υποστήριξη και άλλων look and feel.

Η κλάση `CloseTabIcon` υλοποιεί το interface `Icon` με σκοπό την δημιουργία ενός εικονιδίου το οποίο τοποθετείται πάνω στο Tab και στο οποίο, αν ο χρήστης επιλέξει να

κάνει κλικ, αφαιρείται το συγκεκριμένο Tab από το `ClosableTabbedPane`. Το ιδιαίτερο χαρακτηριστικό αυτού του εικονιδίου είναι ότι σχεδιάζεται ανάλογα με την θέση που βρίσκεται το mouse και αν είναι πατημένο ή όχι. Έτσι το εικονίδιο αυτό παίρνει τρεις μορφές `NORMAL`, `HOVER` και `PRESSED`.



Εικόνα 25: Διάγραμμα κλάσεων του ClosableTabbedPane

Τέλος, η κλάση που κάνει χρήση όλων των παραπάνω κλάσεων είναι φυσικά η `ClosableTabbedPane` που υλοποιεί και το αντικείμενο επαυξημένης λειτουργικότητας. Η κλάση αυτή κληρονομεί το `JTabbedPane` του Swing και υλοποιεί δύο interface, το `MouseListener` και το `MouseMotionListener`. Η υλοποίηση των δύο interface γίνεται με σκοπό να ερμηνευτούν με τον κατάλληλο τρόπο οι κινήσεις και οι ενέργειες του χρήστη με το ποντίκι, ιδιαίτερα όταν βρίσκεται πάνω στην περιοχή στην οποία σχεδιάζεται το εικονίδιο που αφαιρεί ένα πάνελ από το αντικείμενο. Τα γεγονότα λοιπόν που συμβαίνουν πάνω σε αυτήν την περιοχή είναι αυτά που μεταφράζοντας τα κατάλληλα επέρχονται οι κατάλληλες αλλαγές στην κατάσταση του αντικειμένου. Όλα τα γεγονότα του mouse τα διαχειρίζεται μία νέα μέθοδος, η processMouseEvents().

Η κλάση `ClosableTabbedPane` παρέχει τους κατάλληλους κατασκευαστές για την εύκολη χρήση της από τον προγραμματιστή. Μία επιπλέον μέθοδος έχει υλοποιηθεί η οποία επιτρέπει την εισαγωγή ενός πάνελ στο αντικείμενο με μία όμως έξτρα παράμετρο, το αν το πάνελ είναι δυνατό να αφαιρείται ή όχι από το αντικείμενο. Επίσης, η μέθοδος setCloseIcons(), η οποία παίρνει σαν ορίσματα τρία εικονίδια, υποστηρίζει την εισαγωγή άλλων εικονιδίων για την λειτουργία αφαίρεσης των πάνελ από το `ClosableTabbedPane`. Στην Εικόνα 26 μπορούμε να δούμε μία σύνοψη των κλάσεων και των μεθόδων που υλοποιήθηκαν.

Για να δημιουργήσει ένας προγραμματιστής ένα `ClosableTabbedPane` αντικείμενο αρκεί να καλέσει τον πιο απλό κατασκευαστή της κλάσης. Θα πρέπει να γράψει:

```
ClosableTabbedPane pane = new ClosableTabbedPane()
```

ενώ για την εισαγωγή πάνελ με την δυνατότητα αφαίρεσης από το αντικείμενο που δημιουργήσαμε γράφει:

```
pane.addTab("Tab 1", new JPanel(), true)
```

ClosableTabbedPane	
addTab(**String**, **Component**, **boolean**)	Μέθοδος για την εισαγωγή ενός πάνελ στο αντικείμενο
processMouseEvents()	Μέθοδος για την διαχείριση των γεγονότων του mouse
ClosableTabbedPaneListener	
closeTab()	Η μέθοδος αυτή διαχειρίζεται το γεγονός αφαίρεσης ενός πάνελ
CloseIcon	
paintIcon()	Μέθοδος του interface Icon η οποία σχεδιάζει το εικονίδιο
ClosableTabbedPaneUI και ClosableMetalTabbedPaneUI	
layoutLabel()	Παράκαμψη μεθόδου για της υπερκλάσης για σωστή τοποθέτηση του Label πάνω στο Tab

Εικόνα 26: Σύνοψη κλάσεων και μεθόδων που υλοποιούν το ClosableTabbedPane

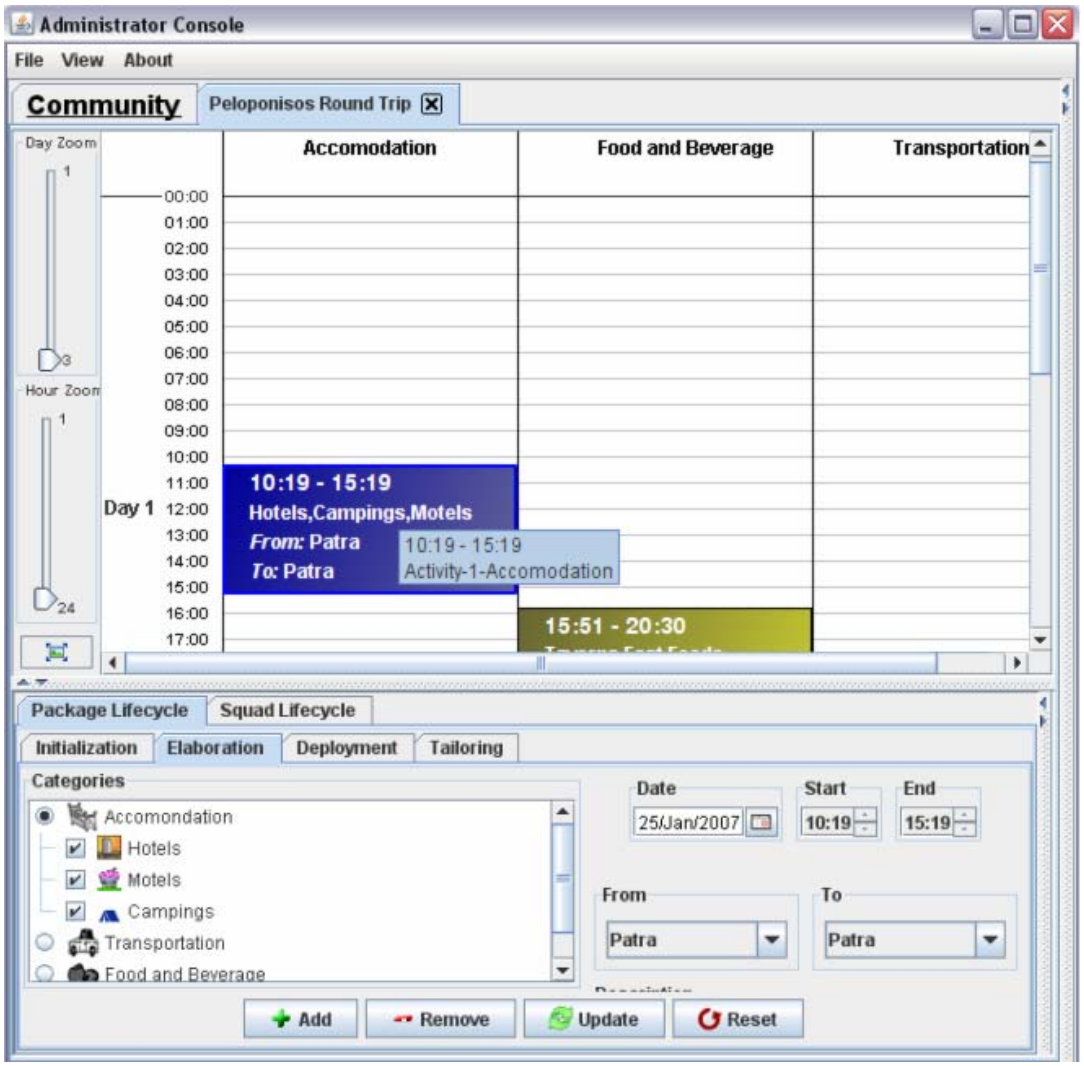
5.3 Παραδείγματα επέκτασης (expansion)

Παρακάτω θα παρουσιάσουμε ένα αντικείμενο το οποίο υλοποιήθηκε επεκτείνοντας την βιβλιοθήκη Swing. Η λειτουργικότητα που χρειαζόμασταν δεν υποστηριζόταν πριν, από κανένα συστατικό της βιβλιοθήκης και για αυτόν τον λόγο θα έπρεπε να υλοποιηθεί ένα νέο αντικείμενο. Σκοπός του αντικειμένου ήταν να δώσει την δυνατότητα χρονοπρογραμματισμού των δραστηριοτήτων (activities) ενός πακέτου με έναν γραφικό τρόπο. Τα συστατικά που υλοποιούν το νέο διαδραστικό αντικείμενο, κληρονομούν κάποιες βασικές κλάσεις όπως την κλάση `JPanel` και την κλάση `JButton`. Το νέο αυτό αντικείμενο προσφέρει δύο τρόπους παρουσίασης της πληροφορίας, ο ένας από τους δύο τρόπους ενδείκνυται για την χρήση του σε διεπαφές λιγότερο έμπειρων χρηστών. Έτσι, ενώ υποστηρίζεται η εναλλαγή μεταξύ των δύο αυτών διαφορετικών τρόπων προβολής, για την παρουσίαση του προγραμματισμού του πακέτου, επιλέχθηκε να χρησιμοποιηθεί η μία εκδοχή της παρουσίασης στην διεπαφή του διαχειριστή και η άλλη για την διεπαφή του εταιρικού συνεργάτη. Οι διαφορά των δύο αυτών παρουσιάσεων έγκειται στον τρόπο ταξινόμησης των δραστηριοτήτων στην οθόνη του χρήστη. Στη μία περίπτωση οι δραστηριότητες είναι χωρισμένες ανά κατηγορία και στην άλλη περίπτωση χρησιμοποιείται η μεταφορά του τηλεοπτικού προγράμματος (TV program metaphor).

5.3.1 ActivityPanel (πάνελ δραστηριοτήτων)

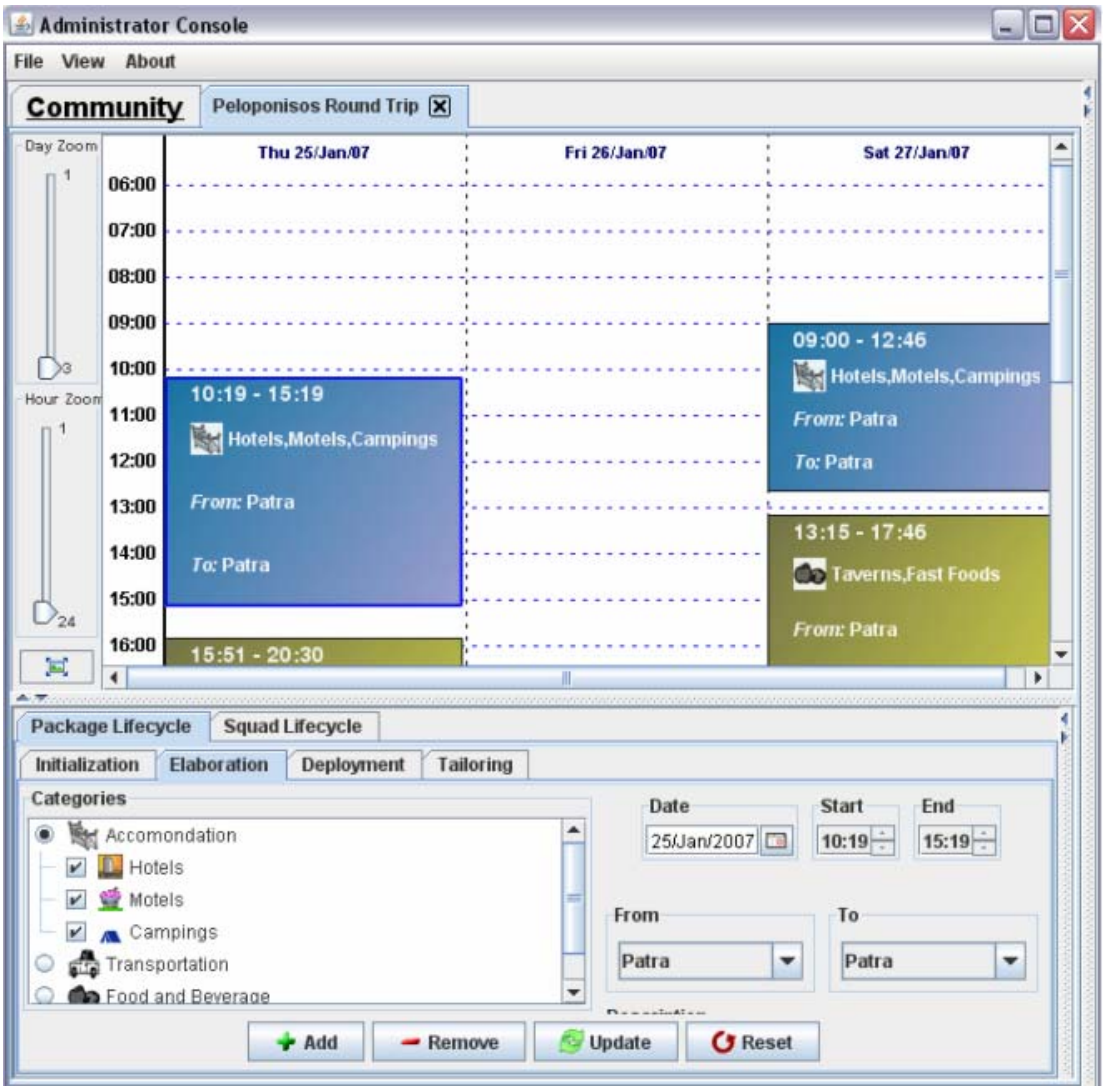
Ένα πακέτο του eKoNES αποτελείται από πολλές δραστηριότητες τις οποίες έχουν καθήκον να πλαισιώσουν οι εταιρικοί συνεργάτες. Για να υλοποιηθεί λοιπόν ένα πακέτο θα πρέπει να γίνει ένας χρονοπρογραμματισμός των δραστηριοτήτων που το αποτελούν. Αυτός ο προγραμματισμός είναι καθήκον του διαχειριστή ο οποίος, σε συνεργασία με τους εταίρους, πρέπει να κατανείμει κατάλληλα χρονικά τις δραστηριότητες σε ένα πακέτο. Η εισαγωγή των δραστηριοτήτων έπρεπε να υποστηριχθεί από έναν γραφικό τρόπο απεικόνισης της δομής του πακέτου έτσι ώστε να γίνεται εφικτή η καλύτερη δυνατή επισκόπηση της συνολικής δομής του πακέτου.

Μία δραστηριότητα καθορίζεται από την ημερομηνία έναρξης και λήξης της, και φυσικά από τον τύπο της (σε ποια γενική κατηγόρια ανήκει). Με δεδομένα τα παραπάνω χαρακτηριστικά μίας δραστηριότητας, ένας γράφος θα έπρεπε να μας δίνει την θέση της δραστηριότητας στα χρονικά όρια του πακέτου. Επίσης, θεωρήθηκε χρήσιμο να μπορούμε να έχουμε μία επισκόπηση των γενικών κατηγοριών που υπάρχουν στο πακέτο μας. Έτσι δημιουργήσαμε έναν γράφο ο οποίος μας παρουσίαζε στον οριζόντιο άξονα τις διαφορετικές κατηγορίες και στον κάθετο άξονα τον χρόνο. Ο διαχειριστής του eKoNES έχει την δυνατότητα να γνωρίζει τι είδους δραστηριότητες έχει προσθέσει στο πακέτο και σε ποιες ημέρες.



Εικόνα 27: ActivityPanel

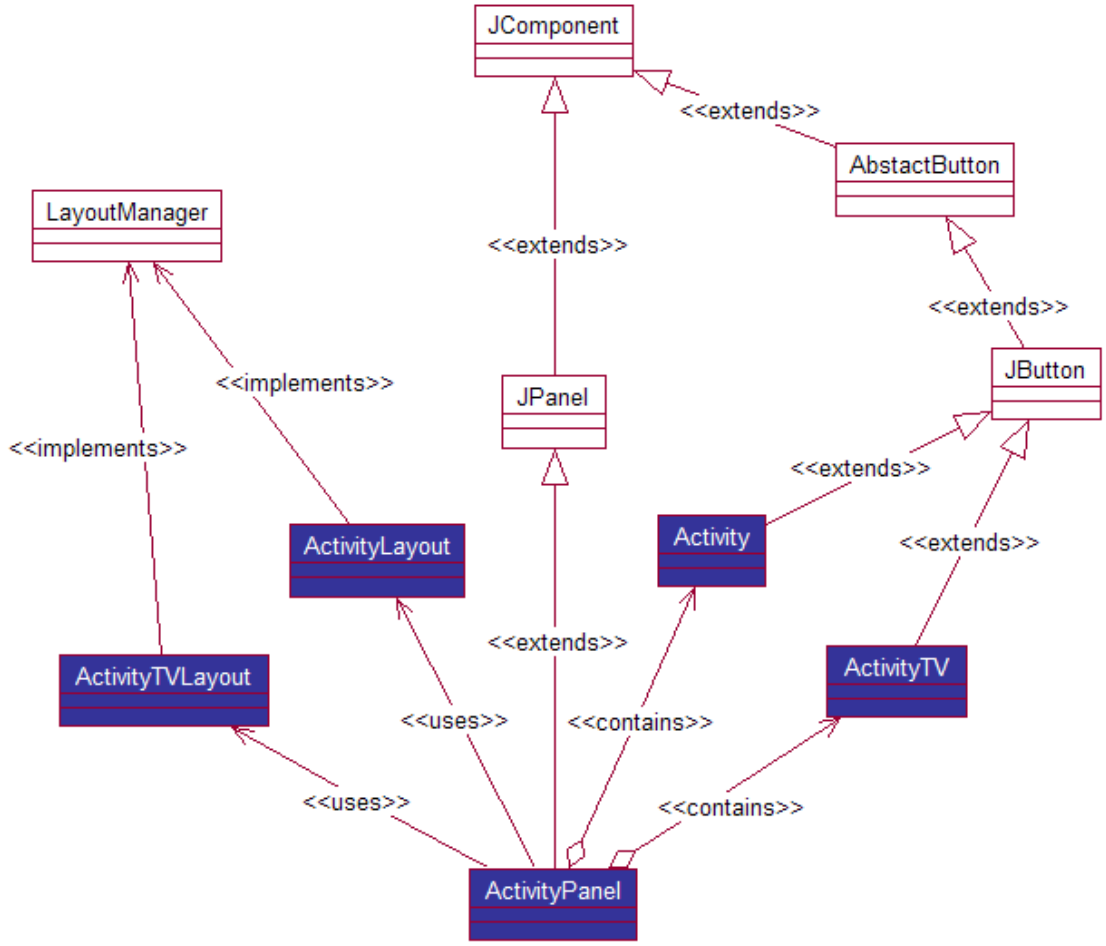
Ο παραπάνω τρόπος προβολής του πακέτου θεωρήθηκε χρήσιμος για την διεπαφή του διαχειριστή αλλά όχι και για την διεπαφή του εταιρικού συνεργάτη. Έτσι για την διεπαφή του εταιρικού συνεργάτη προσφέρεται μία προβολή που είναι πιο απλή και κατανοητή. Η προβολή αυτή κάνει χρήση της μεταφοράς του τηλεοπτικού προγράμματος. Στο οριζόντιο άξονα υπάρχουν οι ημέρες του πακέτου και στο κάθετο άξονα οι ώρες μίας ημέρας. Αυτός ο τρόπος παρουσίασης εκτός ότι είναι πιο φιλικός στον χρήστη, δεν περιέχει και την μη χρήσιμη πληροφορία για τον εταιρικό συνεργάτη. Στην Εικόνα 27 και στην Εικόνα 28 μπορούμε να δούμε το νέο αντικείμενο που υλοποιήθηκε και τους δύο διαφορετικούς τρόπους προβολής.



Εικόνα 28: ActivityPanel (TV metaphor)

Για να υλοποιήσουμε το αντικείμενο χρειάστηκε να δημιουργήσουμε πέντε κλάσεις. Το διάγραμμα κλάσεων του `ActivityPanel` με τις νέες κλάσεις που δημιουργήθηκαν φαίνεται στην Εικόνα 29. Η βασική κλάση που υλοποιεί το αντικείμενο είναι η κλάση `ActivityPanel`. Οι κλάσεις `Activity` και `ActivityTV` περιέχουν όλα τα απαραίτητα στοιχεία της δραστηριότητας και επίσης είναι το γραφικό αντικείμενο που αναπαριστά πάνω στο `ActivityPanel` μία δραστηριότητα. Ο σκοπός των κλάσεων `ActivityLayout` και `ActivityTVLayout` είναι να τοποθετούν τα `Activity` και `ActivityTV` αντικείμενα κατάλληλα πάνω στο `ActivityPanel` ανάλογα με τον τρόπο παρουσίασης που χρησιμοποιείται.

Η κλάση `ActivityPanel` κάνει extends ένα `JPanel`. Η κλαση αυτή στην ουσία είναι ένα πάνελ στο οποίο έχει γίνει override η μέθοδος paintComponent(), η οποία οφείλεται για τον σχεδιασμό του πάνελ στην οθόνη. Παρακάμπτοντας αυτήν την μέθοδο αυτήν μπορούμε να σχεδιάσουμε το πάνελ στην μορφή που θέλουμε ανάλογα με την παρουσίαση που επιθυμούμε. Για να δημιουργήσουμε ένα `ActivityPanel` θα πρέπει να δώσουμε ως όρισμα ένα `Calendar` αντικείμενο το οποίο περιέχει την ημερομηνία έναρξης του πακέτου και ένα ακέραιο (integer) ο οποίος δίνει την συνολική διάρκεια του πακέτου σε ημέρες. Για την πρόσθεση ή την αφαίρεση `Activity` αντικειμένων υλοποιήσαμε δύο μεθόδους, την addActivity() και την removeActivity(). Επίσης υλοποιήθηκε και η μέθοδος getSelectedActivity() η οποία επιστρέφει το `Activity` αντικείμενο που είναι επιλεγμένο στο `ActivityPanel`.



Εικόνα 29: Διάγραμμα κλάσεων του ActivityPanel

Οι κλάσεις `Activity` και `ActivityTV` επιλέχθηκε να κληρονομούν την κλάση `JButton` επειδή η κλάση αυτή έχει υλοποιημένες χρήσιμες λειτουργίες όπως επιλογή (selection). Και σε αυτά τα αντικείμενα, όπως και στο `ActivityPanel`, έχει γίνει override η μέθοδος paintComponent() για λόγους παρουσίασης. Υλοποιήθηκαν μέθοδοι όπως setStartDate() και setEndDate() για τον καθορισμό της αρχικής και τελικής ημερομηνίας όπως επίσης και οι κατάλληλες μέθοδοι για να παίρνουμε τις ημερομηνίες αυτές. Επίσης δόθηκε η δυνατότητα τα αντικείμενα αυτά να περιέχουν ένα αντικείμενο καθορισμένο από τον χρήστη. Με τις μεθόδους getUserObject() και setUserObject() γίνεται η διαχείριση του αντικειμένου αυτού.

Στόχος των κλάσεων `ActivityLayout` και `ActivityTVLayout` είναι να τοποθετήσουν κατάλληλα τα αντικείμενα πάνω στο `ActivityPanel` ανάλογα με τον τρόπο παρουσίασης που χρησιμοποιείτε. Οι δύο αυτές κλάσεις υλοποιούν (implement) το interface `LayoutManager`. Ο `LayoutManager` έχει πέντε μεθόδους που θα πρέπει να υλοποιηθούν από τις κλάσεις που δημιουργήσαμε. Τις μεθόδους layoutContainer(), minimumLayoutSize(), preferredLayoutSize(), addLayoutComponent() και removeLayoutComponent(). Η πιο σημαντική μέθοδος είναι η layoutContainer() στην οποία υλοποιείται όλος ο αλγόριθμος τοποθέτησης των αντικειμένων πάνω στο `ActivityPanel`. Στην Εικόνα 30 φαίνονται συνοπτικά οι κλάσεις και οι μέθοδοι που υλοποιήθηκαν.

ActivityPanel	
paintComponent()	Παράκαμψη μεθόδου του JPanel για κατάλληλη σχεδίαση του αντικειμένου
addActivity(**Activity**)	Μέθοδος για την εισαγωγή δραστηριότητας
removeActivity(**Activity**)	Μέθοδος για την αφαίρεση δραστηριότητας
getSelectedActivity()	Μέθοδος για την επιστροφή της επιλεγμένης δραστηριότητας
setView()	Μέθοδος που ορίζει την επιθυμητή παρουσίαση
Activity ή ActivityTV	
paintComponent()	Παράκαμψη μεθόδου του JButton για κατάλληλη σχεδίαση του αντικειμένου
setStart/EndDate(**Date**)	Μέθοδος για τον καθορισμό αρχικής και τελικής ημερομηνίας
getStart/EndDate()	Μέθοδος για την επιστροφή της αρχικής και της τελικής ημερομηνίας
get/setUserObject()	Μέθοδος για την επιστροφή ενός αντικειμένου καθορισμένο από τον χρήστη
ActivityLayout και ActivityTVLayout	
layoutContainer(**Container**)	Μέθοδος που υλοποιείται από τον LayoutManager για τοποθέτηση των αντικειμένων πάνω στο πάνελ

Εικόνα 30: Σύνοψη των κλάσεων και των μεθόδων που υλοποιούν το ActivityPanel

Για να υλοποιηθεί ένα αντικείμενο τύπου `ActivityPanel` θα πρέπει να καλέσουμε τον κατασκευαστή του αντικειμένου με ορίσματα την αρχική ημερομηνία και το σύνολο των ημερών του πακέτου. Έτσι λοιπόν γράφουμε:

```
Calendar sCal = Calendar.getInstance();
int duration = 5;
ActivityPanel panel = new ActivityPanel(sCal, duration);
```

Για να προσθέσουμε ή να αφαιρέσουμε μία δραστηριότητα από το `ActivityPanel` γράφουμε αντίστοιχα:

```
Activity anActivity = new Activity();
panel.addActivity(anActivity);
panel.removeActivity(anActivity);
```

5.4 Παραδείγματα ενσωμάτωσης (integration)

Για την δημιουργία κάποιων γραφικών αναπαραστάσεων στο eKoNES κρίναμε σκόπιμο να χρησιμοποιήσουμε μία βιβλιοθήκη διαδραστικών αντικειμένων η οποία έχει υλοποιηθεί με σκοπό να προσφέρει γρήγορα και εύκολα πολλών διαφορετικών ειδών διαγράμματα. Τα διαγράμματα που θέλαμε να υλοποιήσουμε θα έπρεπε να είχαν την δυνατότητα να παρουσιάζουν με γραφικό τρόπο μεγάλες ποσότητες δεδομένων. Τέτοια δεδομένα μπορεί να είναι η συμμετοχή των εταιρικών συνεργατών στην δημιουργία ενός πακέτου ή τα μηνύματα που ανταλλάχθηκαν για την δημιουργία του πακέτου. Έτσι, δεδομένα που υπακούουν σε μία ιεραρχική δομή, έπρεπε να παρουσιαστούν με τέτοιο τρόπο ώστε να εξάγονται και τα κατάλληλα συμπεράσματα.

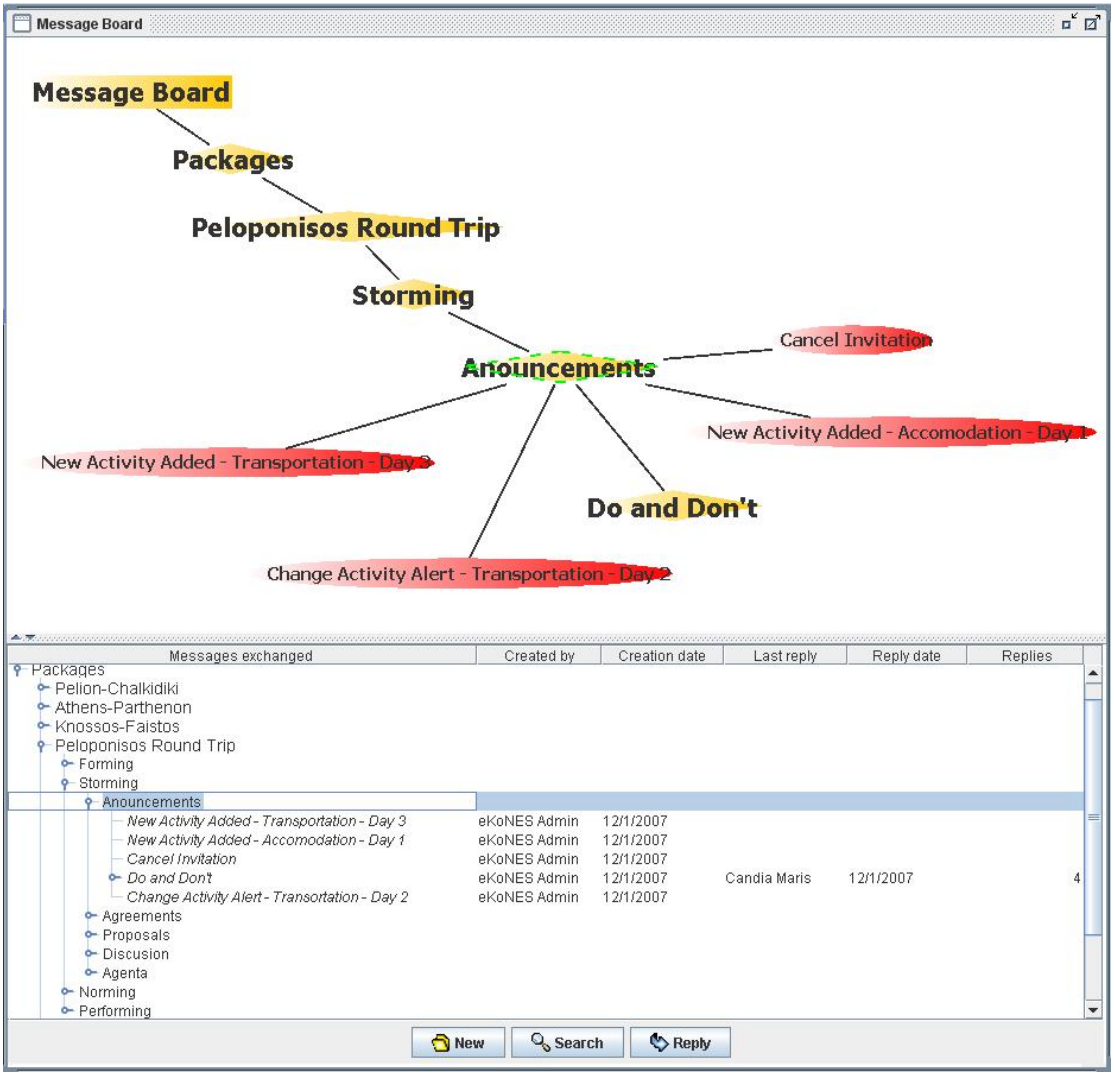
Για την δημιουργία των γραφικών αναπαραστάσεων χρησιμοποιήσαμε μία open source βιβλιοθήκη διαδραστικών αντικειμένων, την JGraph. Η βιβλιοθήκη JGraph είναι μία βιβλιοθήκη η οποία προσφέρει όλα εκείνα τα απαραίτητα συστατικά για την δημιουργία διαγραμμάτων και χρήσιμα εργαλεία για την διαχείριση τους (πχ zooming). Επίσης ενδείκνυται για παρουσιάσεις ιεραρχικών δομών δεδομένων. Στην περίπτωση του eKoNES σκοπός μας ήταν να παρουσιάσουμε τα δεδομένα που υπάρχουν σε ένα εργαλείο ανταλλαγής μηνυμάτων με διαφορετικό τρόπο από τον συνηθισμένο. Τα μηνύματα αυτά υπακούν σε μία δενδροειδή δομή και έτσι η βιβλιοθήκη JGraph κρίθηκε ιδανική λόγο των χαρακτηριστικών που προσφέρει.

Όπως φαίνεται στη Εικόνα 31, στην οποία παρουσιάζεται το message board του eKoNES, η παρουσίαση της πληροφορίας γίνεται με την χρήση ενός συνδυασμού `JTree` και `JTable` (συστατικά τα οποία και τα δύο παρέχονται από την βιβλιοθήκη Swing) και με την χρήση της τρίτης βιβλιοθήκης JGraph. Στο κάτω μέρος χρησιμοποιείται ένα `JTreeTable` το οποίο παρουσιάζει τα μηνύματα που στάλθηκαν με περαιτέρω πληροφορίες όπως δημιουργό μηνύματος, ημερομηνία δημιουργίας κ.α. Στο πάνω μέρος και με την χρήση της βιβλιοθήκης JGraph δημιουργήθηκε ένα αντικείμενο που παρουσιάζει με γραφική μορφή την ίδια πληροφορία.

Πέρα όμως από το να χρησιμοποιηθεί ένα νέο αντικείμενο για την παρουσίαση της ίδιας πληροφορίας αναγκαίο ήταν να υπάρχει και η δυνατότητα συγχρονισμού των δύο αυτών αναπαραστάσεων. Έτσι, όταν επιλέγεται ένα αντικείμενο που αναπαριστά ένα μήνυμα στον γράφο του JGraph θα πρέπει να επιλέγεται και το αντίστοιχο αντικείμενο στο `JTreeTable` και το αντίστροφο. Επίσης έπρεπε να δοθεί η δυνατότητα πλοήγησης στα μηνύματα είτε με την χρήση του JGraph αντικειμένου είτε με την χρήση του `JTreeTable` και να γίνεται αυτόματα η ανανέωση των παρουσιάσεων.

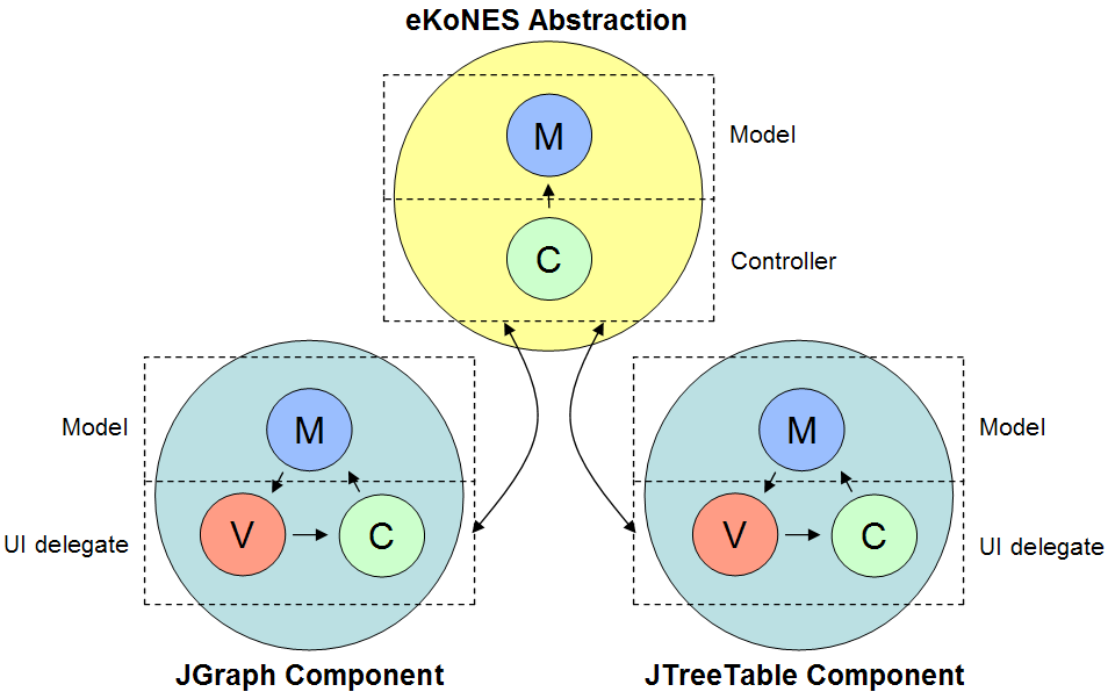
Η όλη υλοποίηση έγινε προσθέτοντας το JGraph σαν επιπλέον view (σε όρους MVC) στο model του JTreeTable αντικειμένου. Έτσι λοιπόν, έχουμε ένα μόνο μοντέλο (model) που κρατά τα δεδομένα μας και δύο όψεις (view) του μοντέλου αυτού. Αυτό επιτρέπει την παρουσίαση των δεδομένων και την επεξεργασία τους κάνοντας χρήση του ενός ή του αλλού τρόπου παρουσίασης. Αξίζει να σημειωθεί ότι όταν μιλάμε για τρόπο παρουσίασης ενός αντικειμένου δεν μιλάμε μόνο για το view του αλλά εννοούμε και τον controller του, όπως αναφέρετε και στο κεφαλαίο 'Βιβλιοθήκες γραφικών διαδραστικών αντικειμένων' και συγκεκριμένα στην παράγραφο που μιλά για το SwingTo Swing και

το αντικείμενο UI delegate. Το αντικείμενο UI delegate ενσωματώνει σε μία οντότητα το view και τον controller. Έτσι η παρουσίαση ενός model σε δύο view σημαίνει και την επεξεργασία του model από δύο controller.



Εικόνα 31: Το message board του eKoNES

Στην Εικόνα 32 βλέπουμε την αρχιτεκτονική που υποστηρίζει την υλοποίηση των δύο διαφορετικών όψεων του message board. Όπως φαίνεται, υπάρχει ένα eKoNES Controller-Model abstraction το οποίο είναι υπεύθυνο για την ανανέωση και τον συγχρονισμό των view, καθώς επίσης και για την διαχείριση των συμβάντων. Αυτό το abstraction στην ουσία υλοποιεί την λειτουργία διαμοιρασμού των γεγονότων έτσι ώστε όταν συμβεί κάποιο γεγονός, κάθε ένα από τα δύο view να ενημερωθεί μέσω του eKoNES controller. Κάθε ένα από τα αντικείμενα μας (Το JGraph αντικείμενο και το `JTreeTable`) ερμηνεύει τα γεγονότα αυτά κατάλληλα και ανάλογα με τις δυνατότητες του. Αυτό προσφέρει συνέπεια των δεδομένων που παρουσιάζονται και επιτρέπει την διαλειτουργικότητα των δύο αντικειμένων.



Εικόνα 32: Ενσωμάτωση βιβλιοθήκης JGraph

Σύνοψη και συμπεράσματα

Σε αυτήν την πτυχιακή εργασία αναλύσαμε τα προβλήματα των σύγχρονων διεπαφών και προτείναμε τρόπους για την επίλυση τους. Όπως είδαμε, η διαδικασία παραγωγής διεπαφών, όσο εύκολη και αν έχει γίνει με τα σύγχρονα εργαλεία, ακόμα εξακολουθεί να ταλαιπωρεί τους προγραμματιστές όταν το πεδίο εφαρμογής είναι σύνθετο και απαιτεί συστατικά πέραν του συνηθισμένου. Οι λύσεις που δίνουν οι βιβλιοθήκες διαδραστικών αντικειμένων τρίτων (third party libraries) πολλές φορές δεν επαρκούν, αλλά και όταν επαρκούν προκύπτουν θέματα συγχρονισμού με την βασική βιβλιοθήκη ή διαλειτουργικότητας των αντικειμένων των δύο βιβλιοθηκών.

Όπως είναι φυσικό, πριν από την εμβάθυνση σε θέματα πολύπλοκα όπως επέκταση μίας βιβλιοθήκης ή επαύξηση της λειτουργικότητας ενός διαδραστικού αντικειμένου, κρίθηκε σκόπιμο μία εισαγωγή σε βασικές έννοιες και σε γνώσεις που θα πρέπει να έχει ο αναγνώστης πριν διαβάσει αυτήν την εργασία. Έτσι, έννοιες όπως βιβλιοθήκη διαδραστικών αντικειμένων, στυλ αλληλεπίδρασης, διαδραστικά συστατικά κ.α., εξηγήθηκαν και αναλύθηκαν στα αρχικά κεφάλαια της εργασίας αυτής.

Η συμμετοχή σε ένα project μεγάλης κλίμακας όπως το eKoNEΣ, μέσα από τα προβλήματα που αντιμετωπίστηκαν και μέσω της εκβάθυνσης σε σύνθετα θέματα της βιβλιοθήκης Swing και της εξοικείωσης με την γλώσσα JAVA, έκανε την συγγραφή αυτής της πτυχιακής πολύ πιο εύκολη. Βέβαια οι υλοποιήσεις που παρουσιάστηκαν δεν είναι οι μόνες που έγιναν κατά την διάρκεια του eKoNEΣ αλλά ούτε και οι μόνες που θα γίνουν από εδώ και στο εξής (σ.σ. το eKoNEΣ κατά την συγγραφή της πτυχιακής αυτής βρισκόταν σε εξέλιξη) αλλά ένα μέρος αυτών, επιλεγμένες με τέτοιον τρόπο ώστε να καταδείξουν τα προβλήματα των σύγχρονων βιβλιοθηκών.

Τα ζητήματα που εξετάστηκαν δεν ήταν απλά και οι λύσεις που προτείνονται κρίνονται επαρκείς όσον αφορά την υλοποίηση των αντικειμένων για την χρήση στο έργο eKoNEΣ. Φυσικά, οι τρόποι διαχείρισης της πλατφόρμας που προτάθηκαν από αυτήν την πτυχιακή επιδέχονται περαιτέρω ανάλυση και βελτίωση, δουλειά η οποία όμως ξεφεύγει από τα όρια αυτής της πτυχιακής εργασίας. Στόχος της εργασίας αυτής ήταν πρώτα από όλα να καταδείξει το πρόβλημα που υπάρχει και μέσω της διαδικασίας επίλυσης να προτείνει πιθανές λύσεις. Η τεχνικές της επαύξησης, της επέκτασης και της ενσωμάτωσης εξετάστηκαν και δοκιμάστηκαν σε μεγάλο βαθμό. Δυστυχώς στα πλαίσια αυτής της πτυχιακής δεν υπήρχε χρόνος για την περαιτέρω εξέταση μίας σημαντικής τεχνικής, αυτήν της αφαίρεσης.

Βιβλιογραφία

Akoumianakis, D., Milolidakis, G., Vellis, G., Kotsalis, D. (2007). Interaction platform administration strategies - Practice and experience, Proceedings of the 11th Panhellenic Conference on Informatics (PCI-2007), 18-23 May, Patras, Greece.

Akoumianakis, D., Stephanidis, C. Multiple Metaphor Environments: designing for diversity, Ergonomics, 46 (1-3), 2003, 88-113.

Ballagas, R., Ringel, M., Stone, M., Borchers, J. iStuff: A Physical User Interface Toolkit for Ubiquitous Computing Environments, Proceedings of CHI 2003, 537-544.

Bederson, B. B., J. Grosjean, & J. Meyer. Toolkit Design for Interactive Structured Graphics, IEEE Transactions on Software Engineering, 30 (8), pp. 535-546, 2004.

Bederson, B.B., J. Meyer, and L. Good. Jazz: An Extensible Zoomable User Interface Graphics Toolkit in Java. Conference on User Interface Software and Technology (UIST'00), pp. 171-180 2000.

Heer, J., Card, S., Landay, J. prefuse: a toolkit for interactive information visualization, Proceedings of CHI 2005, April 2–7, 2005, Portland, Oregon, USA.

Kohlert, D., Rodham, K., Olsen, D. Implementing a graphical multi-user interface toolkit, Software – Practice & Experience, 23 (9), 981-999, 1993.

Lee, C., Helal, S., Lee W. Universal Interactions with Smart Spaces, IEEE Pervasing Computing (January-March), 2006, pp. 16-21.

Moll-Carrillo, H.J., Salomon, G., March, M., Fulton Suri, J., and Spreenber, P. Articulating a Metaphor through user-centred design, Proceedings of CHI 1995, Denver, 1995, 566–572.

Mori, G., Paternò, F., Santoro., Design and Development of Multidevice user interfaces through multiple logical descriptions, IEEE Transactions on Software Engineering, 30(8), 2004, pp. 1-14.

Myers, B., Creating User Interfaces by Demonstration. Academic Press, Boston, 1988.

Myers, B., User interfaces software tools. ACM Transactions on Human-Computer Interaction, 12 (1), 1995, 64–103.

Perry, M., O'hara, K., Sellen, A., Brown, B., Harper, R. Dealing with Mobility: Understanding Access Anytime, Anywhere, ACM Transactions on Computer-Human Interaction, 8 (4), 2001, 323–347.

Savidis A, Stephanidis C, Akoumianakis D., Unifying toolkit programming layers: a multi-purpose toolkit integration module. In: Harrison MD, Torres JC (eds) Proceedings of the 4th eurographics workshop on design, specification, and verification of interactive systems (DSV-IS'97), Granada, Spain, 4–6 June 1997. Springer, Berlin Heidelberg New York, pp 177–192.

Stephanidis C, Savidis A, Akoumianakis D. Universally accessible UIs: the unified user interface development, Tutorial in the 2001 ACM SIGCHI conference on human factors in computing systems (CHI 2001), Seattle, Washington, 31 March–5 April, 2001.

Stephanidis, C., Akoumianakis, D., & Paramythis, A. Coping with Diversity in HCI: Techniques for Adaptable and Adaptive Interaction, Tutorial no 11 in the 8th International Conference on Human-Computer Interaction (HCI International '99), Munich, Germany, 22-26 August, 1999.

Stephanidis, C., Savidis, A., & Akoumianakis, D. Unified Interface Development: Tools for Constructing Accessible and Usable User interfaces, Tutorial no 13 in the 7th International Conference on Human-Computer Interaction (HCI International '97), San Francisco, USA, 24-29 August, 1997.

Πληροφορίες από την δικτυακή εγκυκλοπαίδεια Wikipedia:

http://en.wikipedia.org/wiki/User_interface

http://en.wikipedia.org/wiki/GUI_toolkit

http://en.wikipedia.org/wiki/Swing_java

http://en.wikipedia.org/wiki/Java_%28programming_language%29

Πληροφορίες από ιστοσελίδες στο διαδίκτυο:

http://www.interaction-design.org/encyclopedia/interaction_styles.html

http://lipas.uwasa.fi/~mj/hci/hci11.html

http://www.answers.com/topic/direct-manipulation-interface

http://www.webopedia.com/TERM/G/Graphical_User_Interface_GUI.html

http://www.idemployee.id.tue.nl/g.w.m.rauterberg/lecturenotes/SEMINAR-3%20INTERACTION/

http://java.sun.com/docs/books/tutorial/uiswing/

http://java.sun.com/docs/books/tutorial/

http://www.jgraph.com/paper.html

http://ace-centre.hostinguk.com/index.cfm?pageid=83216478-D613-62F1-C85EEB3021CA6B57