

ΤΕΧΝΟΛΟΓΙΚΟ ΕΚΠΑΙΔΕΥΤΙΚΟ ΙΔΡΥΜΑ ΚΡΗΤΗΣ
Σχολή Τεχνολογικών Εφαρμογών
Τμήμα Εφαρμοσμένης Πληροφορικής και Πολυμέσων



Πτοχιακή εργασία

“Μεταφορές αλληλεπίδρασης και ο ρόλος τους στη
σχεδίαση σύγχρονων διεπαφών”

Λεωνίδας Πατούχας

Επιβλέπων καθηγητής : Ακουμιανάκης Δημοσθένης

Ηράκλειο, 2007

Ευχαριστίες

Θα ήθελα να ευχαριστήσω ιδιαίτερα τον υπεύθυνο καθηγητή για την πτυχιακή αυτή τον κύριο Ακουμιανάκη για την βοήθεια του καθώς και την υπομονή του. Τέλος να ευχαριστήσω θερμά το forum της Java καθώς και το newsgroup του netBeans για την βοήθεια τους.

Πίνακας περιεχομένων

1.	Εισαγωγή.....	6
2.	Μεταφορές αλληλεπίδρασης.....	8
2.1	Ο ρόλος της μεταφοράς στη σχεδίαση	8
2.2	Τεχνικές σχεδίασης διεπαφής βάση μεταφοράς	9
2.2.1	Χαρακτηρίζοντας το συνολικό περιβάλλον χρήσης	9
2.2.2	Ενσωμάτωση μεταφοράς σε μια διεπαφή.....	11
2.3	Παραδείγματα.....	13
2.3.1	Η μεταφορά του desktop	13
2.3.2	Η μεταφορά του δωματίου	14
2.3.3	Η μεταφορά του τηλεφωνικού καταλόγου- ατζέντα.....	15
2.3.4	Το ξεφύλλισμα (browsing)	16
2.3.5	Το καλάθι του supermarket	17
2.3.6	Η εφημερίδα	18
2.4	Μεταφορές αλληλεπίδρασης και αντικείμενα συμπερίληψης	19
2.5	Στάδια σχεδίασης διεπαφής με την χρήση μεταφοράς.....	20
3.	Σχεδίαση του JBook	21
3.1	Η μεταφορά του βιβλίου.....	21
3.2	Ανάλυση ενός κλασικού βιβλίου.....	22
3.3	Απόδοση της κλασικής μεταφοράς σε H/Y.....	24
4.	Υλοποίηση του JBook και σενάρια επίδειξης	26
3.4	Υλοποίηση σελίδων βιβλίου	27
3.5	Συστατικά παρουσίασης περιεχομένων & λειτουργιών.....	30
5.	Επίλογος	51

Πίνακας σχημάτων

Εικόνα 1 : Σχεδιασμός συστήματος.....	8
Εικόνα 2 : Περιβάλλον του γραφείου	9
Εικόνα 3 : Μοντέλο κλάσεων γραφείου	10
Εικόνα 4 : Βήματα σχεδιασμού	10
Εικόνα 5 : Συμβολική απόδοση βασισμένη στα παράθυρα	11
Εικόνα 6 : Συμβολική απόδοση βασισμένη στο ιδεόγραμμα	11
Εικόνα 7 : Υλοποίηση μεθόδου διαγραφής.....	12
Εικόνα 8 : Κουμπι εντολής.....	12
Εικόνα 9 : Μπάρα κύλισης (sliders) και potentiometers	12
Εικόνα 10 : Check box.....	12
Εικόνα 11 : Μενού επιλογής	13
Εικόνα 12 : Εικονικός χώρος δωματίου συζήτησης.....	14
Εικόνα 13 : Εναλλακτική μεταφορά δωματίου	15
Εικόνα 14: Η σχεδιαστική ιδέα της Compac	16
Εικόνα 15: Παράδειγμα λογισμικού πλοήγησης	17
Εικόνα 16 : Το ηλεκτρονικό βιβλίο	21
Εικόνα 17 : Το Webbook των Card et al. 1996.....	22
Εικόνα 18 : Παράδειγμα εναλλακτικής απόδοσης της μεταφοράς του βιβλίου.....	23
Εικόνα 19: Αρχική μορφή του βιβλίου μας	27
Εικόνα 20: Κυρτότητα της σελίδας του βιβλίου	28
Εικόνα 21: Εμφάνιση διαφοράς γραφικών με την χρήση της μεθόδου antialiasing.....	29
Εικόνα 22: Εξώφυλλο βιβλίου	29
Εικόνα 23: Εμφάνιση περιεχομένων	30
Εικόνα 24: Επιλογή μεγέθους γραμματοσειράς	32
Εικόνα 25: Επιλογή γρήγορης πλοήγησης.....	34
Εικόνα 26: Μήνυμα λάθους	36
Εικόνα 27: Frame βοήθειας και επιλογής τρόπου εμφάνισης σχολίων	36
Εικόνα 28: Σχόλια στην θέση των σελίδων του βιβλίου.....	38
Εικόνα 29: Σχόλια στο δεξιό κομμάτι του βιβλίου.....	38
Εικόνα 30: Σχόλια στο κέντρο της σελίδας	39
Εικόνα 31: Μπάρα πλοήγησης	42
Εικόνα 32: Έλεγχος αποθήκευσης σχολίων.....	46

Πίνακας με αποσπάσματα υλοποίησης

Απόσπασμα 1.....	27
Απόσπασμα 2.....	28
Απόσπασμα 3.....	31
Απόσπασμα 4.....	31
Απόσπασμα 5.....	31
Απόσπασμα 6.....	32
Απόσπασμα 7.....	35
Απόσπασμα 8.....	37
Απόσπασμα 9.....	39
Απόσπασμα 10.....	41
Απόσπασμα 11.....	41
Απόσπασμα 12.....	42
Απόσπασμα 13.....	43
Απόσπασμα 14.....	43
Απόσπασμα 15.....	44
Απόσπασμα 16.....	45
Απόσπασμα 17.....	46
Απόσπασμα 18.....	46
Απόσπασμα 19.....	47
Απόσπασμα 20.....	47
Απόσπασμα 21.....	50

1. Εισαγωγή

Η προτεινόμενη πτυχιακή εργασία αφορά τη μελέτη του ρόλου των μεταφορών αλληλεπίδρασης (interaction metaphors) στο σχεδιασμό μοντέρνων διαδραστικών συστημάτων. Κατά καιρούς η έννοια της μεταφοράς έχει αποτελέσει αντικείμενο έρευνας σε ένα ευρύτατο πεδίο γνωστικών αντικειμένων, όπως η φιλοσοφία της γλώσσας, η τεχνητή νοημοσύνη, κλπ. Από το την δεκαετία το '80 απασχόλησε και συνεχίζει να απασχολεί ερευνητές του κλάδου της 'Επικοινωνίας Ανθρώπου-Υπολογιστή' με σημαντικά αποτελέσματα τόσο σε μεθοδολογικό επίπεδο όσο στο επίπεδο της μηχανικής των διεπαφών. Με δεδομένη την καθολική σχεδόν αποδοχή της μεταφοράς του desktop κατά την δεκαετία του '90, σήμερα το ενδιαφέρον εστιάζεται είτε στη επέκταση της αρχικής εκδοχής του desktop, είτε στην διερεύνηση νέων μεταφορών που να 'ταιριάζουν' καλύτερα στις απαιτήσεις του νέου τεχνολογικού περιβάλλοντος (δηλαδή διαδίκτυο, κοινότητες χρηστών, νέες εφαρμογές, κλπ). Σχετικές προτάσεις που μελετήθηκαν κατά το παρελθόν ήταν η μεταφορά του δωματίου, της ορχήστρας, της βιβλιοθήκης, του ψηφιακού χρήματος, κλπ, ωστόσο, με περιορισμένη απήχηση σε εμπορικές εφαρμογές & υπηρεσίες.

Στην παρούσα εργασία θα εξεταστεί η μεταφορά ενός εντύπου όπως το κλασικό βιβλίο ως μέσο σχεδίασης διεπαφών. Η επιλογή της μεταφοράς του βιβλίου βασίστηκε στους εξής λόγους :

- ξεφεύγει από τις παραδοσιακές μεταφορές που μελετήθηκαν στο παρελθόν και εισάγει νέες δομές παρουσίασης και διαχείρισης δεδομένων όπως η δελίδα, το κεφάλαιο, ευρετήρια, κλπ
- είναι μια έννοια ευρέως γνωστή σε όλους τους εν δυνάμει χρήστες ενός διαδραστικού συστήματος και δεν απαιτεί την εις βάθος γνώση του υπολογιστικού περιβάλλοντος αφού αναγάγει σύνθετα γραφικά καθήκοντα σε στοιχειώδεις χειρισμούς ενός κλασικού εγχειριδίου
- προσφέρεται για να μελετηθούν οι έννοιες της συλλογικότητας και της κοινότητας (community) που χαρακτηρίζουν σύγχρονες εφαρμογές του διαδικτύου
- όπως το κλασικό βιβλίο έχει καθολική αποδοχή, έτσι και οι σύγχρονες εφαρμογές πρέπει να υποστηρίζουν καθολική χρήση - είτε από μεμονωμένα άτομα είτε από ομάδες χρηστών με κοινά ενδιαφέροντα
- έχει ευρύτατο φάσμα εφαρμογών (π.χ. ηλεκτρονικούς φακέλους υγείας, ψηφιακές βιβλιοθήκες)

Ο ειδικότερος στόχος της παρούσας πτυχιακής εργασίας είναι η ανάπτυξη μιας εφαρμογής η οποία στηριζόμενη στη κλασική μορφή του βιβλίου δίνει στους χρήστες της τη δυνατότητα να αλληλεπιδράσουν με το περιεχόμενό της. Ταυτόχρονα, επιχειρείται η επέκταση της κλασικής μεταφοράς έτσι ώστε να αρθούν περιορισμοί

που προκύπτουν από τη στατική δομή του κλασικού βιβλίου και έτσι να παρουσιαστεί μια σύγχρονη και δυναμική διεπιφάνεια χρήσης που να αξιοποιεί την προηγούμενη εμπειρία του χρήστη και να επιτρέπει μεγαλύτερη ευχρηστία στη εκτέλεση ενδεικτικών καθηκόντων.

Η υλοποίηση της εφαρμογής έγινε σε τρία στάδια. Στο πρώτο στάδιο (θεωρητικό) μελετήθηκαν οι έννοιες της αλληλεπίδρασης και του διαδραστικού συστήματος. Αναλύθηκε ο ρόλος της μεταφοράς και οι τεχνικές σχεδίασης συστημάτων με βάση αυτή. Μελετήθηκαν επίσης όλα τα παραδείγματα μεταφορών που έχουν χρησιμοποιηθεί μέχρι τώρα, ξεκινώντας από τη μεταφορά του desktop και του βιβλίου και καταλήγοντας σε πιο σύγχρονες μεταφορές, όπως αυτή του δωματίου. Στο δεύτερο στάδιο μελετήθηκε η γενική έννοια του βιβλίου και η χρήση του από την σκοπιά του αναγνώστη. Αναγνωρίστηκαν οι γενικές οντότητές της, τα καθήκοντα χρήσης τους καθώς και ο τρόπος απόδοσης αυτών μέσω μίας διεπαφής. Για το σκοπό αυτό μελετήθηκαν τυπικά σενάρια χρήσης του βιβλίου και έγινε αναγωγή αυτών σε διαδραστικά καθήκοντα με τη χρήση γραφικής διεπαφής. Στο τελευταίο στάδιο το ηλεκτρονικό βιβλίο αναπτύχθηκε σε πλήρη κλίμακα, εγκαταστάθηκε σε χώρο και μηχανήμα του Τμήματος Εφαρμοσμένης Πληροφορικής & Πολυμέσων και επικυρώθηκε με την χρήση επιλεγμένων σεναρίων τα οποία και παρουσιάζουμε στην παρούσα εργασία.

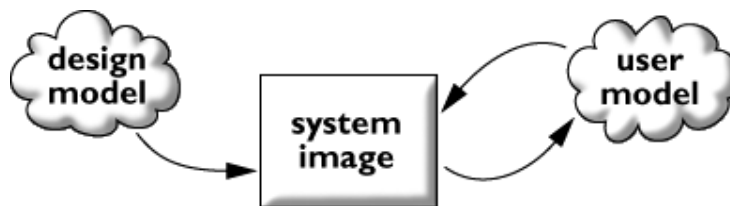
Το αποτέλεσμα της παρούσας πτυχιακής εργασίας είναι μια σύγχρονη διεπαφή που αν και μπορεί να βελτιωθεί σε πολλά σημεία, αποδίδει τόσο την έννοια μιας εναλλακτικής μεταφοράς αλληλεπίδρασης όσο και τις δυνατότητες που προφέρονται από την ενσωμάτωση της μεταφοράς με συμβολικό τρόπο. Το ηλεκτρονικό βιβλίο όπως παρουσιάζεται στην παρούσα πτυχιακή είναι μία εφαρμογή απλή στη χρήση της, εύκολα κατανοητή και απευθύνεται τόσο σε μεμονωμένους χρήστες όσο και σε ομάδες χρηστών, καθώς υποστηρίζει τη συνεργασία και τη συλλογικότητα.

2. Μεταφορές αλληλεπίδρασης

2.1 Ο ρόλος της μεταφοράς στη σχεδίαση

Η σχεδίαση μιας διεπαφής προσομοιάζεται σε πολλά σημεία με άλλα σχεδιαστικά/ κατασκευαστικά πεδία όπως παραδείγματος χάριν η σχεδίαση ενός κτηρίου, η αρχιτεκτονική ενός εσωτερικού ή εξωτερικού χώρου, κλπ. Σε όλες αυτές τις περιπτώσεις αναπτύσσεται από το σχεδιαστή ένα μοντέλο του χώρου το οποίο παρουσιάζεται στους χρήστες για λόγους εξοικείωσης αλλά και επικύρωσης. Το μοντέλο αυτό συνήθως βασίζεται σε προηγούμενη εμπειρία του σχεδιαστή που προσαρμόζεται στο νέο σχεδιαστικό πρόβλημα και σε κανόνες κατασκευής που αφορούν τεχνικά και ποιοτικά στοιχεία του τελικού τεχνήματος.

Στην περίπτωση των διαδραστικών συστημάτων τα μοντέλα που αναπτύσσονται είναι περισσότερα του ενός και αφορούν τόσο το ίδιο το σύστημα όσο και τους χρήστες του. Συγκεκριμένα, για να δημιουργήσει κάποιος ένα μοντέλο από κάτι, πρέπει πρώτα να γνωρίζει πώς δουλεύει αυτό το κάτι. Αυτό δεν συμπεριλαμβάνει μόνο το πώς λειτουργεί το σύστημα ως προς το χρήστη αλλά και πώς αυτό συμπεριφέρεται κάτω από κάποιες συνθήκες. Οι σχεδιαστές αναπτύσσουν ένα μοντέλο σχεδιασμού (design model) το οποίο βασίζεται στους χρήστες του συστήματος και στο υφιστάμενο ή αναμενόμενο τρόπο χρήσης αυτού (βλέπε Εικόνα 1). Περιλαμβάνει επίσης και χαρακτηριστικά γνωρίσματα τους συστήματος όπως τον τρόπο πλοήγησης και τον τρόπο σύνδεσης των επιμέρους μερών.



Εικόνα 1 : Σχεδιασμός συστήματος

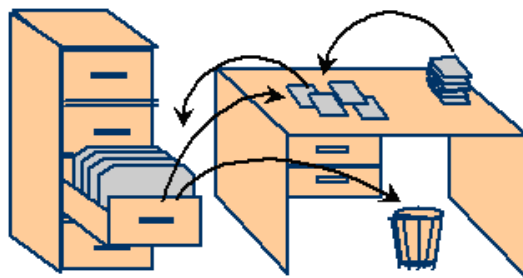
Καθώς ένας χρήστης χρησιμοποιεί το σύστημα αναπτύσσεται το μοντέλο χρήσης του συστήματος (user model). Σε αυτό το μοντέλο παρουσιάζεται η δομή του συστήματος και πώς αυτό λειτουργεί. Μερικές φορές το μοντέλο χρήσης είναι το ακριβές μοντέλο του συστήματος, άλλες όμως όχι. Σκοπός των σχεδιαστών είναι να “επικοινωνήσει” το μοντέλο σχεδίασης με το χρήστη. Ο μόνος τρόπος επικοινωνίας τους, εν τούτοις, είναι η ίδια η εικόνα του συστήματος. Εάν ένα σύστημα έχει σχεδιαστεί επιτυχώς, κατόπιν το μοντέλο χρήσης θα είναι ισοδύναμο με το τελικό σύστημα.

2.2 Τεχνικές σχεδίασης διεπαφής βάση μεταφοράς

Γενικά υπάρχουν δύο τρόποι χρήσης ή ενσωμάτωσης μιας μεταφοράς στο σχεδιασμό ενός διαδραστικού συστήματος. Ο πρώτος αφορά τη χρήση της μεταφοράς για το χαρακτηρισμό ολόκληρου του συστήματος, ενώ ο δεύτερος αφορά τη χρήση της μεταφοράς για την απόδοση επιλεγμένων καθηκόντων ή δραστηριοτήτων ενός συστήματος.

2.2.1 Χαρακτηρίζοντας το συνολικό περιβάλλον χρήσης

Ένα παράδειγμα χαρακτηρισμού ενός συστήματος με τη χρήση μιας μεταφοράς είναι το γραφικό desktop που συναντάμε σε όλους του επιτραπέζιους προσωπικούς υπολογιστές. Στην περίπτωση αυτή χρησιμοποιείται μια καλώς κατανοητή έννοια ή πεδίο αναφοράς (source domain) που είναι το περιβάλλον του γραφείου (βλέπε Εικόνα 2) για να γίνει κατανοητό ένα δύσκολο για τον άνθρωπο πεδίο στόχου (target domain)– αυτό της λειτουργίας ενός H/Y.

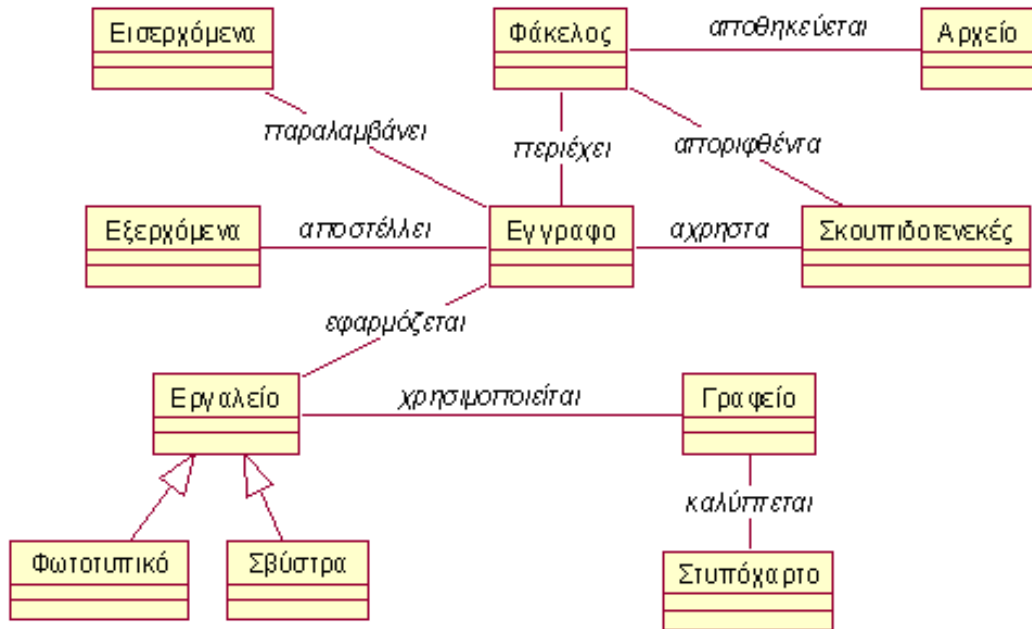


Εικόνα 2 : Περιβάλλον του γραφείου

Η ερμηνεία ενός πεδίου στόχου μέσω της αναγωγής σε ένα πεδίο αναφοράς δεν είναι απλή διαδικασία. Αντίθετα, κρίνεται ότι αποτελεί μέγιστο σχεδιαστικό πρόβλημα με υψηλό βαθμό ρίσκου και κινδύνων. Κάτι τέτοιο τουλάχιστον προκύπτει από τις σχετικές μελέτες που είναι καταγεγραμμένες στην βιβλιογραφία και παρουσιάζουν το χρόνο και τον τρόπο ανάπτυξης της αρχικής έκδοσης του γραφικού περιβάλλοντος χρήσης του STAR και μετέπειτα του LISA. Η δυσκολία συνίσταται στο γεγονός ότι η μετάβαση από ένα πεδίο αναφοράς σε ένα πεδίο στόχου δεν είναι απλή υπόθεση. Κατά καιρούς έχει μελετηθεί από ερευνητές και έχουν υπάρξει διάφορες μεθοδολογικές προτάσεις.

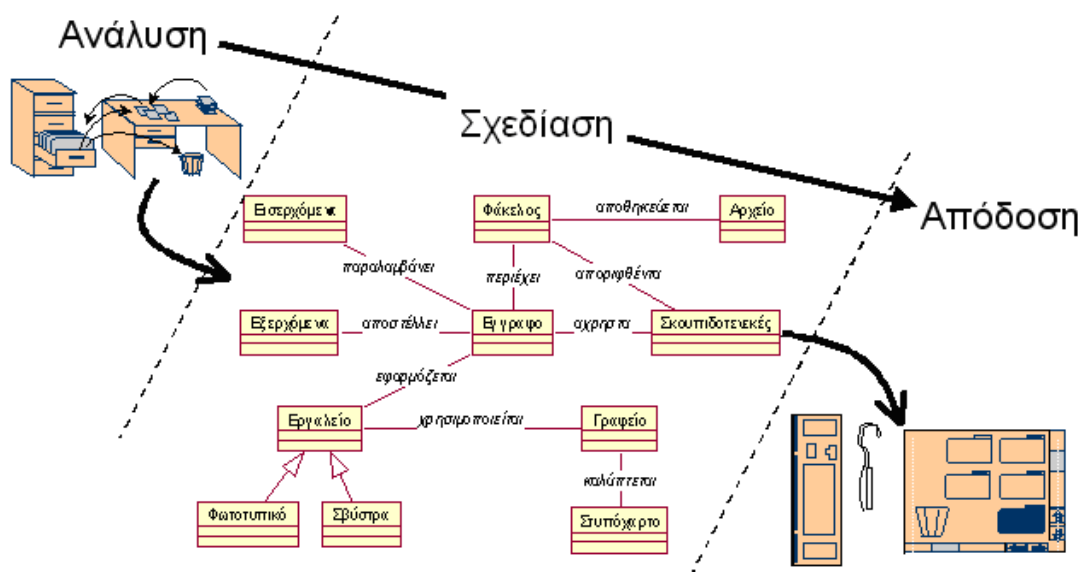
Μια πιο σύγχρονη προσέγγιση στο θέμα αξιώνει ότι η μετάβαση αυτή πραγματοποιείται σε τρία βήματα. Στο πρώτο βήμα, το πεδίο αναφοράς αναλύεται όσον αφορά τα αντικείμενα και τα καθήκοντα που το χαρακτηρίζουν. Με αναφορά στο περιβάλλον του γραφείου θα λέγαμε ότι στη φάση της ανάλυσης αναγνωρίζονται και περιγράφονται τόσο τα αντικείμενα/ οντότητες που χαρακτηρίζουν το πεδίο όπως για παράδειγμα φάκελοι, έγγραφα, συρτάρια, κλπ όσο και τα καθήκοντα που εκτελούν οι χρήστες και αφορούν είτε γενικού τύπου καθήκοντα (επικοινωνία, διάβασμα, συγγραφή) είτε δράσεις στα συγκεκριμένα αντικείμενα (διαγραφή εγγράφου).

Στο δεύτερο στάδιο, αυτό της σχεδίασης τα ευρήματα της ανάλυσης αποτυπώνονται σε ένα αφηρημένο σημασιολογικό μοντέλο. Παράδειγμα τέτοιου μοντέλου για την περίπτωση του περιβάλλοντος γραφείου είναι το παρακάτω (Εικόνα 3) μοντέλο κλάσεων της UML που συνοψίζει με οπτικό τρόπο τις οντότητες του πεδίου αναφοράς και τις χρήσεις τους.



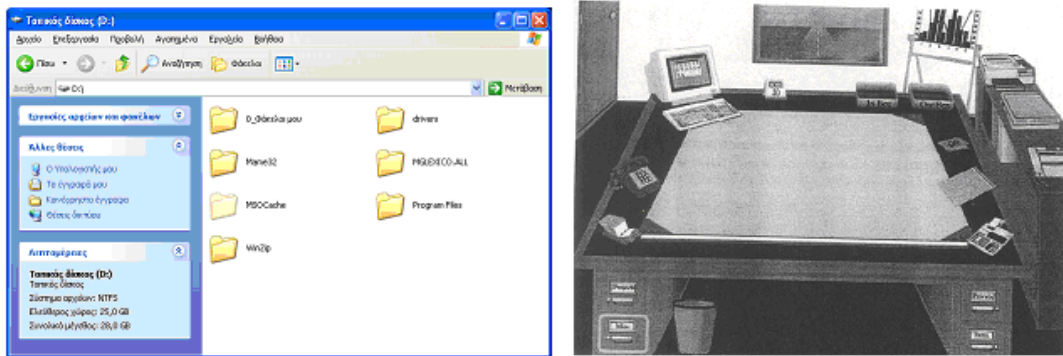
Εικόνα 3 : Μοντέλο κλάσεων γραφείου

Στο τρίτο και τελευταίο βήμα ο σχεδιαστής πρέπει να αποφασίσει ποιο τμήμα του σημασιολογικού μοντέλου αφορά την εφαρμογή του και πώς θα αποδοθεί μέσω υπολογιστή. Στο βήμα αυτό καθορίζονται τα αντικείμενα αλληλεπίδρασης που θα χρησιμοποιηθούν, οι καταστάσεις τους και ο τρόπος χειρισμού αυτών. Σχηματικά, η μέθοδος των τριών βημάτων που παρουσιάσαμε συνοψίζεται στο παρακάτω σχήμα:



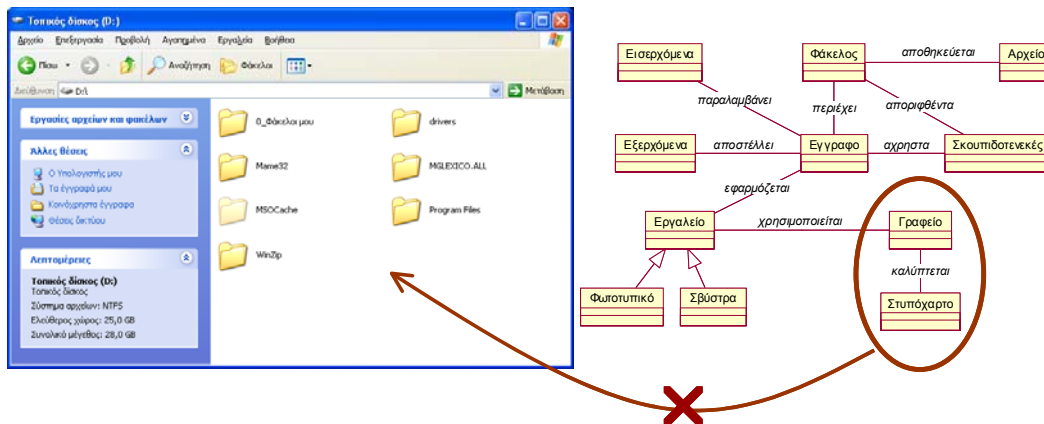
Εικόνα 4 : Βήματα σχεδιασμού

Ωστόσο πρέπει να τονίσουμε δύο σημαντικά στοιχεία που αφορούν την χρήση μεταφορών στην σχεδίαση διεπαφών. Η πρώτη συνίσταται στο γεγονός ότι δεν υπάρχει μία μόνο δυνατή απόδοση των οντοτήτων και των εννοιών ενός πεδίου αναφοράς. Χαρακτηριστικό παράδειγμα είναι η συμβολική απόδοση που βασίζεται στα παράθυρα και η απόδοση βάση ιδεογράμματος του ίδιου ακριβώς πεδίου αναφοράς (βλέπε Εικόνα 5).



Εικόνα 5 : Συμβολική απόδοση βασισμένη στα παράθυρα

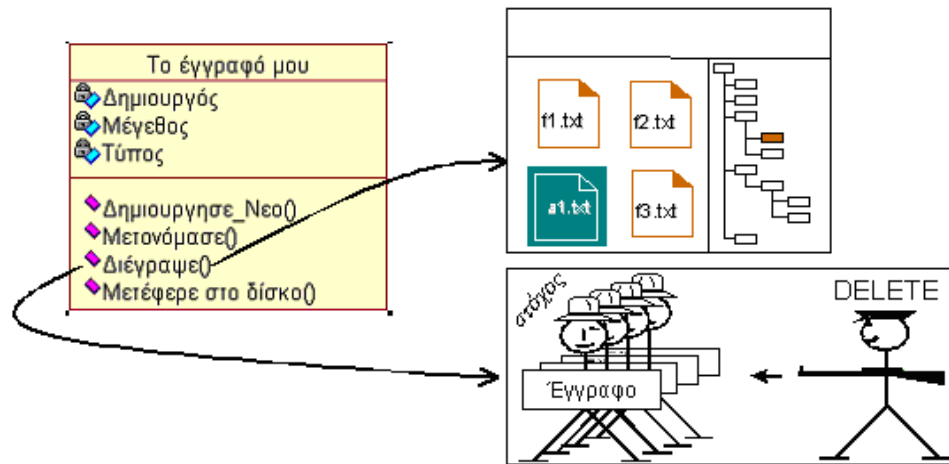
Το δεύτερο σημαντικό στοιχείο είναι ότι στην σχεδίαση διεπαφών μια μεταφορά δεν χρειάζεται και συχνά δεν πρέπει να αξιοποιείται εξ' ολοκλήρου, π.χ. η έννοια του στυπόχαρτου δεν αποδίδεται στη συμβολική απόδοση (βλέπε Εικόνα 6). Επομένως μπορεί κανείς να συμπεράνει ότι η σχεδίαση του σημασιολογικού επιπέδου μιας διεπαφής με την χρήση μιας μεταφοράς είναι ιδιαίτερα απαιτητική διαδικασία τόσο σε επίπεδο ανάλυσης και σχεδίασης όσο και σε επίπεδο απόδοσης της μεταφοράς μέσω υπολογιστή.



Εικόνα 6 : Συμβολική απόδοση βασισμένη στο ιδεόγραμμα

2.2.2 Ενσωμάτωση μεταφοράς σε μια διεπαφή

Αντί του χαρακτηρισμού του συνόλου της διεπαφής, στην περίπτωση αυτή η μεταφορά ενσωματώνεται και χαρακτηρίζει επιμέρους καθήκοντα (βλέπε Εικόνα 7) ή συγκεκριμένα διαδραστικά αντικείμενα (βλέπε Εικόνα 8 - Εικόνα 11).



Εικόνα 7 : Υλοποίηση μεθόδου διαγραφής

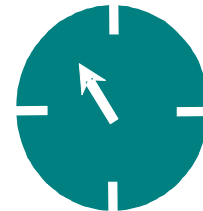
Αξίζει στο σημείο αυτό να αναφέρουμε ότι στην συντριπτική τους πλειοψηφία τα διαδραστικά αντικείμενα που συναντάμε σε σύγχρονες διεπαφές, ενσωματώνουν με τον ένα ή τον άλλο τρόπο κάποια ευρέως γνωστή μεταφορά, π.χ.

- Push buttons – Ηλεκτρικές συσκευές



Εικόνα 8 : Κουμπί εντολής

- Sliders / potentiometers - Ηλεκτρικές συσκευές



Εικόνα 9 : Μπάρα κύλισης (sliders) και potentiometers

- Check- boxes – Συμπλήρωση φόρμας



Εικόνα 10 : Check box

- Menus – Μενού εστιατορίου



Εικόνα 11 : Μενού επιλογής

2.3 Παραδείγματα

2.3.1 Η μεταφορά του desktop

Αναμφισβήτητα, μια από τις πιο γνωστές μεταφορές που χρησιμοποιούνται στο σχέδιο των εικονικών διεπαφών (iconic interface) είναι η μεταφορά του desktop. Ο πρώτος υπολογιστής που χρησιμοποίησε αυτή τη μεταφορά στο σχεδιασμό του περιβάλλοντος εργασίας ήταν ο Apple Macintosh, το 1984.

Η μεταφορά του desktop είναι ένα σύνολο από έννοιες που χρησιμοποιούνται αυτή τη στιγμή από λειτουργικά συστήματα βασισμένα στο GUI (Graphics User Interface). Σε αυτή τη μεταφορά, η οθόνη του υπολογιστή απεικονίζει το γραφείο του χρήστη, επάνω στο οποίο μπορούν να τοποθετηθούν τα έγγραφα, φάκελοι κτλ. Ο χρήστης αλληλεπιδρά με αυτόν τον εικονικό κόσμο με τη χρήση ενός ποντικιού, μια συσκευή εισαγωγής που επιτρέπει στο χρήστη να κινεί ένα βέλος μέσα στην επιφάνεια εργασίας και να επιλέξει ένα ηλεκτρονικό αντικείμενο, με το πάτημα ενός κουμπιού του ποντικιού. Ένα έγγραφο μπορεί να ανοίξει μέσα σε ένα παράθυρο, αντιπροσωπεύοντας το αντίγραφο του εγγράφου που τοποθετείται πάνω στο γραφείο. Για να ξεφορτωθεί ο χρήστης ένα φυσικό έγγραφο από το γραφείο του, το πετάει στα απορρίμματα. Για να διαγράψει ένα ηλεκτρονικό έγγραφο επιλέγει απλά το έγγραφο και το κινεί πάνω στα ηλεκτρονικά απορρίμματα, και το έγγραφο εξαφανίζεται. Και στις δύο περιπτώσεις το έγγραφο δεν χάνεται τελείως μέχρι ο χρήστης να αδειάσει τον κάδο. Εξαρτήματα του γραφείου όπως, σημειωματάριο, calculator, ημερολόγιο είναι και αυτά διαθέσιμα με μικρές εφαρμογές.

Η ίδια μεταφορά του desktop έχει επεκταθεί και σε διάφορες εφαρμογές, καθώς η πρόσβαση και η χρήση των χαρακτηριστικών των υπολογιστών είναι πιο σημαντική από διατήρηση της “αγνότητας” της μεταφοράς. Γι’ αυτό το λόγο συναντάμε τον κάδο απορριμμάτων και τα αποθηκευτικά μέσα (που μπορούν να θεωρηθούν σαν ντουλάπα αρχειοθέτησης), πράγματα τα οποία δεν είναι σίγουρο ότι υπάρχουν σε όλα τα γραφεία. Αλλά και άλλα χαρακτηριστικά όπως menu bars, task bars, ή docks που σίγουρα δεν υπάρχουν σε ένα πραγματικό γραφείο.

Τον τελευταίο καιρό, η μεταφορά του desktop έχει γίνει λιγότερο σημαντική μετά την εμφάνιση μεγάλων αποθηκευτικών μέσων που έχουν κάνει τη διαχείριση των φακέλων πιο εύκολη. Μια πιο προσανατολισμένη προς το χρήστη προσέγγιση κερδίζει έδαφος, καθώς ο χρήστης μπορεί να οργανώσει τα έγγραφά του με τέτοιο τρόπο έτσι ώστε να διευκολύνει τις ιδιαίτερες ανάγκες του, χωρίς να είναι αναγκασμένος να χρησιμοποιήσει ένα file- system σχεδιασμό. Η προσθήκη των “έξυπνων φακέλων” που ανοίγουν οδηγεί σε μια μέθοδο όπου τα αρχεία

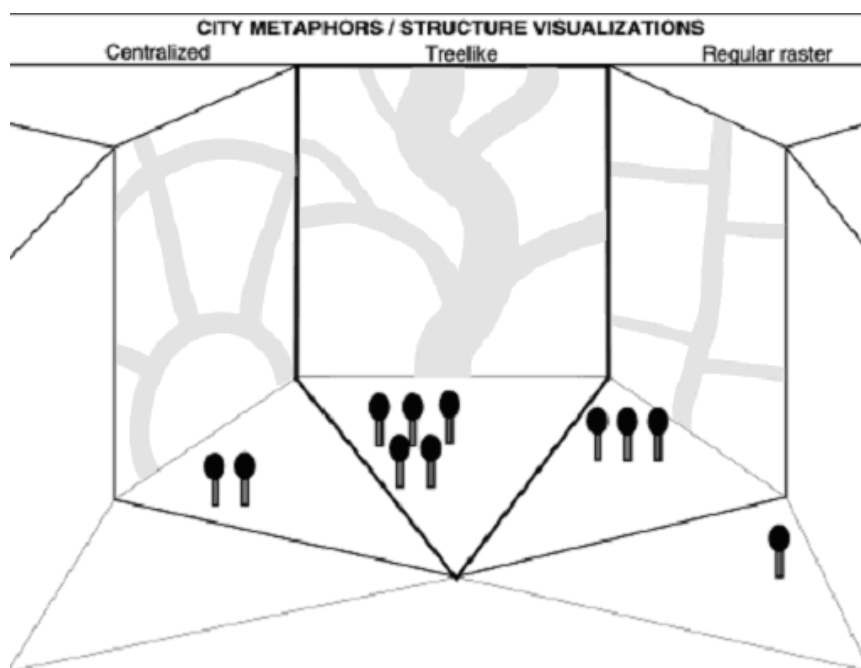
τοποθετούνται με βάση τα κριτήρια αναζήτησης του χρήστη, παρά τη φυσική τους θέση στο δίσκο, η οποία δεν έχει τελικά καμία σπουδαιότητα για το χρήστη.

2.3.2 Η μεταφορά του δωματίου

Πολλές εφαρμογές που απευθύνονται σε ομάδες χρηστών περιέχουν κενά που εμποδίζουν τη συλλογική εργασία των χρηστών. Η μεταφορά του δωματίου καλύπτει τα κενά αυτών των εφαρμογών και επιτρέπει στους χρήστες να συνεργαστούν αρμονικά. Τα δωμάτια που μελετούνται είναι αντίστοιχα των φυσικών δωματίων όπως: γραφεία εταιρειών, γραφεία συνεδριάσεων, γραφεία διαλείμματος κλπ.

Τα δωμάτια είναι οροθετημένοι χώροι που ενεργούν ως containers. Οι τοίχοι χωρίζουν έναν μεγάλο χώρο σε πολλά δωμάτια. Αυτοί οι τοίχοι μπορούν είτε να χωρίσουν είτε να ενώσουν τους ανθρώπους. Κάθε δωμάτιο χωριστά δρα σαν ένας διαφορετικός container, ο οποίος περιέχει, έπιπλα, εργαλεία δουλείας και επικοινωνίας που κάνει την ατομική και την ομαδική εργασία πιο ευχάριστη. Επίσης τα δωμάτια είναι προσπελάσιμα καθώς οι χρήστες μπορούν να εισέλθουν και να εξέλθουν οποιαδήποτε στιγμή. Τα δωμάτια μπορούν να προσαρμοστούν σε κάθε χρήστη ανάλογα με τις ανάγκες του.

Η μεταφορά του δωματίου βρίσκει εφαρμογή σε εφαρμογές όπου είναι απαραίτητη η συνεργασία.

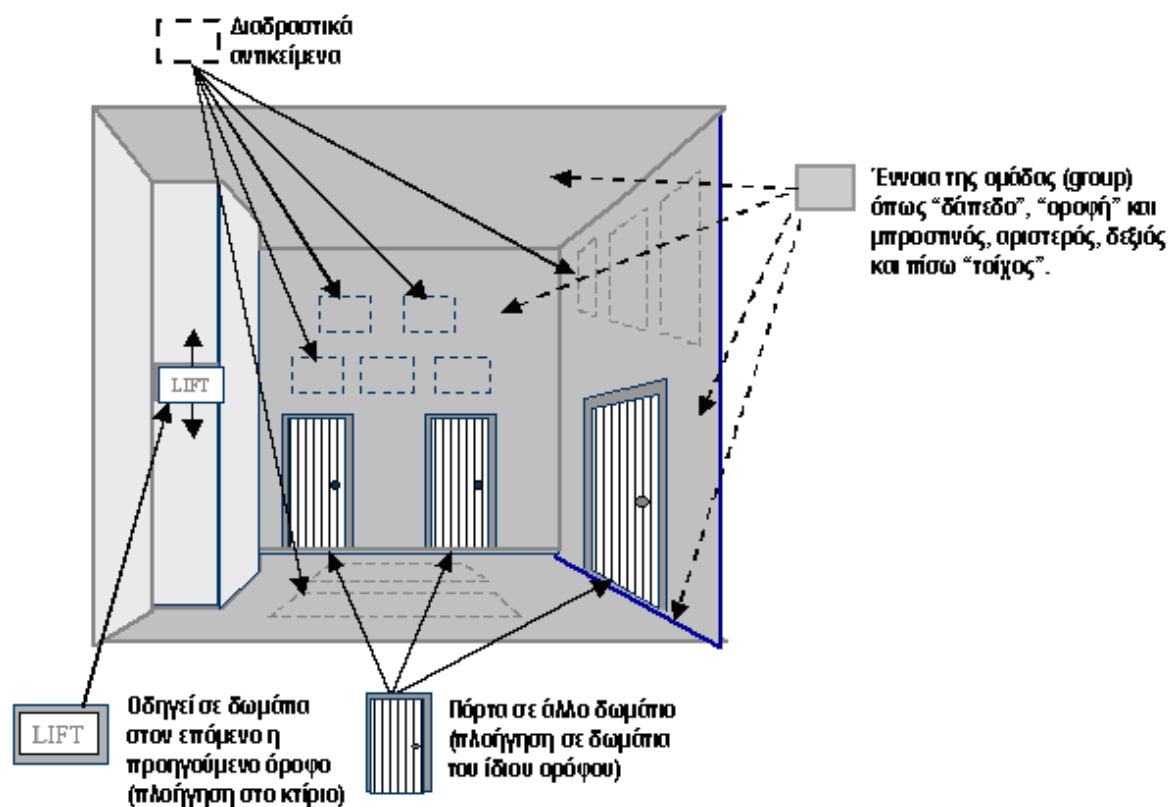


Εικόνα 12 : Εικονικός χώρος δωματίου συζήτησης

Στην Εικόνα 12 παρουσιάζεται ένα δωμάτιο ή χώρος συζήτησης. Υπάρχει ένα ενεργό θέμα συζήτησης που αφορά εναλλακτικές μεθόδους απόδοσης χωρικών μεταφορών. Δύο μετέχοντες έχουν εκφραστεί υπέρ μιας απόδοσης τύπου ομόκεντρων κύκλων (centralised), πέντε μετέχοντες υποστηρίζουν δενδρική δομή (tree-like) και τρεις μετέχοντες υποστηρίζουν απόδοση τύπου regular raster (δεξιά). Ένας μετέχων είναι ο παρατηρητής ή ο συντονιστής. Μπορεί λοιπόν εύκολα να φανταστεί κανείς

ένα σύστημα υποστήριξης συζητήσεων (forum) όπου η βασική μεταφορά είναι η παραπάνω.

Ένα εναλλακτικό σενάριο απόδοσης της μεταφοράς του δωματίου φαίνεται στην Εικόνα 13. Ο χώρος αλληλεπίδρασης είναι δομημένος σύμφωνα με ένα κλασικό δωμάτιο. Τα δωμάτια εσωκλείουν οντότητες, ο τύπος των οποίων καθορίζεται στη φάση υλοποίησης. Τα δωμάτια περιέχουν τις πόρτες και έναν ανελκυστήρα. Οι πόρτες οδηγούν σε άλλα αντικείμενα δωματίων του ίδιου πατώματος, ενώ ο ανελκυστήρας οδηγεί σε αντικείμενα δωματίων που είναι είτε σε επίπεδο ανωτέρω είτε κατωτέρω. Τα αντικείμενα που βρίσκονται σε ένα δωμάτιο μπορούν να ανήκουν σε μια από τις ακόλουθες έξι ομάδες: μπροστινός τοίχος, πίσω τοίχος, αριστερός τοίχος, δεξιός τοίχος, πάτωμα και οροφή. Σύμφωνα με αυτό το σχέδιο, στην υλοποίηση μιας τέτοιου είδους μεταφοράς, τα διάφορα κύρια αντικείμενα μπορούν να κληρονομήσουν τις ιδιαίτερες ιδιότητες αλληλεπίδρασης από τη συγκεκριμένη ομάδα στην οποία ανήκουν.



Εικόνα 13 : Εναλλακτική μεταφορά δωματίου

2.3.3 Η μεταφορά του τηλεφωνικού καταλόγου- ατζέντα

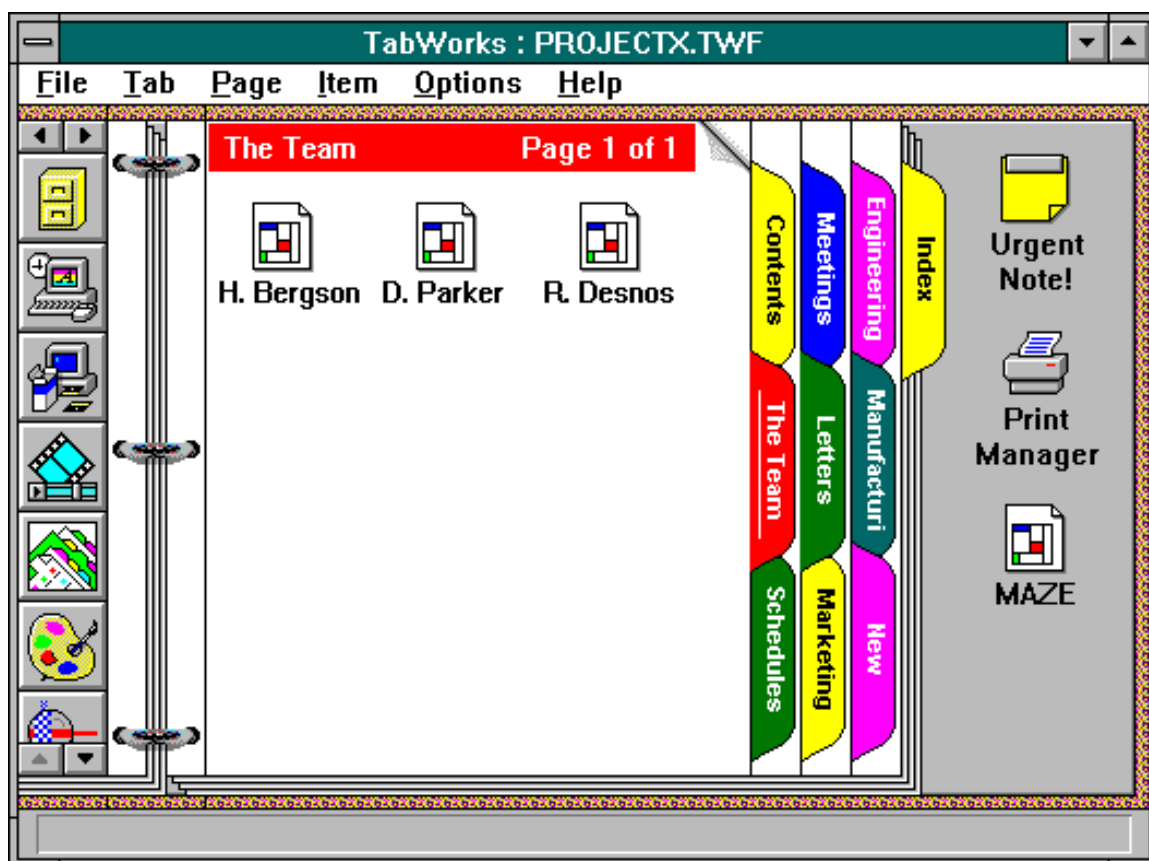
Μια επέκταση της μεταφοράς του βιβλίου, είναι η μεταφορά του τηλεφωνικού καταλόγου- ατζέντα. Όπως κάποιος μπορεί να ανατρέξει σε έναν τηλεφωνικό κατάλογο, με βάση ένα όνομα να αναζητήσει έναν τηλεφωνικό αριθμό, με, ακριβώς, τον ίδιο τρόπο μπορεί να αναζητήσει οποιοδήποτε είδους πληροφορία από μια εφαρμογή. Οι εφαρμογές που χρησιμοποιούν αυτή τη μεταφορά, οργανώνουν τις πληροφορίες με τη χρήση καρτελών, όπως ακριβώς στον τηλεφωνικό κατάλογο. Η

Compac για παράδειγμα έχει χρησιμοποιήσει τη μεταφορά για την οργάνωση των αρχείων ενός φορητού υπολογιστή (βλέπε Εικόνα 14).

Τέλος, κάποιες εφαρμογές ηλεκτρονικού εμπορίου προσπάθησαν να χρησιμοποιήσουν τη μεταφορά του τηλεφωνικού καταλόγου και να συνδέσουν το όνομα χρήστη (username) με το δημόσιο κλειδί του, για την παροχή κάποιου πιστοποιητικού αναγνώρισης. Σ' αυτήν την περίπτωση η μεταφορά απέτυχε λόγω έλλειψης ασφάλειας του συστήματος .

2.3.4 Το ξεφύλλισμα (browsing)

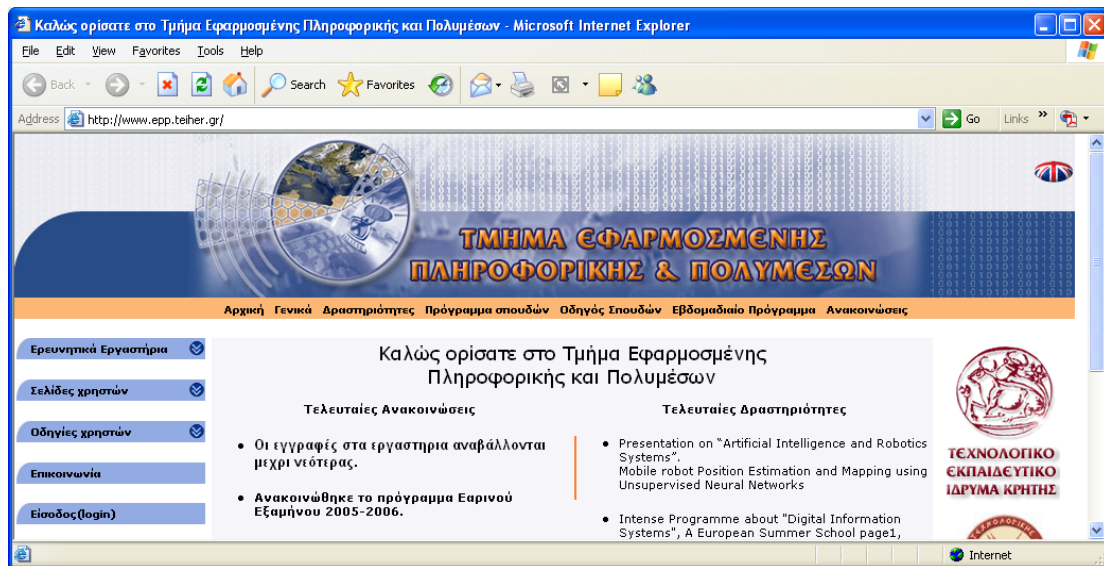
Ο παγκόσμιος ιστός ή World Wide Web (WWW) αναπτύχθηκε και εξακολουθεί να εξελίσσεται για να υποστηρίξει ένα πλούσιο σε περιεχόμενο περιβάλλον που ονομάστηκε 'κυβερνοχώρος'. Η ανάγκη πρόσβασης στο πλήθος των πληροφοριών έγινε υπαρκτή με την διάδοση και ευρύτατη αποδοχή του WWW που μεταφράζεται ακόμη και σήμερα σε πληθώρα δομημένων εγγράφων και πολυμεσικού περιεχομένου, κατανεμημένων σε διαφορετικά υπολογιστικά συστήματα και γεωγραφικά σημεία. Το πρόβλημα ωστόσο που δημιουργήθηκε ήταν η εκ' νέου πρόσβαση στις πληροφορίες αυτές που δεν είχαν συγκεκριμένη δομή.



Εικόνα 14: Η σχεδιαστική ιδέα της Compac

Σε μια προσπάθεια να υπάρξει κάποια μορφή δομής, τουλάχιστον στο λογισμικό πρόσβασης (browser), αφού κάτι τέτοιο δεν ήταν εφικτό σε επίπεδο περιεχομένου των εγγράφων, αξιοποιήθηκε η μεταφορά του ξεφυλλίσματος σελίδων με την χρήση

συνδέσμων (βλέπε Εικόνα 15). Έτσι, αναπτύχθηκαν γλώσσες σήμανσης εγγράφων, όπως η HTML, που επέτρεπαν τη σχεδίαση ενός δομημένου εγγράφου με σαφή διαχωρισμό μεταξύ περιεχομένου και παρουσίασης αλλά και προγράμματα πλοήγησης στον παγκόσμιο ιστό που μπορούσαν να ερμηνεύσουν τα ημι-δομημένα έγγραφα και να ‘μεταφέρουν’ τον χρήστη από ένα σημείο σε ένα άλλο μέσω των διασυνδέσεων. Η γενικευμένη διεπαφή χρήσης των εργαλείων αυτών στηρίχθηκε, στον κλασικό τρόπο ξεφυλλίσματος ενός βιβλίου με σαφείς αναφορές σε έννοιες όπως κεντρική σελίδα, αρχή και τέλος, μετακίνηση προς τα εμπρός και προς τα πίσω, ευρετήρια κα. Η μεταφορά του ‘ξεφυλλίσματος’, επιτρέπει και μια πρόχειρη αναζήτηση περιεχομένου αλλά και συγκεκριμένων αναζητήσεων μέσω των συνδέσεων υπερκειμένων και των μηχανών αναζήτησης.



Εικόνα 15: Παράδειγμα λογισμικού πλοήγησης

Για την υλοποίηση της μεταφοράς χρησιμοποιήθηκαν links(σύνδεσμοι), για την μετάβαση από τη μία σελίδα στην άλλη. Με τη χρήση του link η μετάβαση είναι πολύ εύκολη, αλλά με αυτόν τον τρόπο χάνεται η κλασική δομή της επόμενης σελίδας, καθώς το Link μπορεί να επιτρέψει την περιήγηση του χρήστη σε οποιοδήποτε σημείο, οποιασδήποτε σελίδας.

2.3.5 Το καλάθι του supermarket

Το ηλεκτρονικό εμπόριο είναι πλέον πραγματικότητα. Οι χρήστες θέλουν να πλοηγηθούν μέσα σε ένα ηλεκτρονικό μαγαζί μέχρι να αποφασίσουν τι θέλουν τελικά να αγοράσουν. Στο www.amazon.com, για παράδειγμα, ένας χρήστης μπορεί να διαλέξει ένα υποσύνολο βιβλίων που θέλει να αγοράσει μέσα από τα εκατοντάδες βιβλία που είναι διαθέσιμα. Μια λύση θα ήταν να ζητάει το σύστημα από το χρήστη να αγοράσει το βιβλίο την ίδια ώρα που το κοιτάει. Μια άλλη λύση θα ήταν να βάλει ένα σελιδοδείκτη σε κάθε προϊόν που τον ενδιαφέρει και να τα αγοράσει μέσω κάποιου άλλου συστήματος. Και οι δύο προσεγγίσεις δεν βοηθάνε όταν ο χρήστης θέλει να αγοράσει μια πλειάδα από διαφορετικά πράγματα και το σύστημα απαιτεί μια συναλλαγή για κάθε προϊόν. Επίσης αυτή η λύση έχει ένα άλλο μειονέκτημα, δεδομένου ότι πρέπει να “επισκεφτούμε” τη σελίδα “ελέγχων” πολλές φορές που σπαταλά έτσι το χρόνο σύνδεσης.

Τη λύση σε όλα αυτά ήρθε να δώσει η μεταφορά του καλάθιού αγορών. Ο χρήστης μπορεί με το πάτημα ενός μόνο κουμπιού να προσθέτει στο καλάθι του κάθε

αντικείμενο που τον ενδιαφέρει. Τελειώνοντας τις αγορές του μπορεί να ψάξει μέσα στο καλάθι του και να βρει λεπτομερείς περιγραφές για τα προϊόντα που έχει επιλέξει και έτσι να αποφασίσει πιο εύκολα και πιο γρήγορα. Έτσι θα χρειαστεί να “περάσει από το ταμείο” μία φορά και έχοντας στο καλάθι του μόνο τα προϊόντα που θέλει τελικά να αγοράσει.

Σε πολλές ιστοσελίδες η χρήση του καλαθιού έχει εμπλουτιστεί προσφέροντας στους χρήστες επιπλέον επιλογές όπως επικύρωση διαδικασιών, για αγορές εισιτηρίων, κρατήσεις ξενοδοχείων κτλ. αλλά και επιλογές όπως λίστα προϊόντων που θέλω να αγοράσω στο μέλλον.

Το αποτέλεσμα της μεταφοράς είναι μια πολύ φυσική προσέγγιση δεδομένου ότι μοιάζει με τον τρόπο που οι άνθρωποι ψωνίζουν στο supermarket, προσθέτοντας τα προϊόντα στο καλάθι αγορών τους ενώ περπατούν.

2.3.6 Η εφημερίδα

Η εμφάνιση των εφημερίδων έχει αναπτυχθεί και έχει τυποποιηθεί μέσα από το πέρασμά της από τους αιώνες. Παρά τις κοινωνιολογικές διαφορές, οι εκδότες και οι συντάκτες από τα διαφορετικά μέρη του κόσμου μπορούν να συναντηθούν για να συζητήσουν το περιεχόμενο, το ρόλο, και την τεχνολογία των εφημερίδων- ακριβώς όπως οι αναγνώστες από τα διαφορετικά μέρη του κόσμου αγοράζοντας μία τοπική εφημερίδα ξέρουν αμέσως πώς να την διαβάσουν. Τα διαφορετικά στοιχεία της διεπαφής των εφημερίδων είναι γνωστά ως “μεταφορά της εφημερίδας”.

Η “μεταφορά εφημερίδας” παρέχει μια σειρά σελίδων, όπου κάθε σελίδα παρέχει στις πολλαπλάσιες στήλες, τις πολλαπλάσιες ιστορίες τους ποικίλους μεγέθους τίτλους, και τις ενσωματωμένες φωτογραφίες. Η πλοήγηση μέσω αυτού του “εγγράφου” αποτελείται από τη μετάβαση σε μια νέα σελίδα με “τη στροφή της σελίδας”, επιλέγοντας ένα τμήμα και πηδώντας σε εκείνο το τμήμα, ή ψάχνοντας ενδεχομένως για το επόμενο περιστατικό των λέξεων ή των φράσεων. Αν και οι περισσότεροι συντάκτες ειδήσεων βασίζουν την παρουσίαση στην επιλογή ενός τίτλου βασισμένο στην ειδηση, η υιοθέτηση μερικών χαρακτηριστικών γνωρισμάτων της μεταφοράς εφημερίδας, ιδιαίτερα η χρήση των στηλών και η πλαισίωση αυτών με φωτογραφίες και κείμενο στην μπροστινή σελίδα είναι τώρα κοινή. Αυτή η ενσωμάτωση μερικών χαρακτηριστικών γνωρισμάτων της εφημερίδας μπορεί να γίνει εύκολα κατανοητή για να προσελκύσει τους αναγνώστες. Η ερώτηση που θέλουμε να εξετάσουμε είναι η καταλληλότητα της μεταφοράς παρουσίασης, όχι για να τραβήξει την προσοχή αλλά για την πιο μακροπρόθεσμη ικανοποίηση του στόχου, τις ειδήσεις.

Οι παραδοσιακές εφημερίδες είναι περιορισμένες με διάφορους τρόπους ειδικά στην επικαιρότητα των πληροφοριών που παρουσιάζουν λόγω του ότι το εγγράφου δημοσιεύεται μια φορά την ημέρα. Ένας άλλος περιορισμός είναι το σχήμα της το οποίο είναι ίδιο για ολόκληρο το αναγνωστικό κοινό.

Οι ηλεκτρονικές εφημερίδες έχουν τη δυνατότητα να αντιμετωπίσουν και τους δύο αυτούς περιορισμούς. Οι ηλεκτρονικές εφημερίδες έχουν τη δυνατότητα να ενημερώνονται όποτε τα νέα άρθρα διατίθενται και όχι μόνο σε καθημερινή βάση. Στην πραγματικότητα οι ηλεκτρονικές εφημερίδες θα μπορούσαν να ενημερώνονται συνεχώς αν και αυτό μπορεί να μην είναι στην πραγματικότητα επιθυμητό δεδομένου του ότι μια από τις δυνάμεις της εφημερίδας είναι η δυνατότητά της να ενημερώσει και να παρουσιάσει σε βάθος τα γεγονότα, μια κάλυψη που δεν είναι δυνατή σε άλλα μέσα όπως η τηλεόραση ή το ραδιόφωνο.

Αφετέρου στους χρήστες του Internet έπειτα από κάποια αρχική διαδικασία εγγραφής, τους παρέχεται η δυνατότητα να επέμβουν στο περιεχόμενο των άρθρων, π.χ. σχολιάζοντας το μέσα σε κάποια ομάδα, να επιλέξουν την μορφή του κλπ.

2.4 Μεταφορές αλληλεπίδρασης και αντικείμενα συμπερίληψης

Το βασικό συμπέρασμα που μπορεί να εξαχθεί από την προηγούμενη ανάλυση είναι ο κεντρικός ρόλος των αντικειμένων συμπερίληψης (containers) στη απόδοση μιας διεπαφής. Ο χρήστης μιας σύγχρονης διαδραστικής εφαρμογής αντιλαμβάνεται τον κόσμο του υπολογιστή χρησιμοποιώντας έννοιες από το περιβάλλον του. Όλα αυτά αποδίδονται σε μια εφαρμογή μέσω παραθύρων (για αυτό και η απόδοση της μεταφοράς είναι μέσω ενός παραθυρικού συστήματος). Πρακτικά αυτό σημαίνει ότι το υψηλού επιπέδου αντικείμενο συμπερίληψης (container) είναι το παράθυρο. Μια εναλλακτική απόδοση μη- παραθυρική είναι αυτή του ιδεογράμματος που συναντήσαμε προηγουμένως. Επομένως, η χρήση της μεταφοράς για το χαρακτηρισμό του συνολικού περιβάλλοντος χρήσης απαιτεί να 'πειράξουμε' υψηλού επιπέδου αντικείμενα συμπερίληψης (containers). Για παράδειγμα η υλοποίηση ενός διαδραστικού περιβάλλοντος του οποίου ο container είναι το δωμάτιο απαιτεί μελέτη του πώς αποδίδεται ένα δωμάτιο και στην συνέχεια υλοποίηση ενός room manager (κατ' αντιστοιχία με το window manager που έχουμε σε σύγχρονα συστήματα).

Προκειμένου να χρησιμοποιηθεί μια μεταφορά για το σχεδιασμό μιας εφαρμογής με έναν αποτελεσματικό τρόπο είναι σημαντικό οι χρήστες να αποκτούν ένα κατάλληλο γνωστικό πρότυπο σχετικά με αυτή. Λόγω της σημασίας της, η μεταφορά θα πρέπει να χρησιμοποιηθεί από τα αρχικά στάδια σχεδιασμού του συστήματος.

Οι μεταφορές είναι οι θεμελιώδεις έννοιες, οι όροι, και οι εικόνες από τις οποίες και μέσω των οποίων, οι πληροφορίες αναγνωρίζονται και κατανοούνται εύκολα. Οι μεταφορές περιλαμβάνουν τα ουσιαστικά μέσα, μέσω των οποίων απεικονίζονται οι επιλογές για την εντολή και τον έλεγχο της επικοινωνίας αλλά και η θέση όλων των στοιχείων και των λειτουργιών. Επειδή οι ηλεκτρονικές εφαρμογές μπορούν να μετασχηματιστούν σχετικά εύκολα και γρήγορα, αυτές οι μεταφορικές τεχνικές μπορούν να ποικίλουν στα συστήματα και να αλλάξουν κατά τη διάρκεια του χρόνου.

Τυπικά παραδείγματα μεταφορών που συνδέονται με γνωστά φυσικά αντικείμενα είναι τα παρακάτω:

- Desktop: Συρτάρια, αρχεία, φάκελοι, έγγραφα, συνδετήρες εγγράφου, φύλλα σημειώσεων
- Έγγραφο: Βιβλία, κεφάλαια, σελιδοδείκτες, αριθμοί εφημερίδες, τμήματα, περιοδικά, άρθρα, ενημερωτικά δελτία, μορφές
- Φωτογραφία: Λευκώματα, φωτογραφίες, κορνίζες
- Τηλεόραση: Προγράμματα, κανάλια, δίκτυα, αντιπρόσωποι
- Παιχνίδια: κανόνες παιχνιδιών, κομμάτια παιχνιδιών, πίνακες παιχνιδιών
- Ταινία: Ρόλοι, δίσκοι φωτογραφικών διαφανειών, επιδείξεις, κινηματογράφοι, θέατρα
- Δέντρο: Ρίζες, κορμός, κλαδιά, φύλλα

- Δίκτυο, διάγραμμα, χάρτης: κόμβοι, συνδέσεις, ορόσημα, περιοχές, ετικέτες, βάση (υπόβαθρο), μύθος
- Πόλεις: περιοχές, ορόσημα, διαβάσεις, κτήρια, δωμάτια, παράθυρα, γραφεία

2.5 Στάδια σχεδίασης διεπαφής με την χρήση μεταφοράς

Για το σχεδιασμό μιας διεπαφής με τη χρήση δεδομένης αναφοράς, ακολουθούνται τρία στάδια/ βήματα. Το πρώτο βήμα είναι ο προσδιορισμός ή η επιλογή της μεταφοράς που θα χρησιμοποιηθεί. Η επιλογή μιας μεταφοράς επηρεάζεται από το περιεχόμενο που πρόκειται να αποδοθεί μέσω της διεπαφής και κυρίως από τα λειτουργικά καθήκοντα που οι χρήστες καλούνται να επιτελέσουν. Παράλληλα, η επιλογή αυτή μπορεί να επηρεαστεί είτε από συστήματα υπολογιστών που ήδη διαθέτουν οι χρήστες, είτε από προϋπάρχουσες εμπειρίες. Σε κάθε περίπτωση, η μεταφορά που επιλέγεται θα πρέπει να έχει πλούσια δομή, και να είναι εύκολα κατανοητή ώστε να εξυπηρετήσει το σκοπό για τον οποίο επιλέγεται. Το πιο σημαντικό, όμως, είναι να είναι κατάλληλη για το κοινό στο οποίο θα απευθυνθεί, έτσι ώστε να μπορέσει να αποδώσει.

Αφού επιλεγεί η μεταφορά που θα χρησιμοποιηθεί, το επόμενο βήμα είναι η προσαρμογή της στα λειτουργικά καθήκοντα (σενάρια χρήσης), όπως αυτά έχουν αναπτυχθεί κατά τη διάρκεια της μελέτης του συστήματος. Μια διεπαφή που στηρίζεται σε κάποια μεταφορά δεν αντιγράφει πιστά το αντικείμενο της μεταφοράς, στηρίζεται σε αυτό σαν διεπαφή, αλλά προσαρμόζεται στις απαιτήσεις των χρηστών του, που είναι και οι τελικοί αποδέκτες. Σε αυτό το βήμα ο σχεδιαστής στηρίζεται σε επιλεγμένα σενάρια χρήσης και προσπαθεί να προσαρμόσει την κεντρική μεταφορά έτσι ώστε να εξυπηρετήσει καλύτερα τα σενάρια χρήσης, προσπαθώντας κατά το δυνατόν να περιορίσει την απόκλιση του σχεδίου από το κεντρικό νόημα της μεταφοράς.

Τέλος, στο τελευταίο στάδιο θα πρέπει να γίνει ο προσδιορισμός των διαφορών ανάμεσα στο αρχικό μοντέλο μεταφοράς και σε αυτό που έχει σχεδιαστεί καθώς και επιπτώσεις που έχουν οι διαφορές αυτές στους χρήστες. Οι διαφορές ανάμεσα στο αρχικό και στο τελικό μοντέλο είναι αναπόφευκτες σε όλα τα επίπεδα σχεδίασης. Οι ομοιότητες στην εμφάνιση είναι αυτό που δίνει την αμεσότητα στη μεταφορά. Προκειμένου το σύστημα όμως να ανταποκρίνεται στις απαιτήσεις των χρηστών αυτή η εμφάνιση θα χρειαστεί να αλλάξει. Να αλλάξει τόσο ώστε να μπορεί ο χρήστης να κάνει τη συσχέτιση ανάμεσα στα δύο συστήματα. Όταν όμως οι διαφορές βρίσκονται σε επίπεδο λειτουργικότητας, υπάρχει πρόβλημα. Η λύση του προβλήματος είναι να δώσει ο σχεδιαστής μέσω της διαφοράς ακόμα μια επιλογή στο χρήστη, μια εναλλακτική λύση για να δοκιμάσει- μάθει κάτι καινούριο, κάτι που δεν θα έβρισκε στο αρχικό μοντέλο. Να κάνει τη διαφορά πλεονέκτημα για το σύστημά του. Έτσι το σύστημα γίνεται ακόμα πιο λειτουργικό. Θα πρέπει όμως μέσα από συγκεκριμένα και όχι πολύπλοκα βήματα ο χρήστης να οδηγηθεί στις εναλλακτικές προτάσεις για να κατανοήσει τις καινούριες λειτουργίες του συστήματος.

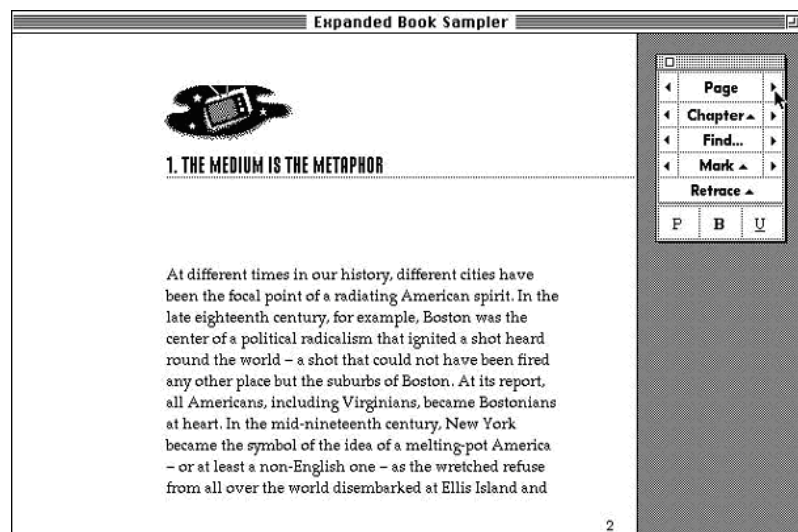
3. Σχεδίαση του JBook

3.1 Η μεταφορά του βιβλίου

Η ιστορία του βιβλίου ξεκινά από την Πέργαμο, μία πόλη της Δυτικής Τουρκίας. Εκεί ο βασιλιάς θέλοντας να φτιάξει μια βιβλιοθήκη μεγαλύτερη από αυτή στην Αλεξάνδρεια ήρθε σε σύγκρουση με τους Αιγύπτιους, οι οποίοι σταμάτησαν να προμηθεύουν την Πέργαμο με πάπυρους. Έχοντας μια πλούσια βιομηχανία μαλλιού άρχισαν να γράφουν πάνω σε δέρματα προβάτων. Αυτό έμεινε γνωστό ως περγαμινή. Επειδή ήταν δύσκολο να γίνει ρολό η περγαμινή σκέφτηκαν να δημιουργήσουν ορθογώνιες σελίδες και να τις ράψουν μαζί. Έτσι κατάφεραν να φτιάξουν το βιβλίο. Την τελική μορφή στο βιβλίο, αυτή που γνωρίζουμε μέχρι σήμερα, την έδωσε ο Γουτεμβέργιος.

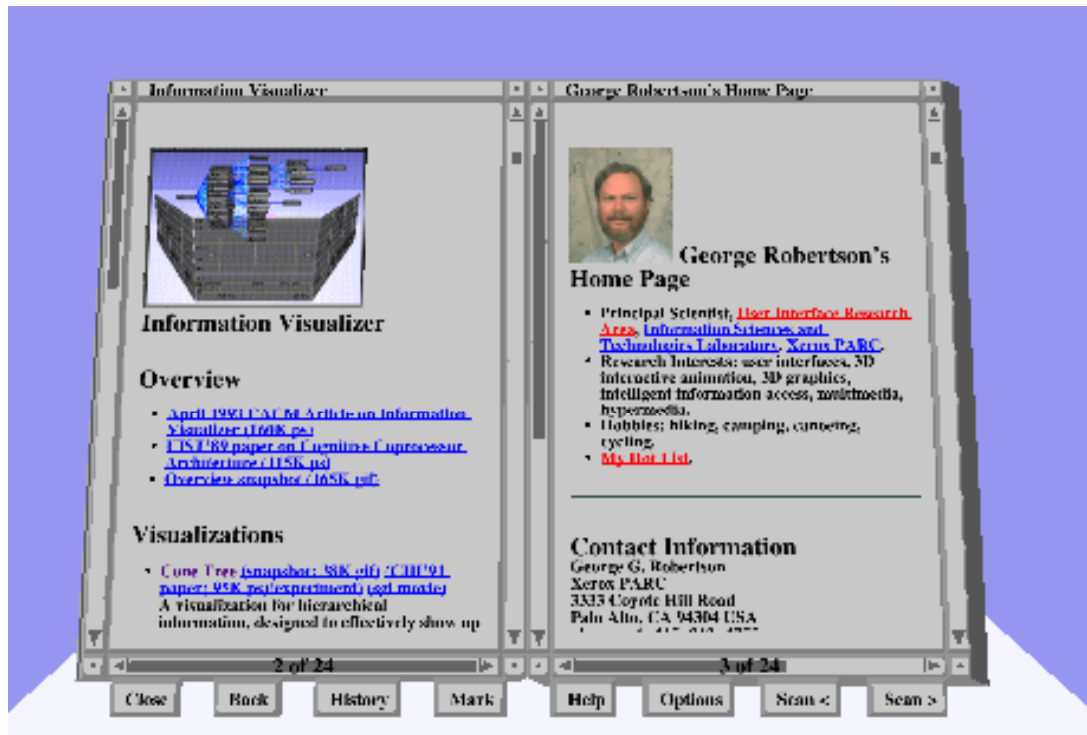
Αφήνοντας πίσω το κλασσικό βιβλίο, συναντάμε τη μεταφορά αυτού. Ένα βιβλίο αποτελείται από σελίδες δεμένες μεταξύ τους που περιβάλλονται από εξώφυλλα. Η μεταφορά του βιβλίου θα πρέπει να δίνει στον χρήστη την εικόνα αυτή. Με άλλα λόγια η μεταφορά παρέχει ένα πλαίσιο για το σχέδιο των εικόνων.

Κύρια χαρακτηριστικά αυτής της μεταφοράς είναι μια κύρια σελίδα με τα πλήκτρα πλοήγησης, τα οποία αντιπροσωπεύουν το γύρισμα της σελίδας. Άλλα χαρακτηριστικά της μεταφοράς είναι η αρίθμηση των σελίδων, μια δομή κεφαλαίων, η αναζήτηση όρων κλπ. Μία τέτοιου είδους μεταφορά δίνει στο χρήστη τη δυνατότητα να επέμβει στο περιεχόμενο του βιβλίου π.χ. να υπογραμμίσει κάτι που τον ενδιαφέρει ή να το τονίσει, χωρίς όμως αυτή η αλλαγή να επηρεάσει μόνιμα τη μορφή του κειμένου, καθώς μπορεί να αναιρέσει τις ενέργειες αυτές και να επιστρέψει το βιβλίο στην κανονική του μορφή.



Εικόνα 16 : Το ηλεκτρονικό βιβλίο

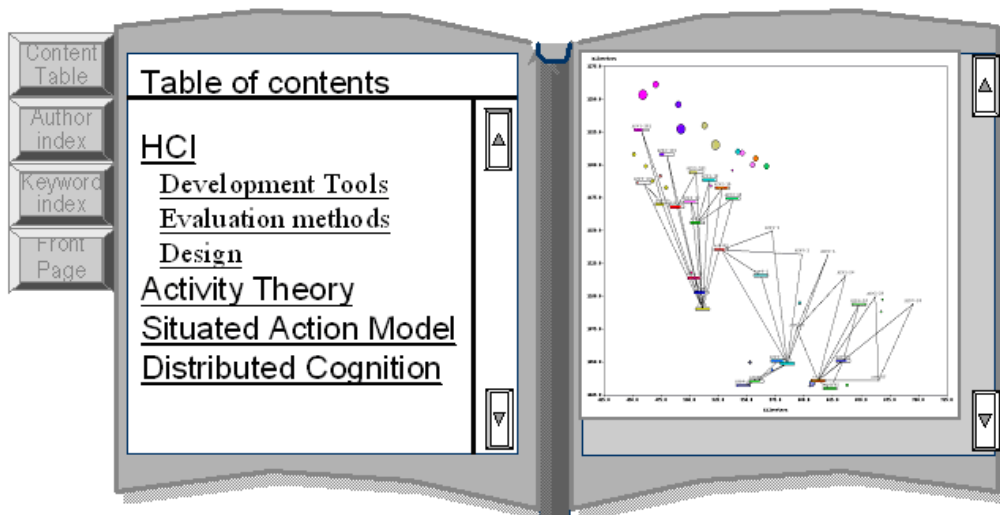
Ένας μεγάλος αριθμός εφαρμογών που ενσωματώνουν τις μεταφορές βιβλίων αναφέρονται συχνά ως "ηλεκτρονικά βιβλία" (βλέπε Εικόνα 16). Παρατηρούμε ότι έννοιες από το πεδίο αναφοράς, όπως σελίδα, κεφάλαιο αποτελούν βασικά στοιχεία της διεπαφής. Μια άλλη υλοποίηση που έχει παρουσιαστεί στην σχετική βιβλιογραφία είναι αυτή της Εικόνα 17. Σε αυτή την περίπτωση η μεταφορά του βιβλίου έχει χρησιμοποιηθεί για την οργάνωση ιστοσελίδων σε τρισδιάστατη μορφή ενός βιβλίου. Κάθε σελίδα του βιβλίου είναι μια ξεχωριστή ιστοσελίδα που παρουσιάζεται με τον παραδοσιακό τρόπο. Οι συνδέσεις με άλλες σελίδες του ίδιου WebBook κωδικοποιούνται με διαφορετικό χρώμα από εκείνες που δείχνουν σε εξωτερικές σελίδες (κόκκινες και μπλε αντίστοιχα). Επιλέγοντας μία κόκκινη σύνδεση το βιβλίο θα "γυρίσει" σελίδα και ο χρήστης μπορεί να έχει πρόσβαση στην επόμενη. Επιλέγοντας μία μπλε σύνδεση το συγκεκριμένο βιβλίο θα κλείσει και θα ανοίξει ένα άλλο που περιέχει τη σελίδα που ζητήθηκε. Επίσης, ο χρήστης έχει τη δυνατότητα να πλοηγηθεί μέσα στο βιβλίο με τη χρήση των αντίστοιχων κουμπιών πλοήγησης, καθώς και να τοποθετήσει σελιδοδείκτες στις σελίδες που τον ενδιαφέρουν. Τέλος, μια ακόμη εναλλακτική έκδοση που διαφοροποιεί τις δύο σελίδες επιτρέποντας την κάθε μια να υποστηρίζει διαφορετική μεταφορά είναι αυτή που παρουσιάζεται στην Εικόνα 18.



Εικόνα 17 : Το Webbook των Card et al. 1996

3.2 Ανάλυση ενός κλασικού βιβλίου

Το συμβατικό βιβλίο όπως όλοι το ξέρουμε είναι μια συλλογή από σελίδες χαρτιού πάνω στα οποία υπάρχει γραμμένο κείμενο. Οι σελίδες αυτές περιβάλλονται από το εξώφυλλο και το οπισθόφυλλο. Το περιεχόμενο του βιβλίου, δηλαδή οι σελίδες στις οποίες έχει γραφτεί το κείμενο χωρίζονται σε κατηγορίες ώστε να έχουμε μια ταξινόμηση της δομής του βιβλίου. Οι κατηγορίες, τα συστατικά του βιβλίου, είναι οι σελίδες, οι γραμμές, οι παράγραφοι, τα κεφάλαια, το ευρετήριο, τα περιεχόμενα καθώς και το εξώφυλλο και το οπισθόφυλλο.



Εικόνα 18 : Παράδειγμα εναλλακτικής απόδοσης της μεταφοράς του βιβλίου

Αρχίζοντας από το πρώτο πράγμα που παρατηρούμε στο βιβλίο, το εξώφυλλο, αρχικά να πούμε πως σε συνδυασμό με το οπισθόφυλλο χρησιμοποιούταν για εκατοντάδες χρόνια κυρίως για να κρατάνε συνδεδεμένες τις σελίδες μεταξύ τους και να τις προστατεύουν. Για αυτόν ακριβώς τον λόγο ήταν συνήθως φτιαγμένα από σκληρό υλικό όπως δέρμα ή χαρτόνι. Καθώς περνούσαν τα χρόνια το εξώφυλλο άρχισε να γίνεται πιο σημαντικό και να περιέχει πληροφορίες για το στίλ, τον τύπο και το θέμα του βιβλίου ενώ ταυτόχρονα πολλοί έκαναν σχεδιαστικές ακρότητες ώστε με αυτόν τον τρόπο να προσελκύσουν αγοραστές. Όσον αφορά το οπισθόφυλλο χρησιμοποιείτε κυρίως για να δώσει μια μικρή περιγραφή του θέματος του βιβλίου και καμία φορά να μας ενημερώσει σχετικά με το συγγραφέα του βιβλίου.

Συνεχίζοντας, στο εσωτερικό του βιβλίου υπάρχουν συνήθως τα περιεχόμενα. Τα περιεχόμενα είναι μία λίστα με τα μέρη του βιβλίου κατηγοριοποιημένα ανάλογα με το πού αυτά εμφανίζονται. Συνήθως έχουμε την περιγραφή τους και δίπλα της την σελίδα στην οποία αυτή εμφανίζεται. Τα περιεχόμενα συνήθως περιέχουν τους τίτλους ή της περιγραφές των εκάστοτε κεφαλαίων και καμία φορά τις επικεφαλίδες των υποκεφαλαίων. Το επίπεδο της λεπτομέρειας της λίστας των περιεχομένων εξαρτάτε από το επίπεδο των διαβαθμίσεων του βιβλίου.

Κάτι παρόμοιο με τα περιεχόμενα είναι και το ευρετήριο. Το ευρετήριο είναι μια αναλυτική λίστα, η οποία είναι συνήθως ταξινομημένη αλφαβητικά, αποτελούμενη από συγκεκριμένες πληροφορίες οι οποίες βρίσκονται μέσα στο βιβλίο. Οι πληροφορίες αυτές μπορεί να είναι από απλές λέξεις, μέχρι και συγκεκριμένες φράσεις. Το ευρετήριο λειτουργεί ως μηχανισμός πλοήγησης και σχεδιάζεται ώστε να βοηθάει τον αναγνώστη να βρίσκει γρήγορα και εύκολα τις πληροφορίες που αναζητεί. Ιδανικά θα μπορούσαμε να πούμε ότι το ευρετήριο δεν είναι απλά μια λίστα με τους σημαντικότερους όρους που περιέχει το βιβλίο αλλά ένας οργανωμένος χάρτης των περιεχομένων του που εμπεριέχει πολλαπλά στοιχεία τα οποία θα εξυπηρετήσουν τον αναγνώστη κατά την ανάγνωση.

Ένα από τα σημαντικότερα συστατικά του βιβλίου είναι τα κεφάλαια. Τα κεφάλαια χωρίζουν το βιβλίο σε τμήματα όμοια μεταξύ κυρίως σε θεματολογία. Με αυτό τον τρόπο διευκολύνεται ο αναγνώστης καθώς έχουμε μια βασική κατηγοριοποίηση των θεμάτων του βιβλίου της οποίας ο αναγνώστης γίνεται γνώστης με βάση τα κεφάλαια.

Πολλά βιβλία εκτός από τα κεφάλαια έχουν και υποκεφάλαια. Αυτά δεν είναι τίποτα άλλο από υποκατηγορίες μέσα σε κάποιο κεφάλαιο που χρίζουν ιδιαίτερης προσοχής ώστε να πρέπει να κατηγοριοποιηθούν εκ νέου. Το σε πόσο βάθος θα εμφανίζονται υποκεφάλαια εξαρτάτε από το περιεχόμενο και τη πολυπλοκότητα του βιβλίου.

Προχωρώντας βαθύτερα στη δομή ενός βιβλίου βρίσκουμε τις παραγράφους. Η παράγραφος είναι υποσύνολο του κεφαλαίου. Είναι μια ανεξάρτητη μονάδα κειμένου η οποία παρουσιάζει μια συγκεκριμένη θέση ή ιδέα. Οι παράγραφοι μεταξύ τους συνήθως χωρίζονται με μια γραμμή κενό απόσταση η μία από την άλλη. Συνήθως η πρώτη γραμμή της παραγράφου είναι αυτή που ορίζει το θέμα της καθώς και τη θέση του συγγραφέα πάνω στην οποία θα αναφερθεί στην παράγραφο.

Τέλος όλα αυτά τα συστατικά του βιβλίου εμπεριέχονται στις σελίδες. Οι σελίδες είναι τα κομμάτια χαρτιού πάνω στα οποία είναι γραμμένες οι σειρές (προτάσεις) του βιβλίου. Ανάλογα με το πλήθος των σελίδων που έχουμε είναι και το μέγεθος του βιβλίου.

3.3 Απόδοση της κλασικής μεταφοράς σε Η/Υ

Ο τρόπος με τον οποίο θα αποδοθούν τα στοιχεία του πραγματικού βιβλίου στην υλοποίηση μέσω υπολογιστή είναι ιδιαίτερος σημαντικός καθώς αυτό θα κρίνει το πόσο καλή θα είναι η διαδραστικότητα της διεπαφής του βιβλίου με τον χρήστη του προγράμματος. Στη διεπαφή του JBook προσπαθήσαμε να κρατήσουμε όλες τις βασικές αρχές της διεπαφής ενός βιβλίου καθώς και της πλοήγησης μας σε αυτό όσο το δυνατόν πιο κοντά στην πραγματικότητα. Το βιβλίο μας έχει αρχικά ένα εξώφυλλο το οποίο παρουσιάζει μια σύντομη περιγραφή του βιβλίου, τον τίτλο, συνοδευμένο από μία εικόνα. Με το πάτημα ενός κουμπιού το οποίο βρίσκετε στο εικονικό εξώφυλλο μας περνάμε στο εσωτερικό του βιβλίου μας.

Με το άνοιγμα του βιβλίου μας βλέπουμε μια διεπαφή που μας θυμίζει το γνωστό κλασικό μας βιβλίο. Συγκεκριμένα, παρουσιάζονται δύο σελίδες ανοιχτές σαν να είχαμε το βιβλίο μας τοποθετημένο, ανοιχτό, σε ένα γραφείο και να το κοιτάζαμε από πάνω. Η πλοήγηση στο εικονικό αυτό βιβλίο υποστηρίζεται από δομές που προσφέρουν:

- Προεπισκόπηση όλου του περιεχομένου
- Προεπισκόπηση ενεργών συνδέσμων
- Προεπισκόπησης και γρήγορης πρόσβασης σε κεφάλαια, σχήματα, κλπ
- Φυλλομέτρησης (ξεφύλλισμα από σελίδα σε σελίδα και προς τις δύο κατευθύνσεις)
- Γρήγορης πρόσβασης σε συγκεκριμένη σελίδα

Το ξεφύλλισμα είναι ιδιαίτερα απλό καθώς στις γωνίες τις κάθε μιας από τις δυο σελίδες μπορούμε να επιλέξουμε να μεταφερθούμε αναλόγως στην επόμενη ή την προηγούμενη σελίδα. Στην πραγματικότητα με το πάτημα του κουμπιού πλοήγησης αλλάζει το περιεχόμενο της κάθε εικονικής σελίδας και όχι η σελίδα αυτή καθ' αυτή.

Το γεγονός ότι έχουμε να κάνουμε με μία ηλεκτρονική μεταφορά της διεπαφής του βιβλίου μας δίνει την δυνατότητα να προσθέσουμε μερικές διευκολύνσεις που υπό πραγματικές συνθήκες στο γνωστό σε όλους μας κλασικό βιβλίο δεν είναι δυνατόν να υποστηριχθούν. Έτσι λοιπόν έχουμε προσθέσει δύο επιπλέον κατηγορίες λειτουργιών που παρέχουν χρηστικές υπηρεσίες στον αναγνώστη του βιβλίου μας. Η μία κατηγορία λειτουργιών αφορά την πλοήγηση στο βιβλίο και που ουσιαστικά

αποσκοπεί στο να κάνει διαθέσιμα στον αναγνώστη με ένα κλικ τα περιεχόμενα και το ευρετήριο όρων. Σε επίπεδο επισκόπησης περιεχομένων, το ηλεκτρονικό βιβλίο υποστηρίζει μια κατηγοριοποιημένη λίστα με τα μέρη του. Η διαφορά είναι ότι ο χρήστης έχει άμεση πρόσβαση με ένα κλικ σε κάθε μέρος του βιβλίου που υποδεικνύουν τα περιεχόμενα. Την ίδια ακριβώς ιδιότητα παρέχει και το Ευρετήριο Όρων. Με αυτές τις διευκολύνσεις και η διεπαφή μας δεν αποκλίνει από την παραδοσιακή και σε όλους μας γνωστή εμφάνιση του βιβλίου αλλά και παράλληλα κάνει την πλοήγηση του χρήστη σε αυτό ακόμα πιο εύκολη και απλή. Να συμπληρώσουμε επίσης ότι στην ανά πάσα στιγμή είναι ορατή η σελιδομέτρηση. Ο χρήστης βλέπει καθ' όλη την διάρκεια της πλοήγησης του στο βιβλίο σε ποια σελίδα βρίσκεται. Επίσης υπάρχει και μία τρίτη επιλογή για τον χρήστη, αυτή της αλλαγής της εμφάνισης των περιεχομένων του βιβλίου με στόχο την υποστήριξη στοιχειώδους προσαρμοστικότητας. Με αυτή την επιλογή ο χρήστης μπορεί να επιλέξει μια διαφορετική εμφάνιση των δεδομένων (πχ μικρότερη γραμματοσειρά) που ίσως τον διευκολύνει στην ανάγνωση.

Η δεύτερη κατηγορία λειτουργιών αφορά υποστηρικτικά καθήκοντα όπως σχολιασμό του περιεχομένου και πληροφορίες σχετικά με τον συγγραφέα. Αυτό το κομμάτι ξεφεύγει κάπως από την παραδοσιακή ιδέα του βιβλίου. Αντιλαμβανόμενοι το γεγονός ότι το ηλεκτρονικό βιβλίο είναι ένα πρόγραμμα στον υπολογιστή δεν μπορούμε να παραλείψουμε το κουμπί της βοήθειας στο οποίο θα ανατρέχει ο χρήστης ώστε να μπορεί να λύσει όποιες τυχόν απορίες και προβλήματα του δημιουργηθούν κατά την χρήση. Τα σχόλια είναι κάτι το πολύ χρήσιμο για τους περισσότερους αναγνώστες βιβλίων. Δεν είναι λίγες οι φορές που διαβάζοντας ένα βιβλίο θέλουμε να σημειώσουμε κάτι, μια παραίτηση, μια επισήμανση για αυτό που διαβάσαμε. Έχουμε λοιπόν την δυνατότητα να το κάνουμε αυτό, το πώς θα το αναλύσουμε στη συνέχεια. Τέλος το κουμπί του συγγραφέα μας δίνει κάποιες γενικές πληροφορίες για το συγγραφέα του βιβλίου το οποίο διαβάζουμε. Αυτό δεν διαφέρει ιδιαίτερα με τα κλασικά βιβλία, απλά όπως και τα περιεχόμενα αλλά και το ευρετήριο όρων είναι προσβάσιμο με ένα μόλις κλικ.

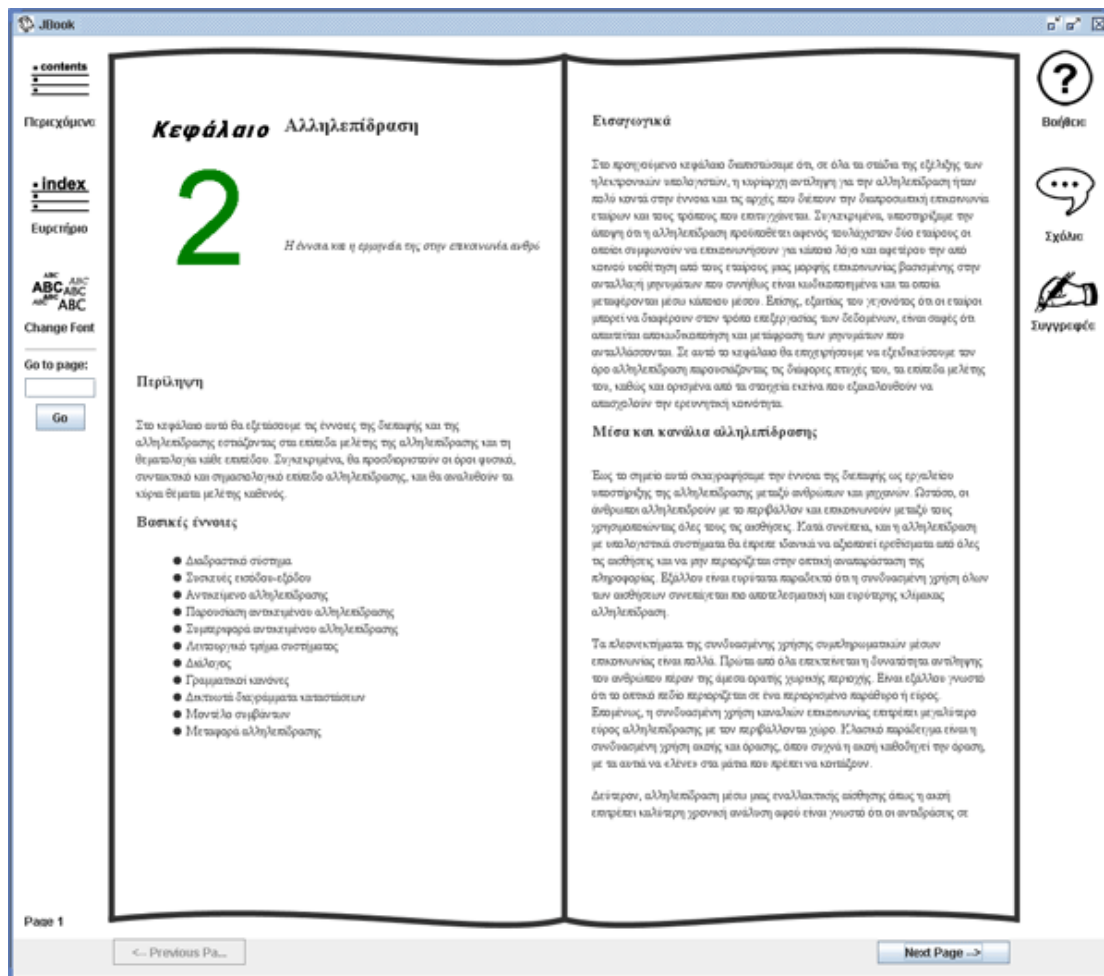
Τα περισσότερα εσωτερικά τμήματα της δομής ενός βιβλίου, δηλαδή οι παράγραφοι, οι σελίδες, τα κεφάλαια, τα υποκεφάλαια και οι γραμμές δεν διαφέρουν σε σχέση με το κανονικό βιβλίο. Άλλωστε και κάποιος που μπορεί να μην είναι εξοικειωμένος με προγράμματα ηλεκτρονικού υπολογιστή είναι πολύ εύκολο να τα αντιληφθεί.

4. Υλοποίηση του JBook και σενάρια επίδειξης

Η υλοποίηση του JBook ξεκίνησε έχοντας ως δεδομένο ότι το βιβλίο μας θα αποτελείται από δυο σελίδες αυτόνομες μεταξύ τους. Πρακτικά λοιπόν θα έχουμε έναν container για κάθε σελίδα. Έτσι δημιουργήθηκαν δυο JPanel (leftPage και rightPage) τα οποία αποτελούν τις δύο σελίδες μας. Επιλέξαμε το JPanel γιατί θα μας διευκολύνει στο επόμενο μας βήμα το οποίο είναι ο σχεδιασμός του γραφικού του βιβλίου.

Με τον σχεδιασμό του γραφικού του βιβλίου ουσιαστικά προσπαθήσαμε να αποτυπώσουμε σε μία εικόνα την μορφή ενός βιβλίου η οποία θα φιλοξενείται σε ένα γραφικό παράθυρο. Ωστόσο, η αντιστοίχιση του βιβλίου σε παράθυρο το οποίο διαθέτει συγκεκριμένες λειτουργίες διαχείρισης απαιτεί την εναρμόνιση της παρουσίασης του βιβλίου με την εκάστοτε κατάσταση που μπορεί να βρεθεί το παράθυρο (π.χ. τα φυσικά του χαρακτηριστικά όπως μέγεθος, θέση) ως αποτέλεσμα της εφαρμογής κάποιων λειτουργιών διαχείρισης (π.χ. ελαχιστοποίηση, μεγέθυνση, αλλαγή θέσης, αλλαγή μεγέθους, κλπ). Έτσι, στόχος από την αρχή ήταν η εμφάνιση του βιβλίου να είναι ανεπηρέαστη από τυχόν αλλαγές στο μέγεθος του παραθύρου που περιέχει το βιβλίο μας. Για αυτόν τον λόγο ο σχεδιασμός θα έπρεπε να γίνει με κάποιον τρόπο ώστε να υπακούει στις αλλαγές που γίνονται στο μέγεθος του παραθύρου. Επίσης από τη στιγμή που οι δυο μας σελίδες είναι αυτόνομες μεταξύ τους θα έπρεπε να σχεδιάσουμε το γραφικό της κάθε σελίδας ξεχωριστά. Για καλή μας τύχη η Java που είναι η γλώσσα προγραμματισμού που επιλέξαμε για την υλοποίηση του προγράμματος, μας παρέχει αρκετές διευκολύνσεις για τον σχεδιασμό δυσδιάστατων γραφικών.

Η πρωτότυπη διεπαφή που σχεδιάσαμε για το βιβλίο συνοψίζεται στην Εικόνα 19. Ακολούθως θα περιγράψουμε τα συστατικά της διεπαφής καθώς και τον τρόπο που υλοποιήθηκαν έτσι ώστε να έχουμε το αποτέλεσμα που παρουσιάζεται στην παραπάνω εικόνα. Θα αρχίσουμε με την λεπτομερή περιγραφή των αντικειμένων συμπερίληψης και θα εστιάσουμε στην περιγραφή της δεξιάς σελίδας. Η αριστερή είναι όμοια με την δεξιά με μια μικρή διαφοροποίηση στις μεταβλητές του αλγορίθμου ώστε να βγει ουσιαστικά καθρεφτισμένη με την δεξιά σελίδα μας. Για την πληρέστερη περιγραφή θα αξιοποιήσουμε αποσπάσματα του κώδικα που αναπτύχθηκε ο οποίος και θα αποτελεί το σημείο αναφοράς και περιγραφής όλων των λειτουργιών της διεπαφής του βιβλίου.



Εικόνα 19: Αρχική μορφή του βιβλίου μας

3.4 Υλοποίηση σελίδων βιβλίου

Για να μπορέσουμε να έχουμε μια δυναμική αλλαγή του περιγράμματος του βιβλίου μας σε σχέση με τις αλλαγές στο μέγεθος του παραθύρου μας έπρεπε να έχουμε μια μεταβλητή στις συναρτήσεις σχεδιασμού δυσδιάστατων γραφικών της Java που να επηρεάζονται από αυτό. Επιλέξαμε τις έτοιμες συναρτήσεις της Java `getSize().width` και `getSize().height`; μέσα από τις οποίες έχουμε άμεση γνώση του μεγέθους των `JPanel` και μπορούμε να σχεδιάσουμε αναλόγως τα περιγράμματα του βιβλίου μας (βλέπε Απόσπασμα 1).

Απόσπασμα 1

```
final class drawLeftPage extends JPanel { //drawing stuff, to
    elaborate to explain
    public void paint (Graphics g)
    {
        super.paintComponent (g) ;
        int w = getSize ().width*2;
        int h = getSize ().height;
        ...
    }
}
```

Στη συνέχεια για να σχεδιάσουμε την πάνω καμπύλη της σελίδας του βιβλίου (βλέπε Εικόνα 20) χρησιμοποιήσαμε την συνάρτηση κυρτότητας `CubicCurve2D` η οποία παίρνει 6 μεταβλητές. `CubicCurve2D.Double (double x1, double y1, double ctrlx1,`

double ctrly1, double ctrlx2, double ctrly2, double x2, double y2). Οι ctrlx μεταβλητές είναι αυτές που ορίζουν το σημείο που θα γίνει η κυρτότητα.



Εικόνα 20: Κυρτότητα της σελίδας του βιβλίου

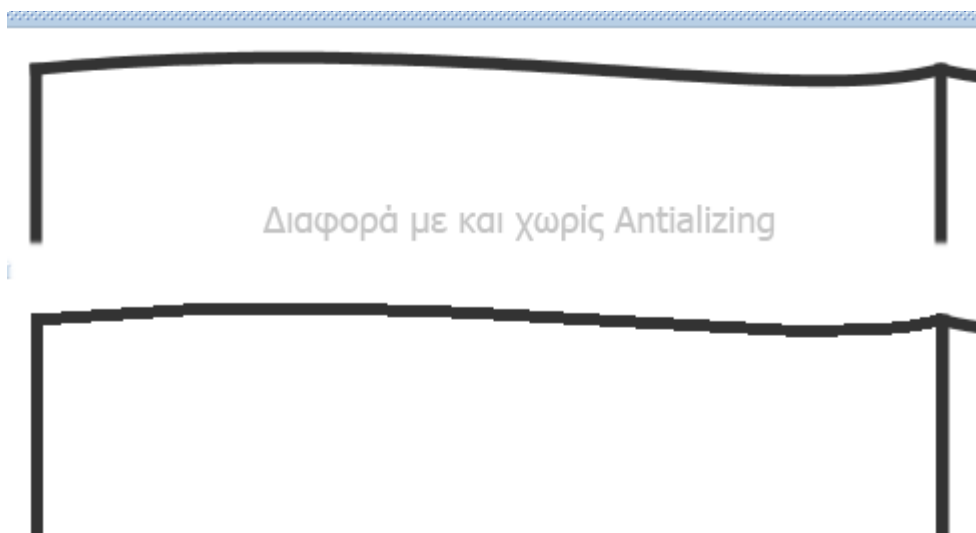
Απόσπασμα 2

```
CubicCurve2D cubicTopLeft = new CubicCurve2D.Double (3,
    20,
    w / 2 - 0.27*w,
    0,
    w / 2 - 0.08*w,
    40,
    w / 2,
    20)
CubicCurve2D cubicBottomLeft = new CubicCurve2D.Double (3,
    h - 20,
    w / 2 - 0.27*w,
    h-40,
    w / 2 - 0.08*w,
    h,
    w / 2,
    h - 20);
Line2D leftLine = new Line2D.Double(3,20,3, h -20);
Line2D centerLine = new Line2D.Double(w / 2,20,w / 2,h -20);
// antialiasing on
RenderingHints renderHints =
new RenderingHints(RenderingHints.KEY_ANTIALIASING,
    RenderingHints.VALUE_ANTIALIAS_ON);
renderHints.put(RenderingHints.KEY_RENDERING,
    RenderingHints.VALUE_RENDER_QUALITY);
// end of antialiasing

Graphics2D g2 = (Graphics2D) g;
g2.setStroke(new BasicStroke(6)); //line thickness
g2.setRenderingHints(renderHints); //turn antialiasing on
super.paintComponent(g); //cleans off-
screen bitmap
g2.draw (cubicTopLeft);
g2.draw (cubicBottomLeft);
g2.draw (leftLine);
g2.draw (centerLine);
paintChildren(g);
}
}
```

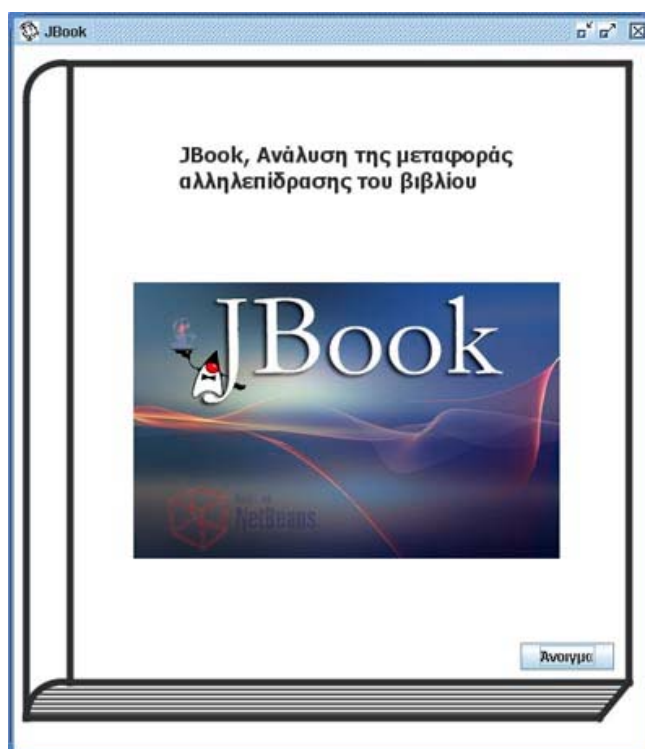
Στο Απόσπασμα 2 συνοψίζεται η υλοποίηση της συγκεκριμένης συνάρτησης. Όπως βλέπουμε από τον κώδικα η συνάρτηση περιέχει τις μεταβλητές x και y που είναι οι δυναμικές μας μεταβλητές αποτέλεσμα των συναρτήσεων Java `getSize ().width` και

`getSize ().height`. Με την συνάρτηση `Line2D` σχεδιάσαμε τις απαραίτητες ευθείες για την ολοκλήρωση του περιγράμματος μας. Σημαντικό παράγοντα στην απαλή εμφάνιση των δυσδιάστατων γραφικών μας έπαιξε η ενεργοποίηση του `antialiasing` πάνω σε αυτά. Με αυτόν τον τρόπο έχουμε μια πιο ομαλή εμφάνιση των γραφικών και εξαλείφεται το λεγόμενο πιξέλιασμα (βλέπε Εικόνα 21:).



Εικόνα 21: Εμφάνιση διαφοράς γραφικών με την χρήση της μεθόδου `antialiasing`

Τέλος πρέπει να πούμε πως για να έχει το `JPanel` μας την δυνατότητα να εμφανίσει μετά το 'ζωγράφισμα' των γραφικών που του ορίσαμε και άλλα `components` είναι απαραίτητο να γράψουμε την τελευταία γραμμή κώδικα `paintChildren(g)`. Με παρόμοια διαδικασία δημιουργήσαμε και το γραφικό για την δεξιά σελίδα. Με βάση την ίδια μεθοδολογία δημιουργήσαμε και το αρχικό τμήμα του βιβλίου, το εξώφυλλο (βλέπε Εικόνα 22:).

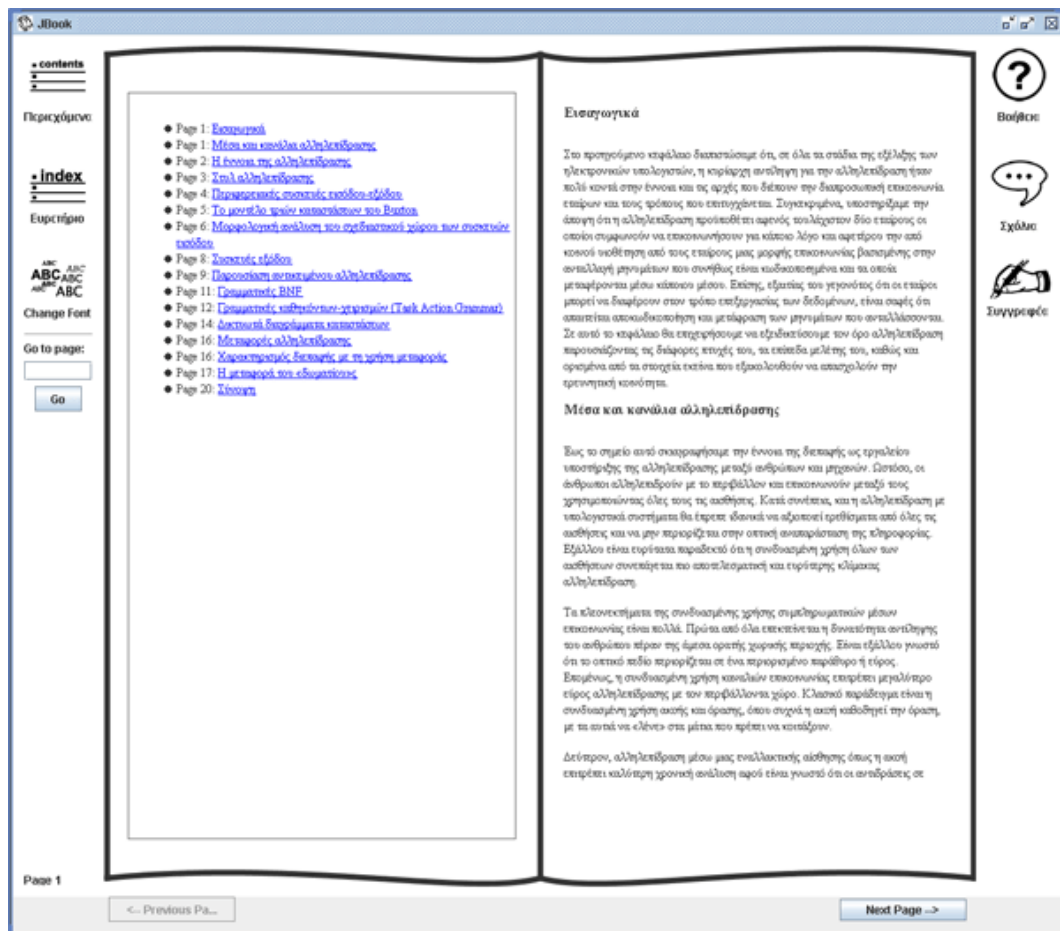


Εικόνα 22: Εξώφυλλο βιβλίου

3.5 Συστατικά παρουσίασης περιεχομένων & λειτουργιών

Αφού έχουμε δημιουργήσει το επιθυμητό γραφικό που δίνει στα JPanel μας την εμφάνιση του βιβλίου στη συνέχεια τους τοποθετούμε από ένα JEditorPane που θα είναι και το συστατικό το οποίο θα εμφανίζει τα περιεχόμενα των σελίδων μας. Εδώ πρέπει να επισημάνουμε πως την συμπεριφορά των JEditorPane μας, αλλά και όλων των συστατικών του Swing της Java, ανάλογα με τις αλλαγές στο μέγεθος του βιβλίου μας τις χειρίζεται το νέο Free layout που εισάγεται στη Java με το νέο JDK 5.0. Η χρήση του είναι εξαιρετικά απλή και εύχρηστη μέσω του εργαλείου Matisse GUI Builder που βοηθά στην δημιουργία των δυσδιάστατων συστατικών του swing καθώς και στον έλεγχο τους.

Έχοντας κάνει αυτά τα απλά βήματα έχουμε υλοποιήσει το βασικό κομμάτι του εσωτερικού του βιβλίου μας. Στη συνέχεια τοποθετήσαμε δίπλα από κάθε σελίδα μας ένα JPanel στον οποίο περιέχονται μερικές από τις λειτουργίες του βιβλίου μας. Στο αριστερό JPanel έχουν τοποθετηθεί οι επιλογές εμφάνισης των περιεχομένων του βιβλίου, του ευρετηρίου, της αλλαγής μεγέθους γραμματοσειράς καθώς και η επιλογή της γρήγορης μετάβασης στη σελίδα της επιλογής μας. Τέλος στο κάτω μέρος του δεξιού JPanel μας βρίσκετε η ένδειξη της σελίδας στην οποία βρισκόμαστε. Στο αριστερό JPanel βρίσκονται οι επιλογές της βοήθειας, των σχόλιων και του συγγραφέα. Στη συνέχεια θα εξετάσουμε τι και πώς κάνουν οι προαναφερθείσες επιλογές.



Εικόνα 23: Εμφάνιση περιεχομένων

Η επιλογή των περιεχομένων εμφανίζει τα περιεχόμενα του βιβλίου στην αριστερή σελίδα μας, αφήνοντας την δεξιά άθικτη.

Με το event handling που συνοψίζεται στο Απόσπασμα 3, με το πάτημα του ποντικιού στο κουμπί μας εμφανίζονται στην αριστερή σελίδα μας τα περιεχόμενα. Αυτό ουσιαστικά το αναλαμβάνει η γραμμή `leftEditorPane.setPage(periexomena)`. Οι δύο πρώτες γραμμές στον κώδικα μας κάνουν κάποιες διεργασίες που αφορούν τα σχόλια του βιβλίου, στα οποία θα αναφερθούμε στη συνέχεια. Η επιλογή του ευρετηρίου εμφανίζει την λίστα των ευρετηρίων όρων στις σελίδες του βιβλίου μας. Ομοίως και σε αυτή την περίπτωση την εμφάνιση του ευρετηρίου αναλαμβάνει το `.setPage()` (βλέπε Απόσπασμα 4).

Απόσπασμα 3

```
if (enterCommentsCheck == true && variablesClass.commentsStyle == 0)
    commentsExitRoutine();

try {
    leftEditorPane.setPage(periexomena);
} catch (IOException e) {
    JOptionPane.showMessageDialog(null, "Attempted to
read a bad URL", "Inane error", JOptionPane.ERROR_MESSAGE);
}
```

Απόσπασμα 4

```
if (enterCommentsCheck == true && variablesClass.commentsStyle == 0)
    commentsExitRoutine();

try {
    leftEditorPane.setPage(index);
    rightEditorPane.setPage(index2);
} catch (IOException e) {
    JOptionPane.showMessageDialog(null, "Attempted to
read a bad URL", "Inane error", JOptionPane.ERROR_MESSAGE);
}
```

Η επιλογή της αλλαγής γραμματοσειράς επιτρέπει στον χρήστη να ορίσει από μόνος του τη γραμματοσειρά που επιθυμεί. Με το πάτημα αυτού του κουμπιού εμφανίζεται από κάτω του ένα drop down μενού με τις επιλογές μεγέθους γραμματοσειράς. Παράλληλα απενεργοποιούνται όλες οι άλλες επιλογές (τα κουμπιά δηλαδή) που υπάρχουν στο βιβλίο. Αυτό γίνεται με το κάλεσμα της συνάρτησης `loadFontChooser()`; στο event handling του κουμπιού αλλαγής της γραμματοσειράς (βλέπε Απόσπασμα 5).

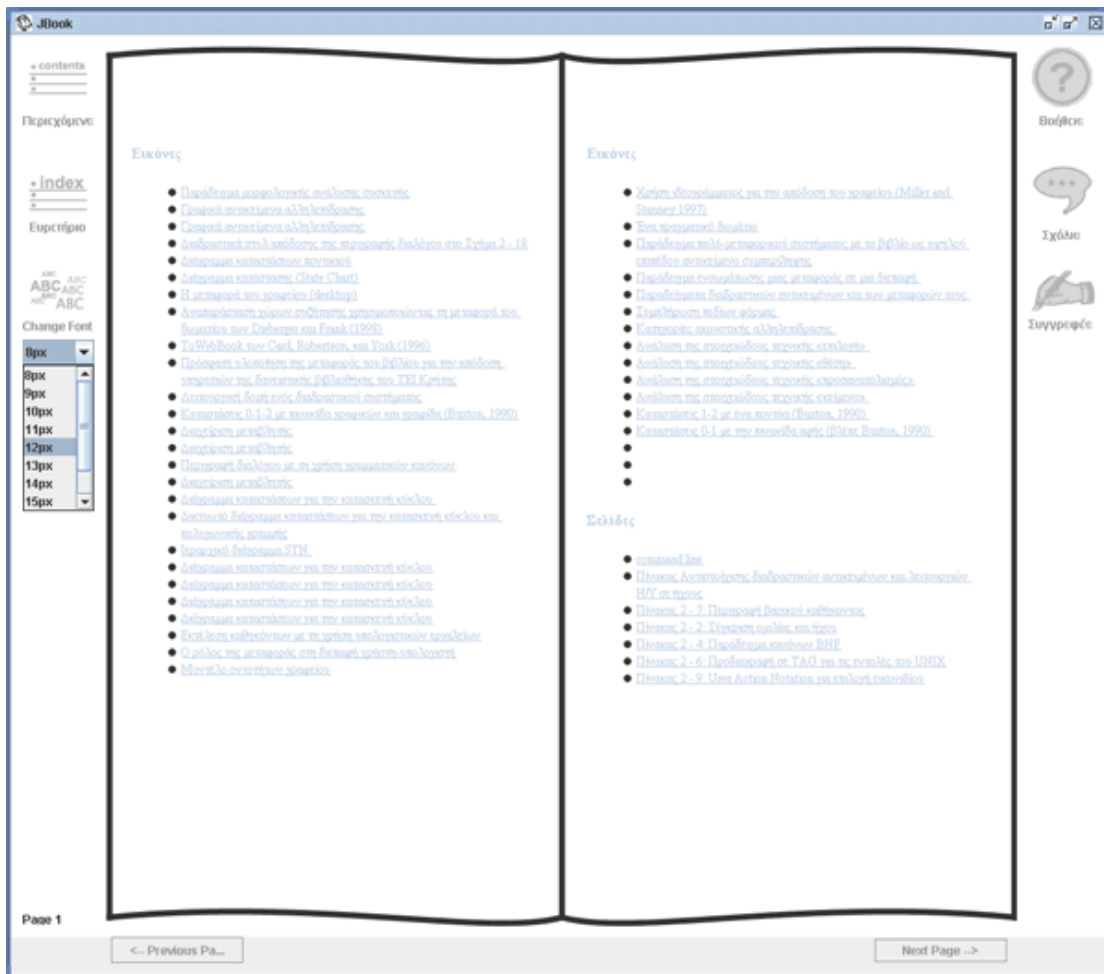
Απόσπασμα 5

```
private void loadFontChooser()
{
    fontSize.setVisible(true);
    leftEditorPane.setEnabled(false);
    rightEditorPane.setEnabled(false);
    previousPageButton.setEnabled(false);
    nextPageButton.setEnabled(false);
    helpImg.setEnabled(false);
    indexImg.setEnabled(false);
}
```

```

authorImg.setEnabled(false);
authorTxt.setEnabled(false);
commentsImg.setEnabled(false);
commentsTxt.setEnabled(false);
contentsTxt.setEnabled(false);
cssChange.setEnabled(false);
helpTxt.setEnabled(false);
indexTxt.setEnabled(false);
cssImage.setEnabled(false);
contentsImg.setEnabled(false);
goToLabel.setEnabled(false);
goToButton.setEnabled(false);
goToTextField.setEnabled(false);
}

```



Εικόνα 24: Επιλογή μεγέθους γραμματοσειράς

Με την επιλογή του μεγέθους της γραμματοσειράς που επιθυμεί ο χρήστης από το drop down μενού έχουμε αμέσως την αλλαγή της στις σελίδες του βιβλίου μας. Αυτό πραγματοποιείτε με τον τρόπο που περιγράφεται στο :

Απόσπασμα 6

```

private void fontSizeActionPerformed(java.awt.event.ActionEvent evt)
{

```



```

if (fontSize.getSelectedIndex() == 0)
{
    leftEditorPane.setFont(new Font("Serif", Font.PLAIN, 8));
    rightEditorPane.setFont(new Font("Serif", Font.PLAIN, 8));
}
if (fontSize.getSelectedIndex() == 1)
{
    leftEditorPane.setFont(new Font("Serif", Font.PLAIN, 9));
    rightEditorPane.setFont(new Font("Serif", Font.PLAIN, 9));
}
if (fontSize.getSelectedIndex() == 2)
{
    leftEditorPane.setFont(new Font("Serif", Font.PLAIN, 10));
    rightEditorPane.setFont(new Font("Serif", Font.PLAIN, 10));
}
if (fontSize.getSelectedIndex() == 3)
{
    leftEditorPane.setFont(new Font("Serif", Font.PLAIN, 11));
    rightEditorPane.setFont(new Font("Serif", Font.PLAIN, 11));
}
if (fontSize.getSelectedIndex() == 4)
{
    leftEditorPane.setFont(new Font("Serif", Font.PLAIN, 12));
    rightEditorPane.setFont(new Font("Serif", Font.PLAIN, 12));
}
if (fontSize.getSelectedIndex() == 5)
{
    leftEditorPane.setFont(new Font("Serif", Font.PLAIN, 13));
    rightEditorPane.setFont(new Font("Serif", Font.PLAIN, 13));
}
if (fontSize.getSelectedIndex() == 6)
{
    leftEditorPane.setFont(new Font("Serif", Font.PLAIN, 14));
    rightEditorPane.setFont(new Font("Serif", Font.PLAIN, 14));
}
if (fontSize.getSelectedIndex() == 7)
{
    leftEditorPane.setFont(new Font("Serif", Font.PLAIN, 15));
    rightEditorPane.setFont(new Font("Serif", Font.PLAIN, 15));
}
if (fontSize.getSelectedIndex() == 8)
{
    leftEditorPane.setFont(new Font("Serif", Font.PLAIN, 16));
    rightEditorPane.setFont(new Font("Serif", Font.PLAIN, 16));
}
if (fontSize.getSelectedIndex() == 9)
{
    leftEditorPane.setFont(new Font("Serif", Font.PLAIN, 17));
    rightEditorPane.setFont(new Font("Serif", Font.PLAIN, 17));
}

fontSize.setVisible(false);
leftEditorPane.setEnabled(true);
rightEditorPane.setEnabled(true);
leftEditorPane.setEnabled(true);
rightEditorPane.setEnabled(true);
    if (counter != 0) //coz if i am in the 1st page and i exit the
comments then by mistake the button will be enabled
        previousPageButton.setEnabled(true);

```

```

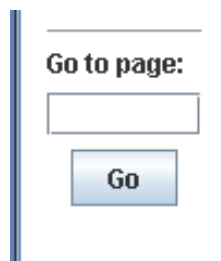
        if (arrayLeft.length-1 != counter) //coz if i am in
the last page and i exit the comments then the button will be by
mistake enabled
            nextPageButton.setEnabled(true);

helpImg.setEnabled(true);
indexImg.setEnabled(true);
authorImg.setEnabled(true);
authorTxt.setEnabled(true);
commentsImg.setEnabled(true);
commentsTxt.setEnabled(true);
contentsTxt.setEnabled(true);
cssChange.setEnabled(true);
helpTxt.setEnabled(true);
indexTxt.setEnabled(true);
cssImage.setEnabled(true);
contentsImg.setEnabled(true);
goToLabel.setEnabled(true);
goToButton.setEnabled(true);
goToTextField.setEnabled(true);

    }

```

Αυτό που κάνει ο παραπάνω κώδικας είναι να πάρει την επιλογή του χρήστη από το drop down μενού που έχουμε ονομάσει `fontSize` με το `if (fontSize.getSelectedIndex() == 0)` και με βάση αυτό να αλλάξει την γραμματοσειρά απλά με το `leftEditorPane.setFont(new Font("Serif", Font.PLAIN, 8))`. Έπειτα ξανά ενεργοποιεί τα κουμπιά του βιβλίου και για τα κουμπιά πλοήγησης στο βιβλίο για τα οποία θα αναφερθούμε σε λίγο κάνει έναν έλεγχο σε πια σελίδα βρίσκετε ο χρήστης ώστε να αποφύγει τυχόν σφάλματα. Με αυτό τον τρόπο αποφεύγει σφάλματα όπως το ο χρήστης να είναι στην τελευταία σελίδα του βιβλίου και το κουμπί που θα τον πηγαίνει στην επόμενη σελίδα να είναι ενεργοποιημένο ενώ δεν θα έπρεπε να είναι. Παρομοίως και αν ο χρήστης είναι στην αρχική σελίδα το κουμπί που θα τον πήγαине στην προηγούμενη σελίδα δεν ενεργοποιήστε. Τέλος είναι πολύ σημαντικό να πούμε ότι για να μπορέσουν τα `JEditorPane` μας να δεχτούν αλλαγές στην γραμματοσειρά τους έπρεπε να συμπεριλάβουμε την εξής εντολή: `.putClientProperty(JEditorPane.HONOR_DISPLAY_PROPERTIES, Boolean.TRUE)` η οποία επιτρέπει αλλαγές στη γραμματοσειρά. Χωρίς αυτή οι αλλαγές μας δεν θα είχαν αντίκρισμα.



Εικόνα 25: Επιλογή γρήγορης πλοήγησης

Με την επιλογή της γρήγορης πλοήγησης ο χρήστης τοποθετεί στο `JTextField` τον αριθμό της σελίδας που θέλει να πάει και πατώντας το κουμπί 'Go' αμέσως πάει στην επιθυμητή σελίδα. Αυτό υλοποιείτε όπως περιγράφεται στο Απόσπασμα 7.

Απόσπασμα 7

```
final class drawLeftPage extends JPanel { //drawing stuff, to
    private void goToButtonActionPerformed(java.awt.event.ActionEvent
    evt) {

        if      (Integer.parseInt(goToTextField.getText())      <
arrayLeft.length + 1 && Integer.parseInt(goToTextField.getText()) >
0)
        {
            try {

leftEditorPane.setPage(arrayLeft[Integer.parseInt(goToTextFie
ld.getText()) - 1]);

rightEditorPane.setPage(arrayRight[Integer.parseInt(goToTextFie
ld.getText()) - 1]);
                } catch (NumberFormatException ex) {
                    ex.printStackTrace();
                } catch (IOException ex) {
                    ex.printStackTrace();
                }
            }

            // ayth h rutina einai gia na energopoihthoun ta koubia se
periptosh pou o xristis alaksei selida enw
            //vriskete eite sthn prwth eite sthn teleftaia selida
            if (counter == 0)
                previousPageButton.setEnabled(true);
            if (counter == arrayLeft.length-1)
                nextPageButton.setEnabled(true);
            //telos routines

            counter = Integer.parseInt(goToTextField.getText());
            pageCounterLabel.setText("Page" + counter);
            counter--;

            //afth h rutina einai gia na apenergopoihtoun ta koubia se
periptosh pou o xristis paei sthn teleftaia eite sthn arxiki selida
                commentsAndNavigationRoutine();

        }
        else
        {
            JOptionPane.showMessageDialog(null, "Page " +
Integer.parseInt(goToTextField.getText()) + " does not exist", "Inane
error", JOptionPane.ERROR_MESSAGE);
        }

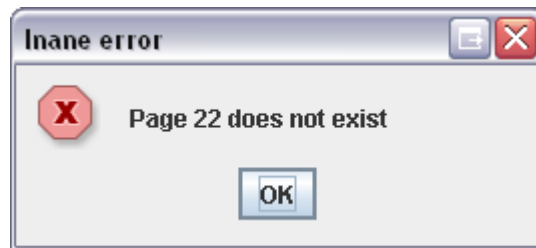
    }
}
```

Ο κώδικας παίρνει αυτό που πληκτρολόγησε ο χρήστης στο JTextField το οποίο έχουμε ονομάσει `goToTextField` με το `goToTextField.getText()`. Επειδή αυτό όμως είναι String μεταβλητή το μετατρέπουμε σε Integer γράφοντας `Integer.parseInt(goToTextField.getText())`. Έχοντας λοιπόν την καταχώριση του χρήστη στο `goToTextField` μας απλά ελέγχουμε αν αυτή η καταχώριση είναι σωστή με βάση τις σελίδες του βιβλίου μας με μία If συνάρτηση. Το είναι ο πίνακας

στον οποίο έχουν καταχωρηθεί οι σελίδες του βιβλίου μας σε μορφή .html σελίδων. Το `arrayLeft.length` μας δίνει το μέγεθος του πίνακα και έτσι έχουμε και τον αριθμό των σελίδων του βιβλίου μας. Επειδή όμως από τον στοιχειώδη προγραμματισμό ξέρουμε ότι η αρίθμηση του πίνακα ξεκινάει από το 0 εμείς στην If συνάρτηση μας προσθέσαμε +1. Έχοντας τσεκάρει αν ο χρήστης έχει ζητήσει υπαρκτή σελίδα ο κώδικας αναλαμβάνει να τον μεταφέρει στην σελίδα της επιλογής του

```
leftEditorPane.setPage(arrayLeft[Integer.parseInt(goToTextField.getText()) - 1]).
```

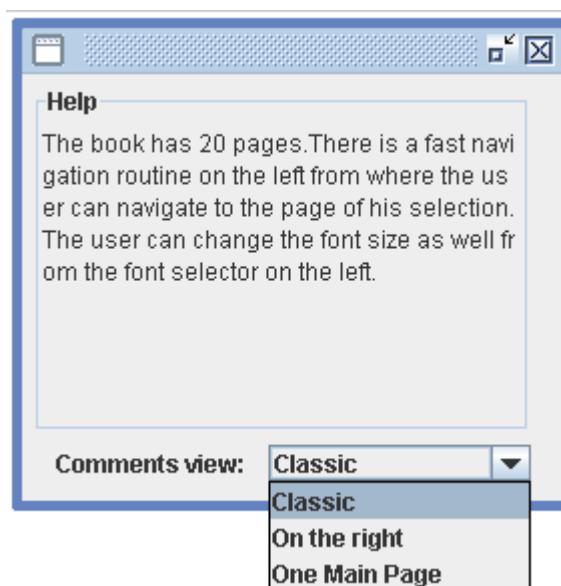
Σε περίπτωση που ο χρήστης εισάγει λάθος δεδομένο κάτι που δεν αντιστοιχεί σε υπάρχον σελίδα τότε το πρόγραμμα του εμφανίζει ένα μήνυμα λάθους και τον ενημερώνει ότι δεν υπάρχει αυτή η σελίδα που ζήτησε.



Εικόνα 26: Μήνυμα λάθους

Τέλος στο JLabel που βρίσκεται στο κάτω μέρος του αριστερού μενού μας και ονομάζεται `pageCounterLabel` εμφανίζεται ο αριθμός της σελίδας που βρισκόμαστε. Στον κώδικα έχουμε ορίσει μια μεταβλητή `counter` που με βάση το σύστημα πλοήγησης του βιβλίου μας καταγράφει σε πια σελίδα είναι ο χρήστης. Έτσι λοιπόν το μόνο που χρειάζεται για να ενημερωθεί ο `pageCounterLabel` είναι να γράψουμε `pageCounterLabel.setText("Page"+counter)`.

Με την επιλογή της βοήθειας που βρίσκουμε στο δεξιό μενού ο χρήστης έχει πρόσβαση σε ένα pop up παράθυρο το οποίο του δίνει γενικές πληροφορίες για το βιβλίο αλλά και τη δυνατότητα επιλογής της εμφάνισης των σχολίων με τον τρόπο που το είναι πιο βολικός. Ας δούμε πως εμφανίζεται το pop up παράθυρο και στη συνέχεια θα αναλύσουμε τις επιλογές εμφάνισης των σχολίων.



Εικόνα 27: Frame βοήθειας και επιλογής τρόπου εμφάνισης σχολίων

Με το πάτημα του κουμπιού της βοήθειας δημιουργούμε το `JToolTip` το οποίο είναι ένα νέο `JFrame` που έχουμε δημιουργήσει.

```
new JToolTip().setVisible(true);
```

Μέσα σε αυτό το `JFrame` το `JToolTip` υπάρχει ένα `JEditorPane` το οποίο περιέχει το κείμενο με τις πληροφορίες για τον χρήστη. Επίσης υπάρχει και ένα `drop down` μενού με τις τρεις διαφορετικές επιλογές εμφάνισης των σχολίων. Η προκαθορισμένη είναι η εμφάνιση 'classic'. Επιλέγοντας ο χρήστης μια από αυτές αλλάζει την μεταβλητή μιας `public` κλάσης η οποία περιέχει κάποιες μεταβλητές που χρησιμοποιούνται και από τα δύο `Frames` το `JToolTip` αλλά και το κεντρικό `Frame` αυτό δηλαδή του βιβλίου που ονομάζουμε `JBookFree`.

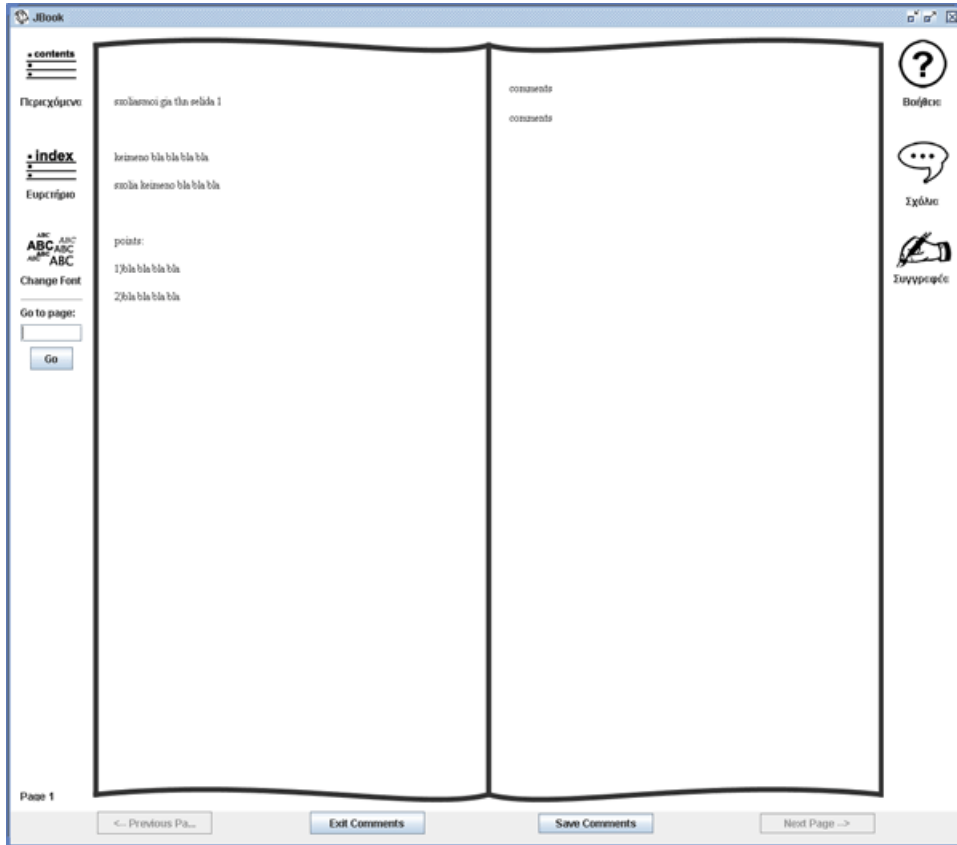
Στη συνέχεια θα εξηγήσουμε πως ο παραπάνω κώδικας επηρεάζει την εμφάνιση των σχολίων (βλέπε Απόσπασμα 8).

Απόσπασμα 8

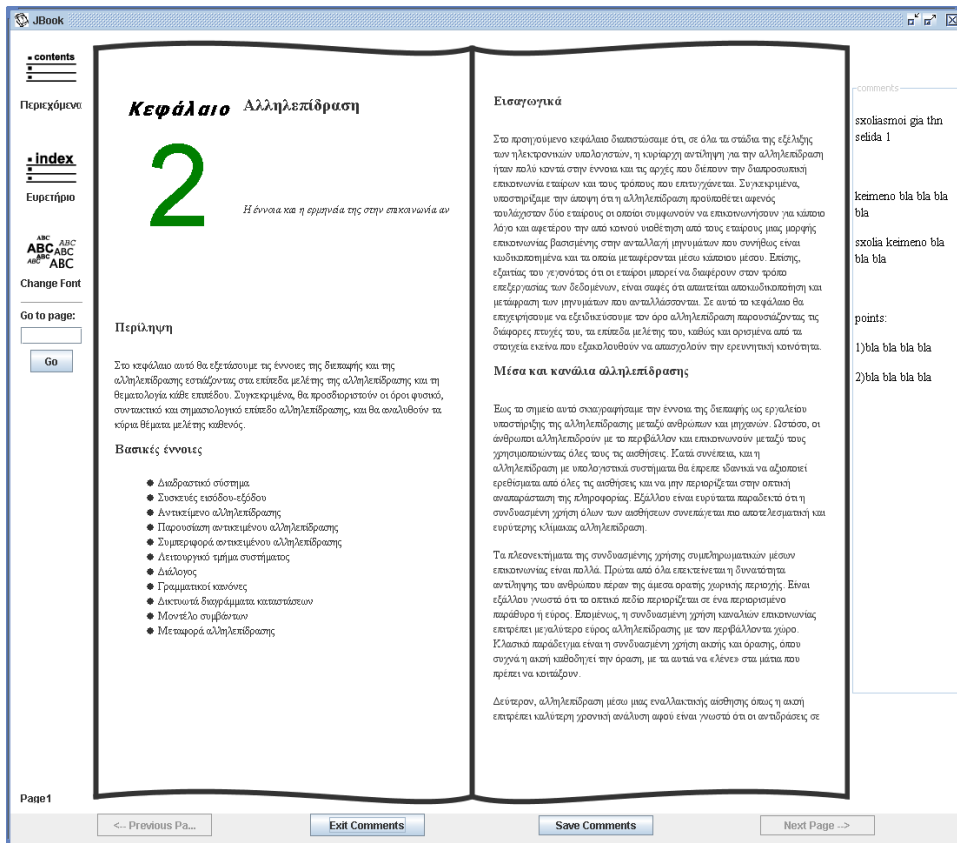
```
private void commentsStyleActionPerformed(java.awt.event.ActionEvent evt) {
    if (commentsStyle.getSelectedIndex() == 0)
        variablesClass.commentsStyle = 0;
    if (commentsStyle.getSelectedIndex() == 1)
        variablesClass.commentsStyle = 1;
    if (commentsStyle.getSelectedIndex() == 2)
        variablesClass.commentsStyle = 2;
}
```

Με το πάτημα του κουμπιού των σχολίων εμφανίζονται τα σχόλια του βιβλίου. Εδώ πρέπει να πούμε πως θέλαμε το πρόγραμμα να δίνει την δυνατότητα στον χρήστη να κρατάει τα σχόλια του καθώς διαβάζει το βιβλίο. Έτσι λοιπόν ο χρήστης πατώντας το κουμπί για τα σχόλια μπορεί να γράψει τις επισημάνσεις του ή ότι άλλο θέλει. Τα σχόλια για την ευκολία του χρήστη τμηματοποιούνται ανά σελίδα. Αυτό σημαίνει ότι αν ο χρήστης βρίσκεται στη δεύτερη σελίδα και εισέλθει στην περιοχή των σχολίων τότε θα γράψει σχόλια που θα αφορούν μόνο την δεύτερη σελίδα.

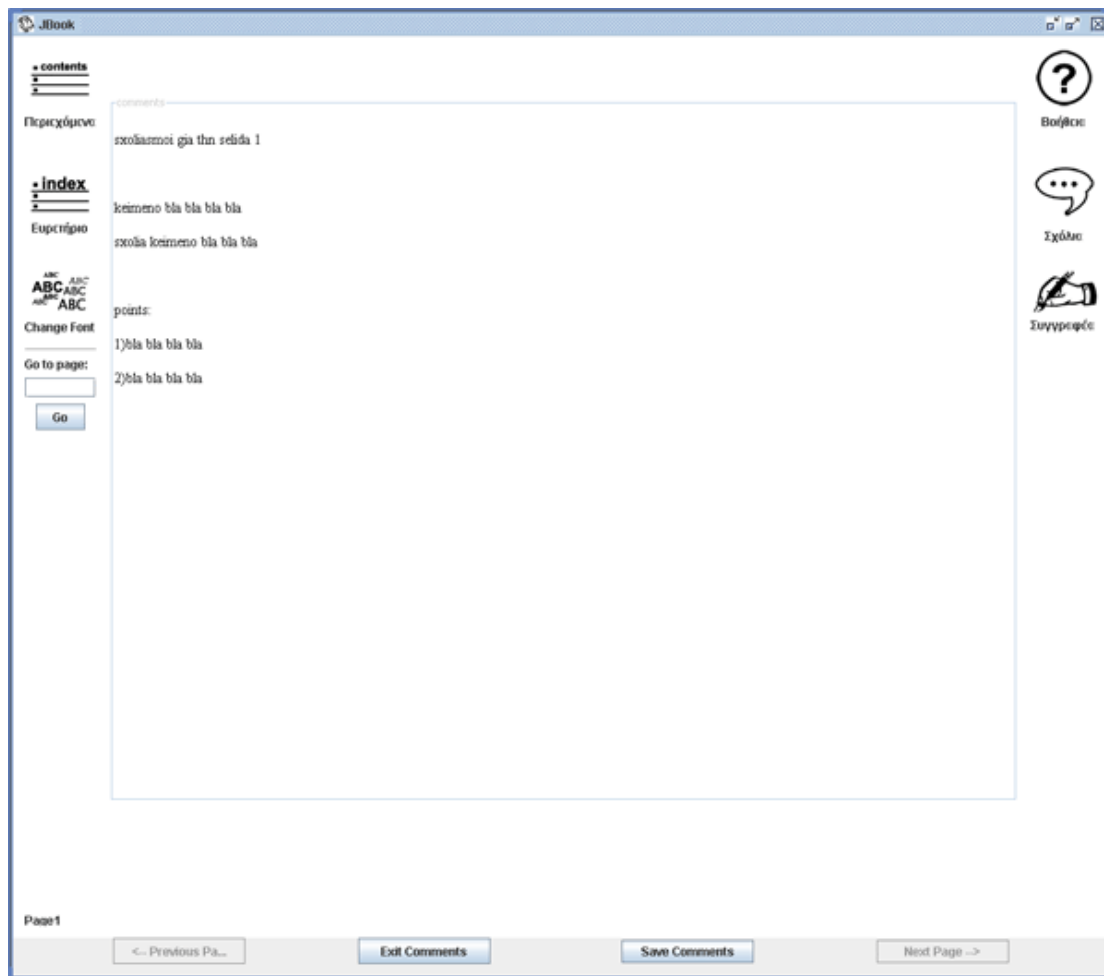
Ο χρήστης όπως αναφέραμε και νωρίτερα έχει τρεις επιλογές για το πώς θέλει να εμφανίζονται τα σχόλια στο βιβλίο. Η πρώτη που είναι και η προεπιλεγμένη είναι τα σχόλια να αντικαθιστούν τις σελίδες του βιβλίου. Έτσι ο χρήστης δεν βλέπει τα περιεχόμενα της σελίδας που βρίσκετε αλλά τα σχόλια σε αυτήν (Εικόνα 28). Βγαίνοντας από τα σχόλια επιστρέφει στη σελίδα που ήταν. Ο δεύτερος εμφανίζει στον χρήστη ένα τρίτο `JEditorPane` δεξιά από δεξιά σελίδα και εξαφανίζει το δεξιό `JPanel` το οποίο και ουσιαστικά αντικαθιστά (Εικόνα 29). Σε αυτό το μικρό `JEditorPane` ο χρήστης βλέπει τα σχόλια της σελίδας που βρίσκεται. Ομοίως βγαίνοντας ο χρήστης από την περιοχή των σχολίων επιστρέφει στην σελίδα που ήταν. Αυτό γίνεται για όλες τις σελίδες. Τέλος ο τρίτος είναι ένα μόνο `JEditorPane` να αντικαταστήσει και τις δύο σελίδες του βιβλίου μας (βλέπε Εικόνα 30).



Εικόνα 28: Σχόλια στην θέση των σελίδων του βιβλίου



Εικόνα 29: Σχόλια στο δεξιό κομμάτι του βιβλίου



Εικόνα 30: Σχόλια στο κέντρο της σελίδας

Πατώντας λοιπόν ο χρήστης το κουμπί γίνονται τα εξής (βλέπε Απόσπασμα 9):

Απόσπασμα 9

```
private void commentsImgMouseClicked(java.awt.event.MouseEvent evt) {
    if (variablesClass.commentsStyle == 2 ) //if users wants the "main
page" comment style this hides the 2 pages
    {
        leftPage.setVisible(false);
        rightPage.setVisible(false);
    }
    if (variablesClass.commentsStyle == 1) //if user wants the "on
the right" style this hides the toolBoxRight menu
        toolBoxRight.setVisible(false);

    if (variablesClass.commentsStyle != 0) ////if user wants the "on
the right" or "main page" style this makes visible the toolBoxRight
JEditorPane
        commentPanel.setVisible(true);

        saveComments.setVisible(true);           //show comments buttons
for all cases
```

```

exitComments.setVisible(true);

        enterCommentsCheck=true;           //inform flag that user has
entered the comments section
        commentsSaveCheck = false;        //for all cases

        if (variablesClass.commentsStyle == 0) //show comments for
classic case
        {
            if ( commentsArrayLeft[counter] != null ) //show comment
pages
            {
                try {
                    leftEditorPane.setPage(commentsArrayLeft[counter]);
                    //saveComments.setVisible(true);// TODO add your
handling code here:
                } catch (IOException e) {
                    JOptionPane.showMessageDialog(null, "Attempted to
read a bad URL","Inane error",JOptionPane.ERROR_MESSAGE);
                }
            } else {
                JOptionPane.showMessageDialog(null, "Couldn't find file:
commentsOne.html","Inane error",JOptionPane.ERROR_MESSAGE);
            }

            if ( commentsArrayRight[counter] != null ) {
                try {
                    rightEditorPane.setPage(commentsArrayRight[counter]);
                    // TODO add your handling code here:
                } catch (IOException e) {
                    JOptionPane.showMessageDialog(null, "Attempted to
read a bad URL","Inane error",JOptionPane.ERROR_MESSAGE);
                }
            } else {
                JOptionPane.showMessageDialog(null, "Couldn't find file:
commentsTwo.html","Inane error",JOptionPane.ERROR_MESSAGE);
            }
        }
        else //show comments for the other two cases
        {
            if ( commentsArrayLeft[counter] != null ) //show comment
pages
            {
                try {
                    commentsText.setPage(commentsArrayLeft[counter]);
                    //saveComments.setVisible(true);// TODO add your
handling code here:
                } catch (IOException e) {
                    JOptionPane.showMessageDialog(null, "Attempted to
read a bad URL","Inane error",JOptionPane.ERROR_MESSAGE);
                }
            } else {
                JOptionPane.showMessageDialog(null, "Couldn't find file:
commentsOne.html","Inane error",JOptionPane.ERROR_MESSAGE);
            }
        }
    }
}

```



```

        saveComments.setVisible(true); //show comment
buttons
        exitComments.setVisible(true); //show comment
buttons
        previousPageButton.setEnabled(false); //disable navigation
buttons
        nextPageButton.setEnabled(false); //disable navigation
buttons
        if (variablesClass.commentsStyle == 0)
        {
            leftEditorPane.setEditable(true); //setEditable to true
so user can change the EditorPanes text
            rightEditorPane.setEditable(true); //setEditable to true
so user can change the EditorPanes text
        }

    }

```

Ο παραπάνω κώδικας κάνει μια πολύπλοκη διαδικασία. Ελέγχει ποιον τρόπο έχει επιλέξει ο χρήστης για να εμφανίζονται τα σχόλια και ανάλογα κάνει τις απαραίτητες διαδικασίες. Αν ο χρήστης έχει επιλέξει να εμφανίζονται τα σχόλια με τον κλασικό τρόπο τότε εμφανίζει τα κουμπιά για έξοδο ή αποθήκευση των σχολίων `saveComments.setVisible(true)` `exitComments.setVisible(true)`. Ενεργοποιεί το flag ότι ο χρήστης έχει εισέλθει στα σχόλια `enterCommentsCheck = true` και εμφανίζει τα σχόλια `leftEditorPane.setPage(commentsArrayLeft[counter]);` Σε περίπτωση που επιλέξει τον δεύτερο τρόπο κρύβει το δεξιό `JPanel` και εμφανίζει τα σχόλια στη θέση του (βλέπε Απόσπασμα 10).

Απόσπασμα 10

```

        if (variablesClass.commentsStyle == 1)
            toolBoxRight.setVisible(false);

        if (variablesClass.commentsStyle != 0)
            commentPanel.setVisible(true);

```

Εδώ βλέπουμε ότι το δεύτερο If είναι κοινό και για αυτή τη περίπτωση αλλά και για την περίπτωση της τρίτης επιλογής. Στη τρίτη επιλογή απλά εξαφανίζονται η σελίδες μας χωρίς να χαθεί το δεξιό `JPanel`.

Απόσπασμα 11

```

final class drawLeftPage extends JPanel { //drawing stuff, to
if (variablesClass.commentsStyle == 2 )
{
    leftPage.setVisible(false);
    rightPage.setVisible(false);
}
}

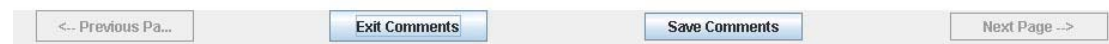
```

Τέλος σε όλες τις περιπτώσεις εμφανίζονται τα σχόλια στα εκάστοτε `JEditorPanes` με το `.setPage()`. Εδώ πρέπει να πούμε ότι για να μπορεί ο χρήστης να γράψει μέσα στα `JEditorPane` πρέπει αυτά να είναι `editable` και για αυτό έχουμε ενεργοποιήσει αυτή την επιλογή με το `.setEditable(true)` στο ανάλογο `JEditorPane`.

Η τελευταία λειτουργία που προσφέρει το δεξιό μενού είναι οι πληροφορίες για τον συγγραφέα. Με το πάτημα του κουμπιού εμφανίζονται στην δεξιά σελίδα οι πληροφορίες για το βιβλίο μας.

Navigation

Στο κάτω τμήμα του Frame μας έχουμε τα κουμπιά πλοήγησης του βιβλίου μας. Αυτά είναι τα κουμπιά με τα οποία αλλάζουμε σελίδες αλλά και στην περίπτωση που βρισκόμαστε στη περιοχή των σχολίων τα κουμπιά για την αποθήκευση και την έξοδο από τα σχόλια.



Εικόνα 31: Μπάρα πλοήγησης

Η αλλαγή στις σελίδες γίνεται με τον παρακάτω κώδικα (βλέπε Απόσπασμα 12):

Απόσπασμα 12

```
counter--; //go to previous page, decrease counter
try {
    rightEditorPane.setPage(arrayRight[counter]);
    leftEditorPane.setPage(arrayLeft[counter]);
} catch (IOException ex) {
    JOptionPane.showMessageDialog(null, "Couldn't find
file", "Inane error", JOptionPane.ERROR_MESSAGE);
}
catch(ArrayIndexOutOfBoundsException e)
{
    counter++;
    JOptionPane.showMessageDialog(null, "You are in the
begining of the book", "Inane error", JOptionPane.ERROR_MESSAGE);
}

//check if the user is in the first or the last page and more
commentsAndNavigationRoutine();
//end of page check
```

Για μία ακόμα φορά χρησιμοποιούμε τη μεταβλητή counter που όπως έχουμε προαναφέρει μας δείχνει σε πια σελίδα βρίσκεται ο χριστής. Ο παραπάνω κώδικας είναι για να πηγαίνει τον χρήστη μια σελίδα πίσω, εξου και το αρχικό counter-; Ακριβώς τον ίδιο κώδικα με την διαφορά ότι έχουμε counter++; έχουμε και για το κουμπί που μας πάει μια σελίδα μπροστά. Αυτό γίνεται γιατί πολύ απλά όταν ορίζουμε την σελίδα που θέλουμε να πάει ο χρήστης έχουμε το rightEditorPane.setPage(arrayRight[counter]) και leftEditorPane.setPage(arrayLeft[counter]); Θα μπορούσαμε εδώ μέσα να αλλάξουμε την μεταβλητή counter αλλά δεν το προτιμήσαμε για να έχουμε καλύτερη διαχείριση στην μεταβλητή μας στην συνέχεια. Επίσης έχουμε καλύψει την απεικόνιση της σελίδας με ένα try – catch το οποίο σε περίπτωση λάθους μας δίνει το σφάλμα. Παρόλα αυτά όμως δεν μείναμε με μόνο στην try – catch. Για να σιγουρευτούμε ότι δεν θα αντιμετωπίσουμε κανένα πρόβλημα με την πλοήγηση μας

όχι μόνο σε αυτή την περίπτωση αλλά και σε όλες της υπόλοιπες (πχ έξοδο από τα σχόλια ή τα περιεχόμενα) δημιουργήσαμε την ρουτίνα `commentsAndNavigationRoutine` την οποία και καλούμε στο τέλος του παραπάνω κώδικα μας. Αυτή η ρουτίνα κάνει τα εξής (βλέπε Απόσπασμα 13):

Απόσπασμα 13

```
private void commentsAndNavigationRoutine() //multi checks in witch
page user is and if comments flag is true or false
{
    //check if the user is in the first or the last page
    if (counter == 0)
        previousPageButton.setEnabled(false);
    else
        previousPageButton.setEnabled(true);

    if (arrayLeft.length-1 == counter)
        nextPageButton.setEnabled(false);
    else
        nextPageButton.setEnabled(true);

    //active deactivate next and previous (plus save and exit
comments buttons if user is in comments area
    if (enterCommentsCheck == false )
    {
        saveComments.setVisible(false);
        exitComments.setVisible(false);
    }
    else
    {
        saveComments.setVisible(true);
        exitComments.setVisible(true);
    }
    counter++;
    pageCounterLabel.setText("Page"+counter);
    counter--;
}
```

Με απλά λόγια αυτή ρουτίνα ελέγχει σε πια σελίδα είναι ο χρήστης και λειτουργεί ανάλογα ώστε τα κουμπιά πλοήγησης να δουλεύουν σωστά.

Το κουμπί `save comments` κάνει αυτό που λέει. Σώζει τα σχόλια με τον εξής τρόπο (βλέπε Απόσπασμα 14):

Απόσπασμα 14

```
//save comments of left page first
counter++; //i raise the counter so i can save the correct
page name coz the counter is always -1 of the page number
// coz the pages tables stars from 0 not 1.
if (variablesClass.commentsStyle == 0)
{
    try {
        leftOutputStream = new
FileWriter("../JBookFree//build//classes//jbookfree//comments//comme
ntsLeft"+counter+".html");
        leftOutputStream.write(leftEditorPane.getText());
        leftOutputStream.close();
    } catch (IOException ex) {
```

```

        ex.printStackTrace();
    }

    // save comments of right page
    try {
        rightOutputStream = new
FileWriter("../JBookFree//build//classes//jbookfree//comments//comme
ntsRight"+counter+".html");
        rightOutputStream.write(rightEditorPane.getText());
        rightOutputStream.close();
    } catch (IOException ex) {
        ex.printStackTrace();
    }
}
else
{
    try {
        leftOutputStream = new
FileWriter("../JBookFree//build//classes//jbookfree//comments//comme
ntsLeft"+counter+".html");
        leftOutputStream.write(commentsText.getText());
        leftOutputStream.close();
    } catch (IOException ex) {
        ex.printStackTrace();
    }
}
JOptionPane.showMessageDialog(null, "Comments
Saved", "information", JOptionPane.INFORMATION_MESSAGE);

    commentsSaveCheck = true; //inform the flag that the user
saved the comments so he program wont ask him "exit with out saving"
    counter--; //move counter to his original position

```

Έχουμε δημιουργήσει ένα OutputStream FileWriter για να μπορούμε να έχουμε πρόσβαση στο αρχείο που θα αποθηκεύουμε. Παίρνουμε τα περιεχόμενα του εκάστοτε JEditorPane μας με το .getText(). Επίσης ο κώδικας τσεκάρει πια επιλογή εμφάνισης σχολίων έχουμε ώστε να σώσει τα περιεχόμενα από το σωστό JEditorPane. Τέλος το flag commentsSaveCheck γίνεται True ώστε να ξέρει το πρόγραμμα ότι ο χρήστης έχει σώσει τις αλλαγές του.

Με το exit comments ο χρήστης βγαίνει από την περιοχή των σχολίων.

Απόσπασμα 15

```

private void exitCommentsActionPerformed(java.awt.event.ActionEvent
evt) {
    if (commentsSaveCheck) //checking if user has saved
the changes to the comments
    {
        commentsExitRoutine(); //runs the above routine
    }
    else
    {
        if (JOptionPane.showConfirmDialog(null,"Exit Comments
With Out Saving?", "Warning", JOptionPane.YES_NO_OPTION) != 1)
            commentsExitRoutine();
    }
}

```

```

        commentPanel.setVisible(false);
        toolBoxRight.setVisible(true);

    if (variablesClass.commentsStyle == 2)
        {
            rightPage.setVisible(true);
            leftPage.setVisible(true);
        }
    }
}

```

Στην αρχή το πρόγραμμα ελέγχει αν ο χρήστης έχει σώσει τις αλλαγές του στα σχόλια. Σε περίπτωση που το έχει κάνει καλεί την `commentsExitRoutine()`:

Απόσπασμα 16

```

private void commentsExitRoutine(){ //some stuff used more than
once while exiting comments
    if (variablesClass.commentsStyle == 0)
        {
            try {

                rightEditorPane.setPage(arrayRight[counter]);
//returns to the page the user was before entering the comments area
                leftEditorPane.setPage(arrayLeft[counter]);
//returns to the page the user was before entering the comments area
            } catch (IOException e) {
                System.err.println("Attempted to read a bad
URL");
            }
        }
        enterCommentsCheck=false; //declare the
exit of the comments area to the flag
        commentsAndNavigationRoutine();
        rightEditorPane.setEditable(false); //coz if Editable
= true then the HyperLinkListener wont work!
        leftEditorPane.setEditable(false); //coz if Editable
= true then the HyperLinkListener wont work!
    }
}

```

Εδώ το πρόγραμμα ελέγχει πιο στυλ σχολίων έχει επιλέξει ο χρήστης και εμφανίζει τις σελίδες αναλόγως. Δηλώνει στο πρόγραμμα ότι ο χρήστης βγήκε από τη περιοχή των σχολίων και καλεί την `commentsAndNavigationRoutine()` για να κάνει τις ρυθμίσεις που έχουμε προαναφέρει. Τέλος απενεργοποιεί την επιλογή να είναι `editable` τα `JEditorPane` μας, δηλαδή οι σελίδες του βιβλίου `rightEditorPane.setEditable(false)`.

Σε περίπτωση που ο χρήστης δεν έχει σώσει τα σχόλια τότε το πρόγραμμα τον ρωτάει αν θέλει σίγουρα να βγει από την περιοχή των σχολίων. Αν ναι τότε βγαίνει χωρίς να σώσει τις αλλαγές αλλιώς επιστρέφει στην περιοχή των σχολίων. Τέλος ενεργοποιεί και απενεργοποιεί τα ανάλογα `components` ώστε να επανέλθει το βιβλίο στην κανονική του μορφή.



Εικόνα 32: Έλεγχος αποθήκευσης σχολίων

Στη συνέχεια θα αναφέρουμε μερικές λεπτομέρειες γενικά για τον κώδικα. Όπως έχουμε προαναφέρει, για να μπορέσει ο χρήστης να γράψει πάνω στα JEditorPane κάποια σχόλια όταν μπει στην περιοχή των σχολίων πρέπει αυτά να είναι editable. Αυτό το κάναμε με το να ελέγξουμε αν το JEditorPane το οποίο εμφανίζεται έχει το focus και στη συνέχεια το ενεργοποιήσαμε.

Απόσπασμα 17

```
private void commentsTextFocusGained(java.awt.event.FocusEvent
evt) {
commentsText.setEditable(true);// thats coz if it was editable from
the begining we had some problems with the other two editorpanes as
well
}
```

Το πρόγραμμα έχει την δυνατότητα να ελέγχει το μέγεθος του παραθύρου και να τροποποιεί αναλόγως την γραμματοσειρά του βιβλίου. Αυτό το κάναμε για να πετύχουμε μια διαδραστικότητα καθώς δεν θα ήταν ιδιαίτερα λειτουργικό με κάθε αλλαγή του μεγέθους να έχουμε ίδια γραμματοσειρά σε μία μεταφορά του βιβλίου που δεν θέλουμε να υπάρχουν scroll bars καθώς θα είχαμε κείμενο εκτός της ορατότητας του χρήστη. Αυτή την λειτουργία την πετύχαμε ως εξής:

Απόσπασμα 18

```
private void formComponentResized(java.awt.event.ComponentEvent evt)
{

    int defSize = 2123;
    int width = getSize().width;
    int size = getSize().height;
    int tmpsize2;
    size = size + width;

    int tmpsize = defSize - size;

    tmpsize2 = (int)(tmpsize/180);

    if (tmpsize2 > 0)
    {
        leftEditorPane.setFont(new Font("Serif", Font.PLAIN, 11 -
tmpsize2));
        rightEditorPane.setFont(new Font("Serif", Font.PLAIN, 11 -
tmpsize2));
    }
}
```

```

    }
    else
    {
        tmpsize2 = (int)(tmpsize/80);
        leftEditorPane.setFont(new Font("Serif", Font.PLAIN, 12 -
tmpsize2));
        rightEditorPane.setFont(new Font("Serif", Font.PLAIN, 12 -
tmpsize2));
    }
}

```

Έχουμε πάρει ως σημείο αναφοράς το αρχικό μέγεθος του παραθύρου μας το `defSize`. Με έναν μαθηματικό αλγόριθμο ελέγχουμε το κατά πόσο έχει μεγαλώσει ή μικρύνει το παράθυρο μας σε σχέση με το αρχικό του μέγεθος και αυξομειώνουμε το μέγεθος της γραμματοσειράς αναλόγως.

Για να μπορέσει ο χρήστης να έχει τον έλεγχο σε ποια σελίδα εστιάζει κάθε φορά που η εκάστοτε σελίδα είναι `focused` την περιβάλλει ένα πλαίσιο και κάθε φορά που χάνει το `focus` το πλαίσιο εξαφανίζεται. Αυτό το κάναμε ως εξής (βλέπε Απόσπασμα 19):

Απόσπασμα 19

```

private void leftEditorPaneFocusLost(java.awt.event.FocusEvent
evt) {
leftEditorPane.setBorder(null);
}

private void leftEditorPaneFocusGained(java.awt.event.FocusEvent
evt) {
leftEditorPane.setBorder(javax.swing.BorderFactory.createLineBorder(n
ew java.awt.Color(100, 100, 100)));
}

```

Τέλος θα κάνουμε μια ιδιαίτερη αναφορά στο πώς δουλεύουν τα `JEditorPane` μας. Δηλαδή το πώς εμφανίζουν τα περιεχόμενα του βιβλίου τα οποία είναι τμηματοποιημένα σε `.html` σελίδες. Αρχικά θα παραθέσουμε τον κώδικα ο οποίος επιτρέπει στα `JEditorPane` μας να εμφανίζουν `.html` σελίδες αλλά και να χειρίζονται τα `events` που αυτές έχουν.

Απόσπασμα 20

```

private void httpThink() { //reads the books pages
//Start of reading a URL page
if ((arrayLeft[counter] != null) && (arrayRight[counter] !=
null) ) { //the counter for start is set to 0
try {
leftEditorPane.setPage(arrayLeft[counter]);
rightEditorPane.setPage(arrayRight[counter]);
} catch (IOException e) {
System.err.println("Attempted to read a bad URL");
}
} else {
System.err.println("Couldn't find file:
TextSamplerDemoHelp.html");
}
}

```

```

//end of read page URL

//lets try to implement a Hyperlink Listener
//it works only with the EditorPanes.editable(false)
class MyHyperlinkListenerLeft implements HyperlinkListener {
public void hyperlinkUpdate(HyperlinkEvent evt) {
if
    (evt.getEventType()
        ==
HyperlinkEvent.EventType.ACTIVATED){ //identify listener

    leftEditorPane = (JEditorPane)evt.getSource();
    httpLinkTest = evt.getDescription(); //i
take the links description (String)

    if ( httpLinkTest.indexOf("image") == -1 )//checks if the
link is a number, if it is its about a page
    {
        httpLinkIntTest = Integer.parseInt(httpLinkTest); //i
parse the string into a number so i can use it as a counter

        try {
            // Show the new page in the editor pane.
            if (periexomena.equals(leftEditorPane.getPage())
== false) //to check if user is in periexomena section
            {

leftEditorPane.setPage(arrayLeft[httpLinkIntTest]); //set the
wantet page

rightEditorPane.setPage(arrayRight[httpLinkIntTest]); //set the
wantet page

                counter = httpLinkIntTest;
                commentsAndNavigationRoutine();// h rutina gia na
exoume ok ta next k previous buttons
            } else
            {

rightEditorPane.setPage(arrayRight[httpLinkIntTest]); //to implement
periexomena navigation
                counter = httpLinkIntTest;
                commentsAndNavigationRoutine(); // h rutina gia
na exoume ok ta next k previous buttons
            }

        } catch (IOException e) {

        }

    }
} else{ //if the link is not a number then it comes here to
point to the image
    variablesClass.imgURL = "/jbookfree/imagesURL/" +
httpLinkTest;
    new popUp().setVisible(true);
}

}
}
}

```


[...] here is the same code as above only for the right page

```
        previousPageButton.setEnabled(false); // coz the book starts
in the first page
        leftEditorPane.addHyperlinkListener(new
MyHyperlinkListenerLeft()); //apply listener
        rightEditorPane.addHyperlinkListener(new
MyHyperlinkListenerRight()); //apply listener
    }
```

Πρέπει να πούμε ότι η παραπάνω συνάρτηση καλείτε στην αρχή του προγράμματος. Για αυτόν τον λόγο λοιπόν αρχικά τσεκάρει μήπως οι πίνακες που περιέχουν τις σελίδες μας είναι άδειες. Στη συνέχεια εμφανίζει στα JEditorPane τις σελίδες με τον απλό τρόπο που μας επιτρέπει η Java `leftEditorPane.setPage(arrayLeft[counter]);` Η μεταβλητή *counter* είναι αρχικά δηλωμένη ως μηδέν.

Στη συνέχεια έχουμε την δημιουργία του event Handling μέσα στην .html που εμφανίζει το JEditorPane. Δημιουργούμε την κλάση μας `MyHyperlinkListenerLeft` το οποίο κάνει implement από το κλασικό `HyperLinkListener` της Java. Με το `if (evt.getEventType() == HyperlinkEvent.EventType.ACTIVATED)` αναγνωρίζουμε το event που έχουμε και στη συνέχεια το παίρνουμε από το JEditorPane μας `leftEditorPane = (JEditorPane)evt.getSource();` Στο επόμενο βήμα παίρνουμε το string που μας επιστρέψει το event μας `httpLinkTest = evt.getDescription()`. Με απλά λόγια αν μέσα στο .html αρχείο μας έχουμε ένα `κάποιο link` εμείς παίρνουμε το string "link".

Έχοντας το string του link μας αρχικά το πρόγραμμα ελέγχει αν αυτό περιέχει την λέξη "image" μέσα του `if (httpLinkTest.indexOf("image") == -1)`. Αν δεν είναι πάει να πει ότι το event μας ζήτησε την εμφάνιση μιας σελίδας. Και έτσι εμφανίζουμε την επιθυμητή σελίδα μας `leftEditorPane.setPage(arrayLeft[httpLinkIntTest]),`
`rightEditorPane.setPage(arrayRight[httpLinkIntTest]).` Έχουμε μετατρέψει το string που έχουμε πάρει από το event μας σε αριθμό (τα link στις .html σελίδες είναι ή αριθμοί ή ονόματα εικόνων) και κάνουμε την μεταβλητή *counter* ίση με αυτόν ώστε να ενημερώσουμε το πρόγραμμα σε πια σελίδα θα πάει ο χρήστης. Επίσης επειδή θέλουμε όταν εμφανίζονται τα περιεχόμενα να εμφανίζονται μόνο στην αριστερή σελίδα και επιλέγοντας μια σελίδα από τα περιεχόμενα αυτή να εμφανίζεται στην δεξιά έχουμε προσθέσει έναν ακόμα έλεγχο `if (perioxomena.equals(leftEditorPane.getPage()) == false)`. Με αυτό ελέγχουμε αν υπάρχουν ήδη τα περιεχόμενα στην δεξιά σελίδα. Αν υπάρχουν τότε απλά γνωρίζουμε ότι το link μας έρχεται από την σελίδα των περιεχομένων και έτσι εμφανίζουμε το αποτέλεσμα του event στην δεξιά σελίδα. Τέλος σε περίπτωση που ο αρχικός μας έλεγχος για τον αν το string που πήραμε από το event μας περιέχει την λέξη "image" ισχύσει τότε το πρόγραμμα καταλαβαίνει ότι το event θέλει να εμφανίσει μια εικόνα και κάνει τα εξής (βλέπε Απόσπασμα 21):

Απόσπασμα 21

```
variablesClass.imgURL = "/jbookfree/imagesURL/" + httpLinkTest;  
    new popUp().setVisible(true);
```

Εδώ πολύ απλά επαναπροσδιορίζει την δημόσια μεταβλητή που ορίζει ποια εικόνα θα ανοίξει το JFrame που καλείτε από κάτω. Το JFrame αυτό έχει ένα JLabel που περιέχει μια εικόνα `jLabel1.setIcon(new javax.swing.ImageIcon(getClass().getResource(variablesClass.imgURL)))` ; Όπως βλέπουμε η εικόνα είναι η μεταβλητή που αλλάξαμε πριν με βάση το string που πήραμε από το event μας.

Με αυτό τον τρόπο λειτουργούν οι διεργασίες με τις .html σελίδες μας και τα event τους στο βιβλίο μας. Εδώ πρέπει να ξανά αναφερθούμε στο Matisse που σε συνδυασμό με το FreeLayout του JDK 5.0 μας έκανε τη ζωή εύκολη. Και αυτό γιατί ήταν ιδιαίτερα απλό το να βάλουμε το JEditorPane που χρησιμεύει για τα σχόλια μας και στην αρχή δεν είναι ορατό. Έτσι λοιπόν όταν αυτό γίνεται ορατό και το δεξί μενού χάνεται η διαχείριση του χώρου στο JFrame μας δεν είχε κανένα πρόβλημα καθώς το JEditorPane απλά πήρε την χώρο του δεξιού μενού μας.

5. Επίλογος

Στα παραπάνω κεφάλαια παραθέσαμε τον τρόπο με τον οποίο αναπτύξαμε την μεταφορά του βιβλίου με την βοήθεια της γλώσσας προγραμματισμού Java. Για μία ακόμα φορά πρέπει να προσθέσουμε ότι η πλατφόρμα netBeans 5.0 σε συνεργασία με το εργαλείο Matisse το οποίο είναι ενσωματωμένο στην πλατφόρμα, έκαναν τη ζωή μας εύκολη γλυτώνοντας μας από περιττό κόπο κατά την διάρκεια του προγραμματισμού. Χάρη στη γλώσσα με την οποία υλοποιήθηκε το JBook το πρόγραμμα μας μπορεί να εκτελεστεί σε οποιοδήποτε λειτουργικό σύστημα κάτι το οποίο το κάνει ιδιαίτερα ευέλικτο.

Το JBook ήταν μια προσπάθεια απόδοσης της μεταφοράς του βιβλίου. Το βιβλίο είναι ένα διαχρονικό αντικείμενο το οποίο έχει υποστεί ελάχιστες ουσιαστικές αλλαγές στη δομή του με το πέρασμα των χρόνων. Ζούμε στην ηλεκτρονική εποχή στην οποία φυσικά έχουν μπει και τα βιβλία με τη μορφή του 'ηλεκτρονικού βιβλίου'. Το βιβλίο είναι ένας δυνατός και παραδοσιακός τρόπος απεικόνισης δεδομένων. Όντας μια γνωστή σε όλους μορφή απεικόνισης η μεταφορά του βιβλίου θα είναι πιο οικεία σε χρήστες που δεν είναι εξοικειωμένοι με τα πολύπλοκα προγράμματα των υπολογιστών.

Το JBook στις επόμενες εκδόσεις του θα μπορούσε να υποστεί πολλές βελτιώσεις οι οποίες θα το έκαναν ιδιαίτερα θελκτικότερο και στους developers αλλά και στους χρήστες. Αρχικά θα ήταν χρησιμότερο να διαβάζει το content του από μια βάση δεδομένων. Με αυτόν τον τρόπο θα μπορεί να αλλάζει αλλά και να τροποποιεί πιο ευέλικτα το περιεχόμενο του. Θα έχει τη δυνατότητα ο χρήστης να διαβάζει περισσότερα του ενός βιβλία μέσα από την μεταφορά του JBook. Επίσης θα είναι ακόμα πιο εύκολο για τον προγραμματιστή να μετατρέψει υπάρχοντα βιβλία στη μορφή που θα είναι αναγνώσιμα από το JBook.

Σε μια διαδικτυακή μορφή του το JBook θα μπορούσε να χρησιμοποιηθεί ως κόμβος ανάγνωσης συγγραμμάτων και όχι μόνο. Με τις ευκολίες που παρέχει η Java το JBook τροποποιημένο θα μπορούσε να τρέχει ως applet σε κάποιον server και να προσφέρει την ανάγνωση στους χρήστες. Μιας και πολλαπλοί χρήστες θα έχουν την πρόσβαση σε αυτό θα μπορούσαν από κοινού να κρατάνε κάποιου είδους σημειώσεις στο ήδη υπάρχον τμήμα σημειώσεων του JBook.

6. Βιβλιογραφία

Stephen Richards, Philip Barker, Ashok Banerji, Charles Lamont and Karim Manji, The Use of Metaphors in Iconic Interface Design

Interactive Systems Research Group, Human-Computer Interaction, Laboratory School of Computing and Mathematics, University of Teesside, Cleve, and, United Kingdom

Ken Mohnkern, Visual Interaction Design: Beyond the Interface Metaphor

George Legrady, Interface Metaphors & New Narratives in Interactive Media

Dan Saffer, The Role of Metaphor in Interaction Design

John G Hedberg and Barry Harper, Creating interface metaphors for interactive multimedia

Katalin Szabó (1995), Metaphors and the user interface

Alan Dix, Janet Finlay, Gregory D. Abowd, Russell Beale, Human-computer interaction

M. Helander, Handbook of Human- Computer Interaction

Νικόλαος Αβούρης, Εισαγωγή στην επικοινωνία ανθρώπου υπολογιστή

Anthony Savidis and Constantine Stephanidis (1995), Building non-visual interaction through the development of the Rooms metaphor, Adjunct Proceedings of ACM SIGCHI.

Card, S., Robertson, G., York, W. (1996): The WebBook and the Web Forager: An Information Workspace for the World-Wide Web. In ACM Annual Conference on Human Factors in Computing Systems (CHI'96).