ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΤΟΜΕΑΣ ΤΕΧΝΟΛΟΓΙΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ

Ανάλυση Τοπολογικής Διάταξης Ιστοσελίδας Για Αποδοτικότερη Γεωκωδικοποίηση

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

του

ΠΑΝΑΓΙΩΤΗ ΜΠΑΛΑ

Επιβλέπων : Ιωάννης Βασιλείου
Καθηγητής Ε.Μ.Π.

Αθήνα, Νοέμβριος 2007

ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΤΕΧΝΟΛΟΓΙΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ
ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ

Ανάλυση Τοπολογικής Διάταξης Ιστοσελίδας Για Αποδοτικότερη Γεωκωδικοποίηση

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

του

ΠΑΝΑΓΙΩΤΗ ΜΠΑΛΑ

Επιβλέπων : Ιωάννης Βασιλείου
Καθηγητής Ε.Μ.Π.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την

….....	.....	.....
Ιωάννης Βασιλείου	Τίμος Σελλής	Ανδρέας-Γεώργιος Σταφυλοπάτης
Καθηγητής Ε.Μ.Π.	Καθηγητής Ε.Μ.Π.	Καθηγητής Ε.Μ.Π.

Αθήνα, Νοέμβριος 2007

.....

ΠΑΝΑΓΙΩΤΗΣ ΜΠΑΛΑ

Διπλωματούχος Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών Ε.Μ.Π.

Περίληψη

Η αποδοτική εξαγωγή γεωγραφικής πληροφορίας (geoparsing) από ιστοσελίδες απαιτεί τη σημασιολογική ανάλυση της σελίδας και την ανακάλυψη της τοπολογικής της δομής. Στην παρούσα διπλωματική εργασία αναπτύσσονται και παρουσιάζονται τεχνικές που επιλύουν τα δυο αυτά θέματα.

Η σημασιολογική επεξεργασία της σελίδας υποδηλώνει την εξέταση των στοιχείων της ως προς το περιεχόμενό τους και τη συσχέτιση μεταξύ τους. Έτσι αναπτύσσονται νέοι αλγόριθμοι και ορίζονται κριτήρια που επιχειρούν ομαδοποίηση των στοιχείων της ιστοσελίδας ελέγχοντας τη συνάφεια του περιεχομένου τους. Αποτέλεσμα της διαδικασίας είναι η απεικόνιση της σελίδας σαν ένα σύνολο από ομάδες στοιχείων οι οποίες χαρακτηρίζονται από περιεχόμενο συγκεκριμένου τύπου.

Για τη μελέτη της τοπολογίας των σελίδων θα στηριχτούμε αποκλειστικά στην επεξεργασία των HTML tags για το λόγο αυτό εξετάζουμε μόνο αμιγώς HTML αρχεία. Οι αλγόριθμοι που θα παρουσιάσουμε εκμεταλλεύονται τις τοπολογικές ιδιότητες των tags και σχηματίζουν μια εικόνα της χωρικής διάταξης των στοιχείων της ιστοσελίδας. Για κάθε στοιχείο πλέον θα γνωρίζουμε τα γειτονικά του στοιχεία καθώς και τη σχετική του θέση στη σελίδα.

Το τελικό αποτέλεσμα μπορεί να δοθεί στον geoparser, ο οποίος μπορεί να επιλέξει ομάδα στοιχείων με βάση το περιεχόμενό τους σε συνδυασμό με τη θέση τους στη σελίδα. Τέλος, οφείλουμε να τονίσουμε ότι οι δυο διαδικασίες, αυτή της σημασιολογικής και εκείνη της τοπολογικής ανάλυσης, είναι τελείως ανεξάρτητες μεταξύ τους.

Λέξεις Κλειδιά: αποδοτική εξαγωγή γεωγραφικής πληροφορίας, σημασιολογική επεξεργασία, ομαδοποίηση στοιχείων, μελέτη τοπολογίας, HTML tags

Abstract

A general observation regarding Web pages is that not all content is of equal importance. Thus, performing any type of content analysis requires one to understand the general structure and layout of a Web page.

This thesis focuses on developing layout analysis techniques in relation to a subsequent geoparsing of a Web page.

This work proposes a technique that (i) analyzes the structure of a Web page in terms of atomic content elements and tags und (ii) subsequently aggregates these elements as much as possible to produce content elements suitable for geoparsing. These elements are meaningful, both, in terms of *semantics* (text dominates images but is not ignored since important for understanding the general layout) and *size* (text elements should be of meaningful size).

The layout analysis focuses only on HTML tags and takes advantage of the topological properties of these tags to create a “picture” of the spatial arrangement of Web page elements. In this process DOM trees are used as a data structure. A visualization tool is developed that allows us to easily assess the output of the analysis algorithm and the varying parameter settings.

The final output is well suited for geoparsing, since depending on the specific application needs, a group of elements is chosen based on to their type of content and position on the page.

Keywords: efficient geoparsing, semantic processing, element grouping, layout analysis, HTML tags

Ευχαριστίες

Θα ήθελα να ευχαριστήσω τους ανθρώπους που με επηρέασαν και με βοήθησαν στην εκπόνηση της παρούσας διπλωματικής εργασίας.

Καταρχάς, θα ήθελα να ευχαριστήσω τον επιβλέποντα καθηγητή τον κ. Ι. Βασιλείου. Επίσης αισθάνομαι την ανάγκη να αναφερθώ στον κ. Τ. Σελλή, του οποίου η υποδειγματική συμπεριφορά και το ύφος του σαν καθηγητή με επηρέασαν καθοριστικά στην επιλογή της διπλωματικής. Άνθρωποι σαν τον κύριο Σελλή μας εμπνέουν και μας δείχνουν τον δρόμο που πρέπει να ακολουθήσουμε ώστε να γίνουμε σωστοί όχι μόνα σαν επιστήμονες αλλά και σαν άνθρωποι.

Η ανάπτυξη της εργασίας δεν θα ήταν δυνατή χωρίς την πολύτιμη υποστήριξη του κ. D. Pfoser. Ο κύριος Pfoser, αφιερώνοντας χρήσιμο προσωπικό χρόνο και πάντα με καλή διάθεση, με καθοδήγησε καθ' όλη τη διάρκεια εκπόνησης της εργασίας. Άλλοτε με παρατηρήσεις και άλλοτε με συμβουλές κατόρθωνε να μου δίνει ώθηση όταν τα πράγματα περιπλέκονταν. Αναμφισβήτητα, το αποτέλεσμα της διπλωματικής δεν θα ήταν το ίδιο χωρίς την σημαντικότατη συνεισφορά του. Του οφείλω, λοιπόν, ένα θερμό 'ευχαριστώ'.

Σε στιγμές δυσκολίας και απογοήτευσης, η ψυχολογική υποστήριξη που μου παρείχαν ο αδερφός μου Νίκος και η αρραβωνιαστικιά του Κατερίνα μου έδωσε δύναμη για να συνεχίσω. Σημαντικότερη πηγή αισιοδοξίας όμως ήταν για μένα ο μικρός τους μπέμπης, ο Λαμπρούκος!

Πίνακας περιεχομένων

1 *Εισαγωγή*

Ο παγκόσμιος ιστός έχει αναδειχθεί σε μια από τη σημαντικότερες πηγές πληροφορίας σήμερα. Όσο όμως η ποσότητα των δεδομένων αυξάνεται, ο εντοπισμός και η προσπέλαση της ακριβούς επιθυμητής πληροφορίας καθίσταται δυσκολότερος. Αν συνυπολογίσουμε και το γεγονός ότι η χρήσιμη πληροφορία, όπως εμφανίζεται στις ιστοσελίδες, περιβάλλεται από στοιχεία μη χρήσιμα όπως διαφημίσεις, εικόνες κ.ά. αντιλαμβανόμαστε ότι η προσπάθεια εξόρυξης χρήσιμων δεδομένων δυσχεραίνεται ακόμη περισσότερο.

Η κωδικοποίηση και η εμφάνιση της πληροφορίας γίνονται κυρίως, τουλάχιστον μέχρι σήμερα, χρησιμοποιώντας τη markup γλώσσα HTML. Ενώ όμως η γλώσσα αυτή σχεδιάστηκε με δυνατότητες δόμησης και παρουσίασης της πληροφορίας, στερείται της δυνατότητας ορισμού σημασιολογικών δομών και σημασιολογικής επεξεργασίας του περιεχομένου του. Η εγγενής αυτή αδυναμία της γλώσσας τροφοδότησε την ανάπτυξη τεχνολογιών αναζήτησης και εξόρυξης χρήσιμου περιεχομένου του Ιστού. Αποτέλεσμα είναι η εμφάνιση υπηρεσιών, όπως οι μηχανές αναζήτησης και τα θεματικά ευρετήρια, που βοηθάνε τον χρήστη στον εντοπισμό της επιθυμητής πληροφορίας και συμβάλλουν κάπως στη σημασιολογική οργάνωση του Ιστού.

Η σημασιολογική οργάνωση του Ιστού με βάση το θεματικό περιεχόμενο δεν αποτελεί μοναδική προσέγγιση. Εναλλακτική ή συμπληρωματική λύση θα μπορούσε να αποτελέσει η οργάνωση βάσει γεωγραφικών χαρακτηριστικών. Μια τέτοια προσπάθεια, σαν παρεχόμενη υπηρεσία επί του περιεχομένου του Ιστού στην παρούσα φάση, αποτελεί η γεωτεχνολόγηση[1]/γεωκωδικοποίηση (geoparsing/geocoding) η οποία εκμεταλλεύεται τυχόν γεωγραφική πληροφορία που υπάρχει στην ιστοσελίδα. Όπως ήδη είπαμε, οι ιστοσελίδες

[1] Όπως μεταφράζεται ο όρος geoparsing στο [Anj06]. Συστήνεται όμως, και εμείς θα το υιοθετήσουμε, να χρησιμοποιείται ο αγγλικός όρος γιατί η μετάφραση δεν είναι επιτυχής.

περιέχουν πολύ θόρυβο (άχρηστη πληροφορία) γεγονός που επηρεάζει αρνητικά την προηγούμενη διαδικασία. Καθίσταται, λοιπόν, ζωτικής σημασίας ο καθαρισμός του θορύβου έτσι ώστε η διαδικασία του geoparsing να μας δώσει αποδεκτά αποτελέσματα.

1.1 Geoparsing

Το geoparsing [Anj06] αναφέρεται στη διαδικασία εντοπισμού και εξαγωγής γεωγραφικής πληροφορίας από ιστοσελίδες. Διακρίνονται γενικά τέσσερα επίπεδα από τα οποία μπορεί να εξαχθεί η πληροφορία αυτή: το επίπεδο δικτύου, το συντακτικό επίπεδο, το σημασιολογικό επίπεδο και το επίπεδο τοπολογίας. Η εύρεση γεωγραφικής πληροφορίας στο επίπεδο δικτύου εκμεταλλεύεται τις IP διευθύνσεις ενώ στο επίπεδο τοπολογίας χρησιμοποιεί τους υπερσυνδέσεις που υπάρχουν σε μια σελίδα. Σε ότι αφορά το συντακτικό και το σημασιολογικό επίπεδο, εδώ εξετάζονται τα διάφορα στοιχεία κειμένου της ιστοσελίδας. Η πρώτη μέθοδος εξετάζει λέξεις του κειμένου και ελέγχει αν αντιστοιχούν σε γεωγραφική πληροφορία π.χ. διεύθυνση, ενώ η δεύτερη αναλύει τη σημασιολογία του κειμένου και εξάγει έμμεσα πληροφορία π.χ. από το όνομα ενός ποταμού.

Το ενδιαφέρον μας επικεντρώνεται στις δυο τελευταίες μεθόδους οι οποίες αποτελούν αντικείμενο μελέτης του [Anj06]. Αναφέρουμε σύντομα κάποιες λεπτομέρειες υλοποίησής τους οι οποίες θα μας φανούν χρήσιμες αργότερα στην ανάλυσή μας.

Η μεθοδολογία που υιοθετείται στο [Anj06] βασίζεται στην εξέταση όλων των λέξεων του HTML αρχείου μια προς μια, αφού πρώτα το αρχείο έχει περάσει από κάποιον parser για να απομονωθούν όλα τα στοιχεία κειμένου και να απαλειφθούν όλα τα υπόλοιπα. Το αποτέλεσμα του parser δίνεται στη συνέχεια για επεξεργασία σε ένα σύνολο διαδοχικών κανονικών γραμματικών όπου η κάθε γραμματική επεξεργάζεται την κάθε λέξη του κειμένου σύμφωνα με τους κανόνες της. Από τη διαδικασία αυτή προκύπτουν λέξεις σε κάποια τυποποιημένη μορφή, οι οποίες στη συνέχεια και μέσω προσεγγιστικού ταιριάσματος αναζητούνται σε μια βάση δεδομένων. Αν βρεθούν αντιστοιχίσεις των παραπάνω λέξεων στη βάση τότε εξάγονται οι συντεταγμένες που αντιπροσωπεύουν το γεωγραφικό σημείο που δηλώνεται από την υπό εξέταση λέξη. Σαν τελευταία φάση, επιδιώκεται η ολοκλήρωση των μέχρι τώρα αποτελεσμάτων. Αυτό σημαίνει ότι σε μια σελίδα μπορεί να υπάρχουν πολλές λέξεις που δηλώνουν διάφορα γεωγραφικά σημεία τα οποία δεν σχετίζονται μεταξύ τους. Η όλη προσπάθεια επικεντρώνεται, λοιπόν, στο να καθοριστεί, αν αυτό είναι δυνατόν, σε ποιό γεωγραφικό σημείο ή έστω σε ποια οριοθετημένη γεωγραφική περιοχή αναφέρεται το κείμενο. Εδώ περατώνεται η όλη διαδικασία εύρεσης γεωγραφική πληροφορίας από την ιστοσελίδα.

Αξίζει στο σημείο αυτό να αναφέρουμε μερικά προβλήματα που αντιμετωπίζει η παραπάνω περιγραφείσα διαδικασία. Καταρχάς, με τη διαδικασία του parsing και της απομόνωσης των στοιχείων κειμένου από το HTML αρχείο χάνεται αυτόματα η δομή της ιστοσελίδας. Αυτό με τη σειρά του μας αναιρεί τη δυνατότητα να συμπεράνουμε τη βαρύτητα μιας γεωγραφικής πληροφορίας ανάλογα με τη θέση του στη σελίδα. Ας το δούμε αυτό αναλυτικότερα μέσω ενός παραδείγματος. Για το λόγο αυτό παρουσιάζουμε την ιστοσελίδα της Εικόνας 1 η οποία είναι ειδησεογραφικού περιεχομένου.

Εικόνα 1: Ειδησεογραφική Σελίδα του in.gr[2]

[2] Η εικόνα είναι από τη διεύθυνση http://www.in.gr/news/article.asp?lngEntityID=839555&lngDtrID=244 το οποίο ίσχυε την 12/10/2007.

Όπως μπορούμε να παρατηρήσουμε, το *κυρίως κείμενο* της ιστοσελίδας βρίσκεται στη μεσαία στήλη και επομένως προκειμένου να βρούμε σε ποιο γεωγραφικό σημείο αναφέρεται το εν λόγο άρθρο θα πρέπει να εξετάσουμε μόνο το συγκεκριμένο κείμενο μέσω του αλγορίθμου γεωτεχνολόγησης και γεωκωδικοποίησης που περιγράφεται στο [Anj06]. Μετά τη διαδικασία του parsing όμως όχι μόνο δεν μπορούμε πλέον να βρούμε το κεντρικό κείμενο αλλά επιπλέον περιορίζεται και η εγκυρότητα του αποτελέσματος αφού αυτό επηρεάζεται σε μεγάλο βαθμό από τα ονόματα των πόλεων (Αθήνα, Θεσσαλονίκη, Πάτρα) που, αν και δεν διακρίνονται, υπάρχουν στην αριστερή στήλη τα οποία προφανώς αντιμετωπίζονται ισότιμα με τις γεωγραφικές πληροφορίες που υπάρχουν στο κυρίως κείμενο.

Δεύτερο πρόβλημα που προκαλεί το parsing είναι ότι *όλες οι λέξεις αντιμετωπίζονται ισότιμα σαν κείμενο χωρίς να γίνεται διάκριση* εάν προήλθαν από κείμενο ή από κάποιον σύνδεσμο. Στο προηγούμενο παράδειγμα, τα ονόματα και των τριών πόλεων (Αθήνα, Θεσσαλονίκη, Πάτρα), αποτελούν σύνδεσμο. Αυτό σαφώς και πρέπει να ληφθεί υπόψη.

Τέλος, ένα τρίτο πρόβλημα *εμφανίζεται στην περίπτωση ύπαρξης πολλών γεωγραφικών σημείων στο κείμενο*, όπου η μεθοδολογία συσχέτισής τους που χρησιμοποιείται βασίζεται στην απόστασή τους σε λέξεις. Αν και μερικές φορές η μέθοδος αυτή αποδίδει, άλλες φορές πάλι μπορεί να οδηγήσει σε λάθος συμπεράσματα εξαιτίας της ίδιας της αναδιαμόρφωσης του αρχείου που προαναφέραμε. Χρησιμοποιώντας το ίδιο παράδειγμα, βλέπουμε ότι τα ονόματα των τριών πόλεων που βρίσκονται στην αριστερή στήλη εμφανώς απέχουν από το κεντρικό κείμενο και είναι τελείως διαφορετικά τμήματα της σελίδας. Αυτές οι ιδιότητες δεν λαμβάνονται υπόψη από τη διαδικασία του geoparsing που περιγράψαμε.

1.2 Αντικείμενο διπλωματικής

Έχοντας παρουσιάσει τον γενικότερο χώρο στο οποίο εντάσσεται η παρούσα διπλωματική καθώς και τα προβλήματα που αυτός αντιμετωπίζει, έχουμε στην ουσία οριοθετήσει το αντικείμενο που διαπραγματευόμαστε. Ας δούμε όμως αναλυτικότερα με ποιόν τρόπο θα επιδιώξουμε την επίλυση των προηγούμενων προβλημάτων.

1. Καταρχάς, είναι λογικό να αποδώσουμε σε κάθε γεωγραφική πληροφορία που εντοπίζεται στην ιστοσελίδα διαφορετική βαρύτητα ανάλογα με τη θέση της.
2. Επιπλέον, η ύπαρξη περισσότερων του ενός γεωγραφικών σημείων στη σελίδα θα πρέπει να μας οδηγεί στη συσχέτισή τους βάση της απόστασής τους στην ιστοσελίδα.

Από τις προηγούμενες παρατηρήσεις συμπεραίνουμε πως πρέπει να εξάγουμε την διάταξη τοπολογίας της σελίδας.

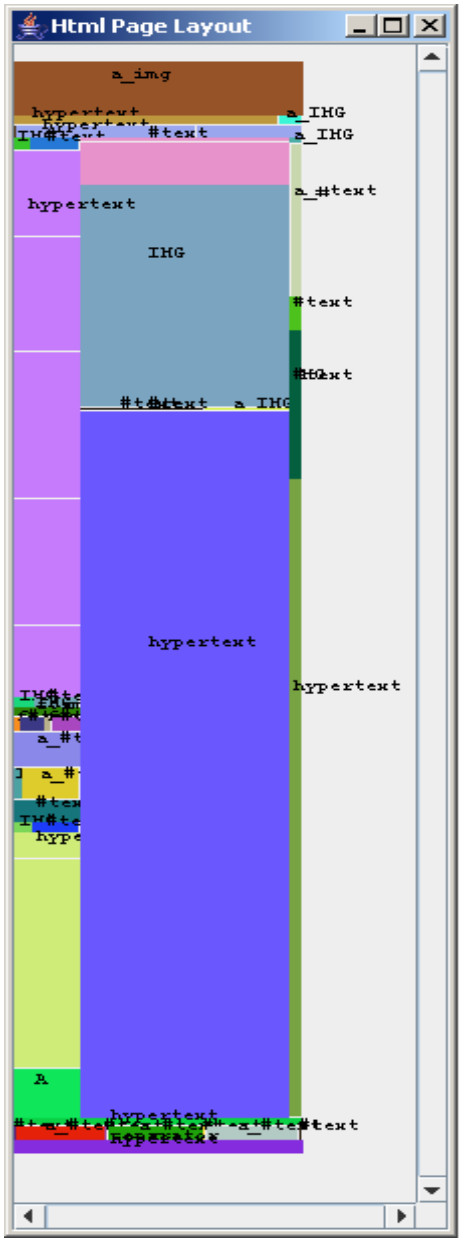
Πράγματι, στόχος της διπλωματικής είναι η εύρεση του τρόπου με τον οποίο διατάσσονται στ χώρο τα διάφορα στοιχεία της ιστοσελίδας. Αυτό σημαίνει πως θα

καταστρώσουμε έναν νοητό χάρτη όπου θα φαίνεται η θέση του κάθε στοιχείου της σελίδας καθώς και τα γειτονικά του στοιχεία.

Με τον τρόπο αυτό επιλύονται τα δυο προαναφερθέντα προβλήματα.

Η χωρική διάταξη, ωστόσο, δεν λύνει το πρόβλημα που σχετίζεται με τον τύπο των στοιχείων που όπως είδαμε στην προηγούμενη ενότητα αντιμετωπίζονται όλα σαν κείμενο. Για το λόγο αυτό επιχειρείται ο χαρακτηρισμός των στοιχείων βάσει του τύπου τους (π.χ. κείμενο, εικόνα, σύνδεσμος) και γίνεται μια προσπάθεια ομαδοποίησής τους βάσει του ίδιου κριτηρίου.

Επειδή δεν υπάρχει καλύτερος τρόπος κατανόησης του στόχου από την παρουσίαση του αποτελέσματος που επιδιώκουμε, παραθέτουμε εδώ το αποτέλεσμα για το παράδειγμα της προηγούμενης ενότητας.



Εικόνα 2: Τοπολογική Διάταξη για τη σελίδα της Εικόνας 1

Στην προηγούμενη εικόνα φαίνεται τόσο η τυπολογική διάταξη της αντίστοιχης ιστοσελίδας όσο και ο τρόπος που έχουν ομαδοποιηθεί τα στοιχεία της βάσει του περιεχομένου τους, φαίνονται δηλαδή υλοποιημένοι οι στόχοι της διπλωματικής.

1.3 Οργάνωση κειμένου

Στο κεφάλαιο αυτό είδαμε ένα μέρος των προβλημάτων του geoparsing και παρουσιάζαμε σε γενικές γραμμές τις λύσεις που προτείνουμε.

Στο κεφάλαιο **2** αναφέρονται εργασίες που ανήκουν στο ίδιο πεδίο μελέτης με την παρούσα διπλωματική. Παρουσιάζονται τα ιδιαίτερα χαρακτηριστικά τους και τα συγκρίνουμε με εκείνα της δικής μας προσέγγισης. Τέλος παρουσιάζουμε την αρχιτεκτονική του λογισμικού που αναπτύσσουμε και το σύνολο των διαδικασιών που αυτό υλοποιεί.

Στο κεφάλαιο **3** μελετάται η σημασιολογική ανάλυση των σελίδων. Συγκεκριμένα, εισάγονται αλγόριθμοι επεξεργασίας και ομαδοποίησης των στοιχείων της σελίδας βάσει του περιεχομένου τους.

Στο κεφάλαιο **4** περιγράφονται οι διαδικασίες και οι αλγόριθμοι που ευθύνονται για την εξαγωγή της τοπολογίας των στοιχείων της ιστοσελίδας. Η όλη επεξεργασία στηρίζεται στις ιδιότητες των HTML tags.

Στο κεφάλαιο **5** παρουσιάζονται λεπτομέρειες υλοποίησης του συστήματος περιγράφοντας τα πακέτα και τις κλάσεις που το συνθέτουν. Επίσης πραγματοποιούμε πειράματα ελέγχου του συστήματος και παραθέτουμε τα αποτελέσματα που παίρνουμε.

Στο κεφάλαιο **6** συνοψίζουμε τα συμπεράσματα από τη ανάπτυξη και μελέτη του συστήματος και προτείνουμε μελλοντικές επεκτάσεις του.

Στο κεφάλαιο **7** παρατίθεται ως παράρτημα ο τρόπος εύρεσης της επιφάνειας των στοιχείων κειμένου που υπάρχουν σε μια ιστοσελίδα.

Τέλος, στο κεφάλαιο **8** γίνεται αναφορά στις βιβλιογραφικές, και όχι μόνο πηγές, που συμβουλευτήκαμε.

Καλή ανάγνωση!

2 *Σχετικές εργασίες*

Στο κεφάλαιο 1 ορίστηκε ο στόχος της διπλωματικής. Αν και δεν ειπώθηκε ρητά, το θέμα που μελετάται εντάσσεται στο γενικότερο πεδίο της *ανάλυσης της διάταξης τοπολογίας των ιστοσελίδων*[3]. Στο παρόν κεφάλαιο παρουσιάζονται σχετικές εργασίες που έχουν γίνει στο συγκεκριμένο πεδίο. Στη συνέχεια παραθέτουμε τη δική μας προσέγγιση.

2.1 Ανάλυση Τοπολογικής Διάταξης Ιστοσελίδων

Οι εργασίες που έχουν γίνει στη συγκεκριμένη θεματική περιοχή στοχεύουν στην ανάλυση της ιστοσελίδας και στην εύρεση της τοπολογίας της. Η τοπολογία μιας σελίδας παρουσιάζει πως έχουν οργανωθεί χωρικά τα διάφορα στοιχεία της και πως σχετίζονται μεταξύ τους. Επομένως θα μπορούσαμε να πούμε πως η όλη προσπάθεια εντάσσεται στο πεδίο της *αντίστροφης μηχανικής (reverse engineering), με την έννοια ότι επιχειρείται να ανακαλυφθεί η διαμόρφωση της ιστοσελίδας όπως την είχε στο μυαλό του ο σχεδιαστής της*. Η κάθε εργασία που παρουσιάζεται ακολούθως έχει αναπτυχθεί για κάποιο ειδικό πεδίο εφαρμογής και υιοθετεί διαφορετική μέθοδο ανάπτυξης.

Ξεκινάμε με το [ZLT06] το οποίο επικεντρώνεται στον κερματισμό σε ζώνες των σελίδων που αποτελούν ιατρικά επιστημονικά άρθρα. Μετά την παρατήρηση πως σε ένα άρθρο υπάρχουν διαφορετικές περιοχές πληροφορίας όπως περιοχή τίτλου, συγγραφέων, αναφορών κ.ά., οι συγγραφείς της εργασίας συμπεραίνουν πως σε αυτού του τύπου τις σελίδες, οι οποίες παρεμπιπτόντως μπορεί να έχουν προέλθει από μετατροπή αρχείων τύπου PDF, η σημασιολογική οργάνωση καθορίζεται σε σημαντικό βαθμό από τη γεωμετρική διάταξη της

[3] Αποτελεί απόδοση στα Ελληνικά της αντίστοιχης Αγγλικής ορολογίας *web page layout analysis*.

σελίδας. Με αφετηρία την παραδοχή αυτή και χρησιμοποιώντας την αναδρομική *μέθοδο αποκοπής X-Y* επί του DOM Tree, κατασκευάζουν ένα *zone tree*. Οι κόμβοι του *zone tree* αντιστοιχούν σε ορατές ζώνες της σελίδας και έχουν προέλθει μετά από ομαδοποίηση των κόμβων του DOM Tree με κριτήριο το αν γειτονεύουν κατά την αναπαράσταση της στην οθόνη. Η διαφορά του zone tree από το DOM Tree είναι ότι το πρώτο προκύπτει από σημασιολογικό κατακερματισμό της σελίδας. Ο κατακερματισμός αυτός στηρίζεται στην παραδοχή , που αποτελεί βάση της όλης ανάλυσης, ότι η συνεκτικότητα των πληροφοριών που υπάρχουν στη σελίδα καθορίζεται από τη σχετική τους θέση σε αυτή. Σε ότι αφορά το DOM Tree, αυτό εκφράζει τη συντακτική οργάνωση της σελίδας. Το αποτέλεσμα που επιτυγχάνεται στη συγκεκριμένη εργασία συμπίπτει με το αντικείμενο της παρούσας διπλωματικής, όπως και οι παραδοχές που γίνονται περί σημασιολογικής οργάνωσης μιας ιστοσελίδας. Η διαφορά είναι ότι εμείς δεν περιοριζόμαστε σε ιατρικά επιστημονικά άρθρα, αλλά εξετάζουμε σελίδες όλων των τύπων.

Στο [SLW+04] αναγνωρίζεται ότι η πληροφορία που περικλείεται σε μια ιστοσελίδα δε είναι της ίδιας σημαντικότητας, αντίθετα διαφέρει από σημείο σε σημείο. Έτσι, ο βαθμός σημαντικότητας ενός στοιχείου της σελίδας εξαρτάται από τη θέση του εν λόγω στοιχείου στη σελίδα, από το συνολικό χώρο που καταλαμβάνει, από το περιεχόμενό του κ.ά. Αναπτύσσεται, λοιπόν, μια μέθοδος απόδοσης σημαντικότητας στα διάφορα τμήματα της σελίδας, που εδώ ονομάζονται *blocks*. Τα blocks προκύπτουν από την ομαδοποίηση των κόμβων του DOM Tree χρησιμοποιώντας τη μέθοδο *Vision-based page segmentation (VIPS)***,** η οποία χρησιμοποιεί οπτικά κριτήρια ομαδοποίησης. Ενώ τα ίδια κριτήρια χρησιμοποιούνται και στην παρούσα διπλωματική για την εύρεση της σημαντικότητας ενός στοιχείου της ιστοσελίδας, δεν αρκούμαστε σε αυτό. Εμείς ενδιαφερόμαστε περισσότερο για τον τρόπο που οργανώνονται χωρικά τα στοιχεία στη σελίδα κάτι το οποίο δεν μελετάται καθόλου στη προαναφερθείσα εργασία.

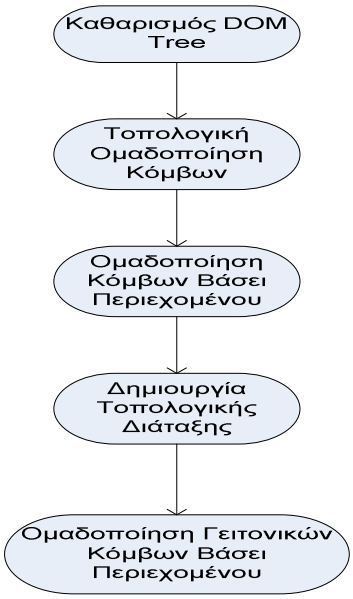
Τέλος, σε μια τρίτη εργασία [CYZ03] μελετάται ο χωρισμός της ιστοσελίδας σε *content blocks*. Σύμφωνα με τη μέθοδο που παρουσιάζεται, τα απλά στοιχεία της σελίδας ομαδοποιούνται μαζί με κριτήριο αν έχουν κοινό μοτίβο εμφάνισης. Αναλυτικότερα, όλα τα απλά στοιχεία τοποθετούνται σε μια σειρά όπως εμφανίζονται στο Html αρχείο. Στη συνέχεια εξετάζονται όλα για την εύρεση μοτίβων που υπάρχουν στην ιστοσελίδα. Στοιχεία που υπακούουν στο ίδιο μοτίβο και είναι γειτονικά εντάσσονται σε έναν κοινό *virtual κόμβο*. Εδώ τελειώνει η σημασιολογική ανάλυση και αρχίζει η χωρική. Συγκεκριμένα, τα blocks που έχουν προκύψει εντάσσονται με τη σειρά τους σε ένα από τα εξής τμήματα της ιστοσελίδας: *header, footer, left side, right side* και *center*. Κριτήριο ένταξης αποτελούν οι συντεταγμένες, οι οποίες δίνονται στο DOM Tree, και ο λόγος ύψος προς πλάτος. Το τελευταίο μέτρο καθορίζει ποια διάσταση του block είναι μεγαλύτερη και ανάλογα με τη μορφή του ευνοείται

η ένταξή του σε συγκεκριμένο τμήμα. Η συγκεκριμένη εργασία υλοποιήθηκε με στόχο να βελτιώσει την παρουσίαση ιστοσελίδων στις μικρές οθόνες που έχουν οι διάφορες κινητές συσκευές. Το γεγονός αυτό ορίζει κάποιες ειδικές απαιτήσεις που δεν υπάρχουν στην παρούσα διπλωματική. Δεν γνωρίζουμε για παράδειγμα τις συντεταγμένες των κόμβων του DOM Tree. Επίσης δεν περιοριζόμαστε στο στατικό μοτίβο που υιοθετείται για τη δομή της σελίδας (header, footer, …) αλλά καταφεύγουμε σε μια πιο ελεύθερη προσέγγιση που, ωστόσο, με μικρές αλλαγές μπορεί να δώσει και το προηγούμενο στατικό μοντέλο.

Είδαμε μερικές εργασίες που εντάσσονται στο ίδιο πεδίο με τη διπλωματική αυτή. Παρουσιάσαμε τις πιο σχετικές και τονίσαμε τις ομοιότητες και τις διαφορές που υπάρχουν. Στην επόμενη ενότητα αναλύουμε θεωρητικά τη δική μας προσέγγιση και παραθέτουμε την αρχιτεκτονική του συστήματος που αναπτύξαμε.

2.2 Η δική μας προσέγγιση

Η ανάλυσή μας στηρίζεται στην επεξεργασία του DOM Tree που αντιστοιχεί στο HTML αρχείο που εξετάζουμε. Το δένδρο το παίρνουμε έτοιμο από κάποιον html parser και θεωρούμε ότι υπακούει στην προτυποποίηση του W3C. Η όλη διαδικασία εξαγωγής της τοπολογίας της σελίδας διακρίνεται σε επιμέρους φάσεις, οι οποίες φαίνονται στην Εικόνα 1.



Εικόνα 1: Φάσεις Συστήματος

Σε κάθε φάση το DOM Tree υφίσταται επεξεργασία και τροποποιείται ανάλογα για να εκφράσει τις αλλαγές που πραγματοποιήθηκαν. Ανεξάρτητα από τις φάσεις, θα μπορούσαμε να πούμε ότι το σύστημα στο σύνολό του αποτελείται από δυο υποσυστήματα τα οποία φαίνονται στην Εικόνα 2 και είναι:

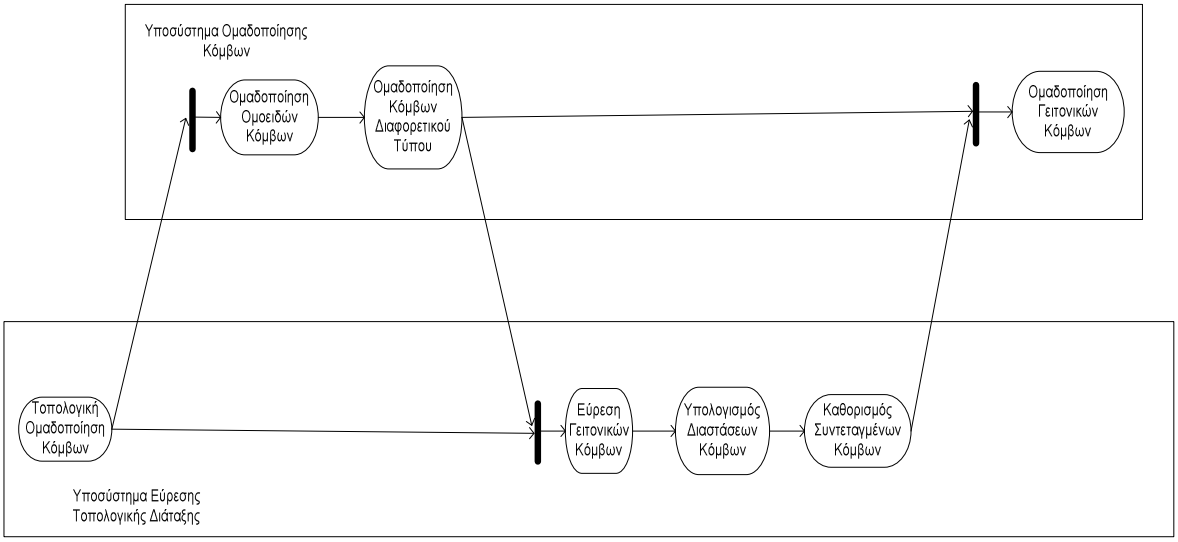
1. το υποσύστημα ομαδοποίησης κόμβων και
2. το υποσύστημα εύρεσης τοπολογικής διάταξης.

Τα υποσυστήματα λειτουργούν ανεξάρτητα με μοναδικό δίαυλο επικοινωνίας μεταξύ τους την ανταλλαγή δεδομένων. Η ροή των δεδομένων επιβάλλει σειριακή εκτέλεση των διαδικασιών των δυο συστημάτων και όχι παράλληλη.

Η τελική έξοδος του συστήματος είναι μια δομή που περιλαμβάνει τις ομάδες των κόμβων όπως αυτές έχουν προκύψει μετά από όλες τις φάσεις ομαδοποίησης. Η εν λόγω δομή δρα επί του DOM Tree παρέχοντας έτσι δυνατότητα διάσχισης όλων των κόμβων του και άμεσης προσπέλασης των περιεχομένων τους για περαιτέρω επεξεργασία.

2.2.1 Ομαδοποίηση Κόμβων

Η ομαδοποίηση κόμβων αναφέρεται σε όλες εκείνες τις λειτουργίες που γίνονται προκειμένου να ενσωματωθούν κόμβοι με περιεχόμενο του ίδιου ή συναφούς τύπου στην ίδια τυπολογική ομάδα. Εδώ εντάσσονται οι διαδικασίες *Ομαδοποίησης Κόμβων Βάσει Περιεχομένου* και *Ομαδοποίησης Γειτονικών Κόμβων Βάσει Περιεχομένου* που φαίνονται στην Εικόνα 1. Η πρώτη διαδικασία αναλύεται περαιτέρω στις φάσεις *Ομαδοποίησης Ομοειδών Κόμβων* και *Ομαδοποίησης Κόμβων Διαφορετικού Τύπου*. Ολόκληρη η διαδικασία ομαδοποίησης ελέγχεται από παραμέτρους τις τιμές των οποίων καθορίζει ο χρήστης ανάλογα με το αποτέλεσμα που επιθυμεί να πετύχει. Στη συνέχεια φαίνεται ένα πλήρες διάγραμμα των διαδικασιών που περιλαμβάνει καθώς και ο τρόπος που αλληλεπιδρά με το υποσύστημα εύρεσης τοπολογικής διάταξης.



Εικόνα 2: Υποσυστήματα και Αλληλεπίδραση μεταξύ τους

2.2.2 Τοπολογική Διάταξη

Το υποσύστημα εύρεσης τοπολογικής διάταξης περιλαμβάνει τις διαδικασίες *Τοπολογικής Ομαδοποίησης Κόμβων* και *Δημιουργίας Τοπολογικής Διάταξης* που φαίνονται στην Εικόνα 1. Η δεύτερη μπορεί να αναλυθεί με μεγαλύτερη λεπτομέρεια και δίνει τις διαδικασίες *Εύρεσης Γειτονικών Κόμβων*, *Υπολογισμού Διαστάσεων Κόμβων* και *Καθορισμός Συντεταγμένων Κόμβων*. Στόχος τους είναι να διαμορφώσουν τη τελική τοπολογική διάταξη των στοιχείων της ιστοσελίδας δίνοντας πληροφορίες για τους γείτονες του κάθε στοιχείου, το μέγεθός του καθώς και τη θέση του στη σελίδα. Όλες οι προηγούμενες πληροφορίες καταγράφονται στους κόμβους του DOM Tree και όχι σε κάποια πρόσθετη δομή. Η αλληλεπίδραση μεταξύ των παραπάνω διαδικασιών αλλά και με το υποσύστημα ομαδοποίησης φαίνονται στην Εικόνα 2.

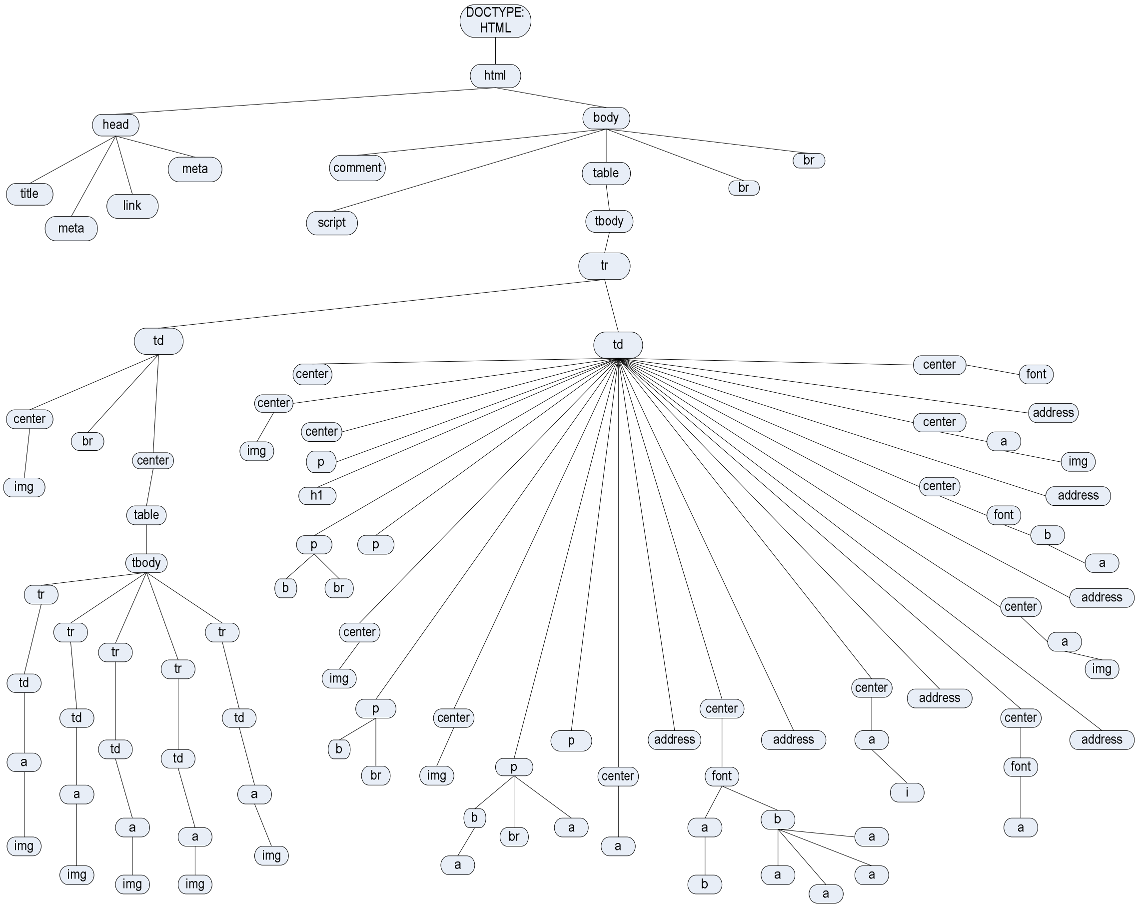
2.2.3 Παράδειγμα Μελέτης και Παραδοχές

Τα όσα αναφέρθηκαν προηγουμένως θα αναλυθούν λεπτομερώς στα επόμενα κεφάλαια της διπλωματικής. Κρίνεται σκόπιμο, για την καλύτερη κατανόησή τους, σε κάθε βήμα να παρατίθεται και το αποτέλεσμά του. Για το λόγο αυτό θα χρησιμοποιήσουμε συγκεκριμένο παράδειγμα ιστοσελίδας, αναφερόμενο εφεξής ως Παράδειγμα 1, και θα παρατηρούμε τη μεταμόρφωση του αντίστοιχου DOM Tree καθώς οι διαδικασίες εξελίσσονται. Η απεικόνιση της ιστοσελίδας παρουσιάζεται στην Εικόνα 3, ενώ στην Εικόνα 4 φαίνεται το DOM Tree που αντιστοιχεί σε αυτήν.

Εικόνα 3: Ιστοσελίδα Παραδείγματος 1[4]

Η επιλογή της προηγούμενης ιστοσελίδας έγινε με κριτήριο τα ιδιαίτερα χαρακτηριστικά της και σε καμία περίπτωση για τη σοβαρότητα του περιεχομένου της! Διαθέτει δηλαδή πολλά γνωρίσματα που θα μας επιτρέψουν να μελετήσουμε σχεδόν όλες τις πτυχές της μεθοδολογίας που πρόκειται να αναπτύξουμε.

[4] Η εικόνα αντιστοιχεί στη σελίδα της διεύθυνσης http://www.godhatesfigs.com/index2.html που ίσχυε την 3/10/2007.



Εικόνα 4: DOM Tree Παραδείγματος 1[5]

Τέλος, θα αναφερθούμε σε κάποιες παραδοχές πάνω στις οποίες αναπτύσσεται η εργασία. Επειδή στόχος μας είναι η βελτίωση του συστήματος geoparsing, το οποίο στηρίζεται στην επεξεργασία κειμένου, κρίνουμε λογικό να εστιάσουμε την προσοχή μας σε στοιχεία κειμένου της ιστοσελίδας και να αγνοήσουμε ή έστω να δώσουμε μικρότερη βαρύτητα στα υπόλοιπα. Πράγματι, η ανάλυσή μας θα μπορούσε να θεωρηθεί κειμενο-κεντρική, το οποίο φυσικά δε σημαίνει ότι τα υπόλοιπα στοιχεία τα διαγράφουμε όπως για παράδειγμα τις εικόνες. Αντίθετα τα διατηρούμε για να έχουμε ακριβή εικόνα της τυπολογικής διάταξης της σελίδας.

[5] Στο DOM Tree δεν εμφανίζονται οι κόμβοι που περιέχουν αποκλειστικά κείμενο έτσι ώστε να μην γίνει πολύπλοκη και δυσνόητη η εικόνα.

3 *Ομαδοποίηση Κόμβων*

Στο κεφάλαιο αυτό θα περιγράψουμε τη διαδικασία ομαδοποίησης των κόμβων του DOM Tree. Θα παρουσιάσουμε και θα αναλύσουμε όλες τις επιμέρους φάσεις που την αποτελούν δείχνοντας παράλληλα πως αυτές επιδρούν στη μορφή του δένδρου. Τέλος, μελετώνται οι παράμετροι που ελέγχουν και καθορίζουν το τελικό αποτέλεσμα της όλης διαδικασίας. Πριν από αυτό όμως κρίνεται σκόπιμο να αναφερθούμε στη λειτουργία εκκαθάρισης του DOM Tree και να εξηγήσουμε την, καθοριστικής σημασίας για τη διαδικασία ομαδοποίησης, έννοια του βάρους.

3.1 Εισαγωγή

Η ομαδοποίηση των κόμβων του DOM Tree είναι μια διαδικασία που διακρίνεται σε πολλές φάσεις και πραγματοποιείται σταδιακά. Σε κάθε βήμα εξετάζονται διαφορετικά χαρακτηριστικά των κόμβων και χρησιμοποιούνται νέα κριτήρια ομαδοποίησης. Προτού όμως αρχίσει η οποιαδήποτε προσπάθεια συνδυασμού των κόμβων κρίνεται σκόπιμο να γίνει εκκαθάριση του DOM Tree από στοιχεία που δεν χρειάζονται στην ανάλυσή μας. Με τον τρόπο αυτό αποφεύγουμε την επεξεργασία μη χρήσιμων κόμβων και βελτιώνουμε την ταχύτητα απόκρισης του συστήματος. Έχοντας πλέον ένα «καθαρό» DOM Tree, εφαρμόζονται επί αυτού αλγόριθμοι ομαδοποίησης, οι οποίοι μπορούν να ενταχθούν σε τρείς φάσεις: ομαδοποίηση ομοειδών κόμβων, ομαδοποίηση κόμβων διαφορετικού τύπου και ομαδοποίηση γειτονικών κόμβων.

Η ομαδοποίηση ομοειδών κόμβων είναι μια απλή διαδικασία η οποία αφενός χαρακτηρίζει τους κόμβους ανάλογα με τον τύπο του περιεχομένου τους π.χ. κείμενο, αφετέρου εντάσσει στην ίδια ομάδα κόμβους του ίδιου τύπου. Περιορίζεται, ωστόσο, από το γεγονός ότι για να επιτευχθεί η ομαδοποίηση θα πρέπει όλοι οι κόμβοι-παιδιά ενός κόμβου-πατέρα να έχουν περιεχόμενο ίδιου τύπου.

Ο προηγούμενος περιορισμός αίρεται κατά τη δεύτερη φάση ομαδοποίησης όπου επιχειρείται συνδυασμός κόμβων διαφορετικού τύπου. Εδώ κεντρικό ρόλο παίζει η έννοια τους βάρους που εκφράζει το μέγεθος των κόμβων στην ιστοσελίδα. Η χρήση της γίνεται σε συνδυασμό με τις παραμέτρους συνδυασμού τύπων οι οποίες ελέγχουν και καθορίζουν εντέλει αν θα πραγματοποιηθεί η ομαδοποίηση ή όχι. Με λίγα λόγια μπορούμε να πούμε ότι οι παράμετροι αυτές συγκρίνουν τα βάρη των κόμβων που ελέγχονται για συνδυασμό. Μια ακόμη παράμετρος που επηρεάζει την ομαδοποίηση των κόμβων είναι η προτεραιότητα τύπων. Με την παράμετρο αυτή εκφράζεται ο βαθμός σημαντικότητας των τύπων όπως εμείς τον θεωρούμε κατάλληλο στην ανάλυσή μας.

Τελευταία φάση στη διαδικασία ομαδοποίησης είναι ο συνδυασμός των γειτονικών κόμβων. Εδώ η έννοια «γειτονικός» αναφέρεται στον τρόπο που διατάσσονται τα στοιχεία στην ιστοσελίδα και όχι στον τρόπο που οργανώνονται στο DOM Tree. Η εφαρμογή της εν λόγω φάσης κρίνεται σκόπιμη αν αναλογιστούμε ότι στοιχεία που γειτονεύουν οπτικά στη σελίδα μπορεί να απέχουν αρκετά στο δένδρο.

3.2 Εκκαθάριση του DOM Tree

Όπως ήδη έχουμε αναφέρει, ο Html Parser μας δίνει σαν αποτέλεσμα το DOM Tree ενός Html αρχείου. Το προκύπτον όμως DOM Tree μπορεί να περιλαμβάνει κόμβους που δεν περιέχουν τιμή, δηλαδή είναι κενοί, ή κόμβους που εντέλει δεν παίζουν καθόλου ρόλο στην ανάλυσή μας. Τονίζουμε εδώ ότι θεωρούμε σωστά δομημένο Html αρχείο και επομένως σωστό DOM Tree και δεν μελετάμε την περίπτωση λανθασμένης σύνταξης της ιστοσελίδας. Ξεκινώντας, λοιπόν, με τη δεύτερη κατηγορία, αναφέρουμε πως οι κόμβοι που δεν περιέχουν ορατή για τον χρήστη πληροφορία, χωρίς ωστόσο να είναι κενοί, αλλά ούτε συμβάλλουν στην διαμόρφωση της τοπολογίας της ιστοσελίδας αντιστοιχούν στα εξής tags: *script*, *noscript*, *comment* και *meta*. Τους κόμβους αυτούς τους διαγράφουμε από το DOM Tree. Επιστρέφοντας στην κατηγορία των κενών κόμβων, διακρίνουμε δυο υποπεριπτώσεις. Στην πρώτη εντάσσονται οι κενοί κόμβοι που δεν περιέχουν κάποια ιδιότητα τοπολογίας, δεν καθορίζουν δηλαδή καμία είδους διάταξη, αλλά ο κύριος λόγος ύπαρξής τους είναι η μορφοποίηση του περιεχομένου μιας ιστοσελίδας. Εδώ περιλαμβάνονται οι κόμβοι που προκύπτουν από τα ακόλουθε tags: *b*, *big*, *em*, *font*, *i*, *small*, *strong*, *span*, *abbr*, *acronym*, *bdo*, *dfn*, *code*, *samp*, *kbd*, *var*, *cite*, *ins*, *del*, *label*, *optgroup*, *param*, *q*, *s*, *strike*, *style*, *sub*, *sup* και *tt*. Για να είμαστε περισσότερο ακριβείς, τα προηγούμενα tags υπονοούν οριζόντια διάταξη χωρίς να αποκλείουν ωστόσο και κατακόρυφη όταν επιπλέον οριζόντια επέκταση δεν είναι εφικτή. Αυτή ακριβώς η παρατήρηση καταδεικνύει την ασάφεια που υπάρχει γύρω από το θέμα της τοπολογικής τους τοποθέτησης.

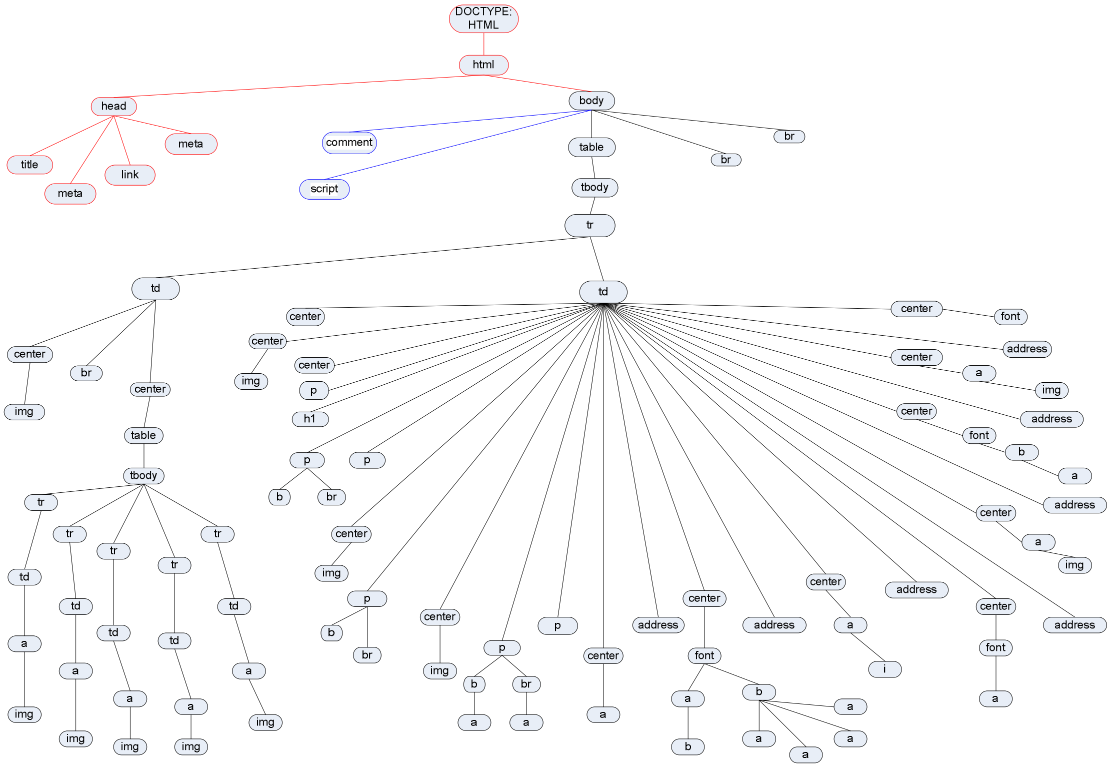
Μεταβαίνοντας στην δεύτερη υποπερίπτωση των κενών κόμβων, αναφέρουμε ότι τα tags που τα αντιπροσωπεύουν είναι: *br, dd, dl, dt, form, h1, h2, h3, h4, h5, h6, li, ol, menu, marquee, p, table, tr, ul, div, center, hr, address, caption, col, colgroup, dir, fieldset, legend, pre, tfoot* και *thead*. Τα προαναφερθέντα tags ορίζουν σαφώς κατακόρυφη διάταξη.

Από τις δυο υποπεριπτώσεις κενών κόμβων, τους κόμβους που ανήκουν στην πρώτη τους διαγράφουμε, ενώ εκείνους της δεύτερης τους διατηρούμε αφού θα τους χρησιμοποιήσουμε στη φάση της εύρεσης της τοπολογίας για να καθορίσουμε τη συνολική διάταξη της σελίδας.

Η παραπάνω περιγραφείσα διαδικασία διαγραφής μη χρήσιμων κόμβων είναι η δεύτερη σε μια σειρά δυο διαδικασιών που στόχο έχουν το «κλάδεμα» του DOM Tree έτσι ώστε αφενός να απαλειφθούν οι κόμβοι που δεν χρειάζονται στην ανάλυσή μας και αφετέρου να μειωθεί το μέγεθος του δένδρου. Η πρώτη διαδικασία αναλαμβάνει να εξάγει από το DOM Tree το κόμβο που αντιστοιχεί στο tag *body* αγνοώντας όλους τους υπόλοιπους κόμβους που βρίσκονται στο ίδιο ή υψηλότερο επίπεδο στο δένδρο. Αυτός είναι και ο κόμβος που συγκεντρώνει ολόκληρη την πληροφορία της ιστοσελίδας γι' αυτό και η όλη μετέπειτα επεξεργασία επί αυτού θα γίνει.

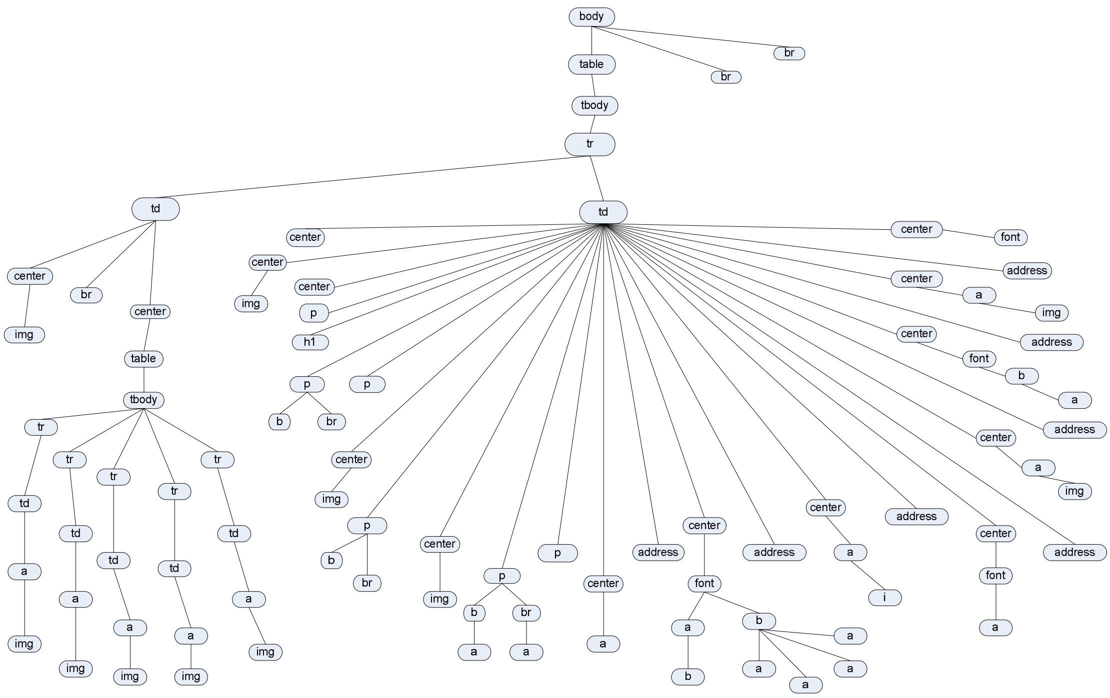
3.2.1 Παράδειγμα Εκκαθάρισης DOM Tree

Η όλη διαδικασία καθαρισμού του DOM Tree παρουσιάζεται στη συνέχεια μέσω ενός παραδείγματος. Συγκεκριμένα, χρησιμοποιούμε το DOM Tree του *Παραδείγματος 1*, το οποίο και παραθέτουμε ξανά χάριν απλότητας. Έχουμε χρησιμοποιήσει διαφορετικούς χρωματισμούς για τους κόμβους που επηρεάζονται από τις δυο διαδικασίες που περιγράφηκαν στην προηγούμενη ενότητα. Η κόμβοι με κόκκινο χρώμα μεταβάλλονται, στην ουσία διαγράφονται, από την (χρονικά) πρώτη διαδικασία που επιλέγει τον κόμβο body, ενώ εκείνοι με μπλε χρώμα διαγράφονται από τη διαδικασία εκκαθάρισης των κενών κόμβων.



Εικόνα 1: DOM Tree Παραδείγματος 1

Το DOM Tree που τελικά προκύπτει από τη φάση εκκαθάρισης και το οποίο στη συνέχεια θα χρησιμοποιηθεί από τη διαδικασία κατηγοριοποίησης των κόμβων βάσει του περιεχομένου τους, φαίνεται στην κάτωθι εικόνα.



Εικόνα 2 :DOM Tree Παραδείγματος 1 μετά από διαδικασία εκκαθάρισης

Αξίζει να παρατηρήσουμε την ύπαρξη στο τελικό δένδρο κόμβων οι οποίοι είναι κενοί. Χαρακτηριστικά αναφέρουμε τους κόμβους br. Αυτό γίνεται επειδή, όπως είπαμε, οι κόμβοι αυτή καθορίζουν τοπολογική διάταξη.

3.3 Βάρος Κόμβων

Σκοπός μας είναι να βρούμε ένα μέγεθος με το οποίο να μπορούμε να χαρακτηρίσουμε τους διάφορους τύπους κόμβων με ομοιόμορφο τρόπο έτσι ώστε τελικά να μας δίνεται η δυνατότητα να τους συγκρίνουμε μεταξύ τους. Όπως θα αναφέρουμε σε επόμενη ενότητα, έχουμε κόμβους τύπου κειμένου, εικόνας, ενσωματωμένου αντικειμένου κ.ά. Η ποικιλία αυτή περιορίζει τις επιλογές μας. Όλοι οι κόμβοι ωστόσο, ανεξαρτήτως τύπου, καταλαμβάνουν κάποιο χώρο στην σελίδα. Η ποσότητα αυτού του χώρου θα είναι το μέγεθος με το οποίο θα χαρακτηρίσουμε τους κόμβους. Στηριζόμαστε δηλαδή σε οπτικά κριτήρια, παρόλα αυτά χρησιμοποιούμε καταχρηστικά τον όρο *βάρος*.

3.3.1 Υπολογισμός Βάρους Κόμβων

Η τιμή του βάρους ενός κόμβου εκφράζει το μέγεθος του κόμβου, το πόσο μεγάλος δηλαδή είναι ο κόμβος και πόση συνολική επιφάνεια καταλαμβάνει στην ιστοσελίδα. Θα μπορούσε κάλλιστα να χρησιμοποιηθεί ο όρος εμβαδόν, επιλέχθηκε όμως το βάρος επειδή θέλουμε να δείξουμε ότι η τιμή αυτή δηλώνει τη βαρύτητα του κόμβου στην ιστοσελίδα, τη σημαντικότητά του. Έπεται προφανώς πως όσο μεγαλύτερο το βάρος ενός κόμβου τόσο μεγαλύτερο τμήμα της ιστοσελίδας καταλαμβάνει και άρα τόσο σημαντικότερος είναι.

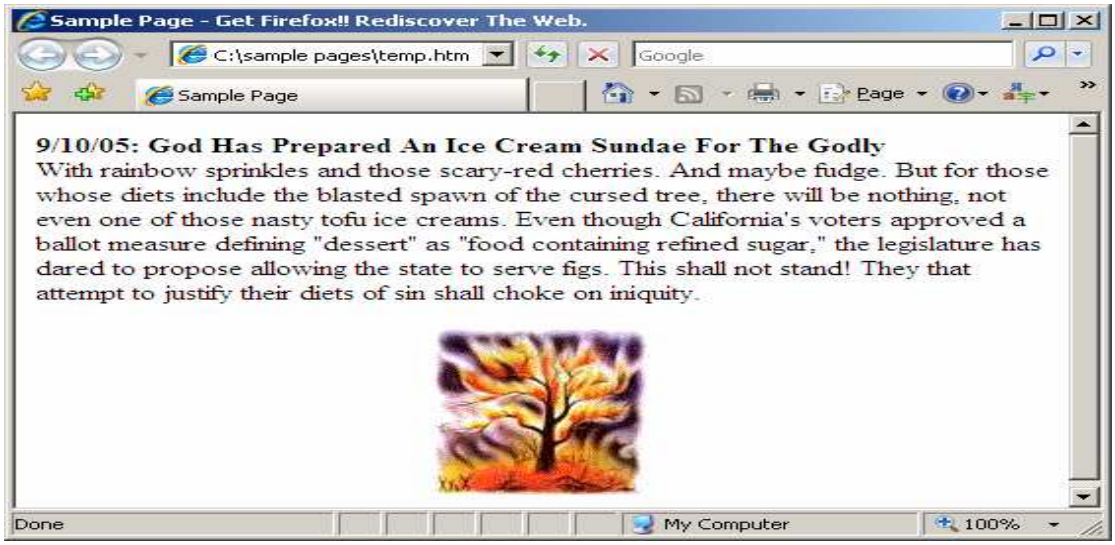
Τυπικά θα μπορούσαμε να πούμε ότι μονάδα μέτρησής του είναι το εικονοστοιχείο[6]. Με αφετηρία, λοιπόν, την τελευταία παρατήρηση μπορούμε να περιγράψουμε πως υπολογίζεται. Πρώτα όμως θα πρέπει να ανακαλέσουμε στη μνήμη μας ότι τα βασικά δομικά συστατικά οιασδήποτε ιστοσελίδας είναι τα ακόλουθα: *κείμενο*, *εικόνα*, *ενσωματωμένο αντικείμενο*[7] και διάφορα *στοιχεία φόρμας*. Εύκολα οδηγείται κανείς στο συμπέρασμα, τουλάχιστον για τα συστατικά *εικόνα* και *ενσωματωμένο αντικείμενο*, ότι το βάρος τους προκύπτει σαν το γινόμενο των δυο διαστάσεών τους: του ύψους και του πλάτους. Αυτός ο ορισμός βέβαια δεν ισχύει για το συστατικό «κείμενο». Αν σκεφτούμε όμως ότι το κείμενο αποτελείται από γράμματα και πως κάθε μέγεθος γραμματοσειράς έχει τυποποιημένες διαστάσεις σε εικονοστοιχεία, συνειρμικά καταλήγουμε στον εξής ορισμό: το βάρος για το συστατικό

[6] Μετάφραση του αγγλικού όρου pixel.

[7] Εκφρασμένο από τα tags <embed> και <object>.

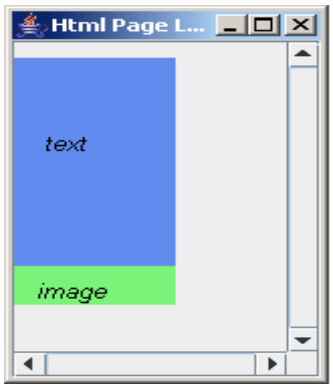
«κείμενο» ισούται με το γινόμενο του πλήθους των χαρακτήρων που αποτελείται επί το μέγεθος του κάθε χαρακτήρα σε εικονοστοιχεία για την αντίστοιχη γραμματοσειρά[8].

Στην Εικόνα 3 φαίνεται ένα κομμάτι της ιστοσελίδας του *Παραδείγματος 1*, ενώ στην Εικόνα 4 παρουσιάζεται το αντίστοιχο αποτέλεσμα για τους δυο κόμβους όπου γίνεται φανερή η διαφορά του βάρους τους.



Εικόνα 3: Τμήμα σελίδας Παραδείγματος 1

Σε αυτό το σημείο, και για να αποφύγουμε τυχόν παρερμηνείες της Εικόνας 4, πρέπει να διευκρινίσουμε ότι η απεικόνιση του μεγέθους των κόμβων με βάση το βάρος τους δεν σχετίζεται με την κάθε διάσταση ξεχωριστά (ύψος και πλάτος) αλλά με το συνολικό εμβαδό. Η διαδικασία που υπολογίζει την κάθε διάσταση περιγράφεται στο Κεφάλαιο 4 και χρησιμοποιεί την έννοια της γειτνίασης.



Εικόνα 4: Βάρος Κόμβων για τη σελίδα της Εικόνας 3

[8] Αυτό δεν είναι τελείως ακριβές. Στο Παράρτημα Α περιγράφεται η μεθοδολογία που ακολουθήθηκε για την εύρεση της επιφάνειας που καταλαμβάνει κάποιο κείμενο. Επίσης σε περίπτωση που δεν ορίζεται ρητά γραμματοσειρά, όλοι οι browsers έχουν κοινή default τιμή.

Σε ότι αφορά τα διάφορα στοιχεία φόρμας, αυτά είναι τα εξής: *text, password, textarea, option, submit, reset, radio, checkbox*. Από τα προηγούμενα στοιχεία, τα τέσσερα πρώτα σχετίζονται με κείμενο και καθορίζουν τις περισσότερες φορές ρητά το μέγεθός τους, σε διαφορετική περίπτωση υπακούουν σε τυποποιημένα μεγέθη. Για τα υπόλοιπα στοιχεία το μέγεθός τους συνήθως είναι τυποποιημένο, οπότε είναι γνωστό εκ των προτέρων. Όταν όμως έχουν ειδική διαμόρφωση, αυτή δηλώνεται μέσω των ιδιοτήτων τους που περικλείονται στα αντίστοιχα tags του Html αρχείου άρα και στην περίπτωση αυτή μπορούμε εύκολα να βρούμε το βάρος τους.

Ένα πρόβλημα που μπορεί να εμφανιστεί εφαρμόζοντας την παραπάνω περιγραφείσα μεθοδολογία υπολογισμού του βάρους αφορά τα συστατικά *εικόνα* και *ενσωματωμένο αντικείμενο* όταν δεν καθορίζεται κάποια ή και οι δυο από τις διαστάσεις τους. Όταν ισχύει το προηγούμενο σενάριο χρησιμοποιούνται διάφορες ευρετικές μέθοδοι οι οποίες λαμβάνουν υπόψη τα γειτνιάζοντα στοιχεία αλλά και το βάρος διάφορων κόμβων-προγόνων που καθορίζεται ρητά στο Html αρχείο (σαν ιχνοστοιχεία προφανώς). Με τον τρόπο αυτό προσπαθούμε να προσεγγίσουμε όσο γίνεται περισσότερο το πραγματικό τους βάρος.

3.4 Ομαδοποίηση Κόμβων

Η διαδικασία της ομαδοποίησης των κόμβων του DOM Tree έγκειται στην προσπάθεια ελέγχου της τυπολογικής σχέσης των διάφορων στοιχείων της σελίδας και στη συνένωση εκείνων που σχετίζονται. Αποτελεί, θα μπορούσαμε να πούμε, ένα εγχείρημα ανακάλυψης της σημασιολογικής δομής της ιστοσελίδας όπως τη φαντάστηκε ο δημιουργός της κατά της φάση σχεδιασμού. Κατά γενική παραδοχή, ο σχεδιαστής μιας σελίδας τοποθετεί στοιχεία που σχετίζονται ή που συνδυαζόμενα συνθέτουν ένα μεγαλύτερο στοιχείο κοντά το ένα στο άλλο. Ενώ στην απεικόνιση της σελίδας η εγγύτητα δυο στοιχείων είναι προφανής σαν έννοια και εμφανής σαν ισχύουσα ιδιότητα, δεν ορίζεται μονοσήμαντα πάνω στη δομή του DOM Tree. Με άλλα λόγια, στοιχεία που γειτονεύουν στη σελίδα συμβαίνει να βρίσκονται μακριά στο δένδρο. Εάν σε αυτό συνυπολογίσουμε και το γεγονός ότι διαφορετικά στοιχεία μπορούν να συνδυαστούν προς σύνθεση ενός νέου, αντιλαμβανόμαστε την ανάγκη για υιοθέτηση μια σταδιακής και βαθμιαίας διαδικασίας ομαδοποίησης, η οποία ξεκινάει με την απλούστερη των περιπτώσεων που είναι η ομαδοποίηση κόμβων με περιεχόμενο του ίδιου τύπου.

3.4.1 Ομαδοποίηση Ομοειδών Κόμβων

Πρώτο βήμα στην ενοποίηση των κόμβων βάσει του περιεχομένου τους είναι ο χαρακτηρισμός, ως προς τον τύπο τους, εκείνων των κόμβων των οποίων τα παιδιά είναι ομοειδή, έχουν δηλαδή όλα περιεχόμενο του ίδιου τύπου. Προφανώς και ο κόμβος-πατέρας

θα αποκτήσει τον ίδιο τύπο με τους κόμβους-παιδιά του. Η διαδικασία αυτή ξεκινάει από τα φύλλα και γίνεται αναδρομικά ανεβαίνοντας προς τη ρίζα. Οι πρωταρχικοί τύποι κόμβων που μπορούμε να έχουμε καταγράφονται στον Πίνακα 1.

Τύπος
κείμενο
εικόνα
σύνδεσμος-κείμενο
σύνδεσμος-εικόνα
κενό
διαχωριστής[9]
ενσωματωμένο αντικείμενο
φόρμα
br

Πίνακας 1: Πρωταρχικοί Τύποι Περιεχομένου Κόμβων

Οι κόμβοι που έχουν κόμβους-παιδιά διαφορετικών τύπων χαρακτηρίζονται ως *σύνθετοι*. Ο χαρακτηρισμός ενός κόμβου ως σύνθετος αυτόματα συνεπάγεται παρόμοιο χαρακτηρισμό όλων των κόμβων-προγόνων του.

Πρέπει να τονίσουμε εδώ ότι δεν εξετάζουμε σε αυτή τη φάση ένωση κόμβων διαφορετικού τύπου το οποίο ενδεχομένως να οδηγούσε στη δημιουργία ενός νέου τύπου. Η εν λόγω διαδικασία γίνεται σε μετέπειτα φάση και προαπαιτεί τη λειτουργία που εξηγούμε σε αυτή την ενότητα.

Η διαδικασία ομαδοποίησης που αναλύουμε δεν επηρεάζει καθόλου τη δομή του DOM Tree, παρά μόνο χαρακτηρίζει τους κόμβους του. Αυτό επιδρά όμως στην επακόλουθη επεξεργασία του δένδρου. Συγκεκριμένα, οι κόμβοι που χαρακτηρίζονται με έναν από τους πρωταρχικούς τύπους του Πίνακα 1, θεωρούνται πλέον «τερματικοί» κόμβοι, αδιαίρετες μονάδες οι οποίες δεν έχουν εσωτερική δομή και δεν επιδέχονται περαιτέρω ανάλυση. Παρόλα αυτά συνεχίζουν να έχουν το συνολικό βάρος που προκύπτει από τους κόμβους-παιδιά τους και το οποίο

[9] Αντιστοιχεί στο tag <separator>. Ξεχωρίζουμε τον τύπο αυτό επειδή παίζει σημαντικό ρόλο στον οπτικό διαχωρισμό των στοιχείων ης σελίδας.

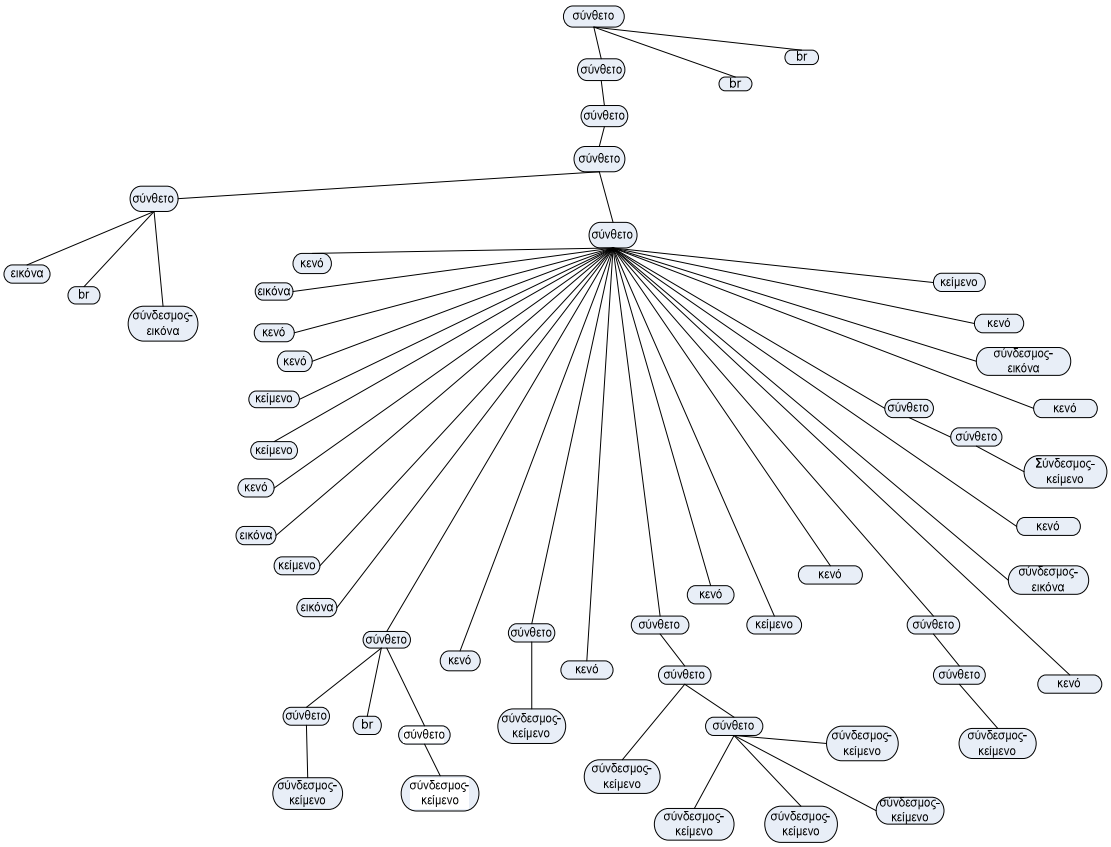
υπολογίστηκε όπως περιγράφει η Ενότητα 3.2 . Στην ουσία οι κόμβοι αυτοί θεωρούνται φύλλα και είναι σαν να έχουμε κλαδέψει το δένδρο. Αυτές οι αδιαίρετες μονάδες αποτελούν τα δομικά στοιχεία του τελικού αποτελέσματος που περιέχει την τοπολογική διάταξη της ιστοσελίδας.

Στην υπό-ενότητα που ακολουθεί παρουσιάζεται ένα παράδειγμα ομαδοποίησης ομοειδών κόμβων.

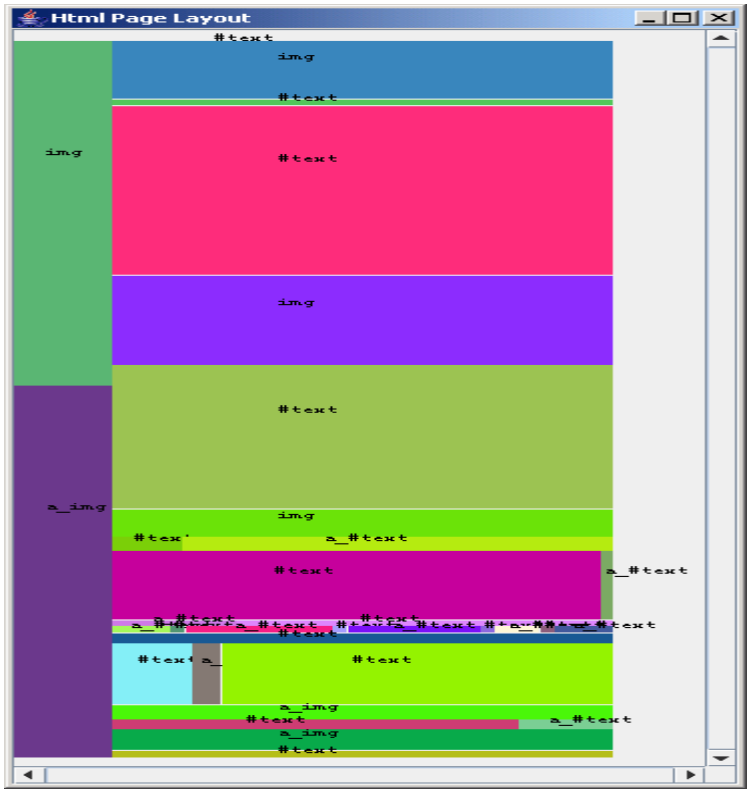
3.4.1.1 Παράδειγμα Ομαδοποίησης Ομοειδών Κόμβων

Παρέχοντας στην διαδικασία ομαδοποίησης ομοειδών κόμβων το DOM Tree του *Παραδείγματος 1* όπως αυτό έχει προκύψει από τη διαδικασία τοπολογικής ομαδοποίησης κόμβων, παίρνουμε το δένδρο που απεικονίζεται στην Εικόνα 5. Στην εικόνα εμφανίζονται μόνο οι τύποι των κόμβων όπως προέκυψαν από την προηγούμενη διαδικασία και όχι τα ονόματά τους. Η αντίστοιχη διάταξη τοπολογίας φαίνεται στην Εικόνα 6.

Αξιοσημείωτη είναι η συρρίκνωση που έχει γίνει στο δένδρο. Αντίστοιχη μείωση των δομικών μονάδων παρατηρείται και στην διάταξη τοπολογίας. Στην πραγματικότητα δεν έχει διαγραφεί κανένας κόμβος από αυτούς που εξαφανίστηκαν. Υπάρχουν και παραμένουν συνδεδεμένοι στο δένδρο, απλώς εμείς δεν θα χρειαστεί να τους επεξεργαστούμε. Ο λόγος που δεν τους διαγράφουμε είναι το γεγονός ότι μας ενδιαφέρει το περιεχόμενό τους, το οποίο τελικά πρέπει να δώσουμε σαν έξοδο του προγράμματος.

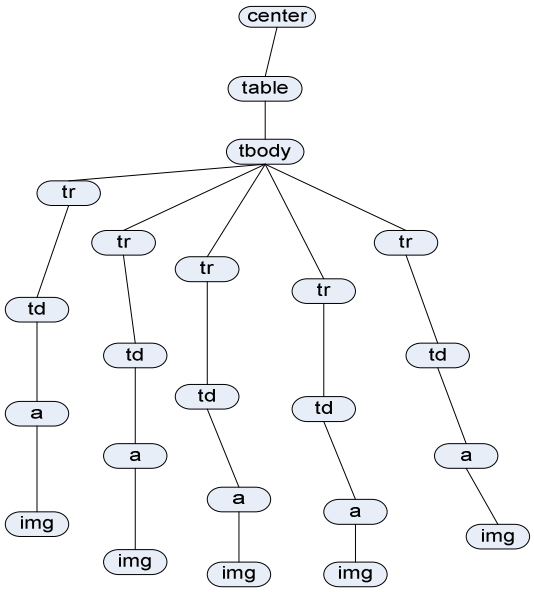


Εικόνα 5: Ομαδοποίηση ομοειδών κόμβων



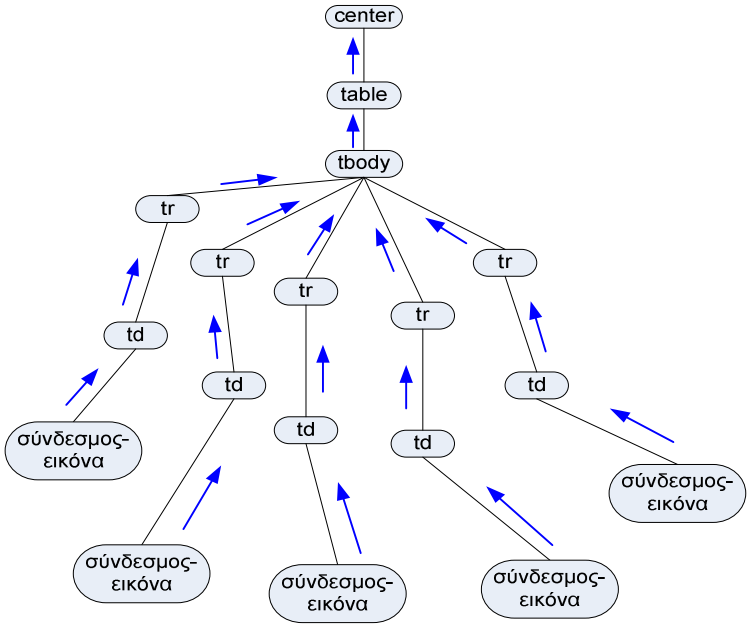
Εικόνα 6: Τοπολογική Διάταξη μετά από Ομαδοποίηση Ομοειδών Κόμβων

Ας δούμε όμως αναλυτικότερα πως ένα κομμάτι του προηγούμενου δένδρου χαρακτηρίζεται ως προς το περιεχόμενό του και συγκεκριμένα το υπό-δένδρο που φαίνεται στην Εικόνα 7.

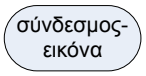


Εικόνα 7: Τμήμα του DOM Tree του Παραδείγματος 1

Όπως είπαμε, η ομαδοποίηση ξεκινάει από τα φύλλα και προχωράει αναδρομικά προς τα πάνω. Σε πρώτη φάση έχουμε, λοιπόν, το αποτέλεσμα της Εικόνας 8. Καθώς η ομαδοποίηση ανεβαίνει σε υψηλότερο επίπεδο, ο χαρακτηρισμός των κόμβων έχει σαν αποτέλεσμα την συρρίκνωση του δένδρου και τελικά προκύπτει ένας μοναδικός κόμβος με τον τύπο που φαίνεται στην Εικόνα 9.



Εικόνα 8: Διαδικασία Ομαδοποίησης Ομοειδών Κόμβων



Εικόνα 9: Αποτέλεσμα Ομαδοποίησης Ομοειδών Κόμβων

3.4.2 Ομαδοποίηση Κόμβων Διαφορετικού Τύπου

Η ομαδοποίηση ομοειδών κόμβων που περιγράφηκε στην προηγούμενη ενότητα είναι απλή και έχει περιορισμένα αποτελέσματα. Συνήθως επιτυγχάνει μικρό βαθμό ομαδοποίησης εξαιτίας της αυστηρής απαίτησης που θέσαμε πως όλοι οι κόμβοι-παιδιά ενός κόμβου πρέπει να είναι του ίδιου τύπου. Υπάρχει όμως η περίπτωση οι κόμβοι-παιδιά ενός κόμβου να δημιουργούν περισσότερες της μια ομάδες (πάντα ως προς το περιεχόμενό τους) ή ακόμη κόμβοι διαφορετικού τύπου να μπορούν να συνδυαστούν για να δώσουν κόμβο νέου τύπου. Για το λόγο αυτό δημιουργήσαμε τη διαδικασία ομαδοποίησης κόμβων διαφορετικού τύπου που στην ουσία ελέγχει για ύπαρξη υπό-ομάδων στο σύνολο των κόμβων-παιδιών ενός κόμβου. Η διαδικασία αυτή χρησιμοποιεί την έννοια τους βάρους του κόμβου που εισάγαμε στην Ενότητα 3.2 .

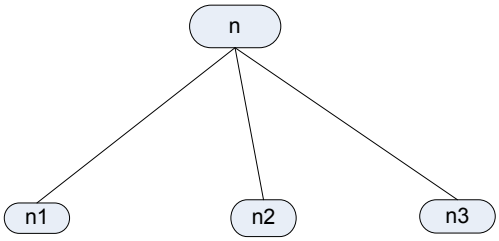
Το αποτέλεσμα της διαδικασίας είναι η δημιουργία δυο ή περισσότερων ομάδων κόμβων εντός του ίδιου κόμβου-πατέρα. Τι τύπο όμως θα έχουν οι ομάδες που θα προκύψουν; Το θέμα αυτό μελετάμε στην επόμενη υπό-ενότητα.

3.4.2.1 Συνδυασμός πρωταρχικών τύπων

Οι πρωταρχικοί τύποι είναι αυτοί που παρουσιάζονται στον Πίνακα 1. Ένας κόμβος θα μπορούσε να έχει λιγότερα παιδιά από το πλήθος των πρωταρχικών τύπων, το ίδιο με το πλήθος τους ή και περισσότερα. Θα περίμενε κανείς επομένως τον έλεγχο όλων των δυνατών συνδυασμών που θα μπορούσαν να προκύψουν από τους πρωταρχικούς τύπους. Ένας τέτοιος έλεγχος σίγουρα θα έλυνε το πρόβλημα, δεν είναι όμως ούτε αποδοτικός (εξαιτίας του τεράστιου πλήθους των δυνατών συνδυασμών), και κυρίως, ούτε αναγκαίος αν λάβουμε υπόψη τα ιδιαίτερα χαρακτηριστικά του προβλήματος που καλούμαστε να λύσουμε.

Αναλυτικότερα, δυο κόμβοι μπορούν να συνδυαστούν, ανεξαρτήτως του αποτελέσματος που θα δώσουν, μόνο και μόνο αν είναι διαδοχικοί κόμβοι-παιδιά ενός κοινού κόμβου-πατέρα. Αυτό σημαίνει πως οι κόμβοι αυτοί πρέπει να γειτονεύουν μεταξύ τους. Από την παρατήρηση αυτή συμπεραίνουμε αμέσως πως χρειάζεται να ελέγξουμε μόνο τους ανά δυο συνδυασμούς πρωταρχικών τύπων. Ας εξηγήσουμε όμως την απαίτηση αυτή με ένα θεωρητικό παράδειγμα.

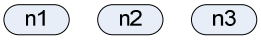
Θεωρούμε το δένδρο της Εικόνας 10. Για την εξήγησή μας δεν έχουν σημασία οι τύποι των κόμβων n1, n2 και n3.



Εικόνα 10: Ομαδοποίηση Γειτονικών Κόμβων

Στην ενότητα εύρεσης της τοπολογικής διάταξης της ιστοσελίδας θα εξηγήσουμε πως οι κόμβοι-παιδιά ενός κόμβου πρέπει να έχουν όλοι τον ίδιο προσανατολισμό: είτε οριζόντιο είτε κατακόρυφο. Επομένως οι τρεις κόμβοι της Εικόνας 10 μπορούν να διαταχθούν τοπολογικά είτε σύμφωνα με την Εικόνα 11 είτε σύμφωνα με την Εικόνα 11.

Εικόνα 11: Κατακόρυφη Τοποθέτηση



Εικόνα 12: Οριζόντια Τοποθέτηση

Συμπεραίνουμε, λοιπόν, πως μπορούμε να έχουμε συνδυασμούς μόνο μεταξύ δυο κόμβων: n1 με n2 ή n2 με n3. Έτσι τεκμηριώνεται η απαίτηση που θέσαμε.

Όπως έχουμε ήδη αναφέρει, οι τύποι *κενό* και *br* αποτελούν ταυτοτικά στοιχεία, δηλαδή απορροφώνται από όλους τους υπόλοιπους τύπους, ενώ οι τύποι *separator*, *embedded object* και *φόρμα* δεν συνδυάζονται με κανέναν άλλο τύπο παρά μόνο με τον εαυτό τους. Στον Πίνακα 2 παρουσιάζονται όλοι οι υπόλοιποι συνδυασμοί.

Τύπος 1	Τύπος 2	Αποτέλεσμα
κείμενο	εικόνα	κείμενο, σύνθετο
κείμενο	σύνδεσμος-κείμενο	υπερκείμενο, σύνθετο
κείμενο	σύνδεσμος-εικόνα	υπερκείμενο, σύνθετο
εικόνα	σύνδεσμος-κείμενο	σύνδεσμος-κείμενο, σύνθετο
εικόνα	σύνδεσμος-εικόνα	σύνδεσμος-εικόνα, σύνθετο
σύνδεσμος-κείμενο	σύνδεσμος-εικόνα	σύνθετο

Πίνακας 2: Συνδυασμοί πρωταρχικών τύπων

Εκτός από τη δημιουργία ενός νέου τύπου (τύπος *υπερκείμενο*), παρατηρούμε από τον πίνακα ότι ο συνδυασμός δυο τύπων δεν είναι μονοσήμαντος αλλά εξαρτάται από κάποια παράμετρο συσχέτισής τους. Τις παραμέτρους αυτές, οι οποίες συνδέονται με τον λόγο των

βαρών των κόμβων, αναλύουμε στην επόμενη υπό-ενότητα. Ανάλογα, λοιπόν, με τη τιμή της παραμέτρου μπορεί να προκύψει διαφορετικό αποτέλεσμα.

Ένα χαρακτηριστικό παράδειγμα συνδυασμού πρωταρχικών τύπων και δημιουργίας ενός νέου φαίνεται στην ακόλουθη εικόνα, τμήμα της σελίδας του Παραδείγματος 1. Τα στοιχεία κειμένου και συνδέσμου-κειμένου που διακρίνονται τελικά θα συνδυαστούν για να δώσουν ένα μόνο στοιχείο τύπου υπερκειμένου.

Εικόνα 13: Παράδειγμα Συνδυασμού Πρωταρχικών Τύπων

Στο σημείο αυτό οφείλουμε να πούμε πως οι συνδυασμοί που παρουσιάζονται στον Πίνακα 2 δεν είναι μοναδικοί, απλά αποτελούν δική μας επιλογή έχοντας κατά νου το αποτέλεσμα που θέλουμε να πετύχουμε. *Συγκεκριμένα, στην ανάλυσή μας ενδιαφερόμαστε περισσότερο για στοιχεία κειμένου, είτε απλού είτε συνδέσμου, αφού οι εικόνες δεν περιέχουν καμία χρήσιμη πληροφορία να επεξεργαστούμε κατά τη διαδικασία του geoparsing.* Γι' αυτόν ακριβώς τον λόγο μικρές εικόνες προσπαθούμε να τις απαλείψουμε ενσωματώνοντάς τες σε μεγάλα στοιχεία κειμένου, ενώ ακόμη και μικρά στοιχεία κειμένου μας είναι χρήσιμα έτσι τα διατηρούμε ακόμη και αν βρίσκονται κοντά σε μια τεράστια εικόνα. Ο λόγος που δεν διαγράφουμε τις εικόνες είναι το γεγονός ότι επηρεάζουν τη τοπολογική διάταξη της ιστοσελίδας.

Η ανάλυση του συνδυασμού των τύπων δεν θα ήταν πλήρης αν δεν παρουσιάσουμε και τον συνδυασμό του νέου τύπου «υπερκείμενο» με τους υπόλοιπους τύπους. Τους συνδυασμούς αυτούς παραθέτουμε στον Πίνακα 3.

Τύπος 1	Τύπος 2	Αποτέλεσμα
υπερκείμενο	εικόνα	υπερκείμενο, σύνθετο
υπερκείμενο	σύνδεσμος-κείμενο	υπερκείμενο, σύνθετο
υπερκείμενο	σύνδεσμος-εικόνα	υπερκείμενο, σύνθετο
υπερκείμενο	κείμενο	υπερκείμενο, σύνθετο

Πίνακας 3: Συνδυασμοί Τύπου 'Υπερκείμενο'

Και στην περίπτωση αυτή ο συνδυασμός δεν είναι μονοσήμαντος ούτε ντετερμινιστικός, αλλά εξαρτάται από παραμέτρους.

Έχοντας εξετάσει τους συνδυασμούς τύπων, μελετάμε στη συνέχεια τις παραμέτρους αυτές που καθορίζουν το τελικό αποτέλεσμα που θα προκύψει από έναν συνδυασμό.

3.4.2.2 Παράμετροι Συνδυασμού τύπων

Ο ντετερμινιστικός συνδυασμός τύπων είναι μια εύκολη και απλή προσέγγιση η οποία σαφώς και δουλεύει, η ορθότητα όμως των αποτελεσμάτων της είναι αμφίβολη. Υπάρχει κίνδυνος να οδηγήσει σε υπεραπλουστεύσεις, πραγματοποιώντας υπερβολικούς συνδυασμούς, ή να μην γίνουν καθόλου συνδυασμοί οπότε και έχουμε υπερβολική ανάλυση. Επιπλέον, αφαιρείται από τον χρήστη η δυνατότητα ελέγχου και ρύθμισης του βαθμού λεπτομέρειας της τοπολογικής διάταξης ανάλογα με το τελικό αποτέλεσμα που θέλει να πετύχει. Για τους προαναφερθέντες λόγους θεωρούμε παραμέτρους που καθορίζουν τον τελικό τύπο που θα προκύψει από τον συνδυασμό δυο άλλων.

Το πλήθος των παραμέτρων θα πρέπει να ισούται με το πλήθος των δυνατών συνδυασμών ανά δυο. Οι παράμετροι αυτές φαίνονται στον Πίνακα 4.

Τύπος 1	Τύπος 2	Παράμετρος Αποτελέσματος
κείμενο	εικόνα	*imageToTextRatio*
κείμενο	σύνδεσμος-κείμενο	*textToA_textRatio*
κείμενο	σύνδεσμος-εικόνα	*a_imageToTextRatio*
εικόνα	σύνδεσμος-κείμενο	*imageToA_textRatio*
εικόνα	σύνδεσμος-εικόνα	*imageToA_imageRatio*
υπερκείμενο	εικόνα	*imageToHypertextRatio*
υπερκείμενο	σύνδεσμος-κείμενο	*a_textToHypertextRatio*
υπερκείμενο	σύνδεσμος-εικόνα	*a_imageToHypertextRatio*
υπερκείμενο	κείμενο	*textToHypertextRatio*

Πίνακας 4: Παράμετροι Ελέγχου Συνδυασμών

Θα εξετάσουμε τώρα πως μεταβάλλεται η τελική τοπολογική διάταξη της ιστοσελίδας του Παραδείγματος 1 για διάφορους συνδυασμούς των παραπάνω παραμέτρων.

Στην πρώτη εκτέλεση παρουσιάζουμε μια αναλυτική εικόνα της τοπολογικής διάταξης. Οι τιμές των παραμέτρων φαίνονται στον Πίνακα 5 και η αντίστοιχη έξοδος στην Εικόνα 14.

Παράμετρος	Τιμή
imageToTextRatio	*0.1*
textToA_textRatio	*0.2*
a_imageToTextRatio	*0.1*
imageToA_textRatio	*0.1*
imageToA_imageRatio	*0.1*
imageToHypertextRatio	*0.1*
a_textToHypertextRatio	*0.2*
a_imageToHypertextRatio	*0.1*
textToHypertextRatio	*0.2*

Πίνακας 5: Τιμές Παραμέτρων για Λεπτομερή Ανάλυση

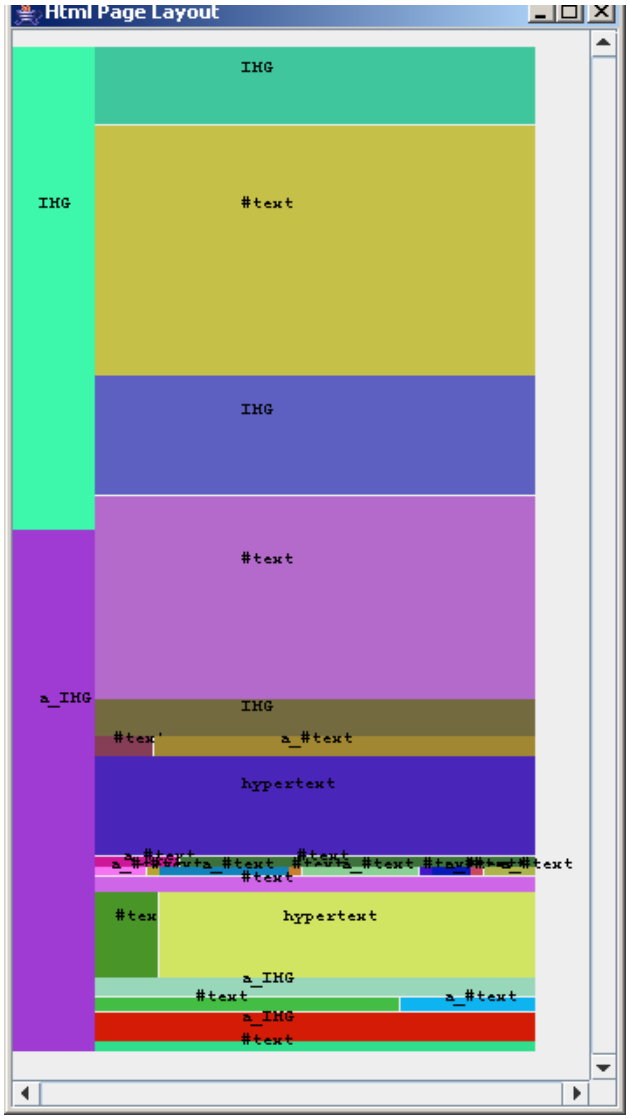
Παράμετρος	Τιμή
imageToTextRatio	*0.4*
textToA_textRatio	*10.2*
a_imageToTextRatio	*5.5*
imageToA_textRatio	*0.3*
imageToA_imageRatio	*0.3*
imageToHypertextRatio	*1.4*
a_textToHypertextRatio	*10.2*
a_imageToHypertextRatio	*5.1*
textToHypertextRatio	*0.0*

Πίνακας 6: Τιμές Παραμέτρων για Χονδρική Ανάλυση

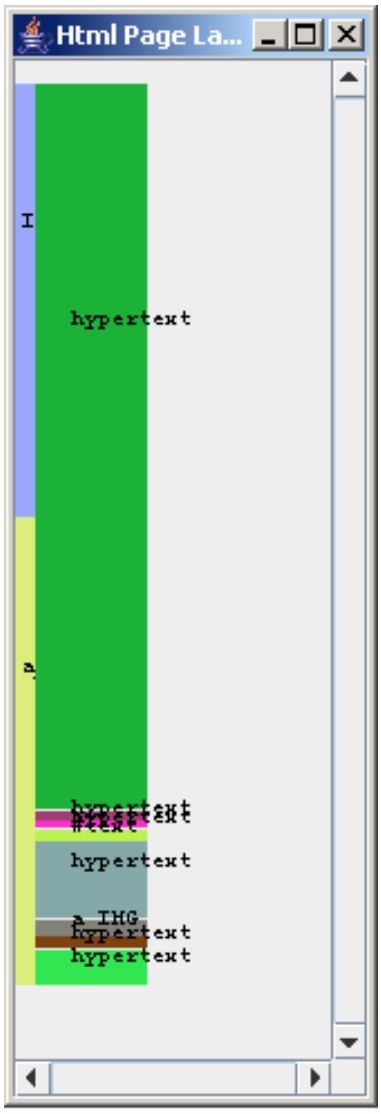
Στην δεύτερη εκτέλεση παρουσιάζουμε μια πιο χονδρική εικόνα της τοπολογικής διάταξης. Οι τιμές των παραμέτρων φαίνονται στον Πίνακα 6 και η αντίστοιχη τοπολογική διάταξη στην Εικόνα 15.

Στο σημείο αυτό πρέπει να αναφερθούμε στον τρόπο που βρήκαμε τις τιμές των πινάκων 5 και 6. Καταρχάς διευκρινίζεται ότι οι τιμές αυτές δεν είναι στατικές απλά αποτελούν κάποιο στιγμιότυπο των παραμέτρων του Πίνακα 4. Οι τιμές οριστικοποιήθηκαν μετά από πολλές δοκιμές κατά τις οποίες επιχειρούσαμε να πετύχουμε διάφορα επίπεδα λεπτομέρειας για την ιστοσελίδα που μελετάμε. Κρίναμε ικανοποιητικές τις προηγούμενες τιμές επειδή δίνουν εμφανώς διαφορετικά αποτελέσματα. Βέβαια για κάθε συνδυασμό τιμών παίρναμε διαφορετικό αποτέλεσμα, απλώς οι διαφοροποιήσεις τις οποίες θέλουμε να επισημάνουμε δεν ήταν αρκετές. Τέλος, οφείλουμε να τονίσουμε ότι άλλες σελίδες απαιτούν άλλους συνδυασμούς τιμών.

Από τα προηγούμενα παραδείγματα διαπιστώνουμε πως οι τιμές των παραμέτρων μπορούν να ρυθμιστούν ανάλογα με την εφαρμογή και το επίπεδο λεπτομέρειας που επιθυμεί ο χρήστης. Σίγουρα όμως δεν υπάρχει ένα σύνολο τιμών για τις παραμέτρους το οποίο να ισχύει για όλες τις ιστοσελίδες. Αντίθετα κάθε ιστοσελίδα έχει κάποια ιδιαίτερα χαρακτηριστικά τα οποία μπορούν να αναδειχθούν μόνο μέσα από τη μεταβολή των προηγούμενων παραμέτρων



Εικόνα 14: Τοπολογία σελίδας Παραδείγματος 1 για Λεπτομερή Ανάλυση



Εικόνα 15: Τοπολογία σελίδας Παραδείγματος 1 για Χονδρική Ανάλυση

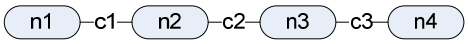
3.4.2.3 Αλγόριθμος Ομαδοποίησης

Όπως έχουμε ήδη αναφέρει, ένας κόμβος μπορεί να ενωθεί μόνο με έναν άλλο που γειτονεύει άμεσα μαζί του. Επιπλέον, η ένωση μπορεί να πραγματοποιηθεί μονάχα αν ο έλεγχος για την αντίστοιχη παράμετρο της ενότητας 3.4.2.2 ικανοποιείται. Αυτές τις δυο προϋποθέσεις ο αλγόριθμος ομαδοποίησης πρέπει να τις υπακούει. Επιπρόσθετα, θα πρέπει να πραγματοποιεί όλους τους δυνατούς συνδυασμούς κόμβων, γεγονός που σημαίνει πως ένας κόμβος μπορεί να συμμετέχει σε περισσότερους του ενός συνδυασμούς. Σε ποιόν τελικά θα ενταχθεί αποφασίζεται σε επόμενη φάση.

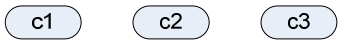
Η περιγραφή του αλγορίθμου θα γίνει αρχικά μέσω ενός αφηρημένου παραδείγματος. Η έξοδός του είναι ένας γράφος συνδυασμών, ο οποίος έχει ιδιαίτερη μορφή. Πρόκειται δηλαδή για έναν γράφο του οποίου οι κόμβοι είναι συνδυασμοί, όχι μόνο κόμβων αλλά και συνδυασμών (ως οντότητες πλέον) μεταξύ τους.

Θεωρούμε τους παρακάτω τέσσερις κόμβους. Όλοι έχουν κοινό πατέρα αλλά δεν έχουν όλοι κοινό τύπο, μπορούν ωστόσο να συνδυαστούν και να δώσουν νέο τύπο.

Έστω ότι όλοι συνδυάζονται με τους γείτονές τους. Τότε προκύπτει η επόμενη εικόνα στην οποία φαίνονται οι συνδυασμοί αυτοί.



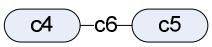
Οι συνδυασμοί c1, c2 και c3 έχουν συγκεκριμένο τύπο ο οποίος έχει προκύψει σύμφωνα με τους κανόνες του Πίνακα 3 και τις τιμές των αντίστοιχων παραμέτρων του Πίνακα 4. Στο στάδιο αυτό θεωρούμε τους παραπάνω συνδυασμούς σαν κόμβους, οπότε έχουμε την ακόλουθη εικόνα:



Εφαρμόζοντας του ίδιους κανόνες συνδυασμών τύπων προκύπτουν οι κάτωθι συνδυασμοί:

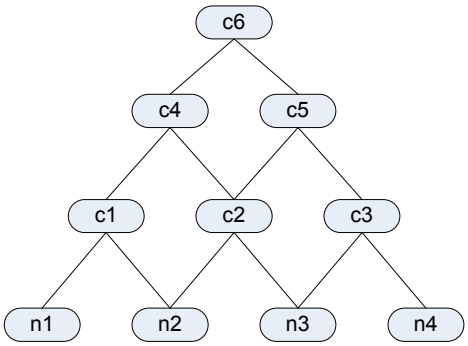
Εργαζόμενοι με τον ίδιο τρόπο παίρνουμε:

Τα οποία συνδέονται:



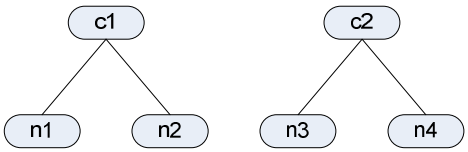
Και δίνουν τον μοναδικό συνδυασμό:

Τελικά προκύπτει ο επόμενος γράφος συνδυασμών:



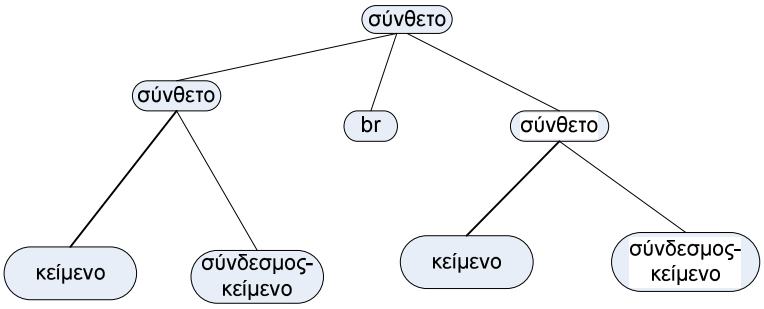
Η έννοια των ακμών είναι η εξής: ο κόμβος-σύνδεση c1 έχει δημιουργηθεί από τον συνδυασμών των κόμβων n1 και n2. Εφαρμόζοντας τον ίδιο ορισμό, αναδρομικά αυτή τη φορά, για τον τελικό κόμβο-σύνδεση c6 έχουμε ότι ο κόμβος αυτός προήλθε από των συνδυασμό των κόμβων n1, n2, n3 και n4. Επομένως και οι τέσσερις κόμβοι, παρόλο που δεν είχαν τον ίδιο τύπο, ενώθηκαν τελικά και έδωσαν νέο τύπο στον κόμβο-πατέρα τους.

Αν θεωρήσουμε ότι πραγματοποιούνται μόνο οι συνδυασμοί των ζευγών n1, n2 και n3, n4 ο τελικός γράφος συνδυασμών θα είχε την ακόλουθη μορφή:



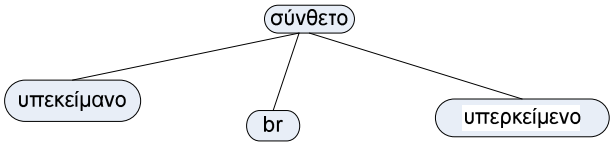
Στην περίπτωση αυτή θα είχαμε δυο ομάδες κόμβων, διαφορετικού τύπου η κάθε μια, οι οποίες δεν μπορούν να συνδυαστούν περαιτέρω. Ο κόμβος-πατέρας των τεσσάρων κόμβων n1, n2, n3 και n4 θα χαρακτηριζόταν ως *σύνθετος*.

Στο σημείο αυτό θα παρουσιάσουμε και ένα πραγματικό παράδειγμα. Για το λόγο αυτό θα χρησιμοποιήσουμε τμήμα της ιστοσελίδας του *Παραδείγματος 1*, την οποία και παρουσιάσαμε στην Εικόνα 13. Το αντίστοιχο κομμάτι του DOM Tree φαίνεται στην Εικόνα 16. Οι κόμβοι έχουν ετικέτα με βάση τον τύπο του περιεχομένου τους.



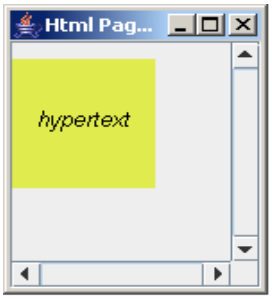
Εικόνα 16: Τμήμα του DOM Tree του Παραδείγματος 1

Σε πρώτη φάση γίνεται η επεξεργασία των σύνθετων κόμβων-παιδιών, οι οποίοι χαρακτηρίζονται ως υπερκείμενο όπως φαίνεται και στην Εικόνα 17.



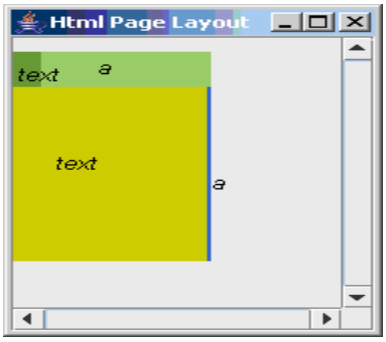
Εικόνα 17: Αποτέλεσμα Ομαδοποίησης Κόμβων Διαφορετικού Τύπου

Στη συνέχεια όλος ο κόμβος χαρακτηρίζεται ως υπερκείμενο. Η τελική έξοδος φαίνεται στην επόμενη εικόνα.



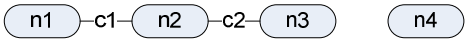
Εικόνα 18: Τελικό Αποτέλεσμα Ομαδοποίησης Κόμβων Διαφορετικού Τύπου

Αξίζει να την αντιπαραθέσουμε με την έξοδο για τον ίδιο κόμβο πριν την εφαρμογή όμως του αλγορίθμου ομαδοποίησης. Η εικόνα της αρχικής κατάστασης φαίνεται ακολούθως.



Εικόνα 19: Τοπολογική Διάταξη πριν την Ομαδοποίηση

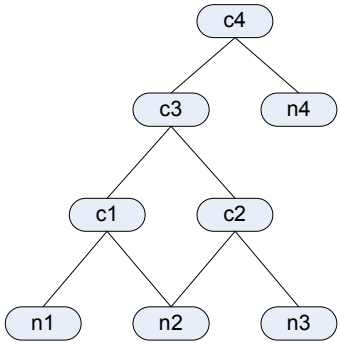
Θεωρούμε χρήσιμο, καθώς θα μας χρειαστεί αργότερα στην ανάλυσή μας, να παρουσιάσουμε και ένα άλλο, συχνά παρατηρούμενο, σενάριο εκτέλεσης του αλγορίθμου ομαδοποίησης. Έχουμε τους ίδιους τέσσερις κόμβους n1, n2, n3 και n4 με τη διαφορά ότι τώρα πραγματοποιούνται μόνο οι τρεις πρώτες συνδέσεις.



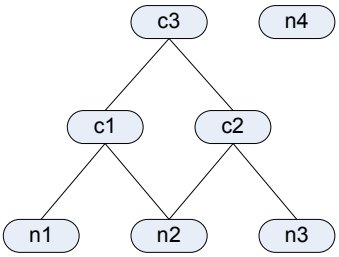
Στο επόμενο βήμα εξετάζουμε όχι μόνο τους νέους κόμβους-συνδέσεις c1 και c2 αλλά και τον κόμβο n4, έχουμε δηλαδή την επόμενη εικόνα.

Θεωρώντας ότι πραγματοποιείται μόνο ο συνδυασμός μεταξύ c1 και c2 έχουμε το αποτέλεσμα της κάτωθι εικόνας.

Σε περίπτωση που ο συνδυασμός μεταξύ αυτών των κόμβων πραγματοποιείται προκύπτει ο επόμενος γράφος συνδυασμών.



Δηλαδή όλοι οι κόμβοι συνδυάζονται τελικά για να δώσουν έναν νέο τύπο. Αν πάλι ο συνδυασμός μεταξύ c3 και n4 δεν επιτυγχάνεται, τότε προκύπτει ο παρακάτω γράφος συνδυασμών.



Δηλαδή θα προκύψουν δυο ομάδες διαφορετικών τύπων με τη πρώτη να αποτελείται από τρεις κόμβους και τη δεύτερη μόνο από έναν.

Πρέπει να πούμε ότι η ίδια ακριβώς διαδικασία ισχύει και στην περίπτωση που οι κόμβοι n1, n2, n3 και n4 δεν είναι «τερματικοί» κόμβοι, αλλά ενδιάμεσοι κόμβοι-συνδέσεις ενός δένδρου συνδυασμών. Αυτό σημαίνει ότι έχουν από κάτω τους κόμβους-παιδιά.

3.4.2.4 Βάρος Κόμβων-Συνδέσεων[10]

Ένα λεπτό, καθοριστικής ωστόσο σημασίας, σημείο του αλγορίθμου ομαδοποίησης που σκοπίμως δεν αναφέραμε έως τώρα είναι ο υπολογισμός του βάρους των νέων κόμβων-συνδέσεων. Λέμε ότι είναι καθοριστικής σημασίας αφού στην ουσία καθορίζει την ποιότητα του αποτελέσματος της φάσης της ομαδοποίησης των κόμβων βάσει του περιεχομένου τους. Ο τρόπος υπολογισμού του βάρους αυτού μπορεί να οδηγήσει σε λογικά ή σε μη αποδεκτά αποτελέσματα.

Ας σκιαγραφήσουμε το πρόβλημα μέσω ενός παραδείγματος. Θεωρούμε ότι έχουμε δυο κόμβους τύπου εικόνα και κείμενο αντίστοιχα οι οποίοι συνδυάζονται για να δώσουν ένα νέο κόμβο-σύνδεση τύπου κείμενο. Αυτό σημαίνει ότι η εικόνα ήταν πολύ μικρή και απορροφήθηκε από το κείμενο. Τι βάρος θα δοθεί στον νέο κόμβο-σύνδεση; Αν του δώσουμε το άθροισμα τον αρχικών κόμβων τότε υπάρχει ο κίνδυνος να απορροφώνται συνέχεια εικόνες με αποτέλεσμα να έχουν βάρος υπολογίσιμο συγκριτικά με το κείμενο, παρόλα αυτά ο κόμβος-σύνδεση θα συνεχίζει να έχει τύπο κείμενο πράγμα που σαφώς δεν θα ήταν ορθό.

Από το προηγούμενο παράδειγμα γίνεται σαφής η επίδραση του τρόπου υπολογισμού του βάρους των νέων κόμβων-συνδέσεων στο τελικό αποτέλεσμα. Η λύση στην οποία καταφύγαμε είναι η εξής. Όταν συνδυάζονται κόμβοι του ίδιου τύπου το βάρος του νέου κόμβου-σύνδεση θα ισούται με το άθροισμα των βαρών των κόμβων αυτών. Αν οι κόμβοι που συνδυάζονται είναι διαφορετικού τύπου και ο νέος κόμβος-σύνδεση έχει τύπο ίδιο με έναν από τους αρχικούς, τότε το βάρος του τίθεται ίσο με το βάρος του αντίστοιχου κόμβου

[10] Αναφέρουμε ως κόμβους-συνδέσεις εκείνους τους κόμβους του γράφου συνδυασμών που δημιουργούνται από την ομαδοποίηση δυο άλλων κόμβων.

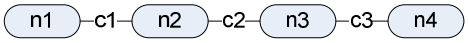
από τον οποίο κληρονομεί τον τύπο, το οποίο προφανώς είναι και το μεγαλύτερο από τα δυο βάρη. Τέλος, αν μετά τον συνδυασμό προκύψει νέος τύπος ο νέος κόμβος-σύνδεση θα αποκτήσει βάρος ίσο με το άθροισμα των βαρών των δυο αρχικών κόμβων.

Πρέπει να τονίσουμε εδώ ότι η προηγούμενη ανάλυση αναφέρεται στους κόμβους-συνδέσεις του γράφου συνδυασμών και όχι στους νέους κόμβους του DOM Tree που πιθανότατα θα προκύψουν. Οι κόμβοι-συνδέσεις παύουν να υπάρχουν μετά την εκτέλεση του αλγορίθμου ομαδοποίησης, αντίθετα οι άλλοι προσδένονται στο DOM Tree και παραμένουν για πάντα. Οι νέοι αυτοί κόμβοι του DOM Tree έχουν πάντα βάρος ίσο με το άθροισμα των βαρών των κόμβων-παιδιών τους.

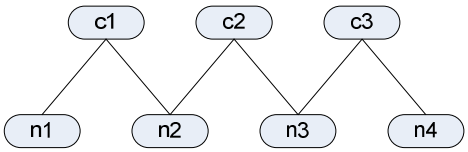
Είναι πολύ πιθανό όμως ένας κόμβος του DOM Tree να συμμετέχει σε δυο συνδυασμούς βάσει του αλγορίθμου ομαδοποίησης που σχεδιάσαμε. Ποιος συνδυασμός τελικά θα επικρατήσει και σε ποιόν θα ενσωματωθεί ο κοινός κόμβος; Αυτό αποτελεί αντικείμενο ανάλυσης της επόμενης υπό-ενότητας.

3.4.2.5 Προτεραιότητα Τύπων

Θα ξεκινήσουμε την περιγραφή της προτεραιότητας (ή βαρύτητας) των τύπων παρουσιάζοντας το πρόβλημα που καλείται η έννοια αυτή να επιλύσει. Για το λόγο αυτό θα χρησιμοποιήσουμε το θεωρητικό παράδειγμα της Ενότητας 3.3.2.3 , όπου πλέον θεωρούμε πως πραγματοποιούνται μόνο οι συνδυασμοί του πρώτου επιπέδου, τους οποίους και παρουσιάζουμε ξανά χάριν ευκολίας.



Επομένως ο γράφος συνδυασμών είναι ο ακόλουθος:



Παρατηρούμε πως οι κόμβοι n2 και n3 συμμετέχουν έκαστος σε δυο συνδυασμούς. Σε ποιόν θα ενσωματωθούν τελικά;

Η προσέγγιση που υιοθετούμε είναι της στατικής απόδοσης προτεραιότητας στους διάφορους τύπους κόμβων που μπορούμε να έχουμε. Ο καθορισμός του βαθμού βαρύτητας γίνεται από το χρήστη με βάση το είδος της ανάλυσης που πραγματοποιεί στην ιστοσελίδα και δεν μπορεί

να αλλάξει από το πρόγραμμα[11]. Στην περίπτωσή μας θέσαμε για κάθε τύπο την προτεραιότητα που φαίνεται στον Πίνακα 7.

Τύπος	Προτεραιότητα
br	0
εικόνα	1
φόρμα	1
Ενσωματωμένο αντικείμενο	1
κείμενο	2
σύνδεσμος-εικόνα	3
σύνδεσμος-κείμενο	4
υπερκείμενο	5
σύνθετο	6
διαχωριστής	6

Πίνακας 7: Προτεραιότητα Τύπων

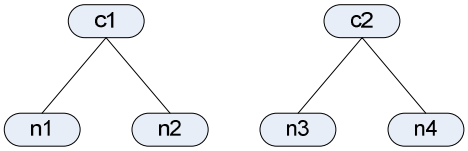
Όπως έχουμε αναφέρει και σε άλλο σημείο, σημαντικότερα για εμάς είναι τα στοιχεία κειμένου, ενώ δεν μας ενδιαφέρουν οι εικόνες και τα ενσωματωμένα αντικείμενα. Με βάση την παρατήρηση αυτή έγινε και η απόδοσης βαθμού προτεραιότητας στους διάφορους τύπους. Έτσι στους τύπους *εικόνα*, *φόρμα* και *ενσωματωμένο αντικείμενο* αποδόθηκαν οι μικρότερες προτεραιότητες, ενώ στους διάφορους τύπους κειμένου αποδόθηκαν μεγαλύτερες. Αυτό πρακτικά σημαίνει πως αν μια εικόνα συμμετέχει σε δυο συνδυασμούς, έναν συνδυασμό κειμένου και έναν συνδυασμό εικόνας, θα ενταχθεί τελικά στον συνδυασμό κειμένου. Ο βαθμός προτεραιότητας του τύπου *διαχωριστής* δεν έχει ιδιαίτερη σημασία και θα μπορούσε να έχει οποιαδήποτε τιμή αφού ο εν λόγω τύπος δεν συνδυάζεται με κανέναν άλλο. Αντίθετα η προτεραιότητα του τύπου *σύνθετος* είναι καθοριστικής σημασίας κάτι το οποίο θα παρουσιάσουμε λεπτομερώς αργότερα. Τέλος, ο τύπος br έχει το μικρότερο βαθμό προτεραιότητας αφού απορροφάται από όλους τους υπόλοιπους τύπους.

[11] Μια εναλλακτική προσέγγιση, πιο δυναμική και ελεγχόμενη από το πρόγραμμα, θα μπορούσε να ελέγξει τη συνεκτικότητα των κόμβων χρησιμοποιώντας κριτήρια μορφοποίησης. Σίγουρα όμως η προσέγγιση αυτή απαιτεί μεγαλύτερη πολυπλοκότητα.

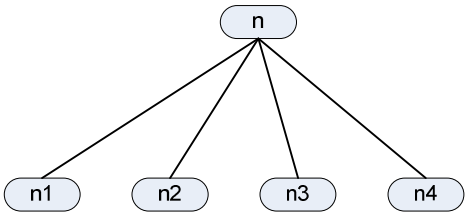
3.4.2.6 Οριστικοποίηση Ομαδοποίησης

Μέχρι στιγμής, η διαδικασία ομαδοποίησης κόμβων διαφορετικού τύπου έχει φτάσει στο σημείο δημιουργίας του γράφου συνδυασμών. Καμία τροποποίηση ή ενημέρωση ωστόσο δεν έχει γίνει στους κόμβους του DOM Tree όπως αυτό διαμορφώθηκε μετά τη φάση ομαδοποίησης ομοειδών κόμβων. Στην παρούσα φάση θα εφαρμοστούν όλες οι αλλαγές στο DOM Tree, σύμφωνα πάντα με τους κόμβους του γράφου συνδυασμών. Η όλη διαδικασία θα παρουσιαστεί με παραδείγματα.

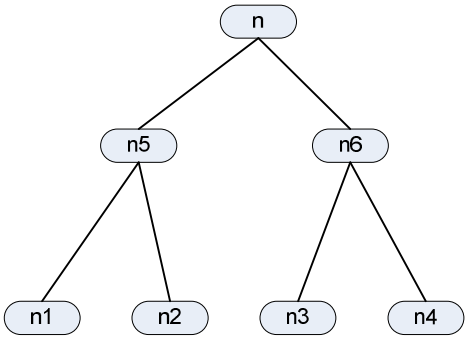
Αρχικά θεωρούμε την περίπτωση εκτέλεσης του αλγορίθμου ομαδοποίησης που φαίνεται στην Ενότητα 3.3.2.3 την οποία παραθέτουμε ξανά χάριν ευκολίας.



Παραθέτουμε επίσης και την αρχική εικόνα των κόμβων που εξετάζονται, μαζί με τον κόμβο-πατέρα αυτή τη φορά.

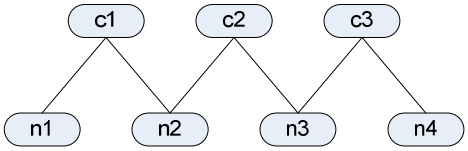


Σύμφωνα με το γράφο συνδυασμών, οι τέσσερις κόμβοι-παιδιά θα δημιουργήσουν δυο ομάδες. Για το λόγο αυτό δημιουργούμε δυο νέους κόμβους στους οποίους προσαρτούμε τους αντίστοιχους κόμβους-παιδιά αφού προηγουμένως τα έχουμε διαγράψει από τον αρχικό κόμβο-πατέρα. Το τελικό αποτέλεσμα φαίνεται στην επόμενη εικόνα.

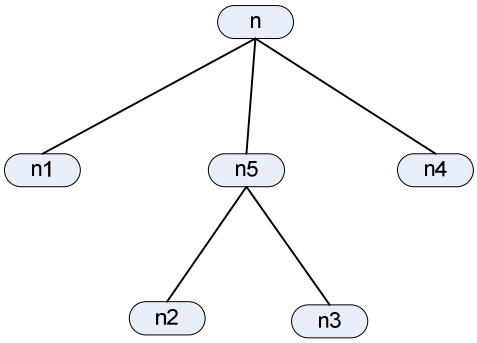


Παρατηρούμε πως το μέγεθος και το βάθος του DOM Tree έχουν αυξηθεί. Αυτό αποτελεί μια αρνητική συνέπεια της διαδικασίας ομαδοποίησης την οποία θα δούμε να εμφανίζεται ξανά στην διαδικασία της τοπολογικής ομαδοποίησης κόμβων στο Κεφάλαιο 4.

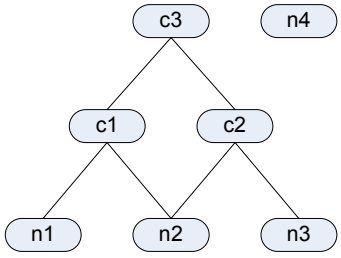
Θέλοντας να παρουσιάσουμε τη χρησιμότητα της έννοιας της προτεραιότητας τύπων, συνεχίζουμε το παράδειγμα που δείξαμε στην αντίστοιχη ενότητα. Το προκύπτον δένδρο συνδυασμών θυμίζουμε ότι είναι αυτό που φαίνεται στην επόμενη εικόνα.



Αν οι νέοι κόμβοι-συνδέσεις διατάσσονται κατά σειρά προτεραιότητας τύπων ως εξής c1, c3 και c2 τότε το αποτέλεσμα είναι το ίδιο με την προηγούμενη περίπτωση. Το ίδιο αποτέλεσμα θα έχουμε και στην περίπτωση που η σειρά διάταξης είναι η ακόλουθη: c1, c2 ,c3. Αυτό ισχύει επειδή ο κόμβος c2 στην ουσία παύει να υπάρχει μετά την ομαδοποίηση των κόμβων-παιδιών του κόμβου c1. Αντίθετα, αν ο κόμβος-σύνδεση c2 έχει τη μεγαλύτερη προτεραιότητα τότε θα έχουμε τρείς ομάδες όπως δείχνει και η επόμενη εικόνα.

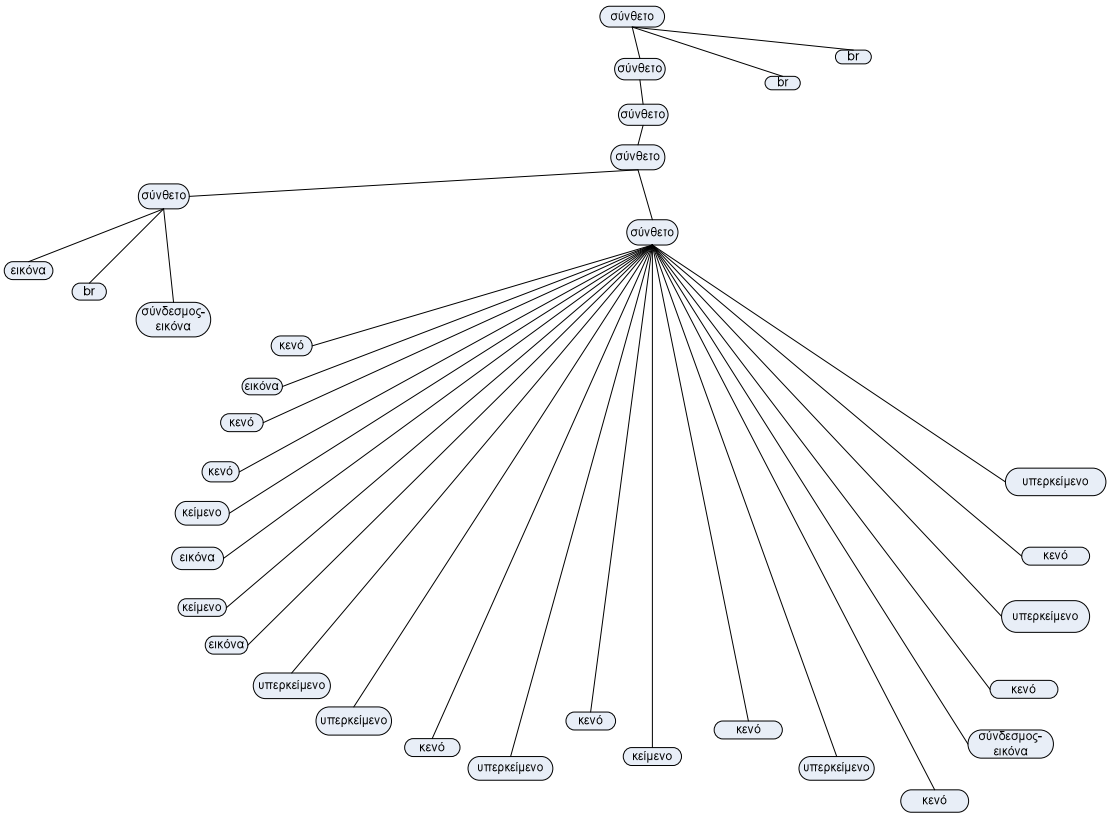


Θα αναφερθούμε τώρα σε μια ειδική περίπτωση, τη τελευταία της Ενότητας 3.3.2.3, της οποίας ο γράφος συνδυασμών είναι ο κάτωθι.



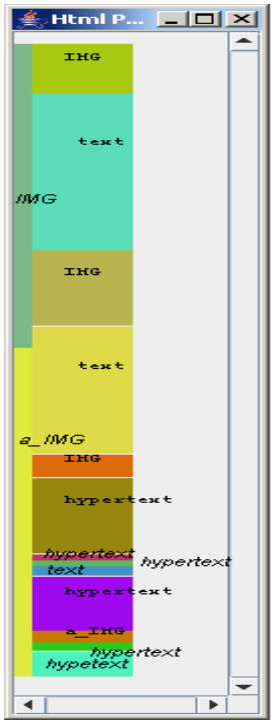
Όπως θυμόμαστε, οι κόμβοι n1, n2, n3 και n4 δεν είναι υποχρεωτικά τερματικοί κόμβοι αλλά μπορεί να έχουν και παιδιά. Σε αυτή τη περίπτωση, λοιπόν, κατά τη φάση ολοκλήρωσης της ομαδοποίησης ελέγχονται οι κόμβοι c3 και n4. Επειδή όμως ο κόμβος n4 έχει προέλθει από συνδυασμό σε χαμηλότερο επίπεδο, δεν χρησιμοποιείται μόνο το κριτήριο της προτεραιότητας τύπου αλλά και το επίπεδο στο οποίο έχει δημιουργηθεί ένας κόμβος-σύνδεση και μάλιστα χρησιμοποιείται σαν το πρώτο κριτήριο. Κόμβοι που έχουν δημιουργηθεί σε υψηλότερο επίπεδο έχουν μεγαλύτερη προτεραιότητα από κόμβους χαμηλότερου επιπέδου. Η επιλογή μας αυτή δηλώνει ότι στόχος μας είναι η ευρύτερη δυνατή ομαδοποίηση κόμβων και όχι η στενότερη του χαμηλότερου επιπέδου που πιθανότατα να είχε και μεγαλύτερη συνάφεια περιεχομένου. Εξάλλου, το βαθμό συνάφειας μπορούμε να τον ελέγξουμε μέσω των παραμέτρων συνδυασμών.

Στο σημείο αυτό θα παρουσιάσουμε ένα πραγματικό παράδειγμα. Θα χρησιμοποιήσουμε το DOM Tree του *Παραδείγματος 1* όπως αυτό έχει διαμορφωθεί μετά τη φάση ομαδοποίησης ομοειδών κόμβων. Μετά την εφαρμογή του αλγορίθμου ομαδοποίησης κόμβων διαφορετικού τύπου το DOM Tree παίρνει τη μορφή που φαίνεται στην Εικόνα 20.



Εικόνα 20: DOM Tree Παραδείγματος 1 μετά από Ομαδοποίηση Κόμβων

Είναι εμφανής η μείωση τόσο του μεγέθους όσο και του βάθους του δένδρου εξαιτίας της διαδικασίας ομαδοποίησης. Η αντίστοιχη τελική τοπολογική διάταξη της σελίδας φαίνεται στην Εικόνα 21.



Εικόνα 21: Τοπολογική Διάταξη Παραδείγματος 1 μετά από Ομαδοποίηση Κόμβων

Σε αυτό το σημείο πρέπει να επισημάνουμε ότι όλα τα φύλλα του DOM Tree θεωρούνται πλέον αδιαίρετες μονάδες, χωρίς εσωτερική δομή ανεξάρτητα από το πλήθος των κόμβων που ομαδοποιήθηκαν σε αυτά. Το βάρος τους ισούται με το άθροισμα των βαρών των παιδιών τους, εκτός από τους κενούς κόμβους οι οποίοι έχουν μηδενικό βάρος γι' αυτό και στην Εικόνα 20 δεν φαίνονται καθόλου

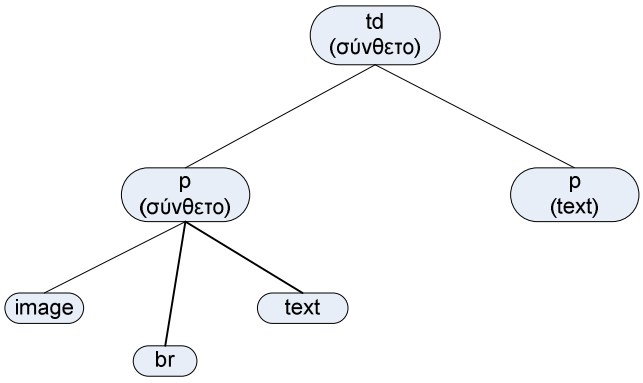
3.4.3 Ομαδοποίηση Γειτονικών Κόμβων

Η ομαδοποίηση κόμβων βάσει περιεχομένου που έχουμε εφαρμόσει μέχρι τώρα στηριζόταν αποκλειστικά στο γεγονός ότι οι κόμβοι ήταν γείτονες στη δομή του DOM Tree. Αυτό σημαίνει ότι είχαν κοινό κόμβο-πατέρα. Όπως έχουμε όμως αναφέρει, τμήματα της ιστοσελίδας που γειτονεύουν και που έχουν περιεχόμενο ίδιου ή ισοδύναμου τύπου μπορεί να απέχουν πολύ στο δένδρο. Την ομαδοποίηση κόμβων που εμπίπτουν σε αυτή τη δεύτερη κατηγορία μελετάμε εδώ.

Οι κόμβοι της πρώτης κατηγορίας ευνοούνται από τη δομή του δένδρου, πράγμα που σημαίνει πως όσοι τελικά ομαδοποιούνται μαζί μπορούν να ενσωματωθούν εύκολα σε έναν νέο κόμβο αφού τους έχουμε αφαιρέσει από τον αρχικό κόμβο-πατέρα τους. Αυτό το είδαμε στην Ενότητα 3.3.2.3 . Αντίθετα, η σχετική θέση στο δένδρο των κόμβων της δεύτερης κατηγορίας καθιστά αδύνατη μια τέτοια τροποποίηση του DOM Tree. Το γεγονός αυτό μας οδηγεί στην υιοθέτηση μιας εναλλακτικής λύσης όπου πλέον δεν αλλάζουμε καθόλου τη μορφή του δένδρου αλλά κατασκευάζουμε μια νέα δομή η οποία δρα επί του DOM Tree. Η

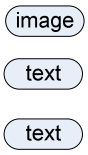
πρόσθετη δομή περιέχει την τελική εικόνα της ιστοσελίδας όπως αυτή έχει οργανωθεί σε τμήματα (ομάδες) με διαφορετικού τύπου περιεχόμενο.

Ας δούμε όμως το πρόβλημα αυτό μέσω ενός παραδείγματος. Παραθέτουμε τμήμα ενός DOM Tree που έχει τα προαναφερθέντα ιδιαίτερα χαρακτηριστικά.



Εικόνα 22: Ομαδοποίηση Γειτονικών Κόμβων

Ο αλγόριθμος της Ενότητας 3.3.2.3 δεν θα είχε καμία επίδραση στο προηγούμενο δένδρο και το αποτέλεσμα της ομαδοποίησης θα ήταν το ακόλουθο.



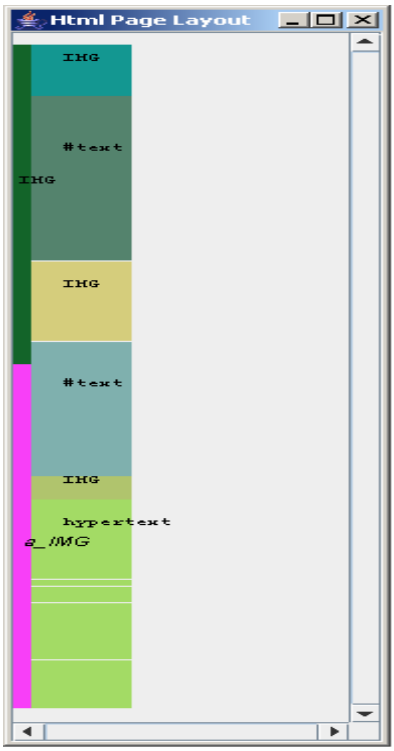
Εικόνα 23: Επίδραση Αλγορίθμου Ομαδοποίησης Ενότητας 3.4.2.3

Είναι πλέον εμφανές το πρόβλημα που υπάρχει. Για το λόγο αυτό προσθέτουμε μια επιπλέον φάση ομαδοποίησης η οποία και θα δώσει το επόμενο σωστό αποτέλεσμα.

Εικόνα 24: Ορθό Αποτέλεσμα Ομαδοποίησης

Όσον αφορά τον αλγόριθμο ομαδοποίησης, αυτός έχει την ίδια λογική με τον αλγόριθμο που χρησιμοποιήσαμε στην ενότητα 3.4.2.3,ελέγχει δηλαδή όλους τους δυνατούς συνδυασμούς και χρησιμοποιεί τη δομή του γράφου συνδυασμών. Διαφέρει ως προς το γεγονός ότι εδώ πρέπει να ελεγχθούν δυο διαστάσεις και όχι μόνο μια. Αυτό γίνεται αφού ένας κόμβος έχει γείτονες τόσο στην κατακόρυφη όσο και στην οριζόντια διεύθυνση. Προφανώς ο κόμβος θα ενταχθεί τελικά μόνο σε μια ομάδα σύμφωνα πάντα με τη προτεραιότητα τύπων που έχουμε αναλύσει στην Ενότητα 3.4.2.5.

Κλείνουμε την ενότητα παρουσιάζοντας το τελικό αποτέλεσμα για το Παράδειγμα 1 όπως αυτό διαμορφώθηκε μετά την εφαρμογής και της νέας φάσης ομαδοποίησης.



Εικόνα 25: Τελική Τοπολογική Διάταξη Παραδείγματος 1

Αξίζει να το συγκρίνουμε με την Εικόνα 21 στην οποία δεν έχει εφαρμοστεί η διαδικασία ομαδοποίησης γειτονικών κόμβων. Παρατηρούμε ότι ο βαθμός ομαδοποίησης είναι σαφώς μεγαλύτερος στην δεύτερη περίπτωση.

3.5 Σύνοψη

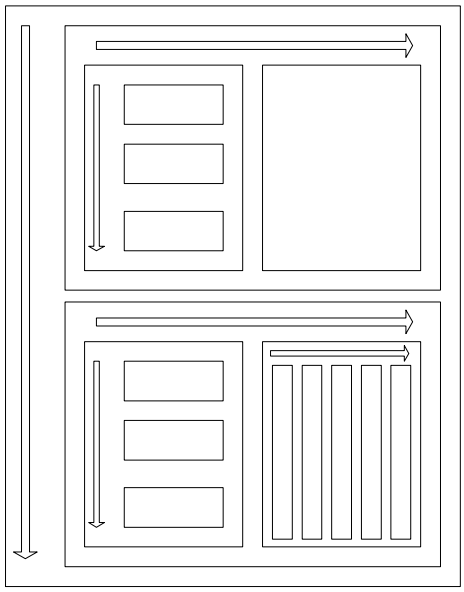
Στο κεφάλαιο αυτό μελετήθηκε η ομαδοποίηση των κόμβων του DOM Tree βάσει του περιεχομένου τους. Η διαδικασία της ομαδοποίησης είδαμε πως χωρίζεται σε τρεις φάσεις. Στην πρώτη πραγματοποιείται ομαδοποίηση ομοειδών κόμβων, ενώ κατά τη δεύτερη φάση ελέγχονται οι συνδυασμοί μεταξύ κόμβων διαφορετικού τύπου περιεχομένου. Στη τρίτη φάση επιχειρείται ομαδοποίηση των γειτονικών κόμβων χρησιμοποιώντας τους δείκτες γειτνίασης. Η δεύτερη και η τρίτη φάση εκ των πραγμάτων δεν μπορεί να είναι ντετερμινιστικές για αυτό και ελέγχονται μέσω παραμέτρων, ενώ χρησιμοποιούν και παρεμφερείς αλγορίθμους. Οι παράμετροι αυτές επιτρέπουν στον χρήστη να καθορίσει το επίπεδο λεπτομέρειας που επιθυμεί.

4 Τοπολογική Διάταξη Ιστοσελίδων

Στο κεφάλαιο αυτό θα περιγράψουμε τις διαδικασίες που είναι υπεύθυνες για τη δημιουργία της τοπολογικής διάταξης μιας ιστοσελίδας. Θα παρουσιάσουμε τους αλγόριθμους που αναπτύξαμε καθώς και παραδείγματα εκτέλεσής τους.

4.1 Τοπολογική Ομαδοποίηση Κόμβων

Στο DOM Tree, όπως αυτό προκύπτει από τη φάση εκκαθάρισης, μπορεί να υπάρχουν κόμβοι των οποίων τα παιδιά-κόμβοι δεν έχουν όλα τον ίδιο προσανατολισμό. Υπάρχουν με άλλα λόγια τόσο κόμβοι με οριζόντιο προσανατολισμό όσο και κόμβοι με κατακόρυφο προσανατολισμό. Το γεγονός αυτό καθιστά περίπλοκες όλες εκείνες τις διαδικασίες που στόχο έχουν τη δημιουργία της τελικής τοπολογικής διάταξης της ιστοσελίδας και τη τοποθέτηση των κόμβων στη σωστή τους θέση. Για να έχουμε, λοιπόν, πιο απλές αλλά και γενικότερης εφαρμογής διαδικασίες, καταφεύγουμε στην ομαδοποίηση των κόμβων βάσει του προσανατολισμού τους. Στόχος είναι να προκύψουν κόμβοι που έχουν παιδιά-κόμβους μόνο με ένα είδος προσανατολισμού: είτε οριζόντιο είτε κατακόρυφο. Αυτό με τη σειρά του σημαίνει πως η δόμηση της σελίδας θα γίνεται αναδρομικά και μόνο προς τη μια κατεύθυνση. Το σχήμα της Εικόνας 1 δείχνει τη διαδικασία δόμησης της σελίδας από τα συστατικά του. Η δόμηση ξεκινάει από τα απλά στοιχεία τα οποία συγκροτούν σύνθετα στοιχεία και αυτά με τη σειρά τους ακόμη μεγαλύτερα κοκ μέχρι τη σύνθεση ολόκληρης της σελίδας.



Εικόνα 1: Σύνθεση σελίδας από τα συστατικά της

4.1.1 Αλγόριθμος Τοπολογικής Ομαδοποίησης

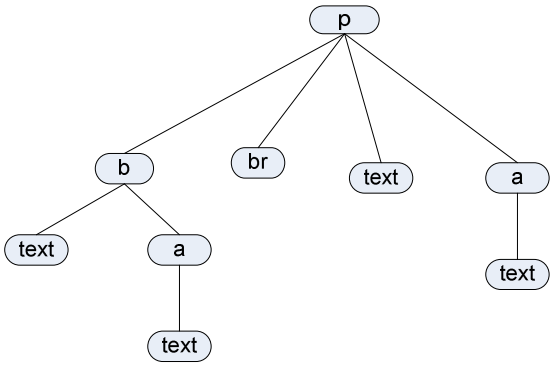
Η ιδέα είναι απλή και εφαρμόζεται αναδρομικά για όλους τους κόμβους του DOM Tree. Εξετάζονται όλοι οι κόμβοι-παιδιά ενός συγκεκριμένου κόμβου. Διαδοχικοί, λοιπόν, κόμβοι με τον ίδιο προσανατολισμό ομαδοποιούνται μαζί. Στη συνέχεια δημιουργείται ένας νέος κόμβος στον οποίον προσαρτούμε τους κόμβους που ομαδοποιήθηκαν μαζί ενώ ταυτόχρονα τους διαγράφουμε από τον αρχικό πατέρα-κόμβο. Ο νέος κόμβος προσαρτάται στον αρχικό πατέρα-κόμβο αφού πρώτα του έχουμε αποδώσει τον κατάλληλο προσανατολισμό.

Η διαδικασία αυτή αν και συμβάλλει στην απλοποίηση μετέπειτα διαδικασιών έχει, ωστόσο, και ένα αρνητικό αποτέλεσμα: αυξάνει το μέγεθος και το βάθος του DOM Tree.

Στην επόμενη υπό-ενότητα επεξηγείται η περιγραφείσα διαδικασία μέσω ενός παραδείγματος.

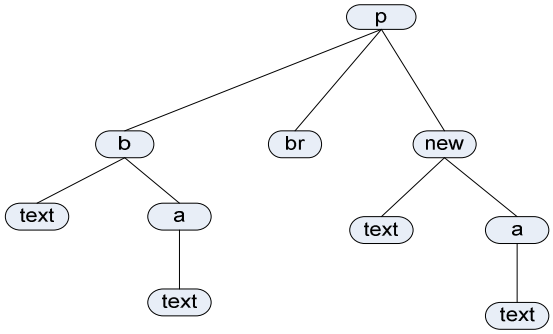
4.1.2 Παράδειγμα Τοπολογικής Ομαδοποίησης Κόμβων

Θα χρησιμοποιήσουμε το DOM Tree του *Παραδείγματος 1*. Στην Εικόνα 2 παρουσιάζεται μόνο εκείνο το τμήμα του DOM Tree που θα τροποποιηθεί από τη διαδικασία ομαδοποίησης. Επιπλέον, για να φανεί αυτή ακριβώς η ομαδοποίηση εμφανίζονται και οι κόμβοι τύπου *text*.



Εικόνα 2: Τμήμα του DOM Tree του Παραδείγματος 1

Είναι προφανές ότι ο κόμβος p έχει παιδιά με διαφορετικούς προσανατολισμούς. Οι κόμβοι-παιδιά *b, text* και *a* έχουν οριζόντιο προσανατολισμό, ενώ ο κόμβος *br* έχει κατακόρυφο. Μετά την εφαρμογή της διαδικασίας τοπολογικής ομαδοποίησης, ο κόμβος *p* έχει την μορφή που φαίνεται στην Εικόνα 3.



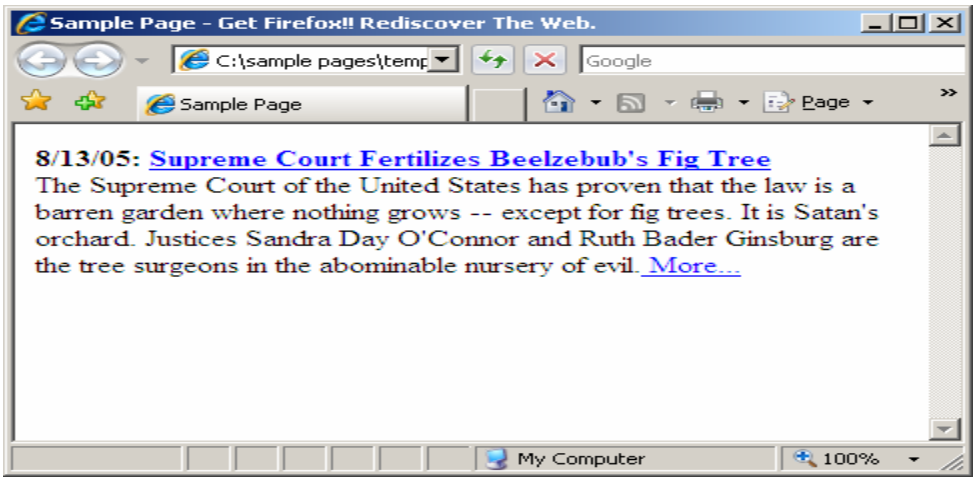
Εικόνα 3: Διαμόρφωση μετά από Τοπολογική Ομαδοποίηση

Ο νέος κόμβος *new* έχει κατακόρυφο προσανατολισμό γεγονός που σημαίνει πως πλέον όλοι οι κόμβοι-παιδιά του *p* έχουν τον ίδιο προσανατολισμό. Η τοπολογική τους διάταξη φαίνεται στην Εικόνα 4, χωρίς να αναλύεται η εσωτερική τους δομή.

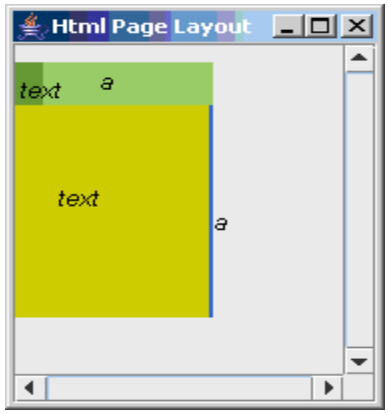
Εικόνα 4: Τοπολογική Διάταξη μετά από Τοπολογική Ομαδοποίηση

Εδώ πρέπει να αναφέρουμε, για να αποφύγουμε τυχόν παρερμηνείες που δημιουργεί η Εικόνα 3 , ότι διαδοχικοί κόμβοι που εναλλάσσουν προσανατολισμό δεν δημιουργούν πρόβλημα. Πρόβλημα δημιουργείται όταν ένας κόμβος έχει κόμβους-παιδιά με διαφορετικό προσανατολισμό και υπάρχουν περισσότεροι του ενός διαδοχικοί κόμβοι με τον ίδιο προσανατολισμό.

Το τμήμα της ιστοσελίδας που αντιστοιχεί στον κόμβο p φαίνεται στην Εικόνα 5 και η αντίστοιχη τοπολογική διάταξη παρουσιάζεται στην Εικόνα 6.



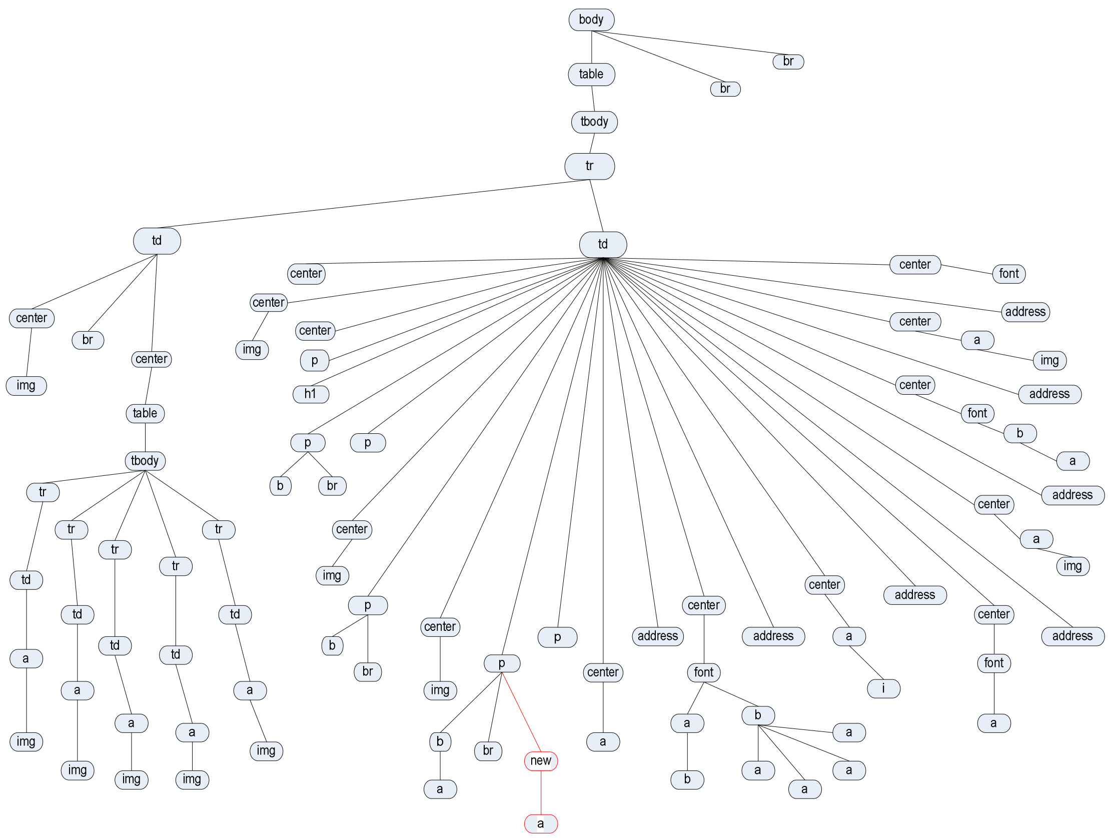
Εικόνα 5: Παράδειγμα Τοπολογικής Ομαδοποίησης



Εικόνα 6: Τοπολογική Διάταξη για τη Σελίδα της Εικόνας 4

Στην Εικόνα 5 φαίνεται και η σχέση μεγέθους των διάφορων κόμβων το οποίο εκφράζεται μέσω του βάρους τους όπως είδαμε στην Ενότητα 3.3.

Τέλος παραθέτουμε, χάριν κατανόησης, το συνολικό DOM Tree του *Παραδείγματος 1* όπως αυτό διαμορφώθηκε μετά τη διαδικασία της τοπολογικής ομαδοποίησης κόμβων. Οι κόμβοι που επηρεάστηκαν έχουν σχεδιαστεί με κόκκινες γραμμές.



Εικόνα 7: Τροποποίηση DOM Tree λόγω Τοπολογικής Ομαδοποίησης

4.2 Τοπολογία Ιστοσελίδας

Μέχρι τώρα έχουμε ομαδοποιήσει τους κόμβους του DOM Tree με κριτήρια αρχικά τον τύπο του περιεχομένου τους και στη συνέχεια χωρικό προσανατολισμό τους. Δεν έχουμε, ωστόσο, ασχοληθεί καθόλου με την σχετική χωρική διάταξη των κόμβων. Δεν έχουμε δηλαδή εξετάσει το θέμα της γειτνίασης, με ποιους κόμβους γειτονεύει ένας κόμβος. Αυτό θα είναι το θέμα της παρούσας ενότητας.

Όλοι οι κόμβοι του DOM Tree αντιστοιχούν σε κάποιο τμήμα της ιστοσελίδας. Η ρίζα αποτελεί ολόκληρη την ιστοσελίδα ενώ τα φύλλα τα δομικά της στοιχεία, αυτά που βλέπει τελικά ο χρήστης. Έτσι ενώ οι τερματικοί κόμβοι (φύλλα) είναι ορατοί μέσω του περιεχομένου τους (κείμενο, εικόνα κ.ά.), οι κόμβοι υψηλότερου επιπέδου συμβάλλουν στη χωρική διαμόρφωση της ιστοσελίδας. Είτε όμως είναι ενδιάμεσος είτε είναι τερματικός, κάθε κόμβος του DOM Tree γειτονεύει με άλλους κόμβους. Αυτή η παρατήρηση καθορίζει και τη διαδικασία που θα χρησιμοποιήσουμε για την εύρεση της χωρικής διάταξης των κόμβων και επομένως και των γειτόνων τους.

Το πώς δυο κόμβοι σχετίζονται χωρικά (ή δεν σχετίζονται καθόλου) εξαρτάται από τη σχετική τους θέση στο DOM Tree καθώς και από τον τύπο τους. Στην παρούσα ενότητα,

όταν αναφερόμαστε στον τύπο ενός κόμβου εννοούμε τον τύπο του αντίστοιχου Html tag από το οποίο έχει προέλθει. Θα ξεκινήσουμε με το δεύτερο κριτήριο το οποίο και αναλύουμε στην επόμενη υπό-ενότητα.

4.2.1 Χωρικός Προσανατολισμός Κόμβων

Η έννοια του χωρικού προσανατολισμού ενός κόμβου σχετίζεται με το πώς ο κόμβος αυτός θα διαταχθεί σχετικά με κάποιον άλλον γειτνιάζοντα κόμβο. Συγκεκριμένα, αν έχουμε δυο κόμβους που γειτονεύουν αυτοί μπορούν να διαταχθούν είτε οριζόντια, ο ένας αριστερά και ο άλλος δεξιά, είτε κατακόρυφα, ο ένας πάνω και ο άλλος κάτω. Το ποιος από τους δυο προσανατολισμούς θα ισχύσει εξαρτάται από τον τύπο του Html tag τον οποίον ο κόμβος αντιπροσωπεύει. Στον Πίνακα 1 καταγράφονται με αλφαβητική σειρά οι διάφοροι τύποι των tags καθώς και ο προσανατολισμός τους.

Html tag	**Προσανατολισμός**
<A -	Οριζόντιος
<ADDRESS>	Κατακόρυφος
<B>	Οριζόντιος
<BIG>	Οριζόντιος
 	Κατακόρυφος
<CAPTION>	Κατακόρυφος
<CENTER>	Κατακόρυφος
<COL>	Κατακόρυφος
<COLGROUP>	Κατακόρυφος
<DD>	Κατακόρυφος
<DIR>	Κατακόρυφος
<DL>	Κατακόρυφος
<DT>	Κατακόρυφος
<EM>	Οριζόντιος
<EMBED>	Οριζόντιος
<FIELDSET>	Κατακόρυφος
<FONT>	Οριζόντιος
<FORM>	Κατακόρυφος

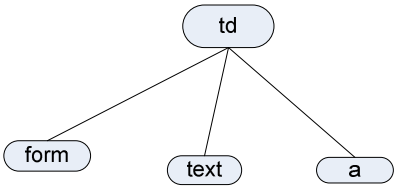
<H1>	Κατακόρυφος
<H2>	Κατακόρυφος
<H3>	Κατακόρυφος
<H4>	Κατακόρυφος
<H5>	Κατακόρυφος
<H6>	Κατακόρυφος
<HR>	Κατακόρυφος
<I>	Οριζόντιος
<IMG>	Οριζόντιος
<INPUT>	Οριζόντιος
<LEGEND>	Κατακόρυφος
<LI>	Κατακόρυφος
<LINK>	Οριζόντιος
<MARQUEE>	Κατακόρυφος
<MENU>	Κατακόρυφος
<OL>	Κατακόρυφος
<P>	Κατακόρυφος
<PRE>	Κατακόρυφος
<SMALL>	Οριζόντιος
<STRONG>	Οριζόντιος
<TABLE>	Κατακόρυφος
<TD>	Οριζόντιος
<TFOOT>	Κατακόρυφος
<TH>	Κατακόρυφος
<THEAD>	Κατακόρυφος
<TR>	Κατακόρυφος
<TT>	Οριζόντιος
<U>	Οριζόντιος
<UL>	Κατακόρυφος
<DIV>	Κατακόρυφος
<SPAN>	Οριζόντιος

TEXT	Οριζόντιος

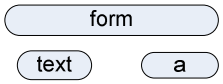
Πίνακας 1: Προσανατολισμός HTML tag

Πρέπει να διευκρινίσουμε ότι δεν είναι μόνο αυτοί οι τύποι των tags που μπορεί να υπάρξουν σε ένα Html αρχείο, μόνο αυτούς μπορούμε ωστόσο να βρούμε στο σώμα μιας ιστοσελίδας μετά και τη διαδικασία καθαρισμού που έχει εφαρμοστεί στο DOM Tree. Επίσης, δεν υπάρχει tag τύπου *TEXT* υπάρχει όμως τέτοιος κόμβος για αυτό και το παρουσιάζουμε.

Ενώ ο προσανατολισμός του κάθε κόμβου, είτε μόνου του είτε σε συνδυασμό με κόμβο του ίδιου προσανατολισμού, είναι σαφής, αξίζει να αναφέρουμε ότι συνδυασμός κόμβων διαφορετικών προσανατολισμών μας δίνει πάντα κατακόρυφο προσανατολισμό. Αυτό φαίνεται και από το επόμενο τυχαίο παράδειγμα. Το DOM Tree φαίνεται στην Εικόνα 8 και η αντίστοιχη χωρική διάταξη στην Εικόνα 9.



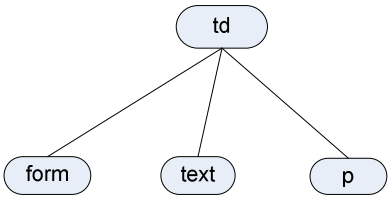
Εικόνα 8: Συνδυασμός Στοιχείων Διαφορετικού Προσανατολισμού



Εικόνα 9: Τοπολογική Διάταξη Στοιχείων Διαφορετικού Προσανατολισμού

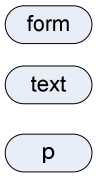
4.2.2 Γειτονικοί Κόμβοι

Τώρα πλέον γνωρίζουμε τον προσανατολισμό του κάθε κόμβου σχετικά με τους γείτονές του, δεν γνωρίζουμε όμως πότε δυο κόμβοι γειτονεύουν οπτικά σαν τμήματα της ίδιας ιστοσελίδας. Όπως αφήνει να υπονοείται το παράδειγμα της προηγούμενης υπό-ενότητας, κόμβοι που έχουν κοινό κόμβο-πατέρα γειτονεύουν μεταξύ τους. Πράγματι, αυτό ισχύει ο ακριβής όμως τρόπος γειτνίασης εξαρτάται από τον τύπο προσανατολισμού τους. Ας το δούμε αυτό αντιπαραβάλλοντας ένα άλλο παράδειγμα με την ίδια μορφή του DOM Tree όπως το προαναφερθέν παράδειγμα αλλά με διαφορετικού τύπου κόμβους όπως δείχνει και η Εικόνα 10.



Εικόνα 10: Ο τύπος των Κόμβων καθορίζει τον τρόπο γειτνίασης

Η αντίστοιχη χωρική διάταξη φαίνεται ακολούθως. Αξίζει να παρατηρήσουμε πως η συνολική εικόνα της 'γειτονιάς' έχει αλλάξει.



Εικόνα 11: Κατακόρυφος Προσανατολισμός

Όπως φαίνεται και από τα δυο παραδείγματα, ο προσδιορισμός των γειτόνων δεν αναφέρεται μόνο σε απλούς κόμβους, αυτούς δηλαδή που έχουν περιεχόμενο ενός τύπου, αλλά και σε σύνθετους, οι οποίοι έχουν επιπλέον εσωτερική δομή. Αυτό όμως θα το αναλύσουμε περαιτέρω αργότερα.

Από την Ενότητα 4.1 ξέρουμε ότι όλοι οι κόμβοι-παιδιά ενός κόμβου διατάσσονται μόνο προς μια κατεύθυνση: είτε οριζόντια είτε κατακόρυφα. Προφανώς το παράδειγμα της προηγούμενης υπό-ενότητας δεν υπακούει σε αυτή την αρχή. Αν λάβουμε υπόψη και το προηγούμενο συμπέρασμα, πως δηλαδή κόμβοι με κοινό κόμβο-πατέρα γειτονεύουν, καταλήγουμε στη διαπίστωση πως, στην καλύτερη περίπτωση, έχουμε βρει τους γείτονες ενός κόμβου μόνο προς τη μια κατεύθυνση. Λέμε στην καλύτερη περίπτωση γιατί υπάρχει το ενδεχόμενο ενός μοναδικού κόμβου-παιδιού οπότε δεν γνωρίζουμε κανένα γείτονα. Αυτό σημαίνει πως οι άλλοι γείτονες του κόμβου βρίσκονται κάπου 'αλλού'. Η αδυναμία να βρούμε όλους τους γείτονες ενός κόμβου ταυτόχρονα μας οδηγεί στην υιοθέτηση μιας διαδικασίας δυο φάσεων.

Κατά την πρώτη φάση, λοιπόν, καθορίζονται οι κόμβοι-γείτονες για έναν κόμβο μόνο προς τη μια κατεύθυνση. Αυτό γίνεται λαμβάνοντας υπόψη μόνο τους κόμβους-αδέρφια, τους κόμβους δηλαδή με τους οποίους ο υπό εξέταση κόμβος έχει κοινό κόμβο-πατέρα. Η διαδικασία αυτή γίνεται αναδρομικά για όλους τους κόμβους, ξεκινώντας από τα φύλλα. Σε κάθε βήμα του αλγορίθμου, οι κόμβοι ενός επιπέδου του DOM Tree μαθαίνουν τους γείτονές τους προς τη μια κατεύθυνση και όλοι, είτε είναι απλοί είτε είναι σύνθετοι, αντιμετωπίζονται σαν απλοί. Στην περίπτωση που ένας κόμβος είναι σύνθετος, η εσωτερική του δομή έχει διευθετηθεί στο προηγούμενο βήμα του αλγορίθμου.

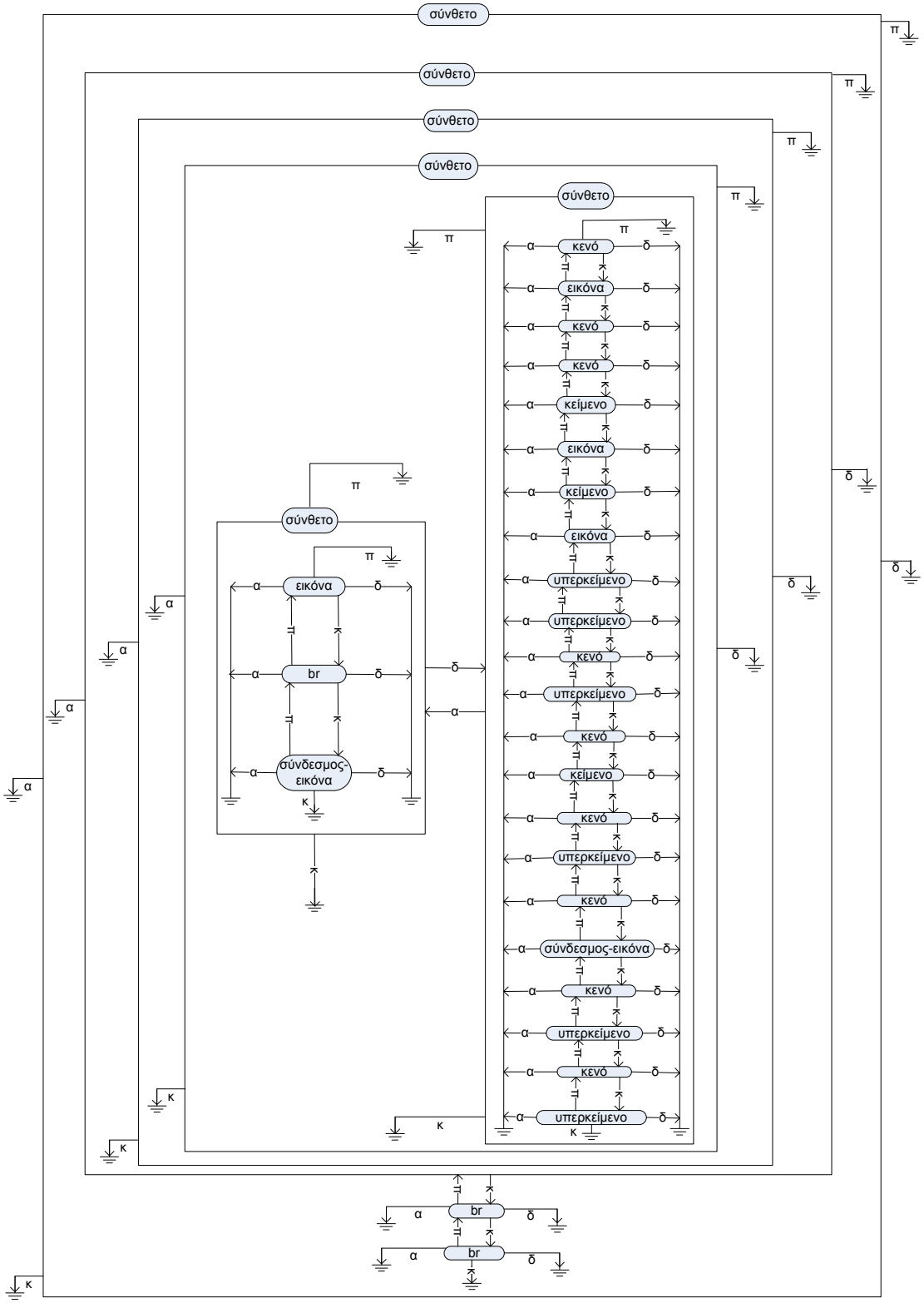
Παραθέτουμε, χάριν κατανόησης, μια σκιαγράφηση του αλγορίθμου σε ψευδοκώδικα.

```
καθορισμός_Γειτονικών_Κόμβων(Κόμβος n) {

        Εάν (ο κόμβος n είναι σύνθετος) {
              Πάρε όλα τα παιδιά του κόμβου n;
              Για κάθε (παιδί του κόμβου n)
                  Καθορισμός_Γειτονικών_Κόμβων(επόμενο παιδί του n)
       }

       εύρεση_Γειτόνων(n);
}

Εύρεση_Γειτόνων( Κόμβος n)
{
          Βρες προσανατολισμό;
          Πάρε τον αριστερό κόμβο;
          Εάν( προσανατολισμός είναι οριζόντιος )
                Βάλε τον αριστερό κόμβο σαν αριστερό γείτονα;
          Αλλιώς
                Βάλε τον αριστερό κόμβο σαν γείτονα από πάνω;
          Πάρε τον δεξιό κόμβο;
          Εάν( προσανατολισμός είναι οριζόντιος )
               Βάλε τον δεξιό κόμβο σαν δεξιό γείτονα;
          Αλλιώς
               Βάλε τον δεξιό κόμβο σαν γείτονα από κάτω;

}
```

Αλγόριθμος εύρεσης γειτονικών κόμβων

Η εφαρμογή του παραπάνω αλγορίθμου στο DOM Tree του Παραδείγματος 1 όπως αυτό έχει διαμορφωθεί από τις διαδικασίες ομαδοποίησης των κόμβων δίνει το επόμενο αποτέλεσμα. Στην εικόνα διακρίνονται τόσο οι απλοί όσο και οι σύνθετοι κόμβοι με τους αντίστοιχους δείκτες γειτνίασης (π = πάνω, κ = κάτω, α = αριστερά, δ = δεξιά). Είναι εμφανές από τους δείκτες ότι η διαδικασία δεν είναι πλήρης. Η σχετική θέση των απλών κόμβων είναι ίδια με την πραγματική τους θέση στην ιστοσελίδα, ενώ ο ρόλος των σύνθετων κόμβων φαίνεται πλέον καθαρά ότι είναι η διαμόρφωση της χωρικής διάταξης.

Κατά την πρώτη φάση καθορίζονται επίσης τα σύνορα, δηλαδή οι συνοριακοί κόμβοι, των σύνθετων κόμβων. Αυτό θα μας χρειαστεί στη δεύτερη φάση και συγκεκριμένα στην περίπτωση που ένας απλός κόμβος γειτονεύει με έναν σύνθετο. Πριν πάμε σε αυτή όμως ας δούμε ένα παράδειγμα καθορισμού συνόρων. Θα χρησιμοποιήσουμε το DOM Tree της Εικόνας 10 και την αντίστοιχη χωρική διάταξη της Εικόνας 11. Σύμφωνα με την τελευταία παρατηρούμε πως ο κόμβος *form* αποτελεί το πάνω σύνορο και ο κόμβος *p* το κάτω σύνορο.



Εικόνα 12: Ημιτελής τοπολογία ιστοσελίδας του Παραδείγματος 1

Τόσο το αριστερό όσο και το δεξιό σύνορο είναι σύνθετα, με την έννοια ότι αποτελούνται από πολλούς κόμβους. Στο παράδειγμα που μελετάμε τυχαίνει να συμπίπτουν και οι κόμβοι που τα αποτελούν είναι οι *form*, *text* και *p*.

Κατά τη δεύτερη φάση της διαδικασίας δημιουργίας της χωρικής διάταξης καθορίζονται οι γείτονες που δεν μπορούσαν να εντοπιστούν κατά την πρώτη φάση. Εύκολα συμπεραίνουμε πως αυτοί αποτελούν γείτονες των συνοριακών κόμβων που συναντάμε στους σύνθετους κόμβους. Για την εύρεσή τους χρησιμοποιούνται χωρικές πληροφορίες των κόμβων-προγόνων. Η διαπίστωση αυτή μας οδηγεί στο να υιοθετήσουμε μια αντίθετη προσέγγιση στον αλγόριθμό μας συγκριτικά με τον αλγόριθμο της πρώτης φάσης. Έτσι ενώ στην πρώτη φάση ξεκινούσαμε από τα φύλλα και ανεβαίναμε προς τη ρίζα, εδώ ξεκινάμε από τη ρίζα του DOM Tree και προχωράμε προς τα παιδία της.

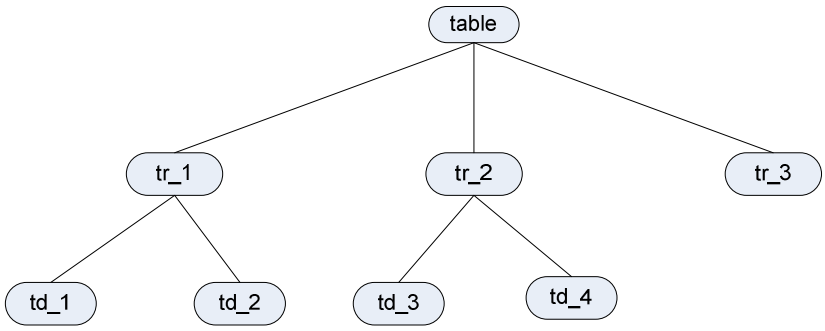
Η προσέγγισή μας είναι λογική αν σκεφτούμε πως οι γείτονες των συνοριακών κόμβων δεν μπορούν παρά να αποτελούν γείτονες ή παιδιά ενός γείτονα του κόμβου-πατέρα.

Στη συνέχεια παρουσιάζουμε μια χονδρική περιγραφή του αλγορίθμου της δεύτερης φάσης σε ψευδοκώδικα.

```
ολοκήρωση_Τοπολογίας(Κόμβος n, Κόμβος up, Κόμβος down, Κόμβος left,
                     Κόμβος right) {

      Εάν (το n δεν έχει γείτονα από πάνω)
            Βάλε γείτονα από πάνω με βάση το up;
      Εάν (το n δεν έχει γείτονα από κάτω)
             Βάλε γείτονα από πάνω με βάση το down;
      Εάν (το n δεν έχει γείτονα από αριστερά)
             Βάλε γείτονα από πάνω με βάση το left;
      Εάν (το n δεν έχει γείτονα από δεξιά)
            Βάλε γείτονα από πάνω με βάση το right;

      Εάν (το n είναι τύπου td)
             εύρεση_Τοπολογίας_Td(n, up, down);

      Εάν (ο κόμβος n είναι σύνθετος) {

         Εάν (το n έχει γείτονα από πάνω)
            up = n;
         Εάν (το n έχει γείτονα από κάτω)
            down = n;
         Εάν (το n έχει γείτονα από αριστερά)
            left = n;
         Εάν (το n έχει γείτονα από δεξιά)
            right = n;

        Πάρε όλα τα παιδιά του κόμβου n;
        Για κάθε (παιδί του κόμβου n)
            ολοκήρωση_Τοπολογίας(επόμενο παιδί του n, up, down, left,
                  right))
}
```

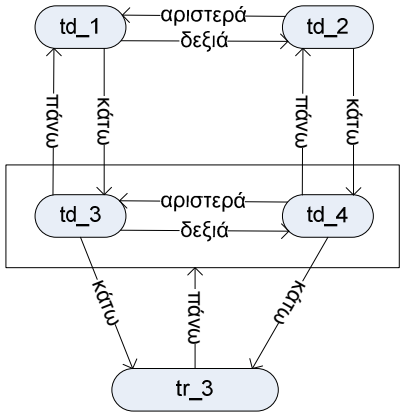
Αλγόριθμος Ολοκλήρωσης Χωρικής Διάταξης

Όπως βλέπουμε από τον παραπάνω αλγόριθμο, λαμβάνεται ειδική μέριμνα για τους κόμβους τύπου *td* μέσω της μεθόδου *εύρεση_Τοπολογίας_Td*. Στόχος μας είναι να εκμεταλλευτούμε και την πιο μικρή πληροφορία χωρικής διάταξης που μας παρέχει το DOM Tree έτσι ώστε να πετύχουμε την μεγαλύτερη δυνατή ευθυγράμμιση των δομικών στοιχείων της ιστοσελίδας. Σίγουρα οι κόμβοι τύπου td παρέχουν χρήσιμη πληροφορία η οποία ωστόσο απαιτεί ειδική διαδικασία εξόρυξης. Θα παρουσιάσουμε τη διαδικασία μέσω ενός θεωρητικού παραδείγματος και συγκεκριμένα θα χρησιμοποιήσουμε το DOM Tree της Εικόνας 13.



Εικόνα 13: Τμήμα ενός DOM Tree

Η σωστή χωρική διάταξη για το δοθέν δένδρο είναι η ακόλουθη.



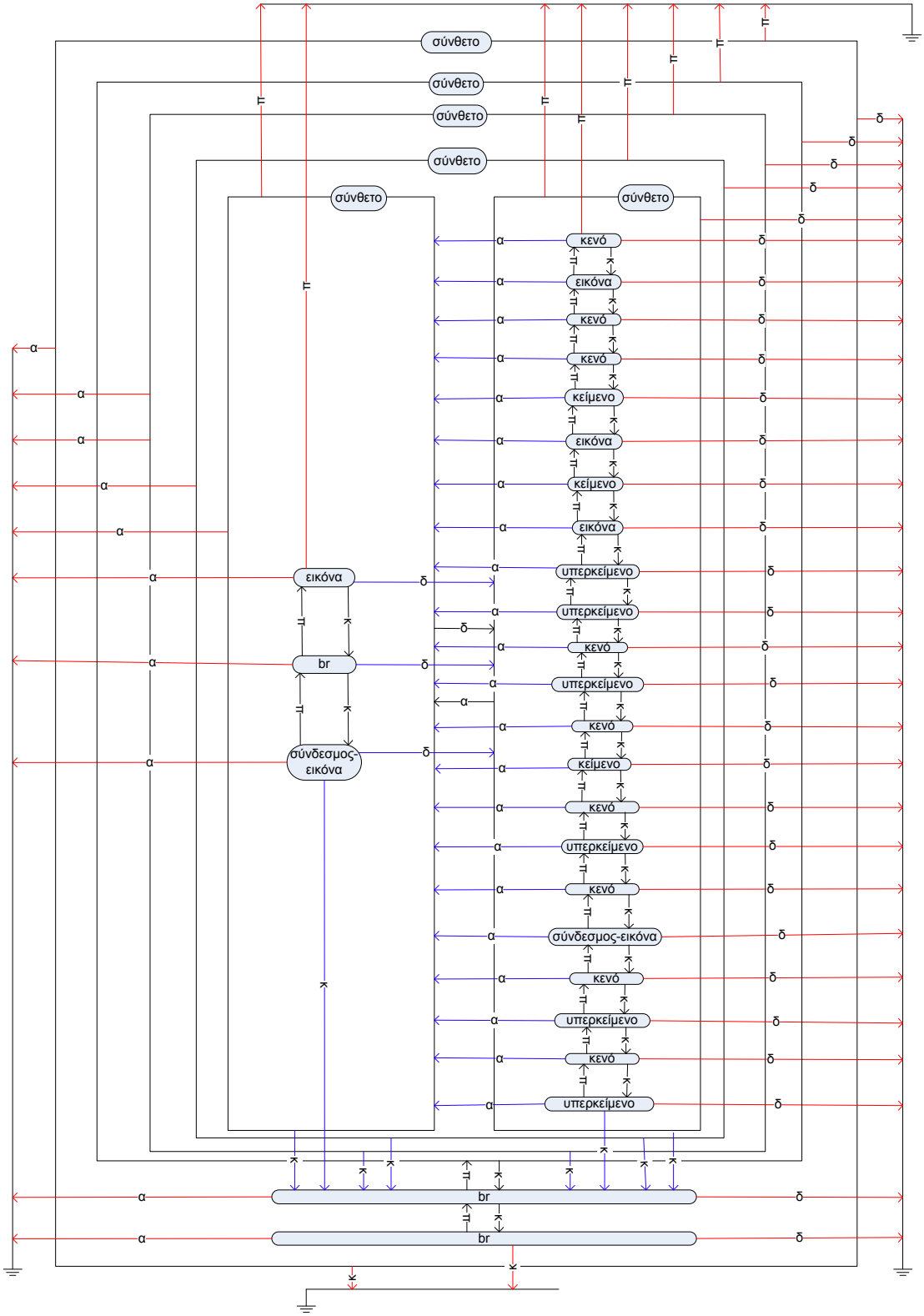
Εικόνα 14: Χωρική τοποθέτηση κόμβων τύπου td

Το αποτέλεσμα αυτό δεν θα μπορούσαμε να το έχουμε αν απλώς εφαρμόζαμε τη γενική διαδικασία που ισχύει για τους κόμβους άλλου τύπου. Αυτό οφείλεται στην σχετική θέση των γειτονικών κόμβων td στο DOM Tree.

Τέλος ας δούμε πως διαμορφώνεται η χωρική διάταξη των κόμβων του Παραδείγματος 1 μετά την εφαρμογή και του δεύτερου αλγορίθμου. Το αποτέλεσμα φαίνεται στην Εικόνα 15. Με μπλε χρώμα έχουν σχεδιαστεί οι δείκτες γειτνίασης των οποίων η τιμή άλλαξε, ενώ με

κόκκινο εκείνοι οι οποίοι αν και τροποποιήθηκαν στην ουσία δεν άλλαξαν τιμή. Οι δείκτες γειτνίασης που παρέμειναν τελείως αμετάβλητοι έχουν σχεδιαστεί με μαύρο χρώμα.

Πριν κλείσουμε την ενότητα αξίζει να αναφερθούμε λίγο στο επίπεδο λεπτομέρειας που έχουμε πετύχει αναφορικά με τους γειτονικούς κόμβους. Όταν, λοιπόν, πρόκειται για απλούς κόμβους που γειτονεύουν γνωρίζουμε την ακριβή τους διάταξη. Όταν όμως έχουμε να κάνουμε με σύνθετους κόμβους, η λεπτομέρεια που μπορούμε να πετύχουμε περιορίζεται. Θα αναφερθούμε στην Εικόνα 15 για να συγκεκριμενοποιήσουμε τη σκέψη μας. Οι δυο σύνθετοι κόμβοι που γειτονεύουν, με τον ένα δίπλα στον άλλον, έχουν πλήρη εικόνα της ‘γειτονιάς’ τους. Τα παιδιά τους όμως δεν γνωρίζουν επακριβώς τις θέσεις των γειτόνων τους. Ξέρουν ότι κάπου απέναντί τους βρίσκονται οι τάδε κόμβοι, αυτοί δηλαδή που αποτελούν το σύνορο του σύνθετου κόμβου, αλλά αγνοούν την ακριβή τους τοποθεσία και τη μεταξύ τους χωρική συσχέτιση. Κάτι τέτοιο βέβαια είναι πέρα από τα όρια της παρούσας διπλωματικής και η λεπτομέρεια που επιτυγχάνουμε είναι αρκετή για να μας δώσει τα επιδιωκόμενα αποτελέσματα.



Εικόνα 15: Τοπολογική διάταξη ιστοσελίδας Παραδείγματος 1

4.2.3 Διαστασιολόγηση Κόμβων

Με την ομαδοποίηση των κόμβων του DOM Tree, είτε με χωρικά κριτήρια είτε με βάση το τύπο του περιεχομένου τους, καθώς και με τη δημιουργία της τοπολογικής διάταξης στην προηγούμενη ενότητα η ανάλυση της ιστοσελίδας έχει ουσιαστικά ολοκληρωθεί. Στην ενότητα αυτή καθώς και στην επόμενη μελετάμε διαδικασίες που μας βοηθούν στην οπτική παρουσίαση του αποτελέσματος που πετύχαμε. Ξεκινάμε με την εύρεση των διαστάσεων των κόμβων.

Όπως έχουμε αναφέρει και σε άλλο σημείο, όλοι οι κόμβοι του DOM Tree αντιστοιχούν σε κάποιο τμήμα της ιστοσελίδας που είναι ορατό στον χρήστη. Κάθε τέτοιο τμήμα καταλαμβάνει κάποιο χώρο και έχει κάποιες διαστάσεις, ύψος και πλάτος. Αυτό σημαίνει ότι μοντελοποιούμε τους κόμβους με ορθογώνια. Ενώ το χώρο που καταλαμβάνει τον γνωρίζουμε, σαν το βάρος του αντίστοιχου κόμβου, οι επιμέρους διαστάσεις του μας είναι άγνωστες. Αυτές θα υπολογίσουμε στη συνέχεια της ενότητας

Πρέπει να τονίσουμε ότι οι τιμές των επιμέρους διαστάσεων που υπολογίζονται για κάθε κόμβο είναι σχετικές και σε καμία περίπτωση οι πραγματικές. Οι τιμές αυτές εκφράζουν το μέγεθος του κόμβου που εξετάζουμε σχετικά με το μέγεθος ενός βασικού δομικού συστατικού της ιστοσελίδας. Σαν βασικό δομικό συστατικό θεωρούμε τους κόμβους απλού τύπου (κείμενο, εικόνα κ.ά.) οι οποίοι δεν έχουν εσωτερική δομή. Οι τιμές που προκύπτουν για την κάθε διάσταση τροποποιούνται κατάλληλα στη συνέχεια έτσι ώστε να ανταποκρίνονται και στην τιμή του βάρους που έχουν οι διάφοροι κόμβοι. Αυτό σημαίνει πως αν δυο κόμβοι έχουν τις ίδιες διαστάσεις αλλά διαφορετικό βάρος, πρέπει οι διαστάσεις τους να μετατραπούν κατά τέτοιο τρόπο που να φανερώνουν τη διαφορά βάρους που υπάρχει.

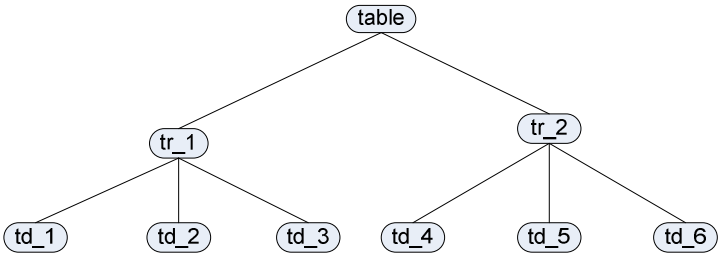
Κεντρική έννοια στην ανάλυσή μας αποτελεί το ελάχιστο περικλείον ορθογώνιο. Είπαμε πως κάθε κόμβο τον μοντελοποιούμε σαν ορθογώνιο όχι όμως σαν οποιοδήποτε ορθογώνιο αλλά σαν το ελάχιστο που απαιτείται έτσι ώστε να περικλείει τα περιεχόμενα του κόμβου. Επομένως οι διαστάσεις του κόμβου αντιστοιχούν στις διαστάσεις του ελαχίστου περικλείοντος ορθογωνίου.

4.2.3.1 Ελάχιστο Περικλείον Ορθογώνιο

Η διαδικασία εύρεσης του ελαχίστου περικλείοντος ορθογωνίου εφαρμόζεται επί του DOM Tree και γίνεται αναδρομικά, ξεκινώντας από τα φύλλα. Αυτό είναι λογικό αν σκεφτούμε πως προκειμένου να βρούμε τις διαστάσεις ενός σύνθετου κόμβου πρέπει πρώτα να έχουμε υπολογίσει τις διαστάσεις των παιδιών του. Σε ότι αφορά τις διαστάσεις των φύλλων, που στην ουσία είναι κόμβοι απλού τύπου, αυτές τίθενται ίσες με τη μονάδα αφού όπως είπαμε

αυτά αποτελούν τη μονάδα αναφοράς. Η ολοκλήρωση της διαδικασία γίνεται σε τρεις φάσεις. Στην πρώτη φάση υπολογίζονται η κατακόρυφη και η οριζόντια διάσταση εξετάζοντας όμως μόνο τους κόμβους-αδέρφια, τους κόμβους δηλαδή που έχουν κοινό κόμβο-πατέρα. Στη δεύτερη φάση υπολογίζεται το πλάτος των κόμβων λαμβάνοντας υπόψη όλους τους κόμβους της ιστοσελίδας. Τέλος, στην τρίτη φάση καταγράφονται οι οριστικές διαστάσεις των κόμβων βάσει του βάρους τους.

Θα εξηγήσουμε την ανάγκη διάκρισης φάσεων με ένα θεωρητικό παράδειγμα. Για το λόγο αυτό χρησιμοποιούμε το DOM Tree της Εικόνας 16. Οι κόμβοι τύπου td έχουν άγνωστη εσωτερική δομή, μάλιστα μπορεί να μην έχουν και καθόλου αν είναι απλοί.



Εικόνα 16: Τμήμα ενός DOM Tree

Παραθέτουμε επίσης, χάριν ευκολίας κατανόησης, τον αλγόριθμο που υπολογίζει την οριστική κατακόρυφη διάσταση των κόμβων.

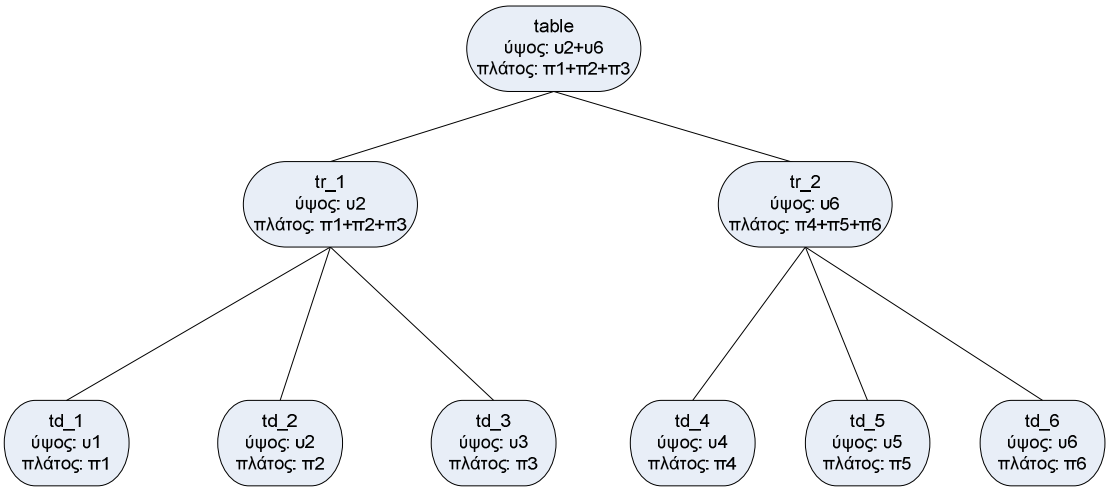
```
κατακόρυφο_Μήκος(Κόμβος n) {

      Μέγιστο_Πλάτος;
      Μέγιστο_Ύψος
      Ουρά Γειτονικών Κόμβων ;
      Κόμβος Δεξιός_Γείτονας, Από_Κάτω_Γείτονας;
      Πρόσθεσε στην ουρά το n;

      Όσο (η ουρά queue δεν είναι κενή) {

            Πάρε το πρώτο στοιχείο της ουράς;
            Πάρε το ύψος του στοιχείου;
            Πάρε το πλάτος του στοιχείου;
            Εάν( το στοιχείο δεν έχει γείτονες)
            {
               Θέσε το Μέγιστο_Πλάτος ίσο με το πλάτος του;
               Θέσε το Μέγιστο_Ύψος ίσο με το ύψος του;
            }
            Αλλιώς
            {
                  Πάρε τον Δεξιό_Γείτονα;
                  Πάρε τον Από_Κάτω_Γείτονα;

                  Εάν( ο Δεξιός_Γείτονας δεν είναι κενός) {
                        Εάν( το Μέγιστο_Ύψος < ύψος του κόμβου n)
                          Μέγιστο_Ύψος = ύψος κόμβου n;
                        Πρόσθεσε στο Μέγιστο_Πλάτος το πλάτος του n;
                        Πρόσθεσε στην ουρά τον Δεξιό_Γείτονα;
                  }
                  Εάν( ο Από_Κάτω_Γείτονας δεν είναι κενός) {
                        Εάν( το Μέγιστο_Πλάτος < πλάτος του κόμβου n)
```

```
                        Μέγιστο_Πλάτος = πλάτος κόμβου n;
                    Πρόσθεσε στο Μέγιστο_Ύψος το ύψος του n;
                    Πρόσθεσε στην ουρά τον Από_Κάτω_Γείτονα;
                }
            }
        }

    Θέσε το ύψος του κόμβου-πατέρα ίσο με το Μέγιστο_Ύψος;
    Θέσε το πλάτος του κόμβου-πατέρα ίσο με το Μέγιστο_Πλάτος;
}
```

Αλγόριθμος που υπολογίζει το ύψος των κόμβων

Όπως μπορούμε να παρατηρήσουμε, ο αλγόριθμος υπολογίζει και το οριζόντιο μήκος (πλάτος) των κόμβων. Τις περισσότερες φορές όμως δεν είναι το τελικό κάτι το οποίο θα φανεί από το παράδειγμα. Μια επιπλέον παρατήρηση που μπορούμε να κάνουμε για τον αλγόριθμο είναι ότι η προσπέλαση των κόμβων δεν γίνεται πλέον μέσω του DOM Tree αλλά μέσω των δεικτών γειτνίασης και μάλιστα ξεκινάει από την πάνω αριστερή γωνία της ιστοσελίδας και κινείται δεξιά και κάτω.

Δίνοντας στον προηγούμενο αλγόριθμο σαν είσοδο το DOM Tree της Εικόνας 16, παίρνουμε το ακόλουθο αποτέλεσμα, το οποίο αποτελεί ένα τυχαίο σενάριο εκτέλεσης.



Εικόνα 17: Καταγραφή ύψους και πλάτους των κόμβων

Έχουμε υποθέσει πως μεταξύ των κόμβων td_1, td_2 και td_3 το μεγαλύτερο ύψος είναι το υ2, ενώ μεταξύ των κόμβων td_4, td_5 και td_6 είναι το υ6. Επίσης μεταξύ των κόμβων tr_1 και tr_2 το μεγαλύτερο πλάτος είναι το π1+π2+π3.

Αν όλοι οι τερματικοί κόμβοι td_1, td_2, td_3, td_4, td_5 και td_6 είχαν ίδιο ύψος και ίδιο πλάτος η διαδικασία διαστασιολόγησης θα είχε τελειώσει εδώ. Επειδή όμως αυτό σπάνια συμβαίνει απαιτούνται και οι μετέπειτα φάσεις. Σε κάθε περίπτωση πάντως, η ρίζα του δένδρου έχει πάντα το σωστό κατακόρυφο μήκος (ύψος).

Πριν εξηγήσουμε την αναγκαιότητα της δεύτερης φάσης, θα αναφερθούμε λίγο στο τι θέλουμε να πετύχουμε προσωρινά. Ενδιάμεσος, λοιπόν, στόχος μας είναι να απεικονίσουμε

την ιστοσελίδα σαν ένα πλέγμα με κελιά όπου κελία στην ίδια στήλη έχουν το ίδιο πλάτος και κελιά στην ίδια γραμμή έχουν το ίδιο ύψος. Για το παράδειγμά μας το πλέγμα αυτό είναι το κάτωθι.

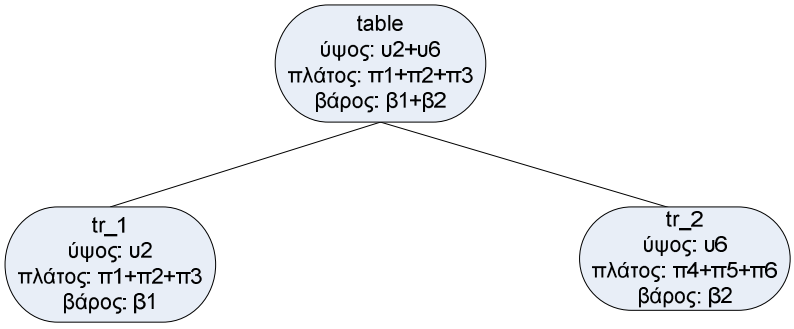
td_1	td_2	td_3
td_4	td_5	td_6

Εικόνα 18: Απεικόνιση της σελίδας σαν πλέγμα κελιών

Μέχρι στιγμής γνωρίζουμε για κάθε γραμμή ποιος κόμβος έχει το μεγαλύτερο ύψος όπως και για κάθε στήλη ποιος κόμβος έχει το μεγαλύτερο πλάτος. Ενώ όμως, λόγω της δομή του DOM Tree, το μεγαλύτερο ύψος σωστά χρησιμοποιείται στους υπολογισμούς για τους κόμβους των ανώτερων επιπέδων, δεν ισχύει το ίδιο και για το μεγαλύτερο πλάτος. Την αδυναμία αυτή έρχεται να καλύψει η δεύτερη φάση της διαδικασίας διαστασιολόγησης η οποία χρησιμοποιεί πλέον το μεγαλύτερο πλάτος από κάθε στήλη. Μετά και τη δεύτερη φάση ο κόμβος-ρίζα έχει σωστό πλάτος. Ο αλγόριθμος που το επιτυγχάνει αυτό είναι παρόμοιος με τον Αλγόριθμο που υπολογίζει το ύψος των κόμβων.

Στο σημείο αυτό πρέπει να πούμε ότι τελικός στόχος των δυο πρώτων φάσεων ήταν η εύρεση του σωστού ύψους και πλάτους της ρίζας του DOM Tree. Οι διαστάσεις των υπόλοιπων κόμβων δεν είναι συνεπείς, αλλά αυτό δεν μας απασχολεί αφού θα ενημερωθούν στη συνέχεια. Τα μόνα δεδομένα που χρειαζόμαστε είναι οι διαστάσεις της ρίζας και το βάρος όλων των κόμβων.

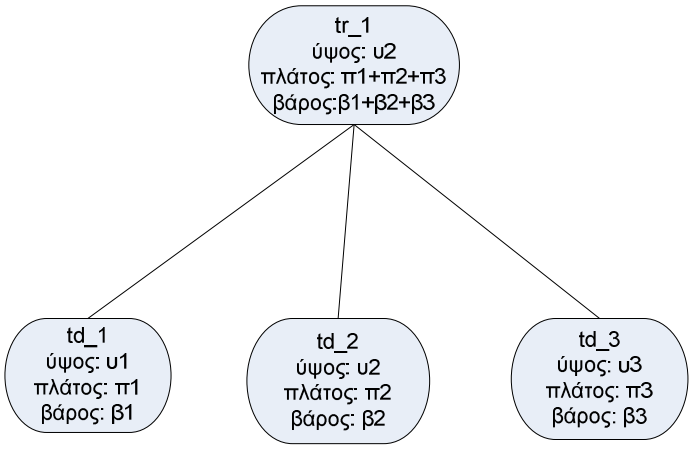
Στην τρίτη, λοιπόν, και τελευταία φάση της διαδικασίας διαστασιολόγησης οριστικοποιούνται οι διαστάσεις των κόμβων με τη βοήθεια του βάρους τους. Ας το δούμε αυτό συνεχίζοντας το προηγούμενο παράδειγμα. Αρχικά θα χρησιμοποιήσουμε μόνο τη ρίζα και τα δυο παιδιά της παρουσιάζοντας επιπλέον και τα βάρη τους.



Εικόνα 19: Οριστικοποίηση των διαστάσεων των κόμβων για κατακόρυφη διάταξη

Όπως έχουμε εξηγήσει στην αντίστοιχη ενότητα, το βάρος εκφράζει την συνολική επιφάνεια που ένα στοιχείο καταλαμβάνει στην ιστοσελίδα. Επομένως δεν θα ήταν λογικό στοιχεία με διαφορετικό βάρος να έχουν τις ίδιες διαστάσεις. Αυτήν ακριβώς την παρατυπία διορθώνουμε σε αυτή τη φάση.

Υπάρχουν δυο περιπτώσεις ανάλογα με τη διάσταση που πρέπει να τροποποιήσουμε που με τη σειρά του εξαρτάται από τον προσανατολισμό των κόμβων-παιδιών. Στην προηγούμενη εικόνα έχουμε κατακόρυφο προσανατολισμό επομένως πρέπει να επαναπροσδιορίσουμε το ύψος των κόμβων-παιδιών. Στην επόμενη εικόνα έχουμε οριζόντιο προσανατολισμό και γι' αυτό πρέπει να υπολογίσουμε πάλι το πλάτος των κόμβων-παιδιών.



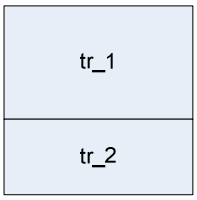
Εικόνα 20: Οριστικοποίηση των διαστάσεων των κόμβων για οριζόντια διάταξη

Τονίσαμε στην αρχή της ενότητας ότι οι διαστάσεις που τελικά θα υπολογίσουμε είναι σχετικές και όχι απόλυτες. Αυτό φαίνεται και στον επόμενο αλγόριθμο όπου οι διαστάσεις των κόμβων-παιδιών υπολογίζονται σαν ποσοστό των διαστάσεων του κόμβου-πατέρα. Το εν λόγο ποσοστό προκύπτει σαν ο λόγος του βάρους του κόμβου-παιδιού προς το βάρος του κόμβου-πατέρα.

```
ολοκλήρωση_Διαστάσεων(Κόμβος n, βάρος_Κόμβου_Πατέρα,
                  ύψος_Κόμβου_Πατέρα, πλάτος_Κόμβου_Πατέρα) {

      Βρες το βάρος_Κόμβου_n;
      λόγος_Βαρών = βάρος_Κόμβου_n προς βάρος_Κόμβου_Πατέρα;
      Βρες τον προσανατολισμό;

      Εάν (ο προσανατολισμός είναι κατακόρυφος) {
            πλάτος_Κόμβου_n = πλάτος_Κόμβου_Πατέρα;
            ύψος_Κόμβου_n = ύψος_Κόμβου_Πατέρα * λόγος_Βαρών;
      }
      Αλλιώς {
            πλάτος_Κόμβου_n = πλάτος_Κόμβου_Πατέρα* λόγος_Βαρών;
            ύψος_Κόμβου_n = ύψος_Κόμβου_Πατέρα;
      }

      Εάν (ο κόμβος n είναι σύνθετος) {
          Πάρε όλα τα παιδιά του κόμβου n;
          Για (κάθε παιδί του κόμβου n)
              ολοκλήρωση_Διαστάσεων(επόμενο παιδί του κόμβου
                            n, βάρος_Κόμβου_n, ύψος_Κόμβου_n,
                            πλάτος_Κόμβου_n);
      }
}
```

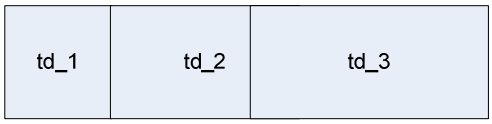
Αλγόριθμος οριστικοποίησης διαστάσεων των κόμβων

Ας επιστρέψουμε στο παράδειγμα της Εικόνας 16 καις ας υποθέσουμε πως το βάρος β1 είναι μεγαλύτερο από το β2. Ένα πιθανό αποτέλεσμα που παρουσιάζει τα περικλείοντα ορθογώνια είναι το κάτωθι.



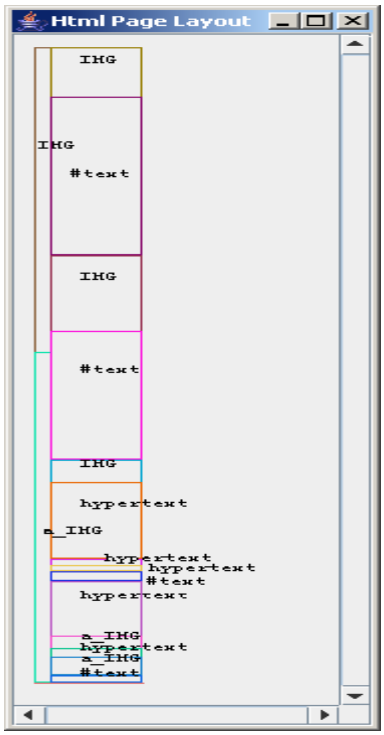
Εικόνα 21: Απεικόνιση κόμβων χρησιμοποιώντας τα περικλείοντα ορθογώνια

Παρατηρούμε ότι και οι δυο κόμβοι έχουν το ίδιο πλάτος. Η περίπτωση στην οποία το ύψος παραμένει αμετάβλητο αντιστοιχεί στην Εικόνα 22. Το αντίστοιχο αποτέλεσμα για τη διάταξη βαρών β1 < β2 < β3 φαίνεται ακολούθως.



Εικόνα 22: Απεικόνιση κόμβων χρησιμοποιώντας τα περικλείοντα ορθογώνια

Σαν τελευταίο παράδειγμα θα παρουσιάσουμε το αποτέλεσμα με τα ελάχιστα περικλείοντα ορθογώνια για την ιστοσελίδα του Παραδείγματος 1. Η έξοδος φαίνεται στην Εικόνα 23.



Εικόνα 23: Απεικόνιση ιστοσελίδας Παραδείγματος 1 με βάση τα περικλείοντα ορθογώνια

4.2.4 Συντεταγμένες Κόμβων

Έχοντας από την προηγούμενη ενότητα τις διαστάσεις των κόμβων, ορθότερα των ελαχίστων περικλειόντων ορθογωνίων, μπορούμε πλέον να μελετήσουμε την τοποθέτησή τους στο σωστό σημείο του χώρου. Προκειμένου να επιτευχθεί αυτό χρειάζεται να υπολογίσουμε πρώτα τις συντεταγμένες του κάθε κόμβου. Αυτό είναι και το αντικείμενο μελέτης της παρούσας ενότητας.

Όπως η προηγούμενη έτσι και αυτή η ενότητα εντάσσονται στην προσπάθεια οπτικοποίησης του αποτελέσματος που προέκυψε από τις διαδικασίες ανάλυσης επί του DOM Tree. Γνωρίζουμε ήδη πως οι κόμβοι που είναι ορατοί είναι οι τερματικοί κόμβοι (φύλλα), οι οποίοι εμπεριέχουν το περιεχόμενο της ιστοσελίδας στο σύνολό του. Έτσι θα περίμενε κανείς πως η απόδοση συντεταγμένων επικεντρώνονται σε αυτούς τους κόμβους, κάτι τέτοιο όμως δεν ισχύει αφού αποδίδουμε συντεταγμένες ακόμη και στους σύνθετους κόμβους. Η χρησιμότητα της ενέργειας αυτής θα εξηγηθεί στη συνέχεια της ενότητας.

Προφανώς η εύρεση των σωστών συντεταγμένων για κάθε κόμβο απαιτεί τη γνώση όχι μόνο των γειτονικών του κόμβων αλλά και των διαστάσεων τους, ακόμη και των διαστάσεων του ίδιου του υπό εξέταση κόμβου. Έχοντας πλέον αυτά τα δεδομένα, η διαδικασία υπολογισμού των συντεταγμένων ξεκινάει από τη ρίζα του δένδρου και προχωράει προς τα παιδιά της. Με την περάτωσή της οι κόμβοι γνωρίζουν τις απόλυτες συντεταγμένες τους.

4.2.4.1 Υπολογισμός Συντεταγμένων

Θα ξεκινήσουμε την ανάλυσή μας παραθέτοντας τον αλγόριθμο που πραγματοποιεί τον υπολογισμό των συντεταγμένων.

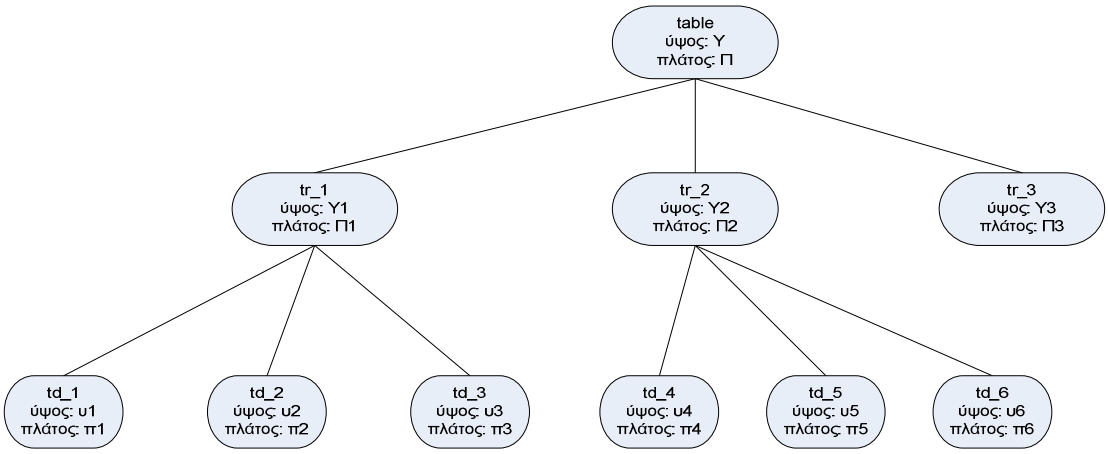
```
υπολογισμός_Συντεταγμένων( Κόμβος n)
{
    ουρά_Κόμβων;
    Βάλε τον κόμβο n στην ουρά_Κόμβων;
    Όσο ( η ουρά δεν είναι κενή)
    {
        Πάρε τον πρώτο κόμβο από την ουρά_Κόμβων;
        Πάρε τον κόμβο-πατέρα του κόμβου αυτού;
        Βρες τον προσανατολισμό;
        Εάν ( ο προσανατολισμός είναι οριζόντιος)
        {
           Θέσε την τετμημένη του κόμβου ίση με τη τετμημένη του
           κόμβου-πατέρα;
           Πάρε τον αριστερό γείτονα;
           Θέσε την τεταγμένη του κόμβου ίση με το άθροισμα της
           τεταγμένης του αριστερού γείτονα και του πλάτους του;
        }
```

```
            Εάν ( ο προσανατολισμός είναι κατακόρυφος)
            {
                Θέσε την τεταγμένη του κόμβου ίση με τη τεταγμένη του
                κόμβου-πατέρα;
                Πάρε τον από πάνω γείτονα;
                Θέσε την τετμημένη του κόμβου ίση με το άθροισμα της
                τετμημένης του από πάνω γείτονα και του ύψους του;
            }
            Πάρε όλα τα παιδιά του κόμβου n και βαλε τα στην
            ουρά_Κόμβων;

        }
}
```

Αλγόριθμος εύρεσης συντεταγμένων

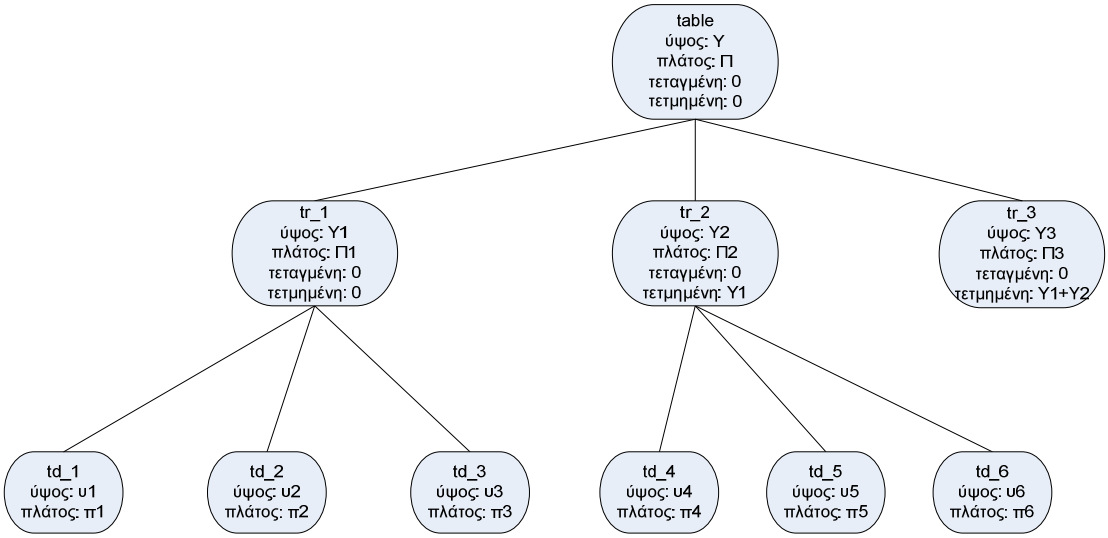
Παρατηρούμε πως ο αλγόριθμος εφαρμόζει κατά πλάτος διάσχιση (breadth first search) του δένδρου. Η επιλογή αυτή ήταν μονόδρομος εξαιτίας της τακτικής που υιοθετήσαμε, πως δηλαδή οι συντεταγμένες έχουν αφετηρία τη ρίζα του δένδρου και μοιράζονται στους κόμβους-παιδιά της βάσει της τοπολογικής τους τοποθέτησης η οποία γίνεται από τα αριστερά προς τα δεξιά και από πάνω προς τα κάτω. Θα το εξηγήσουμε αυτό αναλυτικότερα μέσω ενός παραδείγματος. Για το λόγο αυτό χρησιμοποιούμε το DOM Tree της Εικόνας 24.

Η απόδοση συντεταγμένων στη ρίζα γίνεται πάντα χειροκίνητα και φυσικά ορίζεται ως το σημείο (0,0). Η εκτέλεση του Αλγορίθμου εύρεσης συντεταγμένων για τους κόμβους του δευτέρου επιπέδου δίνει την Εικόνα 25. Το τελικό αποτέλεσμα εμφανίζεται στην Εικόνα 26.

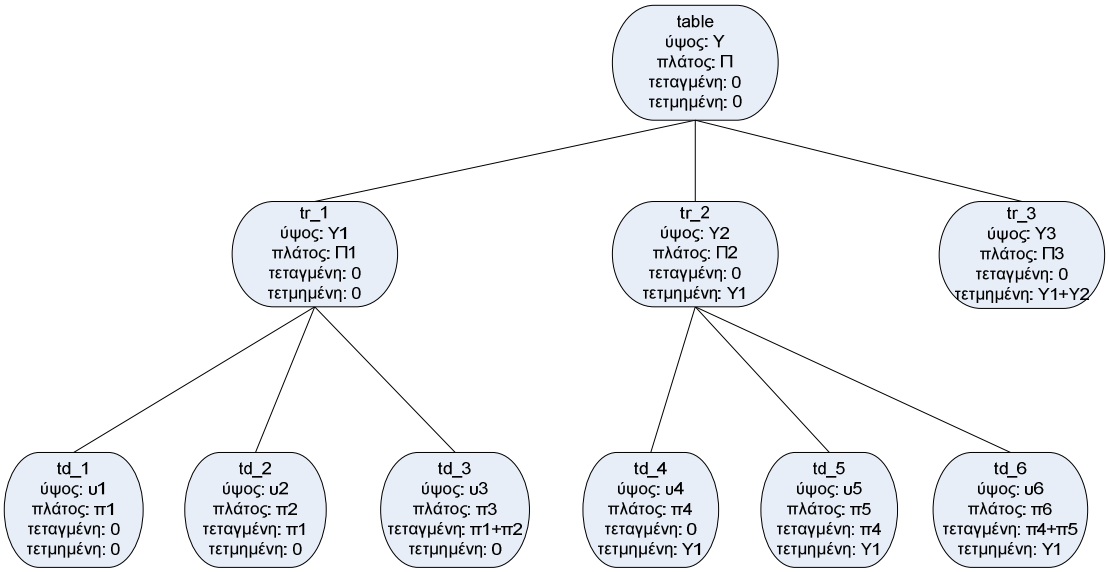


Εικόνα 24: Κόμβοι ενός DOM Tree με τις διαστάσεις τους

Οι εικόνες παρουσιάζουν την πορεία που ακολουθεί ο αλγόριθμος για την καταγραφή των συντεταγμένων. Όπως έχουμε ήδη αναφέρει, αυτή είναι από πάνω προς τα κάτω και από αριστερά προς τα δεξιά. Αυτό ήταν υποχρεωτικό αφού προκειμένου να βρούμε τις συντεταγμένες ενός κόμβου θα πρέπει πρώτα να γνωρίζουμε εκείνες του κόμβου-πατέρα. Όσο για την οριζόντια διεύθυνση, θα μπορούσαμε κάλλιστα να ξεκινήσουμε από τα δεξιά και να προχωρήσουμε προς τα αριστερά κάνοντας κάποιες τροποποιήσεις στον αλγόριθμο.



Εικόνα 25: Ημιτελής καταγραφή συντεταγμένων των κόμβων



Εικόνα 25: Πλήρης καταγραφή των συντεταγμένων των κόμβων

Στο σημείο αυτό θα αναφερθούμε στη χρησιμότητα των συντεταγμένων για τους σύνθετους κόμβους. Πρώτα απ' όλα έχουν πρακτική ωφέλεια αφού διευκολύνουν την όλη διαδικασία απόδοσης συντεταγμένων. Πέρα από αυτό όμως, θα μπορούσαν να χρησιμοποιηθούν από διάφορες εφαρμογές επεξεργασίας ιστοσελίδων οι οποίες εξετάζουν και εξάγουν διάφορα τμήματα της ιστοσελίδας ανάλογα με τη θέση τους σε αυτή. Αυτά τα τμήματα μπορεί να μην είναι ενός μόνο τύπου περιεχομένου αλλά να περιέχουν επιμέρους τμήματα. Αυτό γίνεται εφικτό μέσω της ανάθεσης συντεταγμένων στους σύνθετους κόμβους.

4.3 Σύνοψη

Στο κεφάλαιο αυτό παρουσιάσαμε και αναλύσαμε τη διαδικασία εύρεσης της τοπολογικής διάταξης μιας ιστοσελίδας. Η διαδικασία αυτή χωρίζεται σε επιμέρους φάσεις με τη κάθε μια να επιτελεί συγκεκριμένη λειτουργία και όλες μαζί σε μια αλυσιδωτή σειρά διαμορφώνουν το τελικό αποτέλεσμα , το οποίο καταγράφεται στους κόμβου του DOM Tree και όχι σε κάποια πρόσθετη δομή.

5 *Υλοποίηση Συστήματος*

Στο κεφάλαιο αυτό παρουσιάζονται λεπτομέρειες σχετικές με την υλοποίηση του λογισμικού που αναπτύξαμε. Αναφέρονται τα εργαλεία που χρησιμοποιήθηκαν για την ανάπτυξή του και περιγράφεται η οργάνωση του κώδικα. Τέλος, πραγματοποιούμε μια αξιολόγηση του λογισμικού μέσω εκτέλεσης αντιπροσωπευτικών πειραμάτων και ελέγχοντας τα αποτελέσματα που μας δίνουν.

5.1 Λεπτομέρειες Υλοποίησης

5.1.1 Προγραμματιστικά Εργαλεία

Για την ανάπτυξη της παρούσας διπλωματικής εργασίας χρησιμοποιήθηκε η γλώσσα προγραμματισμού Java στην έκδοση 1.5. Η ανάπτυξη έγινε στο ολοκληρωμένο προγραμματιστικό περιβάλλον Eclipse έκδοση 3.2. Για το parsing των Html αρχείων και την εξαγωγή του αντίστοιχου DOM Tree χρησιμοποιήθηκε ο ανοικτού κώδικα Html parser cobra-0.96.4. Ο εν λόγω parser έχει επίσης αναπτυχθεί σε Java και ενσωματώθηκε στο όλο λογισμικό σαν ένα ξεχωριστό πακέτο.

5.1.2 Υλοποίηση

Στη συνέχεια περιγράφουμε συνοπτικά τα πακέτα και τις κλάσεις που συνθέτουν το λογισμικό που αναπτύξαμε. Παρουσιάζουμε ακόμα τις μεθόδους οι οποίες υλοποιούν τους αλγορίθμους που παραθέσαμε στα προηγούμενα κεφάλαια.

5.1.3 Πακέτα

Πακέτο	**Λειτουργικότητα**
cobra	Λειτουργίες που στο σύνολό τους συμβάλλουν στην εξαγωγή του DOM Tree
tags	Περιλαμβάνει κλάσεις που μοντελοποιούν τα διάφορα tags του Html αρχείου
groups	Ενθυλακώνει τις λειτουργίες που πραγματοποιούν την ομαδοποίηση των κόμβων
topology	Υλοποιεί τις διαδικασίες τοπολογικής επεξεργασίας
test	Περιλαμβάνει κλάσεις για τον έλεγχο της λειτουργίας του συστήματος

5.1.4 Κλάσεις

5.1.4.1 Πακέτο tags

- Interface *NodeContent*: αφηρημένη μοντελοποίηση των διάφορων tags που μπορούν να υπάρξουν σε ένα Html αρχείο.
- Κλάσεις *A_imgNodeContent, TextNodeContent, A_textNodeContent, ANodeContent, BrNodeContent, EmptyNodeContent, FormNodeContent, HypertextNodeContent, ImageNodeContent, SeparatorNodeContent*. Υλοποιούν το Interface NodeContent με βάση τις συγκεκριμένες ιδιότητες του κάθε tag.
- Enumeration *NodeContentTypeEnum*. Η κλάση αυτή περιέχει όλους τους δυνατούς τύπους των κόμβων που μπορούν να υπάρξουν.

5.1.4.2 Πακέτο groups

- Κλάση *ClearDomTree* η οποία παρέχει τη λειτουργία καθαρισμού του DOM Tree μέσω της μεθόδου *public Node clearTree(Node n).*
- Κλάση *Group* η οποία μοντελοποιεί τις ομάδες τν κόμβων όπως αυτές προκύπτουν μετά την ομαδοποίηση βάσει περιεχομένου. Περιλαμβάνει ιδιότητες που

χαρακτηρίζουν την κάθε ομάδα όπως τύπος ομάδας, βάρος, κόμβους που περιλαμβάνει κ.ά.

- Κλάση *MergeNode* η οποία μοντελοποιεί του συνδυασμούς των κόμβων κατά την εφαρμογή του αλγορίθμου ομαδοποίησης της ενότητας 3.3.2.4.
- Κλάση *NodesWeight* η οποία παρέχει τη λειτουργία υπολογισμού του βάρους των κόμβων μέσω της μεθόδου public void *setNodesWeight(Node n)*.
- Κλάση *ClassifyTreeNodes*. Αυτή πραγματοποιεί την ομαδοποίηση των κόμβων μέσω της μεθόδου *classifyNodes(Node n)* η οποία καλεί όλους του αλγορίθμους του κεφαλαίου 3.
- Κλάση MethodCollection. Η κλάση αυτή περιλαμβάνει γενικές μεθόδους που χρησιμοποιούνται ή υλοποιούν μερικούς από τους αλγορίθμους των κεφαλαίων 3 και 4. Οι σημαντικότερες είναι οι εξής:
  - *public static int getNodeDistance(Node n1, Node n2)* η οποία βρίσκει την απόσταση δυο κόμβων στο DOM Tree.
  - *public static int getTypeImprtance(String type)* η οποία επιστρέφει τη σημαντικότητα ενός τύπου όπως αυτή ορίστηκε στην ενότητα 3.3.2.6.
  - *public static String checkCombatibility(String type1, String type2)* . Η μέθοδος αυτή, χρησιμοποιώντας τις παραμέτρους της ενότητας 3.3.2.2 καθώς και την απόσταση των κόμβων, καθορίζει το τελικό αποτέλεσμα της ομαδοποίησης δυο κόμβων το οποίο και επιστρέφει.
  - *public static boolean isVertical(Node n)* η οποία αποφαίνεται για τον προσανατολισμό ενός κόμβου με βάση το όνομα του tag που αντιπροσωπεύει.
  - *public static void createBorders(Node n)*. Η μέθοδος αυτή δέχεται σαν όρισμα κάποιον κόμβο και, στην περίπτωση που είναι σύνθετος, βρίσκει τους συνοριακούς του κόμβους. Αυτό είναι απαραίτητο για την περίπτωση που θέλουμε να βρούμε με ποιους κόμβους συνορεύει ένας κόμβος που έχει σαν γείτονα έναν σύνθετο κόμβο.

5.1.4.3 Πακέτο topology

- Κλάση *TreeTopology* η οποία περιλαμβάνει μεθόδους για την υλοποίηση των αλγορίθμων τοπολογικής επεξεργασίας των κόμβων. Οι μέθοδοι αυτές είναι οι *setNeighbors(Node n), tdTopology(Node n), setVerticalLength(Node n), setHorizontalLength(Node n), setRelativeCoordinates(Node n), setAbsoluteCoordinates(Node n) και setMinimumBoundingBox(Node n)*. Η

αντιστοίχιση των προηγούμενων μεθόδων με τους αλγορίθμους των κεφαλαίων 3 και 4 είναι εμφανής από τα ονόματά τους.

5.1.4.4 Πακέτο test

- Κλάση *HtmlPageLayout*. Η κλάση αυτή είναι βοηθητική και χρησιμοποιείται για την οπτικοποίηση των αποτελεσμάτων του συστήματος μέσω γραφικών.
- Κλάση *PrintMethods*. Περιλαμβάνει μεθόδους για εκτυπώσεις.
- Κλάση *HtmlLayoutStructure*. Η εν λόγω κλάση αποτελεί την έξοδο του συστήματος και μοντελοποιεί τον τρόπο που το σύνολο των ομαδοποιημένων κόμβων είναι οργανωμένα στον χώρο. Παρέχει δηλαδή τη τοπολογία της ιστοσελίδας. Περιλαμβάνει μεθόδους που επιτρέπουν την ανάκτηση δεδομένων με διάφορα κριτήρια όπως ο τύπος τους, το βάρος τους ή η θέση τους στη σελίδα. Αυτές είναι οι *getGroupsBySize(), getGroupsByType(String type), getCenter(), getHeader(), getFooter(), getLeftSide(), getRightSide()*.

5.2 Έλεγχος Συστήματος

5.2.1 Λεπτομέρειες Πειραμάτων

Πρώτο και σημαντικότερο χαρακτηριστικό στοιχείο των πειραμάτων είναι οι παράμετροι που χρησιμοποιούνται για την αξιολόγηση των αποτελεσμάτων τους. Κίνητρο για την ανάπτυξη του συστήματός μας ήταν η βελτίωση της αποδοτικότητας του geoparser. Πέρα από αυτό όμως το σύστημα μπορεί να υπάρξει και μόνο του παρέχοντας λειτουργίες ανάλυσης των ιστοσελίδων παρόμοιες με εκείνες που παρουσιάστηκαν στο κεφάλαιο 2 στις σχετικές εργασίες. Επομένως πρέπει να το ελέγξουμε και ως προς τις δυο προαναφερθείσες λειτουργίες του.

Ξεκινώντας με την επίδραση που έχει στη λειτουργία του geoparser, αυτή μπορεί να γίνει αντιληπτή τόσο από το χρόνο επεξεργασίας που απαιτείται για την κάθε ιστοσελίδα όσο και από την ακρίβεια των αποτελεσμάτων που δίνει. Αναμένουμε ο χρόνος επεξεργασίας να μειωθεί αφού ο geoparser δεν έχει πλέον να επεξεργαστεί ολόκληρη τη σελίδα αλλά μόνο το σημαντικότερο τμήμα της το οποίο δέχεται σαν είσοδο. Βέβαια στο χρόνο αυτό πρέπει να λάβουμε υπόψη και το χρόνο που χρειάζεται το σύστημά μας για να πραγματοποιήσει την ανάλυση της σελίδας. Αναφορικά με την ποιότητα των αποτελεσμάτων, αυτή εξαρτάται από την ιστοσελίδα και μπορεί είτε να βελτιωθεί είτε να παραμείνει ίδια. Η τελευταία περίπτωση είναι το χειρότερο σενάριο λειτουργίας και συμβαίνει όταν όλες οι γεωγραφικές πληροφορίες

της σελίδας οργανώνονται μαζί στο ίδιο τμήμα. Σε κάθε άλλη περίπτωση περιμένουμε το εύρος των γεωγραφικών σημείων που εντοπίζονται να είναι μικρότερο.

Αν θεωρήσουμε το σύστημα που αναπτύξαμε σαν αυτόνομο πρέπει να εξετάσουμε την ορθότητα δυο μεθοδολογιών: της μεθοδολογίας ομαδοποίησης των κόμβων του DOM Tree βάσει του περιεχομένου τους και της μεθοδολογίας εύρεσης της τοπολογίας μιας ιστοσελίδας. Μέτρο σύγκρισης αποτελεί η ομοιότητα των αποτελεσμάτων με την ανθρώπινη κρίση για την ίδια ιστοσελίδα. Έτσι, ενώ για τη δεύτερη μεθοδολογία ο έλεγχος είναι απλός και ευθύς, η έξοδος που παίρνουμε θα είναι ή σωστή ή λάθος, για τη πρώτη δεν υπάρχει μονοσήμαντη κρίση. Όπως έχουμε αναφέρει σε προηγούμενο σημείο, ο χρήστης μπορεί να καθορίσει το επίπεδο λεπτομέρειας μεταβάλλοντας ορισμένες παραμέτρους του συστήματος. Επομένως, ο έλεγχος των αποτελεσμάτων μπορεί να γίνει μόνο σε συνδυασμό με τις παραμέτρους αυτές.

Η επιλογή των δεδομένων εισόδου, δηλαδή των ιστοσελίδων που θα μελετηθούν, έγινε με κριτήριο να καλύψουμε το μεγαλύτερο δυνατό εύρος ως προς τη κατηγορία τους (τουριστικές, εμπορικές ,ειδησεογραφικές) φροντίζοντας παράλληλα να διατηρήσουμε τον αριθμό των δοκιμών μικρό. Όλες οι επιμέρους διαδικασίες του συστήματος ελέχθησαν με το ίδιο σύνολο δεδομένων, εμείς όμως θα παρουσιάσουμε μόνο το τελικό αποτέλεσμα που δίνει το σύστημα στην ολότητά του.

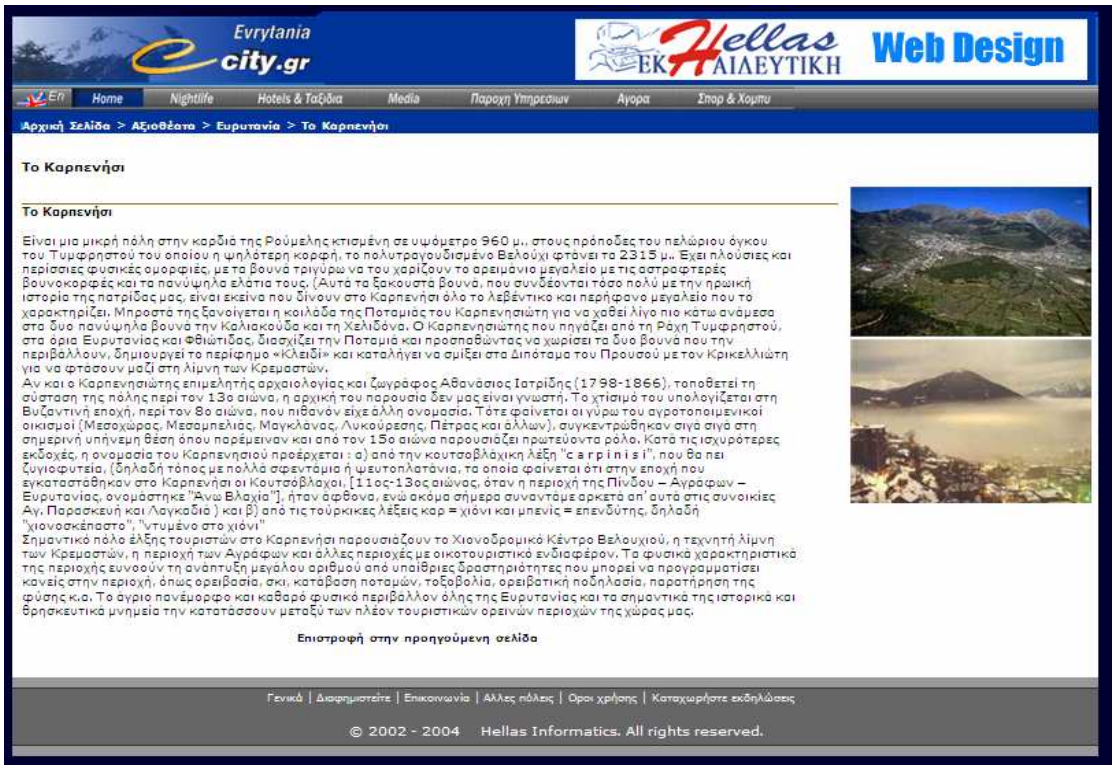
5.2.2 Αποτελέσματα

Ακολουθούν τα αποτελέσματα των πειραμάτων ομαδοποιημένα ανά κατηγορία περιεχομένου της ιστοσελίδας. Για κάθε ιστοσελίδα εκτελούνται δυο δοκιμές με διαφορετικές τιμές των παραμέτρων. Το πρώτο σύνολο τιμών επιδιώκει μια λεπτομερή ανάλυση της σελίδας ενώ το δεύτερο μια πιο χονδρική, από την οποία παίρνουμε το κεντρικό και σημαντικότερο τμήμα για να το δώσουμε σαν είσοδο στον geoparser. Τέλος, οφείλουμε να διευκρινίσουμε ότι τα αποτελέσματα που παρουσιάζονται δεν είναι τίποτε άλλο παρά η οπτικοποίηση της τελικής δομής επί του DOM Tree των ομαδοποιημένων κόμβων που δίνει σαν έξοδο το σύστημα.

5.2.2.1 Τουριστικές Σελίδες

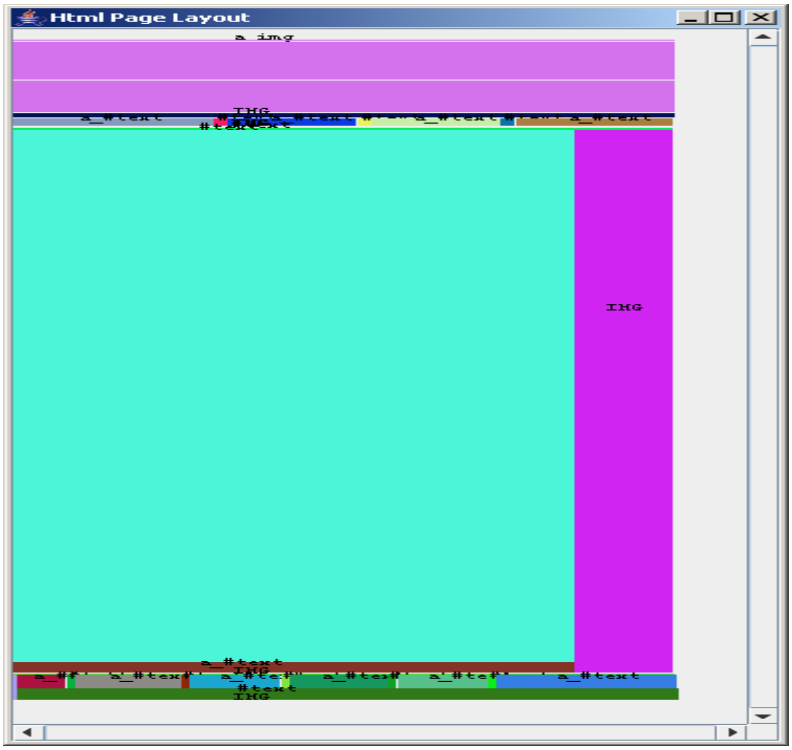
- Εξετάζουμε τη σελίδα www.e-city.gr/evrytania/home/view/1200.php.[12]

[12] Ίσχυε στις 18/10/2007.

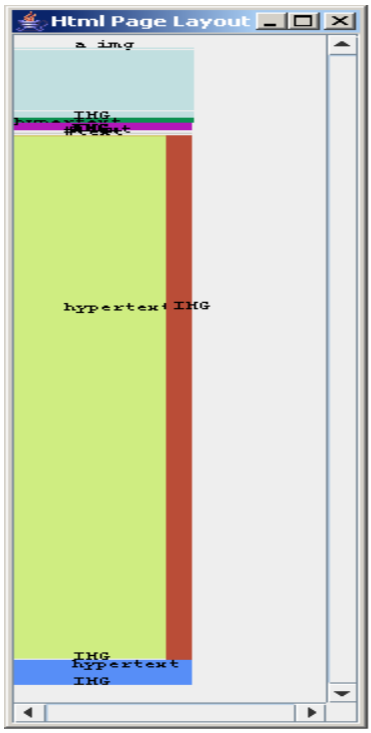


Εικόνα 1: Τουριστική σελίδα για το Καρπενήσι

Εκείνο που πρέπει να τονίσουμε για τη συγκεκριμένη σελίδα είναι ότι δεν ορίζεται το μέγεθος της εικόνας που περιέχει στο κέντρο. Παρόλα αυτά τα αποτέλεσμα που προκύπτουν κρίνονται λογικά.

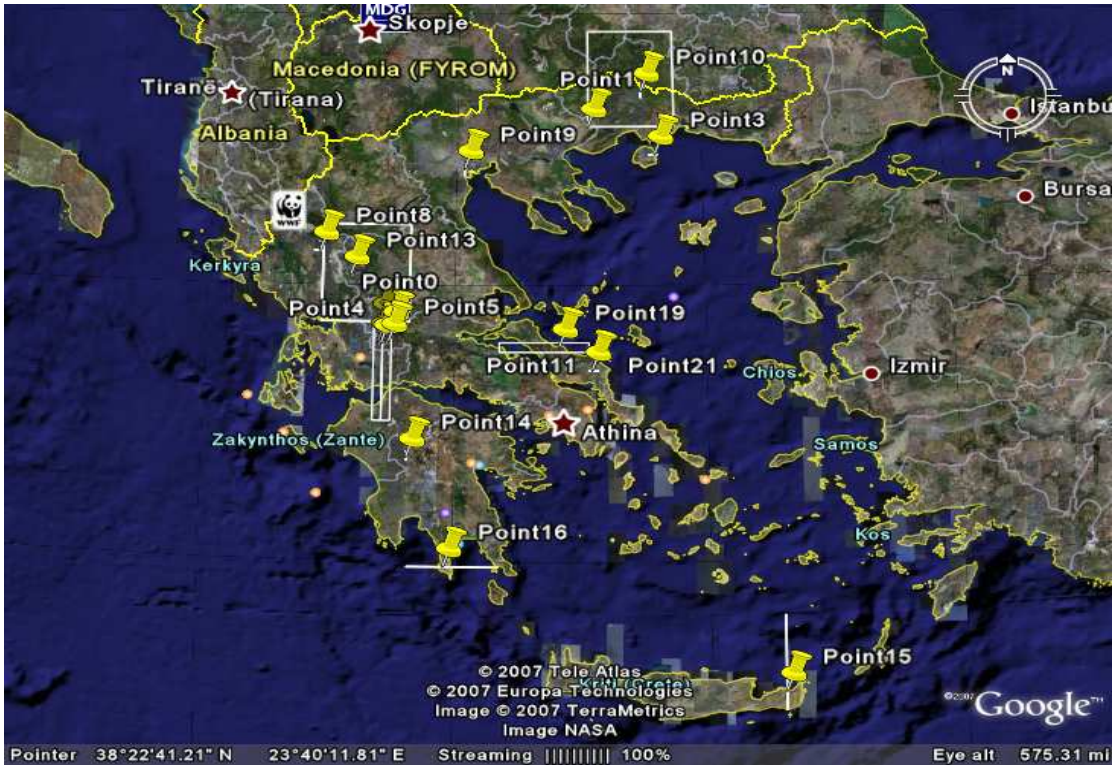


Εικόνα 2: Λεπτομερής ανάλυση σελίδας της Εικόνας 1



Εικόνα 3: Χονδρική ανάλυση σελίδας της Εικόνας 1

Τα γεωγραφικά σημεία που εντοπίζει ο geoparser είναι ίδια τόσο με ανάλυση της σελίδας όσο και χωρίς ανάλυση. Αυτά φαίνονται στο επόμενο σχήμα.



Εικόνα 4: Γεωγραφικά σημεία για τη σελίδα της εικόνας 1

Στην περίπτωση αυτή δεν παρατηρούμε καμία διαφορά επειδή τα γεωγραφικά σημεία βρίσκονται όλα τοποθετημένα στο κεντρικό κείμενο της σελίδας.

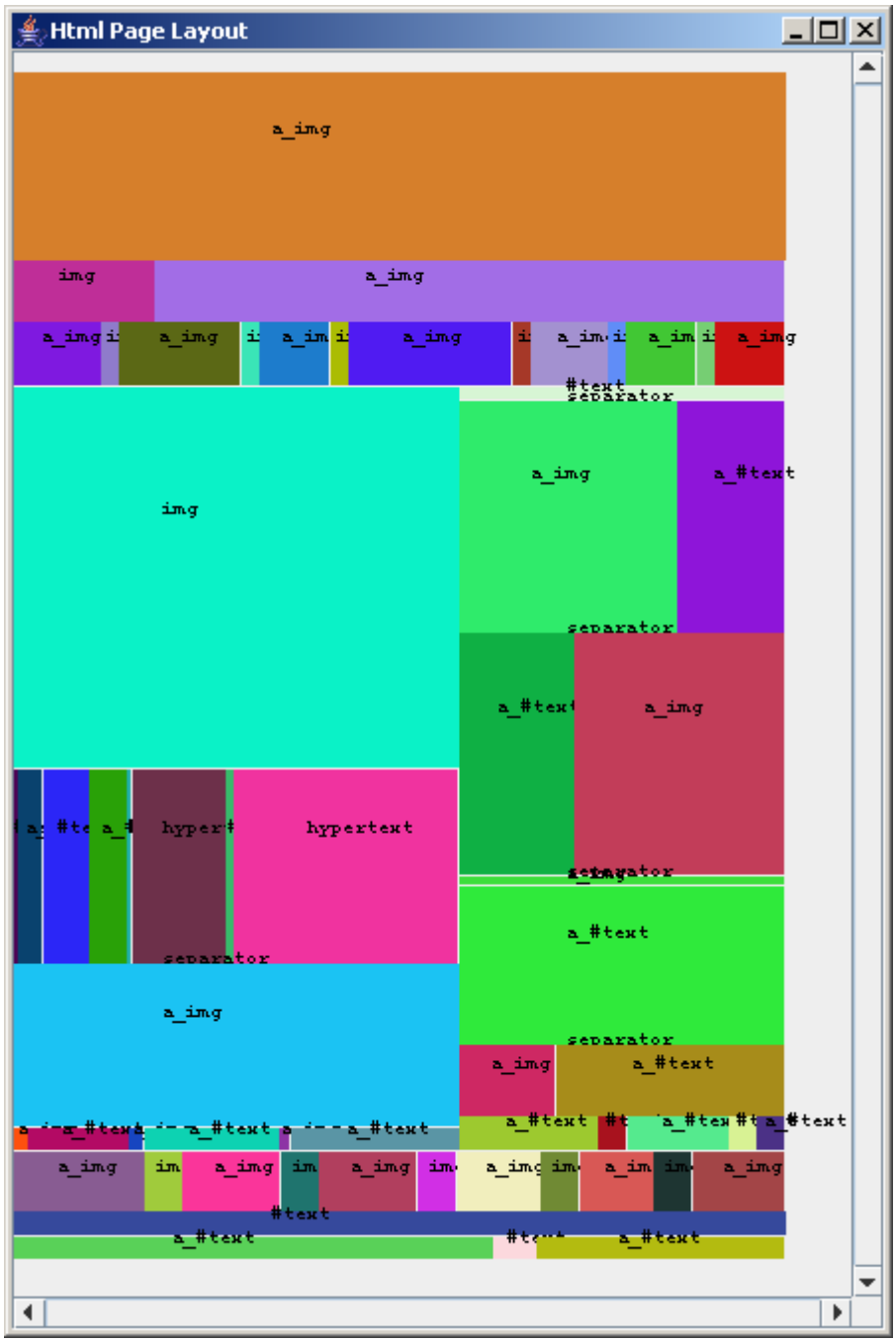
5.2.2.2 Εμπορικές Σελίδες

- Εξετάζουμε τη σελίδα www.harvard.edu.[13]

Εικόνα 5: Εμπορική σελίδα για το Harvard

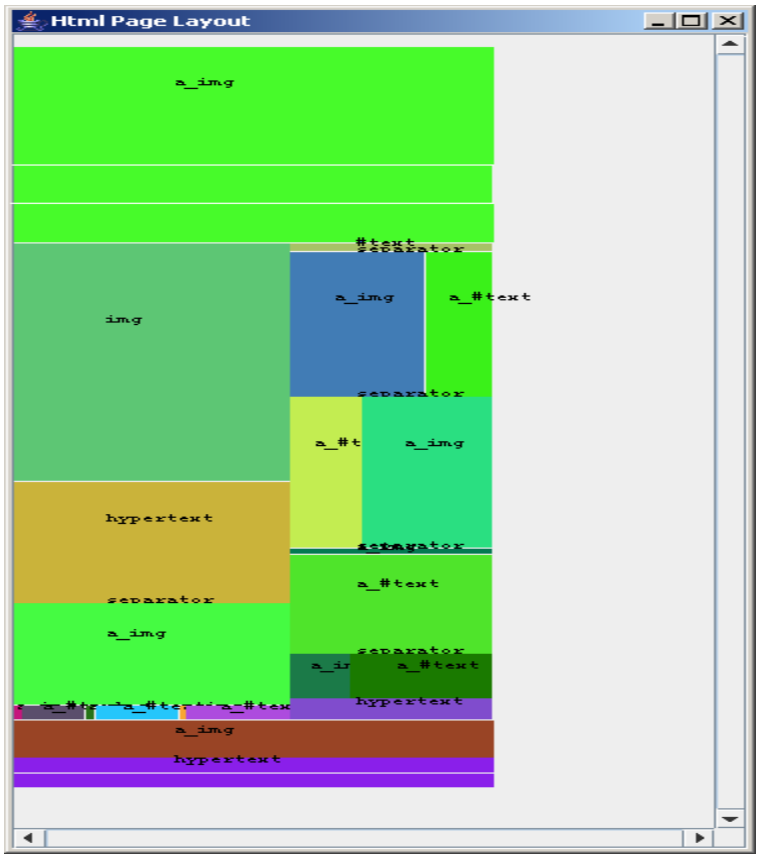
Επιδιώκοντας ένα λεπτομερές αποτέλεσμα παίρνουμε την επόμενη εικόνα.

[13] Ίσχυε στις 8/3/2007.



Εικόνα 6: Λεπτομερής ανάλυση σελίδας της Εικόνας 5

Ενώ αν θέλουμε να πετύχουμε μεγαλύτερο βαθμό ομαδοποίησης, αλλάζουμε τις τιμές των παραμέτρων και προκύπτει το ακόλουθο αποτέλεσμα.



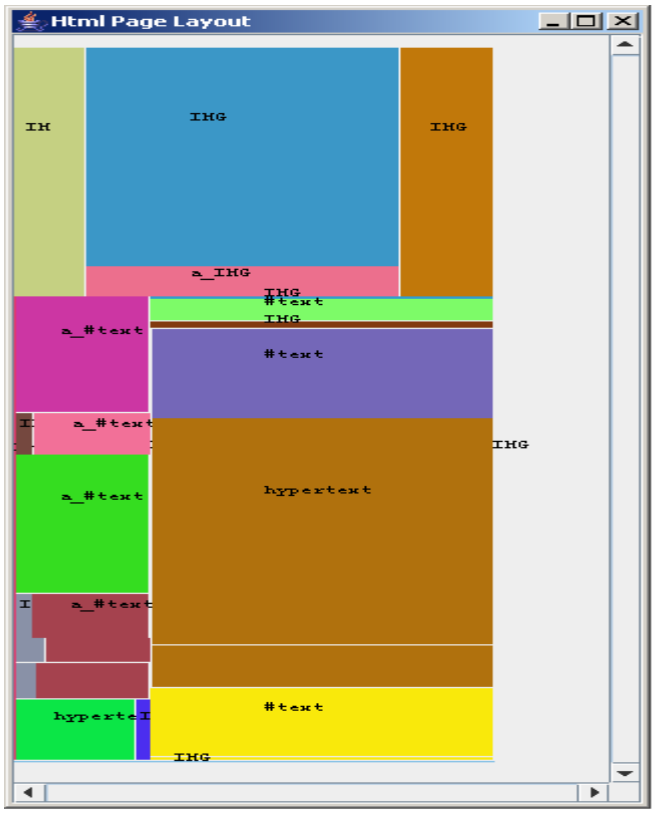
Εικόνα 7: Χονδρική ανάλυση σελίδας της Εικόνας 5

- Εξετάζουμε τη σελίδα www.alfacatering.gr[14].

Εικόνα 8: Εμπορική σελίδα για Catering

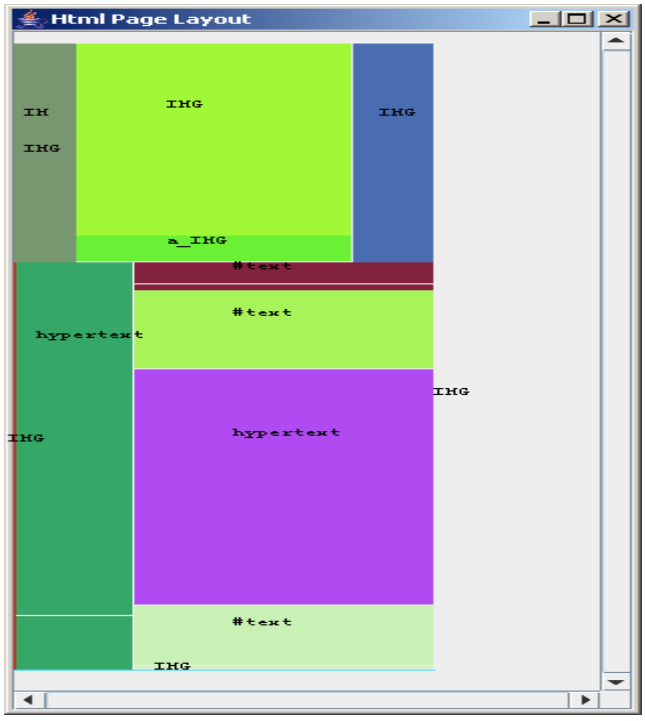
[14] Ίσχυε στις 12/10/2007.

Για λεπτομερή ανάλυση παίρνουμε:



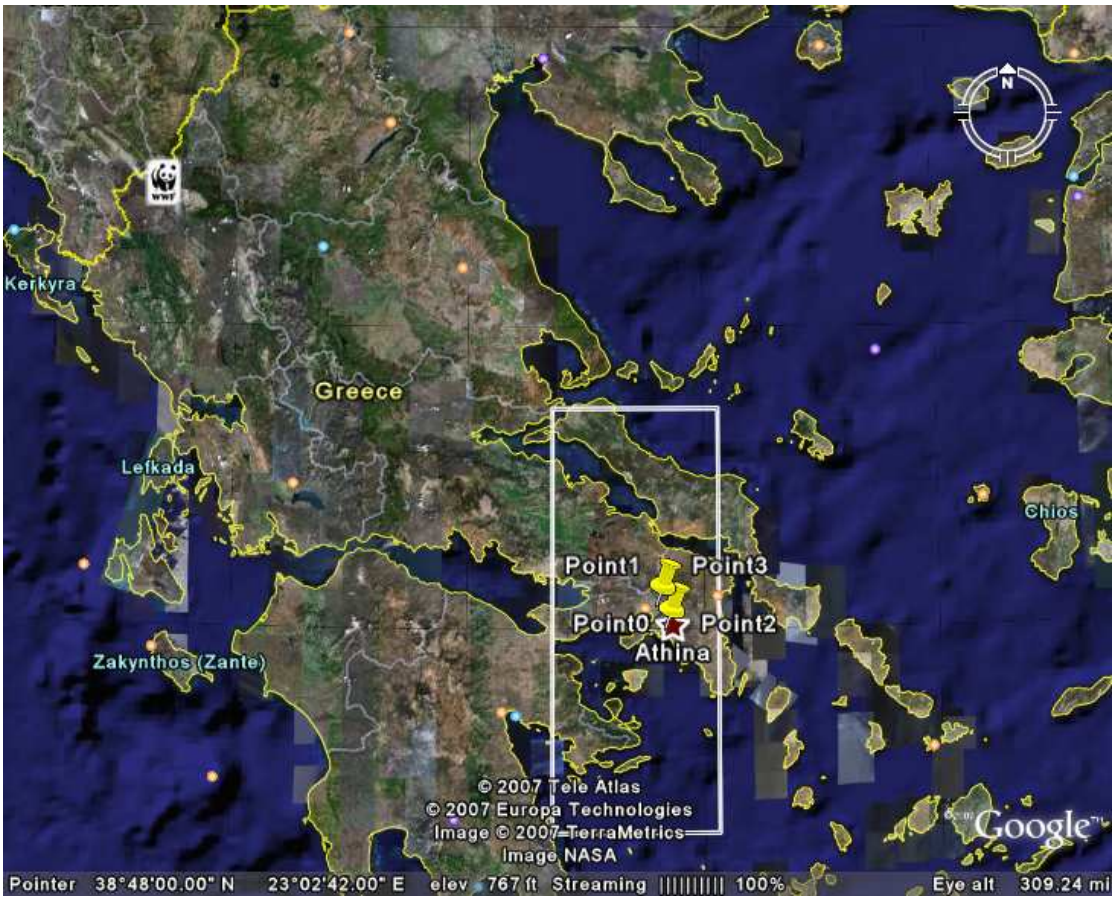
Εικόνα 9: Λεπτομερής ανάλυση σελίδας της Εικόνας 8

Για μεγαλύτερη ομαδοποίηση προκύπτει:



Εικόνα 10: Χονδρική ανάλυση σελίδας της Εικόνας 8

Και στην περίπτωση αυτή τα γεωγραφικά σημεία που εντοπίζει ο geoparser δεν διαφέρουν πριν και μετά την ανάλυση της ιστοσελίδας. Αυτά είναι:



Εικόνα 10: Γεωγραφικά σημεία για τη σελίδα της εικόνας 8

5.2.2.3 *Ειδησεογραφικές Σελίδες*

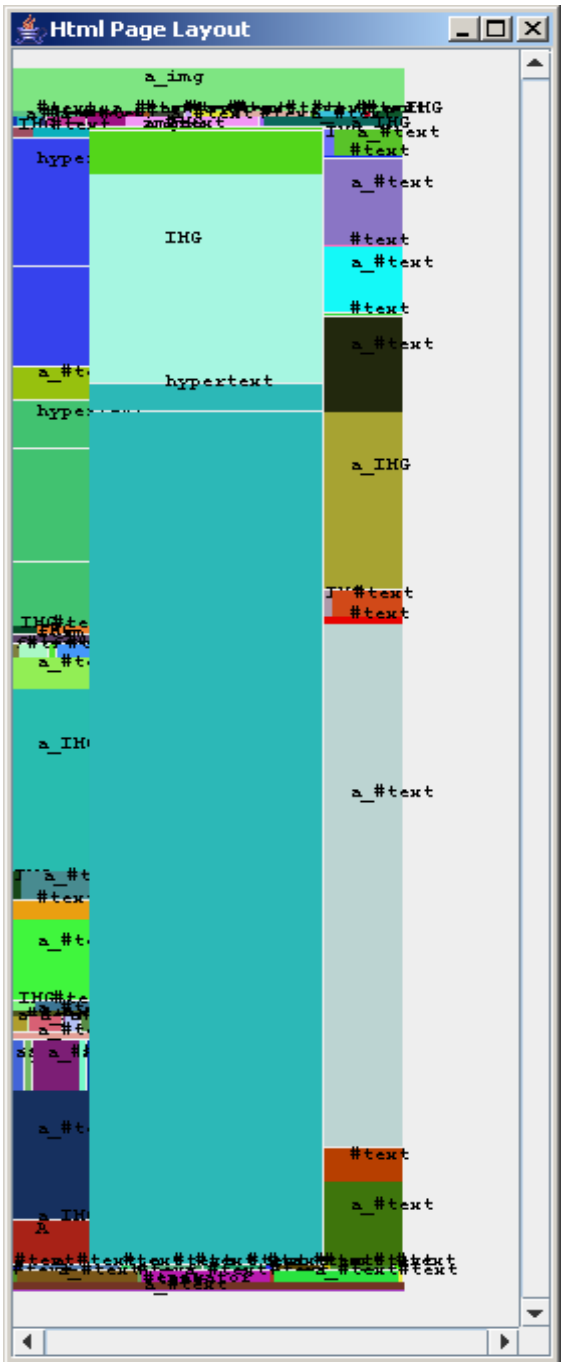
- Εξετάζουμε τη σελίδα `http://www.in.gr/news/article.asp?lngEntityID=839536&lngDtrID=245`[15]

[15] Ίσχυε στις 12/10/2007.

Εικόνα 11: Ειδησεογραφική σελίδα του in.gr

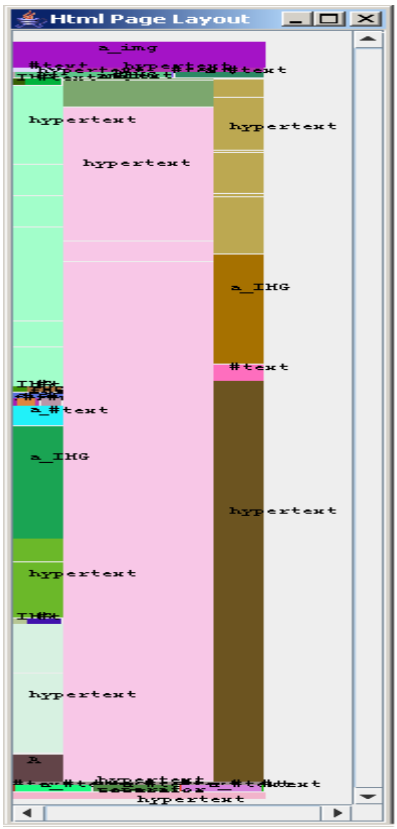
Παρατηρούμε ότι το κυρίως κείμενο του άρθρου βρίσκεται στο κέντρο της σελίδας και καταλαμβάνει το μεγαλύτερο μέρος της. Αναμένουμε αυτό να φαίνεται στο αποτέλεσμα ανεξαρτήτως του επιπέδου λεπτομέρειας που επιλέγουμε.

Για μεγάλη λεπτομέρεια παίρνουμε:



Εικόνα 12: Λεπτομερής ανάλυση σελίδας της Εικόνας 11

Ενώ για μικρότερη λεπτομέρεια προκύπτει:



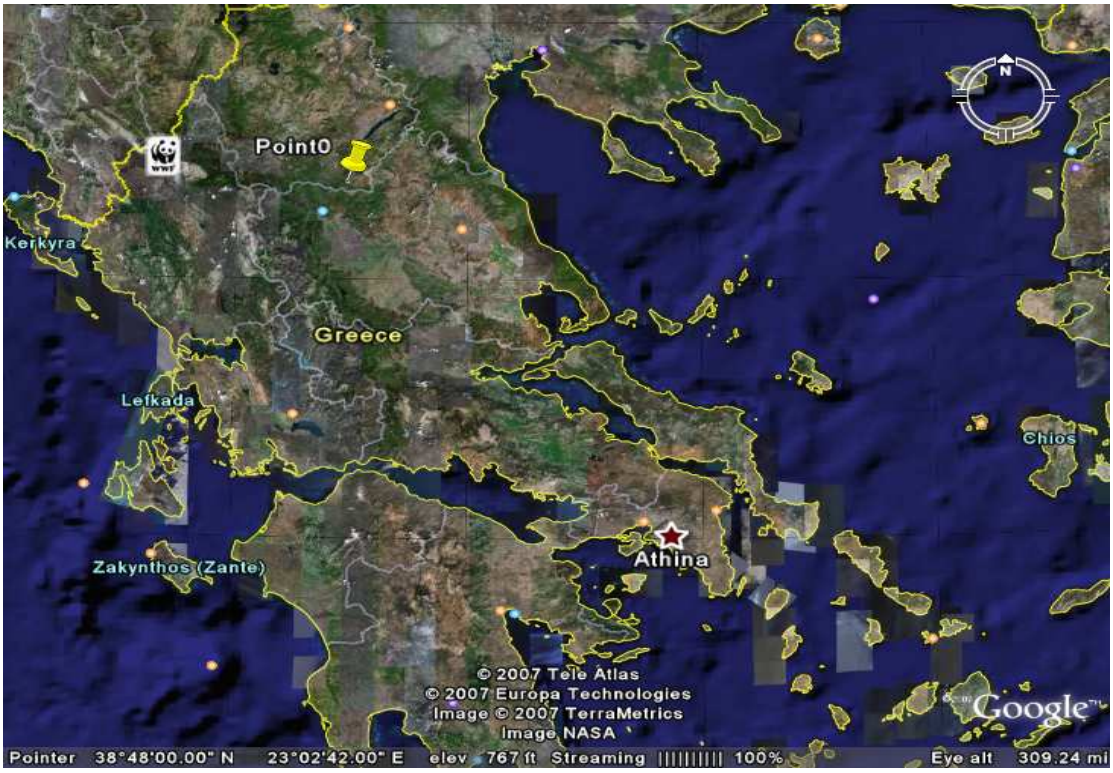
Εικόνα 13: Χονδρική ανάλυση σελίδας της Εικόνας 11

Επειδή η σελίδα αυτή είναι αρκετά μεγάλη και περιέχει πολλά στοιχεία, αυτά δεν είναι πολύ ευδιάκριτα στις προηγούμενες απεικονίσεις. Μπορούμε, ωστόσο, να τα εντοπίσουμε χάρη στους διαφορετικούς χρωματισμούς. Επίσης διαπιστώνουμε ότι αναγνωρίζεται το κεντρικό κείμενο τόσο ως προς τη θέση του όσο και ως προς το μέγεθός του.

Το αποτέλεσμα που δίνει ο geoparser χωρίς ανάλυση της ιστοσελίδας είναι το εξής:

Εικόνα 14: Γεωγραφικά σημεία για τη σελίδα της εικόνας 11

Παρατηρούμε ότι τα σημεία είναι διασκορπισμένα σε ολόκληρη σχεδόν την Ελλάδα. Ας δούμε το αποτέλεσμα που παίρνουμε μετά την ανάλυση της σελίδας.



Εικόνα 15: Γεωγραφικά σημεία για τη σελίδα της εικόνας 11

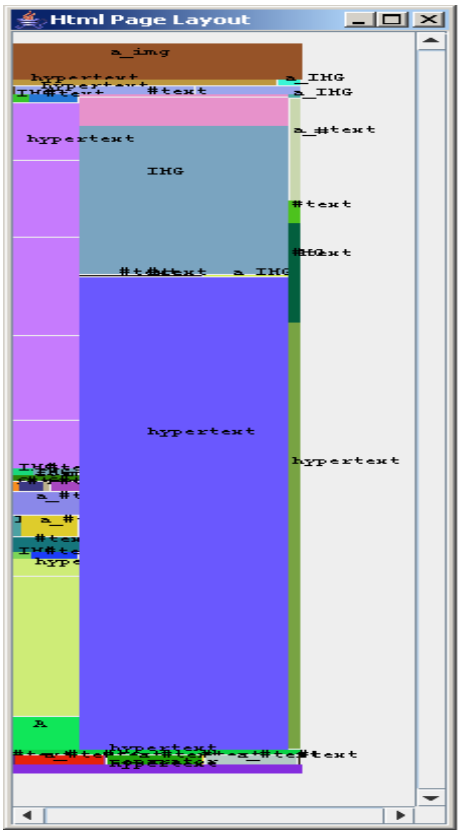
Είναι εμφανής η διαφορά στη δυο αποτελέσματα. Φαίνεται επίσης πόσο πολύ επηρεάζεται ο geoparser από διαφημιστικές πληροφορίες της σελίδας οι οποίες συνήθως δεν βρίσκονται στο κέντρο όπως το κυρίως κείμενο αλλά στις άκρες της σελίδας.

- Εξετάζουμε τη σελίδα http://www.in.gr/news/article.asp?lngEntityID=839555&lngDtrID=244[16]

[16] Ίσχυε την 23/10/2007

Εικόνα 16: Ειδησεογραφική σελίδα του in.gr

Το αποτέλεσμα που παίρνουμε από την τυπολογική και τοπολογική ανάλυση είναι το ακόλουθο:



Εικόνα 17: Αποτέλεσμα για τη σελίδα της Εικόνας 16

Τα σημεία που βρίσκει ο geoparser χωρίς την ανάλυση φαίνονται στην κάτωθι εικόνα.

Εικόνα 18: : Γεωγραφικά σημεία για τη σελίδα της εικόνας 16

Μετά την ανάλυση της σελίδας παίρνουμε το επόμενο αποτέλεσμα.

Εικόνα 19: : Γεωγραφικά σημεία για τη σελίδα της εικόνας 16

Παρατηρούμε ότι τα γεωγραφικά σημεία έχουν περιοριστεί. Το σημαντικότερο βέβαια δεν είναι αυτό αλλά το γεγονός ότι μπορούμε να βρούμε με μεγαλύτερη βεβαιότητα το γεωγραφικό χώρο που αναφέρεται το άρθρο της σελίδας από τη μεγαλύτερη πυκνότητα των σημείων στο χάρτη.

5.2.3 Μετρήσεις[17]

Στην προηγούμενη υπό-ενότητα είδαμε πως επηρεάζονται τα αποτελέσματα του geoparser από την τυπολογική και τοπολογική ανάλυση της ιστοσελίδας. Έχουμε αναφέρει όμως ότι εκτός από τα αποτελέσματα, η διαδικασία του geoparsing επηρεάζεται και ως προς τη διάρκεια εκτέλεσής της. Στον Πίνακα 1 καταγράφεται η χρονική διάρκεια εκτέλεσης για κάθε ένα από τα παραδείγματα της προηγούμενης υπό-ενότητας.

Παρατηρούμε ότι υπάρχει σημαντική μείωση στο χρόνο που απαιτείται για τον έλεγχο μιας σελίδας. Η διαφορά αυτή είναι λογική αν σκεφτούμε ότι ο geoparser έχει πλέον να εξετάσει μικρότερο αριθμό λέξεων και άρα να πραγματοποιήσει λιγότερες συναλλαγές με τη βάση. Το

[17] Οι μετρήσεις καταγράφηκαν σε προσωπικό υπολογιστή με επεξεργαστή στα 2,4 GHz και μνήμη 256 MB.

επιχείρημα αυτό είναι παραπλανητικό και μπορεί να δημιουργήσει λάθος εντυπώσεις καθώς θα μπορούσε να υποστηριχτεί ότι χάνονται σημαντικές πληροφορίες της σελίδας απορρίπτοντας τα άλλα τμήματά της. Κάτι τέτοιο όμως δεν ισχύσει αφού η διαδικασία τυπολογικής και τοπολογικής ανάλυσης εντοπίζει το σημαντικότερο στοιχείο. Επίσης σε περίπτωση αμφιβολίας θα μπορούσαν να εξεταστούν διάφορα στοιχεία της σελίδας και στη συνέχεια να συγκριθούν τα αποτελέσματα του geoparsing που προκύπτουν για του κάθε ένα ξεχωριστά.

Παράδειγμα	Χρόνος Geoparsing Σελίδας (sec)	Χρόνος Geoparsing Τμήματος (sec)	Χρόνος Ανάλυση Σελίδας (msec)
Εικόνα 1	16,313	13,966	313
Εικόνα 8	6,188	4,25	250
Εικόνα 11	14,125	6,469	328
Εικόνα 16	14,484	6,657	297

Πίνακας 1: Μετρήσεις χρονικής διάρκειας εκτέλεσης προγράμματος

5.3 Σύνοψη συμπερασμάτων αξιολόγησης

Όπως μπορούμε να παρατηρήσουμε από τα προηγούμενα πειράματα, το σύστημα δίνει ορθά αποτελέσματα. Σε ότι αφορά το σκέλος της τοπολογικής διάταξης και του σχετικού μεγέθους των στοιχείων της σελίδας, διαπιστώνουμε ότι η έξοδος ανταποκρίνεται στην πραγματική εικόνα της σελίδας και εκφράζει ικανοποιητικά τη σχέση μεγέθους των διάφορων στοιχείων. Το άλλο σκέλος, της ομαδοποίησης δηλαδή των κόμβων, συμπεραίνουμε ότι το σύστημα υλοποιεί με αρκετή επιτυχία τις απαιτήσεις του χρήστη για διαφορετικό βαθμό λεπτομέρειας στην ανάλυσή του. Βασιζόμενοι στην ορθότητα των προαναφερθέντων, εξετάσαμε την επίδραση του συστήματός μας στη διαδικασία του geoparsing. Από τα αποτελέσματα που πήραμε διαπιστώσαμε βελτίωση στην ακρίβεια που επιτυγχάνει ο geoparser καθώς και στο χρόνο που χρειάζεται για την ανάλυση μιας ιστοσελίδας.

6 *Επίλογος*

Στο κεφάλαιο αυτό θα συνοψίσουμε τους στόχους και τα αποτελέσματα της διπλωματικής καθώς θα παρουσιάσουμε και πιθανές μελλοντικές επεκτάσεις του συστήματος που αναπτύξαμε με στόχο τη βελτίωσή του.

6.1 Σύνοψη και συμπεράσματα

Όπως αναφέραμε και στην Εισαγωγή, στόχος της παρούσας διπλωματικής ήταν η βελτίωση της απόδοσης του συστήματος του geoparsing μέσω της ανάλυσης της τοπολογίας των σελίδων. Για το σκοπό αυτό αναπτύξαμε ένα σύστημα το οποίο λειτουργεί πάνω σε δυο άξονες, αμφότεροι δε στηρίζονται στην επεξεργασία του DOM Tree που αντιστοιχεί στο HTML αρχείο .

Ο πρώτος άξονας σχετίζεται με τη σημασιολογική διαχείριση της σελίδας. Συγκεκριμένα, τα δομικά στοιχεία της (κείμενα, εικόνες, ενσωματωμένα αντικείμενα κ.ά.) ελέγχονται και συγκρίνονται μεταξύ τους για πιθανή κατάταξη στην ίδια σημασιολογική ομάδα. Τα κριτήρια που χρησιμοποιήσαμε για τον έλεγχο της παραπάνω ομαδοποίησης ήταν αρκετά. Πρώτο και σημαντικότερο ήταν φυσικά ο τύπος του περιεχομένου των στοιχείων. Στη συνέχεια χρησιμοποιήσαμε οπτικούς κανόνες όπως η σχετική τους θέση, αν γειτονεύουν δηλαδή ή όχι, και το μέγεθός τους εκφρασμένο με τη τιμή του βάρους που εισάγαμε. Ένα τελευταίο κριτήριο ήταν η απόσταση των αντίστοιχων κόμβων τους στο DOM Tree. Πέρα όμως από τα προηγούμενα κριτήρια ομαδοποίησης, η όλη διαδικασία παραμετροποιήθηκε δίνοντας στον χρήστη τη δυνατότητα επηρεασμού του αποτελέσματος κατά βούληση.

Ο δεύτερος άξονας ενασχόλησής μας αφορούσε τη τοπολογική διάταξη της σελίδας. Εργαστήκαμε πάνω στην παραδοχή πως έχουμε αμιγώς HTML αρχεία, χωρίς δηλαδή να λάβουμε υπόψη επιπρόσθετα αρχεία μορφοποίησης όπως τα CSS. Επιπλέον αποκλείσαμε και αρχεία που η οργάνωσή τους στηριζόταν στη λογική των Frames. Τα προηγούμενα

περιορίζουν σαφώς την εμβέλεια του συστήματός μας αλλά εντούτοις μας επιτρέπουν να στηριχθούμε αποκλειστικά στις ιδιότητες των HTML tags για την κατάστρωση της τοπολογική διάταξης. Πράγματι, η χρήση τους ήταν επαρκής για να εξάγουμε τη διαμόρφωση της σελίδας και τον τρόπο που τα στοιχεία της γειτονεύουν. Καταφέραμε δηλαδή να ανακαλύψουμε τη δομή της σελίδας μέχρι κάποιο βαθμό λεπτομέρειας που, ωστόσο, ήταν αρκετός για την ικανοποίηση του αντικειμενικού μας στόχου, το αποτελεσματικότερο geoparsing.

6.2 Μελλοντικές επεκτάσεις

Αναφέραμε στην προηγούμενη ενότητα μερικούς περιορισμούς στους οποίους υπόκειται το σύστημα που αναπτύξαμε. Πέρα αυτών όμως υπάρχουν κάποιες βελτιώσεις που μπορούν να γίνουν για να ενισχύσουν την αξιοπιστία των αποτελεσμάτων αλλά και να επεκτείνουν τα πεδία εφαρμογής του συστήματος.

6.2.1 Χρήση Αρχείων Μορφοποίησης CSS

Είπαμε πως το σύστημα που αναπτύξαμε δεν επεξεργάζεται σελίδες που μορφοποιούνται από αρχεία CSS (Cascading Style Sheets) ή ακριβέστερα δεν δίνει ορθό αποτέλεσμα. Αυτό ισχύει μόνο στην περίπτωση που επηρεάζεται η τοπολογία της σελίδας και όχι μόνο τα οπτικά χαρακτηριστικά της. Στην περίπτωση που ισχύει μόνο το τελευταίο το αποτέλεσμα είναι σωστό. Σε κάθε περίπτωση όμως, αν λάβουμε υπόψη και το γεγονός ότι το μεγαλύτερο ποσοστό των ιστοσελίδων μορφοποιείται από αρχεία css, θα αποτελούσε σαφή βελτίωση του συστήματος ως προς την εμβέλεια εφαρμογής της αν περιελάμβανε τη δυνατότητα επεξεργασίας τους.

6.2.2 Οργάνωση Σελίδων με Frames

Η οργάνωση των σελίδων χρησιμοποιώντας frames είναι ένα συχνά συναντούμενο μοτίβο. Τα frames επεκτείνουν τις δυνατότητες της γλώσσας HTML δίνοντας περισσότερες επιλογές οπτικής παρουσίασης και οργάνωσης των στοιχείων μιας σελίδας. Εντούτοις, στο σύστημά μας δεν λαμβάνονται υπόψη εξαιτίας της διαφοροποίησής της συμπεριφοράς τους από τα υπόλοιπα HTML tags. Η προσθήκη τους στο σύστημα θα επέκτεινε το εύρος των σελίδων που μπορεί να χειριστεί.

6.2.3 Σημασιολογική Επεξεργασία

Η σημασιολογική επεξεργασία της σελίδας και των στοιχείων της δεν ήταν ο κύριος στόχος μας, παρόλα αυτά ασχοληθήκαμε μερικώς και με το θέμα αυτό γιατί συμβάλλει στο

αποδοτικότερο geoparsing. Το πεδίο αυτό είναι εξαιρετικά σημαντικό και πολλές μελέτες έχουν πραγματοποιηθεί για τη διερεύνησή του. Η ενσωμάτωση των αποτελεσμάτων τους στο αναπτυχθέν σύστημα αποτελεί μια πιθανή επέκτασή του. Αναφέρουμε χαρακτηριστικά ένα παράδειγμα. Η σημασιολογική ομαδοποίηση των στοιχείων της σελίδας θα έδινε βελτιωμένα αποτελέσματα αν επιπλέον χρησιμοποιούνταν και το κριτήριο της οπτικής μορφοποίησής τους.

6.2.4 Χρήση Μοτίβων

Είδαμε πως κάθε σελίδα έχει τα ιδιαίτερα χαρακτηριστικά της. Στο σύστημά μας επιλέξαμε σαν τρόπο ανακάλυψής τους την παραμετροποίηση της μεθοδολογίας και είδαμε ότι η μέθοδος αυτή αποδίδει. Θέλοντας να αυτοματοποιήσουμε σε κάποιο βαθμό τη διαδικασία, είναι εφικτός ο καθορισμός μοτίβων για σελίδες των διάφορων κατηγοριών. Παρατηρήσαμε ότι οι ειδησεογραφικές σελίδες ακολουθούν λίγο πολύ έναν συγκεκριμένο τρόπο οργάνωσης με το άρθρο να βρίσκεται στο κέντρο και να περιβάλλεται από σχετικούς ή όχι συνδέσμους και διαφημίσεις. Παρόμοια διαπίστωση μπορεί να γίνει και για τις σελίδες άλλων κατηγοριών. Μπορούμε επομένως να προτείνουμε την προσθήκη στο σύστημα μιας νέας διαδικασίας η οποία θα κάνει χρήση των διάφορων μοτίβων για την ομαδοποίηση των κόμβων δρώντας όμως επικουρικά στην ήδη υπάρχουσα διαδικασία.

7

ΠΑΡΑΡΤΗΜΑ Α:

Επιφάνεια Στοιχείων Κειμένου

Το μέγεθος των γραμμάτων σε εικονοστοιχεία, για κάθε γραμματοσειρά, είναι τυποποιημένο και δεδομένο. Επομένως, μπορούμε εύκολα να βρούμε τη συνολική επιφάνεια που καταλαμβάνει κάθε γράμμα ξεχωριστά χρησιμοποιώντας τους αντίστοιχους πίνακες. Ο επόμενος πίνακας δίνει τις τιμές αυτές για τις πιο κοινές γραμματοσειρές.

Μέγεθος Γραμματοσειράς	Μέγεθος σε pixels
1	10
2	13
3	16
4	20
5	24
6	32

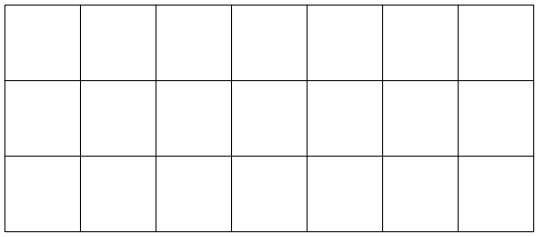
Πίνακας 1

Όταν όμως έχουμε κείμενο, δηλαδή πολλά γράμματα μαζί, δεν μπορούμε να πούμε ότι η συνολική του επιφάνεια ισούται με το άθροισμα της επιφάνειας όλων των γραμμάτων. Αυτό οφείλεται στο γεγονός ότι μεταξύ των γραμμάτων υπάρχουν κενά. Η κατάσταση δυσκολεύει ακόμη περισσότερο όταν έχουμε γράμματα σε παραπάνω από μια σειρές οπότε πρέπει να συνυπολογίσουμε και το κενό ενδιάμεσα των γραμμών. Λαμβάνοντας υπόψη αυτές τις

δυσκολίες καταφύγαμε στον καθορισμό νέων τιμών για τον προηγούμενο πίνακα όπου πλέον θα εντάξουμε και τον κενό χώρο που περιβάλλει το κάθε γράμμα.

Η μέθοδος που υιοθετήσαμε για την εύρεση των νέων επιφανειών ήταν τα δοκιμαστικά πειράματα. Συγκεκριμένα, έχοντας μια εικόνα γνωστών διαστάσεων, άρα γνωρίζουμε και την επιφάνειά της, τη συγκρίναμε με κείμενο στο οποίο μεταβάλλαμε το μέγεθος της γραμματοσειράς. Επειδή όμως ξέρουμε το πλήθος των γραμμάτων μπορούμε, διαιρώντας το με τη συνολική επιφάνεια να βρούμε την επιφάνεια που αντιστοιχεί σε ένα γράμμα. Προφανώς, το κείμενο αποτελείται από γράμματα του ίδιου τύπου (απλός ή bold) και του ίδιου μεγέθους. Ας δούμε όμως τη διαδικασία και οπτικά μέσω των επόμενων δυο εικόνων.

Εικόνα 1: Επιφάνεια εικόνας



Εικόνα 2: Επιφάνεια κειμένου

Στην Εικόνα 2 το κάθε κελί αντιπροσωπεύει ένα γράμμα και το περιβάλλον κενό, ενώ το ορθογώνια παριστάνει όλο το κείμενο. Όπως προκύπτει από το δεύτερο σχήμα, τα γράμματα είναι κατάλληλα στοιχισμένα και όλες οι σειρές έχουν τον ίδιο αριθμό γραμμάτων. Μετά από όλες αυτές τις υποθέσεις είναι πλέον ξεκάθαρο ότι εύκολα υπολογίζουμε την επιφάνεια του κάθε κελιού, άρα και ότι αυτό αντιπροσωπεύει.

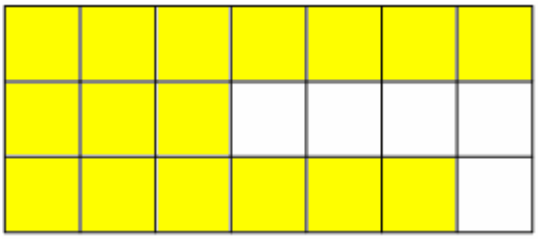
Εκτελέσαμε την παραπάνω διαδικασία για τις πιο κοινές γραμματοσειρές και καταγράψαμε τις νέες τιμές για την επιφάνειά τους, οι οποίες φαίνονται στον κάτωθι πίνακα.

Μέγεθος Γραμματοσειράς	Μέγεθος σε pixels
1	62
2	107
3	103
4	140
5	240
6	440
7	800

Πίνακας 2

Παρατηρούμε ότι υπάρχει μεγάλη διαφορά μεταξύ των τιμών του Πίνακα 1 και του Πίνακα 2 γεγονός που σημαίνει πως κάθε γράμμα συνοδεύεται από υπολογίσιμο κενό χώρο γύρω του.

Ένα τελευταίο σημείο που χρήζει διευκρίνισης είναι ο τρόπος τοποθέτησης των γραμμάτων, δηλαδή σε ισομήκεις γραμμές. Αυτό έγινε απλά για ευκολία σύγκρισης των επιφανειών της εικόνας και του κειμένου. Με τις νέες τιμές επιφάνειας για κάθε γράμμα και στην ίδια σειρά να τα τοποθετήσουμε πάλι την ίδια συνολική επιφάνεια θα έχουμε. Δεν παίζει ρόλο, με άλλα λόγια, ο τρόπος διάταξης των γραμμάτων. Ας δούμε όμως καλύτερο το επόμενο παράδειγμα.



Εικόνα 3

Στην περίπτωση αυτή η συνολική επιφάνεια του κειμένου ισούται με το άθροισμα των επιφανειών μόνο των κίτρινων κελιών, ενώ τα άσπρα δεν λαμβάνονται υπόψη. Τελικά μπορούμε πλέον να πούμε ότι η επιφάνεια ενός κειμένου προκύπτει από το γινόμενο του πλήθους των γραμμάτων που το αποτελούν επί τη νέα επιφάνεια που υπολογίσαμε για κάθε γράμμα και η οποία φαίνεται στον Πίνακα 2.

8

Βιβλιογραφία και Αναφορές

Anj06 Αλβέρτος-Δαυίδ Α. Άντζελ, Διπλωματική Εργασία: Εξαγωγή Γεωγραφική Πληροφορίας από Ημιδομημένο Κείμενο, Εθνικό Μετσόβιο Πολυτεχνείο, 2006

CYZ03 Yu Chen, Wei-Ying Ma, Hong-Jiang Zhang, Detecting web page structure for adaptive viewing on small form factor devices, Proceedings of the 12th international conference on World Wide Web, p. 225-233, May 20-24, 2003, Budapest, Hungary

GKN+03 Suhit Gupta , Gail Kaiser , David Neistadt , Peter Grimm, DOM-based content extraction of HTML documents, Proceedings of the 12th international conference on World Wide Web, May 20-24, 2003, Budapest, Hungary

LH02 Shian-Hua Lin , Jan-Ming Ho, Discovering informative content blocks from Web documents, Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining, July 23-26, 2002, Edmonton, Alberta, Canada

SLW+04 Ruihua Song , Haifeng Liu , Ji-Rong Wen , Wei-Ying Ma, Learning important models for web page blocks based on layout and content analysis, ACM SIGKDD Explorations Newsletter, v.6 n.2, p.14-23, December 2004

XMS+05 Xing Xie , Gengxin Miao , Ruihua Song , Ji-Rong Wen , Wei-Ying Ma, Efficient Browsing of Web Search Results on Mobile Devices Based on Block Importance Model, Proceedings of the Third IEEE International

Conference on Pervasive Computing and Communications, p.17-26, March 08-12, 2005

ZLT06 Jie Zou, Daniel Le, George R. Thoma, Combining DOM tree and geometric layout analysis for online medical journal article segmentation, Proceedings of the 6th ACM/IEEE-CS joint conference on Digital libraries, p. 119-128, June 11-15, 2006, Chapel Hill, NC, USA

http://html.xamjwg.org/index.jsp Από τη συγκεκριμένη ιστοσελίδα χρησιμοποιήσαμε τον parser cobra-0.96.4 ίσχυε την 12/07/2007.