



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΕΦΑΡΜΟΣΜΕΝΩΝ ΜΑΘΗΜΑΤΙΚΩΝ ΚΑΙ ΦΥΣΙΚΩΝ ΕΠΙΣΤΗΜΩΝ

Τεχνολογίες Οπτικοποίησης Ιστοριογραμμών και Γραφημάτων Παραλλαγών

Visualization Technologies for Storylines and Variant Graphs

Διπλωματική Εργασία του
Αθανάσιου Γεωργίου

Αθήνα, 8 Φεβρουαρίου 2017



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΕΦΑΡΜΟΣΜΕΝΩΝ ΜΑΘΗΜΑΤΙΚΩΝ ΚΑΙ ΦΥΣΙΚΩΝ ΕΠΙΣΤΗΜΩΝ

Τεχνολογίες Οπτικοποίησης Ιστοριογραμμών και Γραφημάτων Παραλλαγών

Visualization Technologies for Storylines and Variant Graphs

Διπλωματική Εργασία του
Αθανάσιου Γεωργίου

Επιβλέπων:

Αντώνιος Συμβώνης
Καθηγητής Ε.Μ.Π.

Τριμελής Επιτροπή

Α. Συμβώνης	Π. Στεφανέας	Ι. Κολέτσος
Καθηγητής	Επ. Καθηγητής	Επ. Καθηγητής
Ε.Μ.Π.	Ε.Μ.Π.	Ε.Μ.Π.

Αθήνα, 8 Φεβρουαρίου 2017

Γεωργίου Αθανάσιος

Σχολή Εφαρμοσμένων Μαθηματικών και Φυσικών Επιστημών

Copyright©Γεωργίου Αθανάσιος, 2016.

Με επιφύλαξη παντός δικαιώματος. All rights reserved.

Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας, εξ ολοκλήρου ή τμήματος αυτής για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, εκπαίδευσης ή ερευνητικής φύσης, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να διατηρείται η παρούσα σημείωση. Ερωτήματα που αφορούν τη χρήση της εργασίας για κερδοσκοπικό σκοπό πρέπει να απευθύνονται προς τον συγγραφέα. Οι απόψεις και τα συμπεράσματα που περιέχονται σε αυτή τη διατριβή εκφράζουν τον συγγραφέα και δεν πρέπει να θεωρηθεί ότι αντιπροσωπεύουν τις επίσημες θέσεις του Εθνικού Μετσόβιου Πολυτεχνείου.

Στην οικογένειά μου.

Περίληψη

Η παρούσα διπλωματική εργασία πραγματεύεται τις τεχνολογικές μεθόδους για την οπτικοποίηση ιστοριογραμμών ή χρονολογικών γραφημάτων (storyline graphs) και των γραφημάτων παραλλαγών (variant graphs). Η ακριβέστερη περιγραφή των μεθόδων αυτών γίνεται με την αντίστοιχη ανάλυση των εφαρμογών τους σε επιστημονικούς τομείς και κυρίως σε αυτόν των Ανθρωπιστικών Επιστημών, δίνοντας έμφαση στη βιβλιοθήκη της JavaScript, Raphaël. Δεν πρέπει όμως να λησμονούμε ότι υπάρχουν και ορισμένοι περιορισμοί αυτών των τύπων γραφημάτων, τους οποίους κάθε επίδοξος χρήστης-ερευνητής οφείλει να λαμβάνει υπ' όψιν ώστε να διευκολύνεται στο έργο του. Παρόλα αυτά, δεν είναι διόλου απίθανο να επεκταθούν στο μέλλον οι χρήσεις των προαναφερθέντων γραφημάτων σε περαιτέρω εφαρμογές και άλλων επιστημονικών τομέων, από το χώρο του Διαδικτύου και των μέσων κοινωνικής δικτύωσης, έως την Ιατρική και τη βιολογία, ακόμα και την γενική και ειδική εκπαίδευση.

Abstract

The present thesis is about the visualization technologies for storyline and variant graphs, with emphasis on how Raphaël library is used in the analysis of data that come from sciences, especially in Humanities. Despite their positive aspects, it is necessary to be mentioned that these types of graphs could also show some kind of restrictions that should be definitely taken into account. However, these restrictions do not imply that they cannot be used in the near future in further scientific areas, such as social media, medicine and biology, even education for children with special needs.

Ευχαριστίες

Αρχικά θέλω να ευχαριστήσω τον καθηγητή μου, κ. Αντώνιο Συμβώνη, για την ευκαιρία που μου έδωσε αναθέτοντάς μου την παρούσα διπλωματική εργασία και την διδάκτορα, Χρυσάνθη Ραφτοπούλου, για τις συμβουλές της. Επίσης ευχαριστώ τα υπόλοιπα μέλη της τριμελούς επιτροπής για το χρόνο που διέθεσαν. Τέλος, θέλω να ευχαριστήσω τους δικούς μου ανθρώπους, την οικογένειά μου, για την υπομονή, τη συμπαράσταση και τις συμβουλές τους όλα αυτά τα χρόνια και τη Μαρίτα για την πολύτιμη βοήθεια και στήριξή της.

Περιεχόμενα

1	Εισαγωγή	15
1.1	Σκοπός της Διπλωματικής Εργασίας	16
1.2	Δομή της Εργασίας	16
2	Θεωρία Γραφημάτων	19
2.1	Είδη Γραφημάτων	19
2.2	Ειδικές Μορφές Γραφημάτων	25
2.3	Γραφοθεωρητικές δομές	26
2.4	Σχεδίαση Γραφημάτων	27
2.5	Γραφοθεωρητικά Προβλήματα	27
2.6	Εφαρμογές της Θεωρίας Γραφημάτων	28
3	Διαδικτυακές Τεχνολογίες Οπτικοποίησης	31
3.1	Η Γλώσσα Προγραμματισμού JavaScript	31
3.2	Η Βιβλιοθήκη JQuery	33
3.3	Η Βιβλιοθήκη AngularJS	34
3.4	Η Βιβλιοθήκη Raphaël	36
4	Ιστοριογραμμές	43
4.1	Ορισμός	43

4.2	Εφαρμογές	47
4.3	StorylineViewer	48
5	Γραφήματα Παραλλαγών	53
5.1	Ορισμός	53
5.2	Εφαρμογές	54
6	Μελλοντικές Εφαρμογές	57
7	Συμπεράσματα	59
A'	Κώδικας Εφαρμογής StorylineViewer	61
	Βιβλιογραφία	66

Κεφάλαιο 1

Εισαγωγή

Στη εποχή της ραγδαίας τεχνολογικής ανάπτυξης, τα γραφήματα και η θεωρία που τα συνοδεύει αναδεικνύονται σε ιδιαιτέρως χρήσιμα εργαλεία μοντελοποίησης και μελέτης προβλημάτων που αφορούν κάθε τομέα επιστήμης. Το γεγονός αυτό οδήγησε και στην διεύρυνση της θεωρίας γραφημάτων σε πολύ υψηλά επίπεδα γνώσης. Ειδικά τις τελευταίες δεκαετίες, αυξήθηκαν οι απαιτήσεις για μεθόδους οπτικοποίησης δεδομένων οι οποίες οδηγούν σε πιο εύκολη και συστηματική μελέτη των δεδομένων αυτών. Πλέον, υπάρχει μια πληθώρα γραφοθεωρητικών αλγορίθμων που ως στόχο τους έχουν την βέλτιστη δυνατή απεικόνιση γραφημάτων με τους όποιους περιορισμούς θέτει το εκάστοτε, υπό μελέτη, πρόβλημα.

Με την ανάπτυξη της τεχνολογίας του διαδικτύου διευκολύνεται η δημιουργία διαδικτυακών εφαρμογών και διευρύνονται οι δυνατότητες τους έτσι ώστε στις μέρες μας να αποτελούν ένα αναπόσπαστο κομμάτι των εργασιών, της μελέτης, αλλά και της ίδιας της ζωής μας. Πολλές φορές, κρίνεται απαραίτητο να υπάρχει η δυνατότητα επικοινωνίας των εφαρμογών με το διαδίκτυο, ώστε να λαμβάνουμε πιο αξιόπιστα αποτελέσματα σε μικρότερο χρόνο, ενώ παράλληλα πολλοί χρήστες να έχουν τη δυνατότητα να εργάζονται σε μία κοινή εργασία βλέποντας σε ζωντανό χρόνο τις διάφορες αλλαγές ή αποτελέσματα. Η κατάσταση αυτή δεν διαφοροποιείται ιδιαίτερα και στην περίπτωση των εφαρμογών απεικόνισης γραφημάτων. Έτσι η υλοποίηση των αλγορίθμων οπτικοποίησης γραφημάτων με τη μορφή διαδικτυακών εφαρμογών είναι ένας βασικός τομέας έρευνας και ενασχόλησης.

1.1 Σκοπός της Διπλωματικής Εργασίας

Η παρούσα εργασία πραγματεύεται τη μελέτη δύο ειδών γραφημάτων με πολλές εφαρμογές στις Ανθρωπιστικές Επιστήμες, τις Ιστοριογραμμές ή αλλιώς Χρονολογικά Γραφήματα (*Storyline Graphs*) και τα Γραφήματα Παραλλαγών (*Variant Graphs*) και τις διαθέσιμες τεχνολογίες οπτικοποίησής τους. Κάθε ένα από αυτά τα γραφήματα έχει ως στόχο να βελτιώσει την προβολή των δεδομένων που προέρχονται από την ανάλυση κειμένων, με τρόπο τέτοιο ώστε να δώσει μία πληρέστερη εικόνα στον ερευνητή σχετικά με το προς εξέταση αντικείμενο. Όμως τα παραπάνω γραφήματα υπόκεινται σε κανόνες οι οποίοι περιορίζουν τη δράση τους. Έτσι, για παράδειγμα, δεν μπορεί να χρησιμοποιηθεί ένα χρονολογικό γράφημα στη θέση ενός γραφήματος παραλλαγών ή το αντίστροφο. Γίνεται αντιληπτό, λοιπόν, ότι δεν πρέπει να γίνεται σύγχυση των εννοιών τους, αντιθέτως είναι απαραίτητη η διάκριση μεταξύ τους.

1.2 Δομή της Εργασίας

Στο δεύτερο κεφάλαιο γίνεται μια σύντομη εισαγωγή στις βασικές έννοιες της θεωρίας γραφημάτων και πώς αυτές χρησιμοποιούνται για την ανάλυση και υλοποίηση διαφόρων μεθόδων οπτικοποίησης.

Στο τρίτο κεφάλαιο περιγράφεται πώς η γλώσσα προγραμματισμού *JavaScript* μπορεί να αποτελέσει ένα πολύ χρήσιμο εργαλείο για την οπτικοποίηση ιστοριογραμμών και γραφημάτων παραλλαγών, με έμφαση σε μία από τις βιβλιοθήκες της που θα μπορούσε να φέρει εις πέρας το έργο αυτό με μεγαλύτερη επιτυχία, τη βιβλιοθήκη *Raphaël*. Επιπλέον δίνονται πληροφορίες και για δύο άλλες, πολύ χρήσιμες, βιβλιοθήκες της γλώσσας αυτής: *JQuery* και *AngularJS*.

Στο τέταρτο κεφάλαιο δίνεται αρχικά ο πλήρης ορισμός των ιστοριογραμμών και ακολούθως αναλύονται τα θετικά τους στοιχεία και οι εφαρμογές που βρίσκουν τα γραφήματα αυτά στην ερευνητική διαδικασία. Επιπλέον παρουσιάζεται η εφαρμογή *Storyline-Viewer* η οποία υλοποιήθηκε στα πλαίσια αυτής της εργασίας για την προβολή γραφημάτων ιστοριογραμμών.

Στο πέμπτο κεφάλαιο, αφού δοθεί ο ορισμός των γραφημάτων παραλλαγών, γίνεται εκτενής αναφορά στις εφαρμογές τους στις Ανθρωπιστικές Επιστήμες και ειδικότερα στη σύγκριση κειμένων ή παραλλαγών ενός γραφήματος, με ιδιαίτερη μνεία στη συμβολή του λογισμικού *CollateX* και της εφαρμογής *TRAViz*.

Στο έκτο κεφάλαιο αναφέρονται μελλοντικές εφαρμογές που εκμεταλλεύονται τις τεχνολογίες οπτικοποίησης με σκοπό να βελτιώσουν τις μεθόδους έρευνας σε διάφορους τομείς της επιστήμης.

Στο έβδομο κεφάλαιο καταγράφονται τα συμπεράσματα της παρούσας διπλωματικής και στο παράρτημα Α' δίνεται ο πλήρης κώδικας της εφαρμογής *StorylineViewer*.

Κεφάλαιο 2

Θεωρία Γραφημάτων

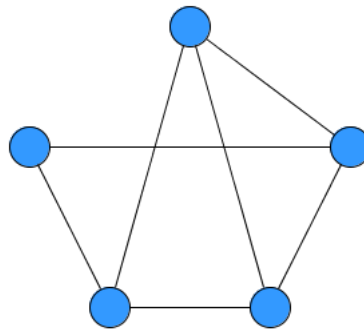
Με τον όρο «Θεωρία Γραφημάτων» εννοούμε τη μελέτη των γραφημάτων, των μαθηματικών κατασκευών που παρουσιάζουν μεμονωμένα ή πολλαπλά ζεύγη δεδομένων και τις σχέσεις που αναπτύσσονται μεταξύ τους, οπτικοποιώντας και συγκρίνοντάς τις. Αποτελεί βασική μέθοδο έρευνας στον τομέα των μαθηματικών, η οποία, χρησιμοποιώντας κάθετες και οριζόντιες γραμμές, τόξα, σημεία, κορυφές και κόμβους, αναπαριστά τις προαναφερθείσες σχέσεις μεταξύ διαφόρων δεδομένων. Πιο αναλυτικά, κάθε δεδομένο αναπαρίσταται με σημεία ή σχήματα, τα οποία ενώνονται μεταξύ τους με γραμμές ή καμπύλες, σχηματίζοντας έτσι κόμβους και ακμές. Το τελικό σχήμα που προκύπτει από την παραπάνω απεικόνιση δίνει πολύτιμα στοιχεία στον ερευνητή για τις σχέσεις των υπό μελέτη αντικειμένων.

2.1 Είδη Γραφημάτων

Υπάρχουν πολλοί τρόποι απεικόνισης των σχέσεων ανάμεσα στα δεδομένα και συνεπώς υπάρχουν πολλά είδη γραφημάτων. Κατά τον μαθηματικό ορισμό, γράφημα ονομάζεται ένα διατεταγμένο ζεύγος συνόλων $G = (V, E)$ έτσι ώστε το V να είναι το σύνολο των κόμβων του γραφήματος και το E το σύνολο των ακμών του [1]. Γενικά ο ορισμός αυτός γενικεύεται με ποικίλους τρόπους, κυρίως όσων αφορούν το σύνολο των ακμών του γραφήματος αλλά και τους κανόνες απεικόνισης κόμβων και ακμών, και έτσι λαμβάνουμε τα διάφορα είδη των γραφημάτων. Παρακάτω θα περιγραφούν τα βασικότερα από αυτά, εκείνα δηλαδή που συναντάμε πιο συχνά στα μαθηματικά και τις επιστήμες και που έχουν τις περισσότερες εφαρμογές.

Απλά, μη κατευθυνόμενα γραφήματα

Σύμφωνα με τον γενικό ορισμό μπορούμε να θεωρήσουμε το σύνολο των ακμών E ως ένα σύνολο δυάδων του συνόλου των κόμβων V . Εδώ, με τον όρο δυάδα, εννοούμε ένα σύνολο δύο στοιχείων του V που δεν μας ενδιαφέρει η σειρά με την οποία εμφανίζονται. Δηλαδή δεν είναι απαραίτητο να καθορίσουμε ποιος κόμβος είναι η αρχή και ποιος το πέρας της ακμής, αλλά ποιοι κόμβοι ενώνονται με ακμή στο γράφημα. Αυτά τα γραφήματα ονομάζονται απλά, μη κατευθυνόμενα (simple, undirected graphs) [1]. Σε αυτή την περίπτωση, συνηθίζεται να απεικονίζουμε τους κόμβους ως σημεία του επιπέδου ή με κάποιο γεωμετρικό σχήμα (συνήθως κύκλο) και τις ακμές ως ευθύγραμμα τμήματα με άκρα κόμβους που ενώνονται στο γράφημα.

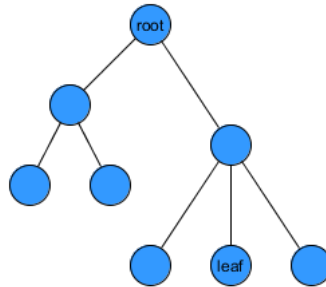


Σχήμα 2.1: Απλό, μη κατευθυνόμενο γράφημα.

Γραφήματα δένδρα

Μια πολύ χρήσιμη, από άποψη εφαρμογών, υποκατηγορία των απλών, μη κατευθυνόμενων γραφημάτων, είναι και τα γραφήματα-δένδρα ή απλά δένδρα (trees). Ενα γράφημα ονομάζεται δένδρο αν είναι απλό και δεν περιέχει κύκλους [1]. Τα γραφήματα αυτά πήραν το όνομά τους από τη μορφή των απεικονίσεών τους που θυμίζουν δέντρα ή θάμνους λόγω των διακλαδώσεων που σχηματίζουν οι ακμές τους. Κάθε δένδρο διαθέτει κόμβους από τους οποίους διέρχεται μόνο μία ακμή. Αυτοί οι κόμβοι ονομάζονται φύλλα (leaves) ενώ όλοι οι υπόλοιποι ονομάζονται εσωτερικοί κόμβοι (inner vertices) ή κόμβοι κλαδιά (branch vertices) του δένδρου. Επίσης κάθε κόμβος που δεν είναι φύλλο μπορεί να επιλεγθεί ως ρίζα (root) του δένδρου. Η ρίζα συνήθως παίζει το ρόλο του κόμβου-δείκτη σε βασικούς αλγορίθμους αναζήτησης και ταξινόμησης. Η απεικόνιση των δένδρων γίνεται όπως και στα απλά, μη κατευθυνόμενα γραφήματα με τον περιορισμό η ρίζα κάθε υποδένδρου (ο κόμβος του υποδένδρου που απέχει την ελάχιστη από-

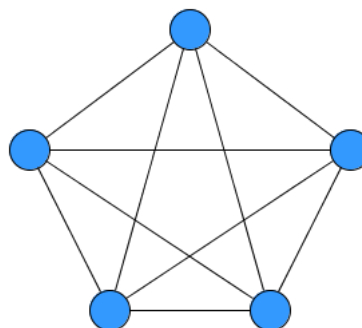
σταση από την ρίζα του κυρίως δένδρου, σε σύγκριση με τους υπόλοιπους κόμβους του υποδένδρου) να είναι πιο ψηλά (σε μεγαλύτερη γ -συντεταγμένη) από όλους τους άλλους κόμβους του.



Σχήμα 2.2: Γράφημα δένδρο.

Πλήρη γραφήματα

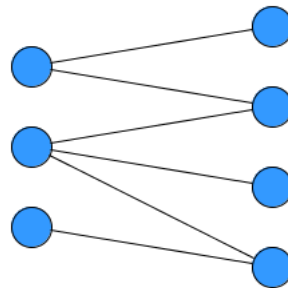
Πλήρη (complete) ή κλίκες (cliques) ονομάζονται τα απλά, μη κατευθυνόμενα γραφήματα που κάθε δύο κόμβοι του ενώνονται με ακμή [1]. Μία σημαντική ιδιότητά τους είναι ότι αποκόπτοντας κάποιον κόμβο από ένα πλήρες γράφημα παράγεται ένα επίσης πλήρες γράφημα. Επιπλέον είναι γραφήματα με τη μέγιστη δυνατή πυκνότητα ακμών, δηλαδή ο αριθμός των ακμών τους είναι ο μέγιστος δυνατός δεδομένου του αριθμού των κορυφών τους. Συγκεκριμένα το πλήθος των ακμών ενός πλήρους γραφήματος n κόμβων έχει ακριβώς $\frac{n(n-1)}{2}$ ακμές.



Σχήμα 2.3: Πλήρες γράφημα.

Διμερή γραφήματα

Εξίσου χρήσιμα αποδεικνύονται και τα διμερή γραφήματα (bipartite graphs). Τα διμερή γραφήματα είναι απλά, μη κατευθυνόμενα γραφήματα, οι κόμβοι των οποίων μπορούν να χωριστούν σε δύο ξένα σύνολα, που ονομάζονται μέρη (parts), έτσι ώστε κάθε ακμή του γραφήματος να συνδέει έναν κόμβο από το ένα σύνολο με έναν δεύτερο κόμβο από το άλλο [1]. Πολλά προβλήματα, όπως αυτό του τέλειου ταιριάσματος, λύνονται ευκολότερα στην περίπτωση των διμερών γραφημάτων. Το γεγονός αυτό υποδεικνύει ότι η αναγωγή ενός γραφήματος σε διμερές βελτιώνει την απόδοση των αλγορίθμων εύρεσης ενός τέλειου ταιριάσματος.



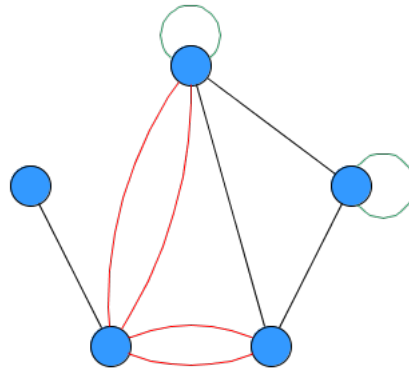
Σχήμα 2.4: Διμερές γράφημα.

Πολυγραφήματα

Είναι, σε κάποιες περιπτώσεις, χρήσιμο να επιτρέπουμε στο σύνολο E να περιέχει περισσότερα του ενός αντίγραφα κάποιων ακμών (πολλαπλές ακμές), καθώς και ακμές με αρχή και πέρας τον ίδιο κόμβο (βρόγχοι). Αυτού του είδους τα γραφήματα ονομάζονται πολυγραφήματα (multigraphs) [1] και απεικονίζονται όπως τα απλά, μη κατευθυνόμενα γραφήματα με μόνη διαφορά ότι στην περίπτωση πολλαπλών ακμών ή βρόγχου οι ακμές απεικονίζονται ως καμπύλες για να ξεχωρίζουν μεταξύ τους.

Κατευθυνόμενα γραφήματα

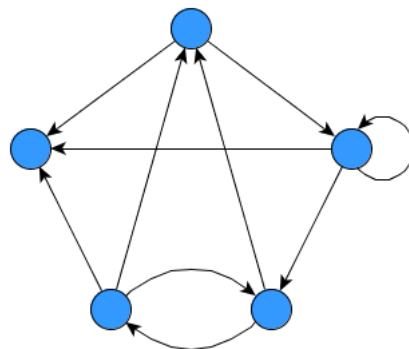
Συχνά, στις εφαρμογές, είναι απαραίτητο να λαμβάνουμε υπόψη την κατεύθυνση των ακμών ενός γραφήματος, δηλαδή να γνωρίζουμε ποιος κόμβος βρίσκεται στην αρχή και ποιος στο πέρας κάθε ακμής. Λόγω αυτής της απαίτησης προέκυψαν τα κατευθυνόμενα γραφήματα [1]. Είναι φανερό ότι σε ένα κατευθυνόμενο γράφημα μπορεί να εμ-



Σχήμα 2.5: Πολυγράφημα.

Οι κόκκινες ακμές ονομάζονται παράλληλες ακμές, ενώ οι πράσινες βρόγχοι.

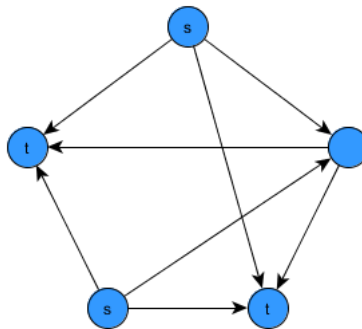
φανίζονται δύο ακμές που ενώνουν τους ίδιους κόμβους αλλά έχουν διαφορετική φορά και επιπλέον μπορούμε να επιτρέψουμε στο γράφημα να διαθέτει βρόγχους. Ο ορισμός των κατευθυνόμενων γραφημάτων δεν επιτρέπει, εν γένει, πολλαπλές ακμές με ίδια κατεύθυνση, ή πολλαπλούς βρόγχους, ωστόσο μπορεί να γενικευτεί ώστε να συμπεριλαμβάνει και αυτές τις περιπτώσεις και έτσι να προκύψει ο ορισμός των κατευθυνόμενων πολυγραφημάτων (directed multigraphs). Η απεικόνιση των κατευθυνόμενων γραφημάτων γίνεται όπως και στην περίπτωση των μη κατευθυνόμενων πολυγραφημάτων, με μόνη διαφορά ότι στο πέρας ή σε κάποια άλλη θέση των ακμών σχεδιάζουμε ένα βέλος με φορά από την αρχή προς το πέρας της.



Σχήμα 2.6: Κατευθυνόμενο γράφημα.

Άκυκλα κατευθυνόμενα γραφήματα

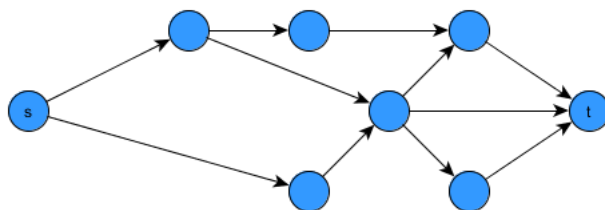
Κύκλος, στα κατευθυνόμενα γραφήματα, ονομάζεται μια αλληλουχία κόμβων που ενώνονται μεταξύ τους με ακμές ίδιας φοράς και ο πρώτος κόμβος ταυτίζεται με τον τελευταίο [1]. Ένας περιορισμός που μπορούμε να θέσουμε στην τοπολογία των κατευθυνόμενων γραφημάτων είναι και η απουσία κύκλων. Με αυτόν τον τρόπο ορίζουμε τα άκυκλα κατευθυνόμενα γραφήματα τα οποία διατίθενται για μια πληθώρα εφαρμογών, όπως για την μοντελοποίηση ιεραρχιών, ακολουθιών γεγονότων ή πληροφοριών καθώς και στοχαστικών διαδικασιών [2]. Αποδεικνύεται ότι τα άκυκλα κατευθυνόμενα γραφήματα έχουν πάντα τουλάχιστον έναν κόμβο με έσω-βαθμό ίσο με μηδέν (0) και τουλάχιστον έναν κόμβο με έξω-βαθμό ίσο με μηδέν (0) [2]. (Ως έσω-βαθμός ενός κόμβου ορίζεται το πλήθος των εισερχόμενων σε αυτόν ακμών και ως έξω-βαθμός το πλήθος των εξερχόμενων από αυτόν ακμών.) Οι κόμβοι με τις παραπάνω ιδιότητες ονομάζονται πηγές (sources) και καταλήξεις (sinks) αντίστοιχα.



Σχήμα 2.7: Άκυκλο κατευθυνόμενο γράφημα.

ST-γραφήματα

Μια υποκατηγορία των άκυκλων κατευθυνόμενων γραφημάτων είναι και τα st-γραφήματα (st-graphs). Χαρακτηριστικό τους είναι ότι διαθέτουν ακριβώς μία πηγή και ακριβώς μία κατάληξη. Μία από τις σημαντικές ιδιότητες αυτών των γραφημάτων είναι ότι κάθε κόμβος τους ανήκει σε κάποιο μονοπάτι που ξεκινάει πάντα από τον κόμβο πηγή και καταλήγει πάντα στον κόμβο κατάληξη [2].

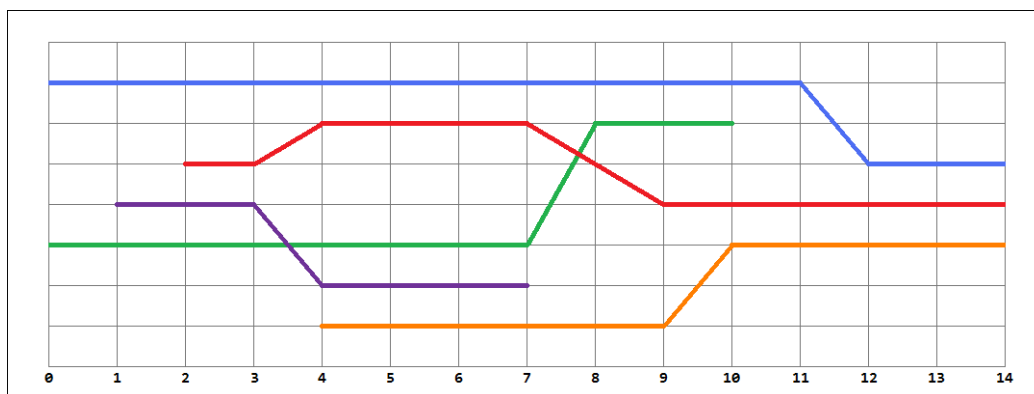


Σχήμα 2.8: S-T γράφημα.

2.2 Ειδικές Μορφές Γραφημάτων

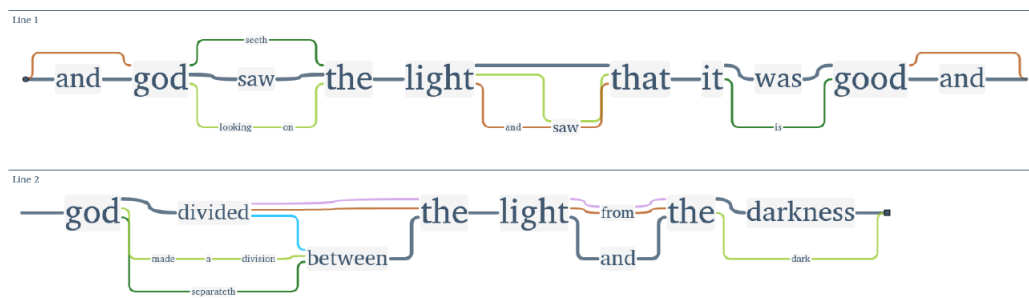
Δύο από τα προβλήματα που αντιμετωπίζονται πιο συχνά στη σύγχρονη έρευνα των ανθρωπιστικών σπουδών, και όχι μόνο, είναι πρώτον, η μελέτη των συναντήσεων ή συσχετίσεων ατόμων στο πλαίσιο ιστορικών ή φανταστικών γεγονότων και δεύτερον η καταγραφή των διαφορών ή παραλλαγών ανάμεσα σε διαφορετικές εκδόσεις γραπτών κειμένων. Σε αυτές τις περιπτώσεις η οπτικοποίηση τέτοιων πληροφοριών κρίνεται απαραίτητη αφού σε μεγάλης έκτασης γραπτά έργα η απλή καταγραφή τους δημιουργεί μια χαοτική λίστα που πολύ δύσκολα μελετάται χωρίς σφάλματα. Το πρόβλημα αυτό ανάγεται στον ορισμό γραφημάτων που διαθέτουν συγκεκριμένη δομή και διάταξη για κάθε περίπτωση και στην ανάπτυξη μεθόδων για την καλλίτερη απεικόνισή τους.

Στην περίπτωση της μελέτης των συναντήσεων-συσχετίσεων χρησιμοποιούνται τα γραφήματα “ιστοριογραμμής”. Αυτά τα γραφήματα, εποπτικά, μοιάζουν με απλά διμερή γραφήματα τα μέρη των οποίων είναι πάντα ισοπληθικά και οι ακμές τους ακολουθούν μια ένα-προς-ένα σχέση όσον αφορά τους κόμβους που συνδέουν, δηλαδή κάθε κόμβος είναι συνδεδεμένος με έναν και μόνο κόμβο από το άλλο μέρος του διμερούς γραφήματος και όλοι οι κόμβοι τελικά είναι συνδεδεμένοι.



Σχήμα 2.9: Μία απλή απεικόνιση ιστοριογραμμής.

Στην περίπτωση της μελέτης των παραλλαγών διαφόρων εκδόσεων του ίδιου κειμένου χρησιμοποιούνται τα γραφήματα “κειμενικών παραλλαγών” ή απλά “γραφήματα παραλλαγών”. Εν γένει αυτά τα γραφήματα συγγενεύουν με τα st-γραφήματα. Δηλαδή υπάρχει ένας κόμβος πηγή που ονομάζεται αρχή (start) και ένας κόμβος κατάληξη που ονομάζεται τέλος (end) και κάθε άλλος κόμβος ανήκει σε ένα μονοπάτι που ξεκινάει από την “αρχή” και καταλήγει στο “τέλος”. Οι υπόλοιποι κόμβοι σε αυτά τα γραφήματα παίζουν το ρόλο των διαφόρων λέξεων που ανήκουν στα υπό μελέτη κείμενα, ενώ τα μονοπάτια αντιπροσωπεύουν την ταυτότητα του εκάστοτε κειμένου.



Σχήμα 2.10: Μία απλή απεικόνιση γραφήματος κειμενικών παραλλαγών.

Στα επόμενα κεφάλαια γίνεται μια εμπειριστατωμένη ανάλυση στα δύο αυτά είδη γραφημάτων και δίνεται έμφαση στον ορισμό τους, τα προβλήματα που αντιμετωπίζονται κατά την απεικόνισή τους και τις εφαρμογές που τα υλοποιούν.

2.3 Γραφοθεωρητικές δομές

Η κωδικοποίηση των γραφημάτων με σκοπό την επεξεργασία και την αποθήκευσή τους στον ηλεκτρονικό υπολογιστή είναι ένα πολύ βασικό βήμα για την εκτενή μελέτη τους. Στην περίπτωση μεγάλων γραφημάτων ($> 10^6$ κόμβων), ακόμη και για την παρούσα τεχνολογία, είναι πολύ δύσκολη, έως και αδύνατη, η επεξεργασία τους, αν δεν έχουν κωδικοποιηθεί με συμπαγείς δομές που χαρακτηρίζονται από γρήγορη προσπέλασή τους.

Υπάρχουν δύο θεωρητικά ευσταθείς τρόποι κωδικοποίησης. Πρώτον, με τη χρήση ενός πίνακα που ονομάζεται πίνακας γειτνίασης του οποίου οι τιμές χαρακτηρίζουν τη σχέση δύο κόμβων (αν τους ενώνει κάποια ακμή ή όχι και αν αυτή διαθέτει κάποια τιμή, σήμανση ή βάρος). Δεύτερον, με τη χρήση μίας λίστας, η οποία ονομάζεται λίστα γειτνίασης και κάθε στοιχείο της, που αντιπροσωπεύει έναν κόμβο του γραφήματος, διατηρεί

την διεύθυνση στη μνήμη του υπολογιστή, των γειτονικών σε αυτόν κόμβων. Ωστόσο, πρακτικά, πιο αποδοτική μέθοδος είναι ένας συνδυασμός των παραπάνω, δηλαδή δομές που είναι ευέλικτες και η επεξεργασία τους γίνεται σε μικρό σχετικά χρόνο.

2.4 Σχεδίαση Γραφημάτων

Η σχεδίαση ή οπτικοποίηση ή οπτική αναπαράσταση των γραφημάτων είναι ένας τομέας της θεωρίας γραφημάτων με μεγάλο ερευνητικό ενδιαφέρον ειδικά τα τελευταία χρόνια. Σκοπός του είναι να μελετήσει μεθόδους και αλγορίθμους για την οπτικοποίηση διαφόρων ειδών γραφημάτων χρησιμοποιώντας κλειστά σχήματα ως κόμβους (πολύγωνα, κύκλους) και γραμμές (καμπύλες, ευθείες, τεθλασμένες γραμμές) ως ακμές. Επιπλέον μπορούν να προστεθούν χρωματισμοί και διαφοροποιήσεις μεγεθών για την αναπαράσταση διαφόρων πληροφοριών που αφορούν τους κόμβους και τις ακμές.

Η οπτικοποίηση ενός γραφήματος υπόκειται σε περιορισμούς που τίθενται από το σκοπό της. Κάθε πρόβλημα οπτικοποίησης θέτει ένα σύνολο περιορισμών που αφορούν, μεταξύ άλλων, την προβολή ιδιοτήτων των προκειμένων γραφημάτων, του συνολικού εμβαδού (ή όγκου στην περίπτωση τρισδιάστατων απεικονίσεων) της παραγόμενης οπτικοποίησης, των τομών ακμών και των επικαλυπτόμενων κόμβων. Έτσι, για παράδειγμα, εάν δίνεται ένα επίπεδο γράφημα, τότε πρέπει να οπτικοποιηθεί με τρόπο τέτοιο ώστε να είναι φανερό η ιδιότητά του αυτή, ή στην περίπτωση ενός δένδρου μπορεί να ζητείται να οπτικοποιηθεί έτσι ώστε οποιοσδήποτε κόμβος να βρίσκεται πάντα σε υψηλότερο σημείο από τους θυγατρικούς κόμβους του. Γενικά η επίλυση τέτοιων προβλημάτων αλγοριθμικά είναι πολύ δύσκολη αφού τα περισσότερα από αυτά ανήκουν στις κλάσεις NP-hard και NP-complete. Η μελέτη, ωστόσο, τέτοιων μεθόδων είναι απαραίτητη, αφού καλά οπτικοποιημένα γραφήματα μπορούν να παρέχουν μεγάλη γκάμα πληροφοριών και να χρησιμοποιηθούν ως πρότυπα για ποικίλες εφαρμογές, από προγραμματισμό εργασιών και καταγραφή πληροφοριών έως πολύπλοκα δίκτυα υπολογιστών και διαρρύθμιση χώρων.

2.5 Γραφοθεωρητικά Προβλήματα

Όπως σημειώθηκε και προηγουμένως, πολλά από τα προβλήματα οπτικοποίησης γραφημάτων ανήκουν στις κλάσεις NP-hard και NP-complete όπως ένας μεγάλος, άλλωστε, αριθμός σημαντικών προβλημάτων της θεωρίας γραφημάτων. Αξίζει να παρου-

σιασθούν με συντομία μερικά από αυτά, η αποδοτική επίλυση των οποίων αποτελεί βασική προϋπόθεση για την επιτυχή απεικόνιση ιστοριογραμμών και γραφημάτων παραλλαγών.

Το πρόβλημα του Περιορισμού των Τομών

Crossing Minimization Problem

Αριθμός τομών ενός γραφήματος ονομάζεται το ελάχιστο πλήθος των τομών μεταξύ των ακμών που μπορεί να έχει το γράφημα. Ένα επίπεδο γράφημα, για παράδειγμα έχει αριθμό τομών ίσο με μηδέν. Το πρόβλημα του περιορισμού των τομών ενός γραφήματος έγκειται στην αλγοριθμική εύρεση του αριθμού των τομών για αυτό το γράφημα. Το πρόβλημα αυτό ανήκει στην κλάση των NP-hard προβλημάτων [4]. Ο περιορισμός των τομών είναι, στις περισσότερες περιπτώσεις, ένα απαραίτητο βήμα που πρέπει να εκτελεσθεί κατά την οπτικοποίηση ενός γραφήματος. Έτσι είναι και στην περίπτωση των ιστοριογραμμών, όπου μικρός αριθμός τομών υποδεικνύει μία καλή απεικόνιση.

Το πρόβλημα της 1-Επιπεδότητας

1-Planarity Problem

1-Επίπεδο ονομάζεται το γράφημα εκείνο που μπορεί να σχεδιαστεί στο επίπεδο με τρόπο τέτοιο ώστε κάθε ακμή του να τέμνεται από το πολύ μία άλλη. Το πρόβλημα αυτό είναι NP-complete και μάλιστα παραμένει σε αυτή την κλάση ακόμα και αν το δοσμένο γράφημα είναι ένα επίπεδο γράφημα με την προσθήκη μιας ακμής [3]. Επιπλέον σχετίζεται σε μεγάλο βαθμό και με το πρόβλημα του περιορισμού των τομών που περιγράφηκε παραπάνω, αφού σε πολλές περιπτώσεις οι αλγόριθμοι που περιορίζουν τις τομές ενός γραφήματος, βρίσκουν εφαρμογή και σε αυτό το πρόβλημα. Παρόμοια, με αυτό, προβλήματα εμφανίζονται κατά την απεικόνιση γραφημάτων παραλλαγών και ιστοριογραμμών.

2.6 Εφαρμογές της Θεωρίας Γραφημάτων

Οι χρήσεις των γραφημάτων είναι ποικίλες και βρίσκουν εφαρμογή στο σύνολο των επιστημών, καθώς αποτελούν τον κατεξοχήν τρόπο απόδοσης των ερευνητικών δεδομένων και αποτελεσμάτων, αλλά και μοντελοποίησης πρακτικών προβλημάτων. Εφαρμογή

της θεωρίας γραφημάτων γίνεται πρωταρχικά στον τομέα της Πληροφορικής, όπου συχνά τα διάφορα προβλήματα που προκύπτουν ανάγονται σε γραφοθεωρητικά μοντέλα με σκοπό την ευκολότερη μελέτη ή και επίλυσή τους. Επιπλέον τα γραφήματα στην πληροφορική μπορούν να απεικονίσουν επικοινωνιακά δίκτυα, όπως τα μέσα κοινωνικής δικτύωσης, ή την οργάνωση παντός είδους δεδομένων σε εύκολα διαχειρίσιμες δομές.

Στη Χημεία, για τη μοντελοποίηση των μοριακών δεσμών και της δομής των χημικών ενώσεων και κατ' επέκταση για την καλύτερη κατανόηση της των χημικών αντιδράσεων, του σχηματισμού νέων προϊόντων από τις αντιδράσεις αυτές και των ιδιοτήτων τόσο των στοιχείων και των ενώσεων, όσο και των νέων προϊόντων. Στη Βιολογία, για τη μελέτη του γενετικού υλικού και των τρόπων κωδικοποίησης, αποθήκευσης και μετάφρασης της γενετικής πληροφορίας μέσα στους έμβιους οργανισμούς καθώς και φαινομένων που αφορούν μεταδόσεις χαρακτηριστικών ή και ασθενειών μεταξύ οργανισμών και πληθυσμών. Επιπλέον, είναι δυνατό, σε συνδυασμό με την ιατρική, να οδηγήσει σε επαναστατικές μεθόδους θεραπείας, ανίατων μέχρι σήμερα, ασθενειών.

Ενας λιγότερο γνωστός τομέας, αλλά εξίσου σπουδαίος, στον οποίο χρησιμοποιούνται κατά κόρον τα γραφήματα, είναι η Γλωσσολογία και ειδικά η Συγκριτική Γλωσσολογία και η Υπολογιστική Γλωσσολογία. Η Συγκριτική Γλωσσολογία, αν και στο παρελθόν ήταν ταυτόσημη με τη Γλωσσολογία, πλέον έχει διαχωριστεί από αυτήν και ασχολείται αποκλειστικά με τη διεξοδική μελέτη (φωνολογίας και μορφολογίας) των γλωσσών της ίδιας γλωσσικής οικογένειας, με απώτερο σκοπό την εύρεση κοινών στοιχείων μεταξύ τους, τα οποία θα δώσουν πληροφορίες για το πώς εμφανίστηκε η πρώτη γλώσσα στον κόσμο. Αυτές ακριβώς τις σχέσεις σχηματοποιούν και τα γραφήματα, ταξινομώντας τα δεδομένα και παρέχοντας τη δυνατότητα στους γλωσσολόγους να τα συγκρίνουν στην πράξη, διευκολύνοντας και συντομεύοντας το έργο τους. Αντίστοιχα, η Υπολογιστική Γλωσσολογία, που έχει ως αντικείμενο τη μοντελοποίηση της φυσικής γλώσσας μέσω του ηλεκτρονικού υπολογιστή για ποικίλες εφαρμογές, όπως τα συστήματα αναγνώρισης φωνής ή οι αυτόματες μεταφράσεις, χρησιμοποιεί τα γραφήματα για να δημιουργήσει σημασιολογικά δίκτυα (semantic network) προκειμένου να αναπαραστήσει τις σημασιολογικές σχέσεις ανάμεσα στα δεδομένα. Έτσι απλοποιείται η διαδικασία μοντελοποίησης της σημασίας κάθε λέξης, αφού είναι πολύ πιο εύκολο να γίνει αντιληπτή δίνοντας συγχρόνως άλλες, συναφείς με αυτήν, λέξεις, που οδηγούν περιγραφικά και περιφραστικά στην ερμηνεία της.

Γίνεται, λοιπόν, φανερό ότι τα τελευταία χρόνια τα γραφήματα έχουν διεισδύσει σε πολλούς κλάδους της επιστήμης. Η σπουδαιότητά τους έγκειται στην ιδιότητα που έχουν να παρουσιάζουν δεδομένα με τρόπο που συμβάλλει στην βέλτιστη εποπτεία τους, κα-

θώς οπτικοποιούν τόσο τα ίδια, όσο και τις σχέσεις μεταξύ τους. Επίσης, η ραγδαία τεχνολογική πρόοδος έχει αναγάγει τον ηλεκτρονικό υπολογιστή στο κυριότερο εργαλείο έρευνας σε όλους τους τομείς της επιστήμης και χρήσει των εφαρμογών της θεωρίας γραφημάτων έχει πετύχει τη βελτιστοποίηση των διαδικασιών που ακολουθούνται κατά την ερευνητική διαδικασία. Συνεπώς είναι απαραίτητη η μελέτη της θεωρίας γραφημάτων και των αντίστοιχων εφαρμογών της.

Κεφάλαιο 3

Διαδικτυακές Τεχνολογίες Οπτικοποίησης

Το διαδίκτυο αποτελεί πλέον ένα αναπόσπαστο κομμάτι της ζωής μας. Ενα μεγάλο ποσοστό των εργασιών μας σχετίζεται άμεσα και έμμεσα με αυτό. Ειδικά με την ανάπτυξη της τεχνολογίας το διαδίκτυο επεκτείνει ολοένα και πιο γρήγορα τις δυνατότητες και τις εφαρμογές του στην καθημερινότητα.

Κυρίαρχο ρόλο στην ανάπτυξη διαδικτυακών εφαρμογών παίζει η γλώσσα προγραμματισμού javascript. Είναι μια πολύ ευέλικτη και συγχρόνως κατανοητή γλώσσα με απεριόριστες δυνατότητες. Πρόσθετες δυνατότητες παρέχονται μέσω βιβλιοθηκών, πολλές εκ των οποίων είναι ανοιχτού κώδικα. Με τον τρόπο αυτό δίνεται η δυνατότητα να αναπτύξει κανείς πολύ εύκολα διαδικτυακές εφαρμογές.

3.1 Η Γλώσσα Προγραμματισμού JavaScript

Όπως ήδη αναφέρθηκε, η JavaScript είναι μία γλώσσα προγραμματισμού με τη συμβολή της οποίας είναι δυνατόν να δημιουργήσουμε ειδικά προγράμματα με στόχο την προβολή ιστοριογραμμών. Αυτό το τελευταίο μπορεί να επιτευχθεί χρησιμοποιώντας τις ποικίλες βιβλιοθήκες της JavaScript.

Πρωταρχικά, στο συγκεκριμένο σημείο της εργασίας θα περιγράψουμε με όσο πιο απλό και περιεκτικό τρόπο μπορούμε τη γλώσσα JavaScript και την εξέλιξή της στο χρόνο. Η JavaScript ανήκει στις διερμηνευμένες γλώσσες προγραμματισμού για ηλεκτρονικούς υπολογιστές. Η βασική της λειτουργία είναι ο ρόλος του "διερμηνέα", η μετάφραση

δηλαδή των διαφόρων εντολών σε προγραμματιστική γλώσσα προκειμένου ένας ηλεκτρονικός υπολογιστής να πραγματοποιήσει συγκεκριμένες ενέργειες. Παρόλο που έχει δανειστεί κάποια στοιχεία από τη γλώσσα C και από τη Java, έχει διαφοροποιηθεί ολοκληρωτικά από αυτές και σε καμία περίπτωση δε θα πρέπει να τις συγχέουμε ή να τις συγκρίνουμε. Το ίδιο ισχύει και με τις γλώσσες Self και Scheme, στις οποίες βασίζονται οι θεμελιώδεις αρχές σχεδιασμού της.

Ο άνθρωπος που πρώτος εισήγαγε τη JavaScript στον τομέα της πληροφορικής ήταν ο Brendan Eich το 1995. Σε εκείνα τα αρχικά στάδια της εισαγωγής της ονομαζόταν Mocha, αργότερα μετονομάστηκε σε LiveScript, για να καταλήξει στη σημερινή της ονομασία με την οποία έγινε και ευρέως γνωστή. Ιδιαίτερη επιτυχία γνώρισε ως γλώσσα στην πλευρά του πελάτη/χρήστη (client-side) για εκτέλεση κώδικα σε ιστοσελίδες, με πολλές μεγάλες εταιρείες να την συμπεριλαμβάνουν στα προγράμματα φυλλομετρητών τους.

Ωστόσο η JavaScript δεν περιορίστηκε μόνο στον τομέα των φυλλομετρητών ιστού, αλλά με το πέρασμα των ετών και καθώς γινόταν όλο και πιο δημοφιλής, απέκτησε όλο και περισσότερες εφαρμογές στον ευρύτερο τομέα της τεχνολογίας. Πιο συγκεκριμένα, πλέον όχι απλώς χρησιμοποιείται από τις περισσότερες ιστοσελίδες στο διαδίκτυο, αλλά με την πρόοδο της τεχνολογίας βρίσκει εφαρμογές στα κινητά τηλέφωνα smartphones, στις κονσόλες βιντεοπαιχνιδιών, στα tablets και γενικά στο σύνολο των φορητών ηλεκτρονικών συσκευών επικοινωνίας και ψυχαγωγίας/διασκέδασης. Ο λόγος της ευρείας διάδοσής της είναι το γεγονός ότι ουσιαστικά πρόκειται για μία "ελαφριά" και εύχρηστη γλώσσα προγραμματισμού, που παράλληλα παρουσιάζει και σημαντικά πλεονεκτήματα έναντι των υπολοίπων. Ο κώδικάς της ενσωματώνεται σε αυτόν της γλώσσας HTML, εκτελείται απευθείας και το μόνο που απαιτεί είναι να υποστηρίζεται από τον εκάστοτε φυλλομετρητή, ενώ χάρη στη JavaScript γίνεται εφικτή η αλληλεπίδραση των χρηστών με μία ιστοσελίδα, μειώνοντας συγχρόνως την εξάρτηση από τον εξυπηρετητή (server).

Βιβλιοθήκες JavaScript

Παρακάτω θα περιγραφούν ορισμένες από τις βασικότερες βιβλιοθήκες JavaScript.

jQuery: Η σημαντικότερη και δημοφιλέστερη cross browser βιβλιοθήκη της JavaScript για τη διαχείριση του περιεχομένου των ιστοσελίδων, με ευρεία χρήση στον κόσμο του διαδικτύου. Απλή στο χειρισμό της, "ελαφριά", αποτελεί την πρώτη επιλογή των ιστοσελίδων και των εφαρμογών στα κινητά και τα tablets και θεωρείται ότι το διαδίκτυο δε θα ήταν αυτό που είναι σήμερα, ούτε θα είχε την ίδια θέση στη ζωή μας χωρίς αυτήν.

AngularJS: Χρησιμοποιείται ως επί το πλείστον από επιχειρήσεις για πιο πολύπλοκες και απαιτητικές επαγγελματικές ιστοσελίδες. Κύριο χαρακτηριστικό της η αυτόματη ενημέρωση όλων των δεδομένων όταν υφίσταται αλλαγή έστω και ένα από αυτά, καθώς προσφέρει αμφίδρομο “δέσιμο” δεδομένων.

Raphaël: Η πιο διαδεδομένη βιβλιοθήκη της γλώσσας JavaScript για δημιουργία γραφικών. Προτιμάται από χιλιάδες ιστοσελίδες λόγω της ιδιότητάς της να ρυθμίζει τη λειτουργία της ανταποκρινόμενη εξαιρετικά σε περιβάλλοντα ιστοσελίδων που υποστηρίζουν τα χαρακτηριστικά και μειώνοντας αντίστοιχα την ανταπόκρισή της όταν τα απαιτούμενα περιβάλλοντα παρουσιάζουν μερικές ή ολικές ελλείψεις (cross-browser library).

3.2 Η Βιβλιοθήκη JQuery

Όπως σημειώθηκε και προηγουμένως, η javascript αποτελεί την πιο διαδεδομένη γλώσσα προγραμματισμού στον τομέα των διαδικτυακών εφαρμογών, με πληθώρα βιβλιοθηκών που διευκολύνουν τη χρήση της και παρέχουν πολλές μεθόδους υλοποίησης απαιτητικών ενεργειών. Η επικρατέστερη, τα τελευταία χρόνια, JavaScript βιβλιοθήκη, ονομάζεται JQuery και αποτελεί ένα ισχυρό εργαλείο για ανάπτυξη διαδικτυακών εφαρμογών ενώ ταυτόχρονα παραμένει εξαιρετικά απλή στη χρήση της.

Η JQuery είναι μια JavaScript βιβλιοθήκη ανοιχτού κώδικα, η οποία σχεδιάστηκε από τον John Resig και η πρώτη έκδοσή της (1.0) δημοσιεύτηκε την 26η Αυγούστου 2006 (τρέχουσα έκδοση: 3.1.1). Ο J. Resig, επηρεασμένος από την CSSQuery, μια βιβλιοθήκη που διευκόλυνε την μορφοποίηση του περιεχομένου μιας ιστοσελίδας, δημιούργησε την JQuery ως μια “οικουμενική” βιβλιοθήκη, η οποία θα παρείχε τη δυνατότητα στον χρήστη, με λίγες γραμμές κώδικα και σε μικρό χρόνο, να αξιοποιήσει όλες τις δυνατότητες επεξεργασίας και μορφοποίησης του περιεχομένου μιας ιστοσελίδας. Μάλιστα, στο λογότυπο της βιβλιοθήκης, αναγράφεται η φράση: “write less, do more”, η ελληνική απόδοση της οποίας θα ήταν “γράψτε λιγότερα, κάντε περισσότερα”.

Τα βασικότερα χαρακτηριστικά της JQuery είναι η εύρεση στοιχείων DOM (Document Object Model), η μορφοποίησή τους μέσω CSS (Cascading Style Sheets), η διαχείριση συμβάντων (event handling), η προβολή κινούμενων γραφικών (animations) και διαφόρων εφέ (effects), η χρήση της τεχνικής AJAX (Asynchronous JavaScript And XML) και η ανάλυση αρχείων JSON (JavaScript Object Notation). Επιπλέον υπάρχει η δυνατότητα επέκτασης των λειτουργιών της μέσω επιμέρους επεκτάσεων (plug-ins).

3.3 Η Βιβλιοθήκη AngularJS

Η AngularJS (αλλιώς Angular ή Angular.js) δημιουργήθηκε το 2009 από τον Miško Hevery και ο αρχικός σκοπός της ήταν να αποτελέσει το λογισμικό που θα εξυπηρετούσε τις ανάγκες για δυνατότητα online JSON (JavaScript Object Notation) αποθήκευσης δεδομένων, προκειμένου εν συνεχεία τα τελευταία να χρησιμοποιηθούν για τη δημιουργία απλών εφαρμογών για επιχειρήσεις. Επρόκειτο για ένα πρωτότυπο, θα λέγαμε, εγχείρημα στο οποίο μπορούσε κανείς να αποκτήσει πρόσβαση μέσω του ιστότοπου (web domain) "GetAngular.com". Η σελίδα μετρούσε ήδη κάποιες εγγραφές χρηστών, ωστόσο η ιδέα αυτή σύντομα εγκαταλείφθηκε και αποφασίστηκε η έκδοση της Angular ως βιβλιοθήκης ανοιχτού κώδικα, κύρια χρήση της οποίας θα είναι η δημιουργία και υποστήριξη πληθώρας εφαρμογών web, όπως για παράδειγμα ιστοσελίδων. Βασίζει τη λειτουργία της στη γλώσσα προγραμματισμού JavaScript και ανήκει, στο μεγαλύτερο μέρος της, στη Google. Μάλιστα, τον Ιούλιο του 2012 δημιουργήθηκε μία επέκταση στον περιηγητή Google Chrome ονόματι Batarang, με στόχο τον αποτελεσματικότερο εντοπισμό και διόρθωση των σφαλμάτων για τις εφαρμογές web που είναι συνυφασμένες με την Angular [6].

Ο τρόπος λειτουργίας της Angular έχει ως εξής: πρωταρχικά «διαβάζεται» η σελίδα HTML, η οποία συνήθως περιέχει ενσωματωμένα κάποια επιπλέον χαρακτηριστικά που διευκολύνουν την τροποποίηση της σελίδας HTML. Αυτά τα χαρακτηριστικά ερμηνεύονται ως πληροφορίες (directives) για το πώς πρέπει να «δεθούν» τα εισαγόμενα (inputs) ή εξαγόμενα (outputs) στοιχεία της ιστοσελίδας, αναπαριστώμενα από συγκεκριμένες μεταβλητές JavaScript. Οι τιμές των μεταβλητών με τη σειρά τους μπορούν να ενσωματωθούν στον κώδικα χειροκίνητα ή να ανακτηθούν από στατικές ή δυναμικές πηγές JSON [6]. Βασικό στοιχείο για τη λειτουργία της Angular είναι το scope, δηλαδή ένα αντικείμενο το οποίο μπορεί να είναι εντός ή εκτός του πεδίου δράσης σε οποιοδήποτε σημείο του προγράμματος και το οποίο ανιχνεύει τυχόν αλλαγές στη φάση της μοντελοποίησης, τροποποιώντας αντίστοιχα τον προγραμματισμό HTML. Οι διάφορες εργασίες που επιτελεί η εκκίνηση προγράμματος της Angular (bootstrapper) λαμβάνουν χώρα σε τρεις φάσεις και έπονται της φόρτωσης του DOM: αρχικά δημιουργείται ένας νέος injector, έπειτα συγκεντρώνονται οι directives που αφορούν στο DOM και τέλος όλες αυτές οι πληροφορίες συνδέονται με το scope. Σαν τελικό αποτέλεσμα των άνω διενεργειών έχουμε τον καθορισμό προσαρμοσμένων και επαναχρησιμοποιήσιμων HTML στοιχείων και χαρακτηριστικών που καθορίζουν τις συνδέσεις δεδομένων και τη συμπεριφορά των στοιχείων παρουσίασης [6].

Η Angular σχεδιάστηκε με σκοπό να ξεχωρίσει τον χειρισμό DOM (document object

model) από τη λογική της εφαρμογής, καθώς και την πλευρά των χρηστών μιας εφαρμογής από την πλευρά του διακομιστή της (client side – server side), παρέχοντας συγχρόνως όλες τις απαραίτητες πληροφορίες για τη δομή, το «χτίσιμο» της εφαρμογής αυτής. Ουσιαστικά υλοποιεί το πρότυπο MVC (Model View Controller) για να διαχωρίσει την παρουσίαση, τα δεδομένα, και τις λογικές συνιστώσες. Έτσι κατάφερε να φέρει υπηρεσίες που κανονικά ανήκαν στην πλευρά του διακομιστή (server side services) σε εφαρμογές που δίνουν έμφαση και στηρίζουν τη λειτουργία τους στην πλευρά των χρηστών [6]. Με λίγα λόγια, η Angular αποτελεί τον απλούστερο, ευκολότερο και αποτελεσματικότερο τρόπο «δεσίματος» δεδομένων σε περιβάλλοντα HTML DOM. Το δέσιμο των δεδομένων πραγματοποιείται αμφίδρομα (two-way data binding) και αυτό αποτελεί το χαρακτηριστικό γνώρισμα της Angular. Χάρη σε αυτό το χαρακτηριστικό ο server αποδεσμεύεται από την ευθύνη της μοντελοποίησης (templating responsibilities) και συνεπώς τα πρότυπα παρέχονται σε απλή HTML μορφή σύμφωνα με τα στοιχεία που περιέχονται σε ένα πεδίο εφαρμογής που ορίζεται στο μοντέλο. Το γεγονός αυτό σε συνδυασμό με το αμφίδρομο «δέσιμο» δεδομένων δίνει την ευχέρεια στην Angular να εντοπίζει τυχόν αλλαγές στα μοντέλα της μέσω της σύγκρισης των τιμών των τρεχόντων δεδομένων με τις τιμές των δεδομένων που είχαν εισαχθεί σε προγενέστερο στάδιο [6].

Τον Σεπτέμβριο του 2014, κατά τη διάρκεια του «ng-Europe conference 22-23», ανακοινώθηκε η κυκλοφορία της έκδοσης AngularJS 2.0, προκειμένου να διαδεχθεί την προηγούμενη AngularJS 1.5.8. Η AngularJS 2.0 δεν αποτελεί μια απλή αναβάθμιση της AngularJS 1.5.8, αλλά πρόκειται για μία εξολοκλήρου καινούρια κυκλοφορία, γραμμένη από την αρχή, με τελείως διαφορετικό τρόπο επεξεργασίας και αποθήκευσης δεδομένων [6]. Ενδεικτικά ορισμένες από τις δραστικές αλλαγές που περιλαμβάνει η νέα έκδοση έναντι της προηγούμενης: 1)η λειτουργία της είναι συνυφασμένη με αυτήν των κινητών τηλεφώνων (smartphones) και μάλιστα διευκολύνεται αισθητά όταν πρώτα έχουν λυθεί τυχόν προβλήματα ή δυσλειτουργίες στο λογισμικό τους, 2)η βασική της λειτουργικότητα έχει μετατεθεί στα modules, καθιστώντας την πιο γρήγορη και «ελαφριά», 3)υποστηρίζεται μόνο από σύγχρονους browsers κι έτσι δεν τίθεται πια ζήτημα συμβατότητας με παλαιότερους (π.χ. δεν υποστηρίζει πια τον Internet Explorer), 4)χρησιμοποιεί ως προεπιλεγμένη την ελεύθερη και ανοιχτού κώδικα γλώσσα προγραμματισμού TypeScript, προσφέροντας μεταξύ άλλων αντικειμενοστραφή προγραμματισμό (object oriented programming), στατικό έλεγχο τύπων (static type checking), λειτουργία ανώνυμου (anonymous function), for/of loops, python-style generators, lambdas, generics και δυνατότητα επιτόπου αυτοεξέτασης, αυτοδιόρθωσης και τροποποίησης της δομής και λειτουργίας του προγράμματος από το ίδιο το πρόγραμμα ενώ αυτό «τρέχει» (reflection), 5)βελτιωμένη αντιστροφή του ελέγχου για την επίλυση των εξαρτήσεων (dependency

injection), 6) απλούστερη δρομολόγηση, 7) δυνατότητα φόρτωσης στη μνήμη δεδομένων, ακόμα και ολόκληρης της βιβλιοθήκης στο χρόνο που «τρέχει» το πρόγραμμα, εξαγωγής όσων το πρόγραμμα θεωρεί χρήσιμα για αυτό και αποφόρτωσης από τη μνήμη των υπολοίπων (dynamic loading) και 8) asynchronous template compilation.

Οι διαφοροποιήσεις αυτές αμφισβητήθηκαν από πολλούς και προκάλεσαν έντονη διαμάχη ανάμεσα στους προγραμματιστές. Μετά από ορισμένες διορθώσεις και αφού προηγήθηκε η κυκλοφορία διαφόρων preview, η τελική έκδοση της AngularJS 2.0 κυκλοφόρησε στις 14 Σεπτεμβρίου 2016 και είναι διαθέσιμη στην επίσημη ιστοσελίδα της Angular, ωστόσο εν τέλει δεν κατάφερε να διαδεχθεί την AngularJS 1.5.8, που παραμένει μέχρι σήμερα η στάνταρ τρέχουσα έκδοσή της [6].

3.4 Η Βιβλιοθήκη Raphaël

Η βιβλιοθήκη Raphaël αποτελεί ένα εύχρηστο εργαλείο δημιουργίας γραφικών τύπου SVG (Scalable Vector Graphics), δηλαδή αρχείων τύπου XML (Extensible Markup Language) που περιέχουν πληροφορίες εικόνας και κινούμενων γραφικών που επιτρέπουν αλληλεπίδραση με τον χρήστη. Πλεονέκτημα αυτού του τύπου γραφικών αποτελεί το ότι αποθηκεύονται, επεξεργάζονται και συμπιέζονται όπως ακριβώς και τα αρχεία τύπου XML επιτρέποντας έτσι την απλή υλοποίηση προγραμμάτων προβολής γραφικών με πολλές επιλογές επεξεργασίας τους. Το χαρακτηριστικό εκείνο που καθιστά την Raphaël πολυχρηστική είναι η δυνατότητα που δίνεται από τον κώδικα για την απευθείας επικοινωνία του χρήστη με τα SVG γραφικά καθώς και η πληθώρα μεθόδων δημιουργίας κινούμενων γραφικών (animations).

Πώς δουλεύει

Είναι εξαιρετικά εύκολο να χρησιμοποιήσει κανείς την βιβλιοθήκη Raphaël, ακόμα και αν δεν γνωρίζει καλά την γλώσσα javascript. Το μόνο που πρέπει να κάνει είναι να εισάγει τον κώδικά της στο αρχείο όπου γράφει τον δικό του κώδικα και στη συνέχεια, ανατρέχοντας στην επίσημη ιστοσελίδα της βιβλιοθήκης (<https://dmitrybaranovskiy.github.io/raphael/>) για πληροφορίες της χρήσης των εντολών, μπορεί με λίγες γραμμές κώδικα να προβάλει διάφορα γραφικά. Παρακάτω δίνονται μερικά απλά παραδείγματα κώδικα που κάνει χρήση των δυνατοτήτων της βιβλιοθήκης.

Παραδείγματα χρήσης της βιβλιοθήκης

Πρώτο βήμα για τη χρήση της βιβλιοθήκης Raphael είναι να αποθηκεύσουμε το αρχείο *raphael.min.js*, το οποίο μπορούμε να κατεβάσουμε από την επίσημη ιστοσελίδα του project, στον φάκελο όπου θα αποθηκεύσουμε και τον κωδικά μας.

Διαθέτουμε παρακάτω κώδικα HTML που εμπεριέχει απλά παραδείγματα χρήσης της βιβλιοθήκης.

```
1 <html>
2   <head>
3     <title> Raphael: A powerful SVG library </title>
4     <meta charset='UTF-8'>
5     <script src='raphael.min.js'></script>
6
7     <script>
8       window.onload = function() {
9         var p = new Raphael(10, 10, 500, 300);
10
11         p.rect(50, 20, 100, 80);
12
13         p.rect(0, 0, p.width, p.height)
14           .attr({
15             'stroke-width' : 5
16           });
17
18         var circle = p.circle(250, 60, 40);
19         circle.attr({
20             'stroke' : '#0000FF',
21             'stroke-width' : 3,
22             'fill' : '#FFFF00'
23         });
24
25         p.text(250, 60, 'Raphael');
26
27         p.path('M 340 50 L 340 80 440 100 200 150');
28
29       };
30     </script>
31   </head>
32
33   <body></body>
34 </html>
```

Κώδικας 3.1: Απλό παράδειγμα χρήσης της βιβλιοθήκης

Για να εισαχθεί η βιβλιοθήκη στο αρχείο html, εντός του tag `<head>` γράφτηκε ο κώδικας της γραμμής 5:

```
5 <script src='raphael.min.js'></script>
```

Κώδικας 3.2: Εισαγωγή της βιβλιοθήκης

Αμέσως μετά μπορούμε εντός του tag `<script>` της γραμμής 7 να γράψουμε τον κώδικά μας. Συνηθίζεται ωστόσο ο κώδικάς να γράφεται εντός μίας συνάρτησης η οποία θα καλείται αμέσως μετά την φόρτωση HTML κώδικα.

```
1 window.onload = function() {  
2     ...  
3 };
```

Αυτό αποτρέπει πιθανά σφάλματα κατά την εκτέλεση του προγράμματος, τα οποία θα οφείλονταν, κυρίως, στην αποτυχία εντοπισμού ορισμένων αντικειμένων του HTML κώδικα που δεν θα είχαν οριστεί εγκαίρως από το πρόγραμμα περιήγησης.

Πρωταρχικό ρόλο στη χρήση της βιβλιοθήκης Raphael παίζει ένα αντικείμενο τύπου *Paper*. Το αντικείμενο αποτελεί ουσιαστικά τον καμβά πάνω στον οποίο σχεδιάζονται τα γραφικά. Ο ορισμός του γίνεται με τη βοήθεια του κατασκευαστή *Raphael(...)* ο οποίος λαμβάνει ένα σύνολο παραμέτρων ανάλογα με τη χρήση που επιδιώκεται. Στο παράδειγμα, στη γραμμή 9, καλείται ο κατασκευαστής με παραμέτρους τέσσερις ακέραιους αριθμούς από τους οποίους οι δύο πρώτοι καθορίζουν τις συντεταγμένες της πάνω αριστερής γωνίας του καμβά, μετρημένες σε pixels, λαμβάνοντας υπόψη ότι ο πρώτος αριθμός είναι η θέση στον οριζόντιο και ο δεύτερος η θέση στον κατακόρυφο άξονα του κυρίως σώματος του προγράμματος περιήγησης). Αξίζει, στο σημείο αυτό, να αναφέρουμε ότι ο οριζόντιος άξονας ορίζεται από τα αριστερά προς τα δεξιά της οθόνης του ηλεκτρονικού υπολογιστή, ενώ ο κατακόρυφος από την πάνω προς την κάτω πλευρά της οθόνης. Οι δύο επόμενοι αριθμοί καθορίζουν το πλάτος και το ύψος του καμβά, μετρημένα σε pixels.

```
9 var p = new Raphael(10, 10, 500, 300);
```

Κώδικας 3.3: Ορισμός του καμβά

Στη γραμμή 9, του παραδείγματος, ορίσαμε την πάνω αριστερή γωνία του καμβά να βρίσκεται στο σημείο (10, 10), δηλαδή 10px από τα αριστερά και 10px από την πάνω πλευρά, ενώ αυτός να έχει μήκος 700px και πλάτος 500px.

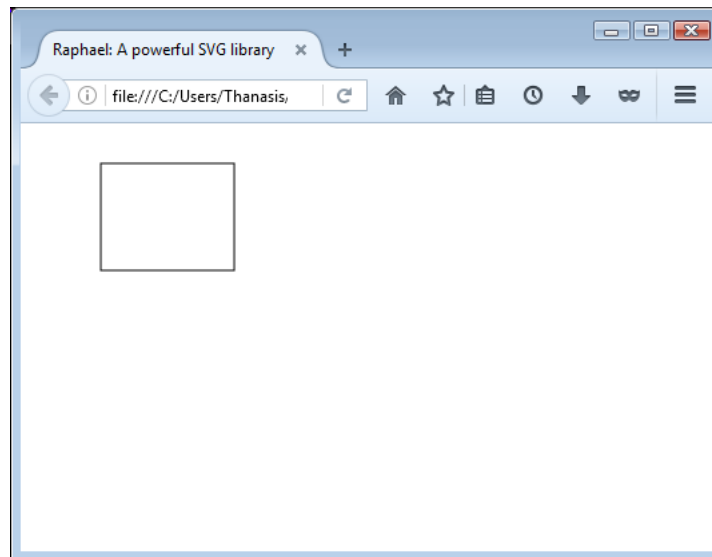
Στη συνέχεια μπορούμε να σχεδιάσουμε γραφικά απλά καλώντας τις μεθόδους που παρέχει η κλάση *Paper*. Στο παράδειγμα, περιέχονται μερικές από τις πιο βασικές μεθόδους

δους, τις οποίες και θα εξηγήσουμε με συντομία.

```
11 p.rect(50, 20, 100, 80);
```

Κώδικας 3.4: Κατασκευή ορθογωνίων

Στην γραμμή 11 του παραδείγματος κατασκευάζεται ένα ορθογώνιο χρήσει της μεθόδου *rect(...)*. Οι παράμετροι που λαμβάνει η μέθοδος θυμίζουν εκείνες του κατασκευαστή. Είναι τέσσερις ακέραιοι αριθμοί εκ των οποίων οι δύο πρώτοι περιγράφουν τη θέση της πάνω αριστερής γωνίας του ορθογωνίου, ως προς την πάνω αριστερή γωνία του καμβά όμως αυτή τη φορά, ενώ οι επόμενοι δύο περιγράφουν το μήκος και το ύψος, αντίστοιχα, του ορθογωνίου. Το αποτέλεσμα φαίνεται στο σχήμα 3.1.

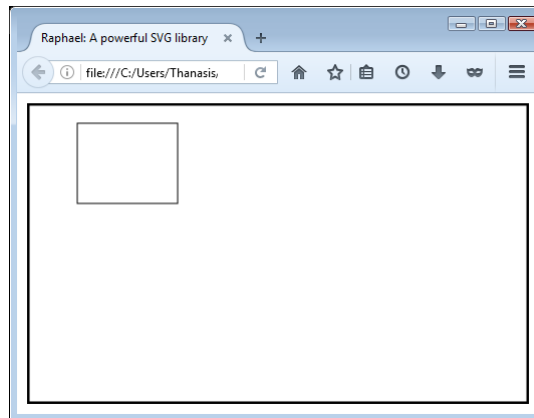


Σχήμα 3.1: Δημιουργία ορθογωνίου.

```
13 p.rect(0, 0, p.width, p.height)  
14 .attr({  
15     'stroke-width' : 5  
16 });
```

Κώδικας 3.5: Κατασκευή ορθογωνίων

Στην γραμμή 13 κατασκευάζεται ένα ορθογώνιο το οποίο περικλείει όλο τον καμβά, αφού το άνω αριστερό του άκρο ταυτίζεται με αυτό του καμβά και έχει διαστάσεις ίσες με αυτές του καμβά. Επιπλέον όμως κάνουμε χρήση της μεθόδου *attr(...)*. Η μέθοδος αυτή μορφοποιεί τα χαρακτηριστικά του αντικειμένου από το οποίο καλείται. Εδώ μεταβάλλουμε το πάχος της γραμμής σχεδίασης από *1px* που είναι προεπιλεγμένο, σε *5px*. Το αποτέλεσμα φαίνεται στο σχήμα 3.2.



Σχήμα 3.2: Δημιουργία περιγράμματος καμβά.

```

18     var circle = p.circle(250, 60, 40);
19     circle.attr({
20         'stroke' : '#0000FF',
21         'stroke-width' : 3,
22         'fill' : '#FFFF00'
23     });
  
```

Κώδικας 3.6: Κατασκευή κύκλων

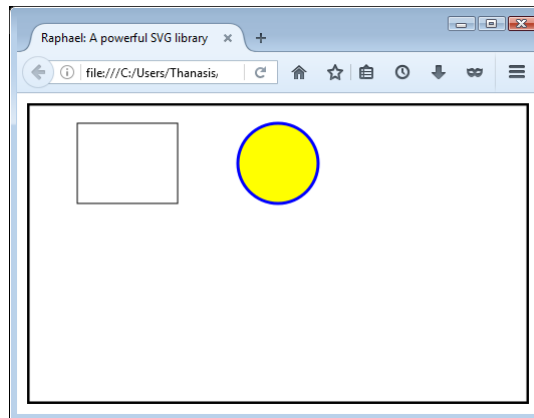
Στην γραμμή 15 κατασκευάζεται ένας κύκλος με τη μέθοδο *circle(...)* ο οποίος ανατίθεται στην μεταβλητή *circle*. Οι παράμετροι που δέχεται η μέθοδος είναι οι συντεταγμένες του κέντρου του κύκλου και η ακτίνα του. Η ανάθεση ενός οποιουδήποτε αντικειμένου της βιβλιοθήκης σε μια μεταβλητή είναι πάντα εφικτή και χαρίζει μεγάλη ευελιξία στο πρόγραμμα. Στο παρόν παράδειγμα εκμεταλλευόμαστε αυτή τη δυνατότητα για να μεταβάλουμε πιο κάτω, στις γραμμές 17 με 21 του κώδικα, μερικά χαρακτηριστικά του και συγκεκριμένα το πάχος της γραμμής σχεδίασης, το χρώμα της γραμμής σχεδίασης και το χρώμα γεμίματος του κύκλου. Το αποτέλεσμα φαίνεται στο σχήμα 3.3.

```

25     p.text(250, 60, 'Raphael');
  
```

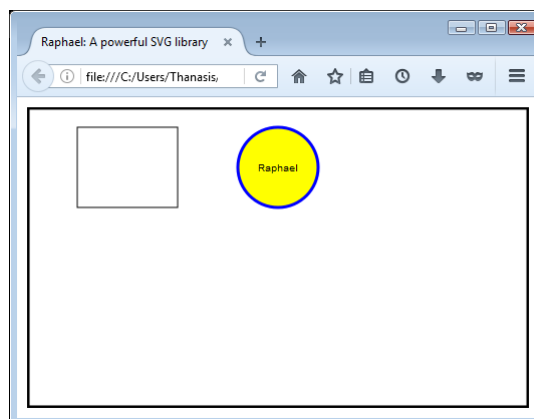
Κώδικας 3.7: Εισαγωγή κειμένου

Η Βιβλιοθήκη Raphaël μας δίνει την δυνατότητα να εισάγουμε κείμενο στα γραφικά καλώντας τη μέθοδο *text(...)*. Η μέθοδος αυτή λαμβάνει ως ορίσματα το σημείο του καμβά που θα τοποθετηθεί το κείμενο και το ίδιο το κείμενο με τη μορφή αλφαριθμητικού. Η τοποθέτησή του γίνεται με τρόπο παρόμοιο με αυτόν του κύκλου, δηλαδή ως προς το κέντρο του. Ως κέντρο του κειμένου λαμβάνεται το κέντρο του μικρότερου νοητού ορθογώνιου που περιβάλλει ολόκληρο το κείμενο. Ο τρόπος εισαγωγής κειμένου στον καμβά δίνεται στην γραμμή 23 του παραδείγματος και η απεικόνισή του φαίνεται στο σχήμα



Σχήμα 3.3: Δημιουργία κύκλου.

3.4. Αξίζει να σημειωθεί ότι στο παράδειγμα δόθηκαν οι συντεταγμένες ίδιες με αυτές του κέντρου του κύκλου του παραδείγματος, με σκοπό να δοθεί έμφαση στον τρόπο τοποθέτησής του κειμένου στον καμβά.



Σχήμα 3.4: Εισαγωγή κειμένου.

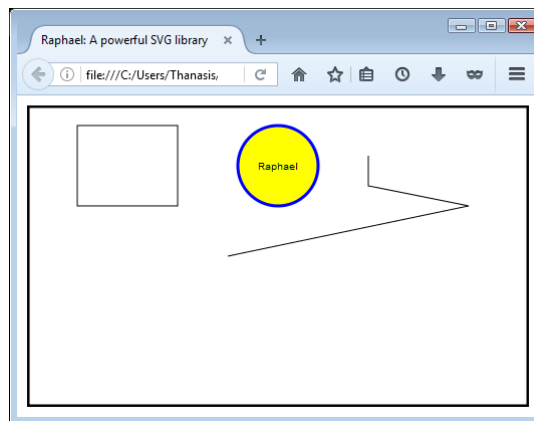
27

```
p.path('M 340 50 L 340 80 440 100 200 150');
```

Κώδικας 3.8: Δημιουργία Μονοπατιού

Ενα από τα σημαντικότερα στοιχεία της βιβλιοθήκης είναι και η μέθοδος δημιουργίας μονοπατιών. Ενα μονοπάτι αποτελείται από καμπύλες που ενώνουν ένα σύνολο σημείων του καμβά. Η κατασκευή ενός μονοπατιού γίνεται με τη χρήση της μεθόδου *path(...)*. Η μέθοδος αυτή παίρνει ως όρισμα μόνο ένα αλφαριθμητικό. Όπως φαίνεται και στη γραμμή 25 του παραδείγματος το αλφαριθμητικό αυτό αποτελείται από γράμματα και ζεύγη ακέραιων αριθμών. Το γράμμα *M* μεταφράζεται από τη μέθοδο ως μια εντολή μετακίνησης του αρχικού σημείου σχεδίασης στο νέο σημείο που ορίζεται από τους δύο ακέραιους που ακολουθούν. Το γράμμα *L* μεταφράζεται ως εντολή σχεδίασης ευθείας

γραμμής από το τρέχον σημείο σχεδίασης στο σημείο που ορίζεται από του δύο ακέραιους που ακολουθούν. Αν υπάρχουν περισσότερα του ενός ζεύγους ακεραίων που ακολουθούν ένα γράμμα, τότε η μέθοδος θεωρεί ότι ισχύει η εντολή του προηγούμενου γράμματος. Στη γραμμή 25 του παραδείγματος βλέπουμε τη χρήση της μεθόδου κατασκευής μονοπατιών. Επιπλέον και σε αυτή την περίπτωση μπορούμε να μορφοποιήσουμε όπως θέλουμε τα χαρακτηριστικά του μονοπατιού. Το αποτέλεσμα είναι αυτό του σχήματος 3.5.



Σχήμα 3.5: Δημιουργία μονοπατιού.

Με τη βοήθεια αυτής της βιβλιοθήκης έχουν δημιουργηθεί πολλά προγράμματα οπτικοποίησης, εξειδικευμένα να οπτικοποιούν δεδομένα με τη μορφή γραφημάτων, με πολύ αποδοτικές μεθόδους και αλγόριθμους. Μερικά χαρακτηριστικά παραδείγματα τέτοιας υλοποίησης θα μελετηθούν στα επόμενα κεφάλαια. Είναι ωστόσο φανερό ότι τόσο η ευελιξία της *JavaScript*, όσο και οι δυνατότητες της βιβλιοθήκης *Raphaël* αποτελούν κίνητρο για την ανάπτυξη ισχυρών εφαρμογών οπτικοποίησης, που στο μέλλον θα αποτελούν θεμελιώδη εργαλεία έρευνας, μελέτης και μάθησης.

Κεφάλαιο 4

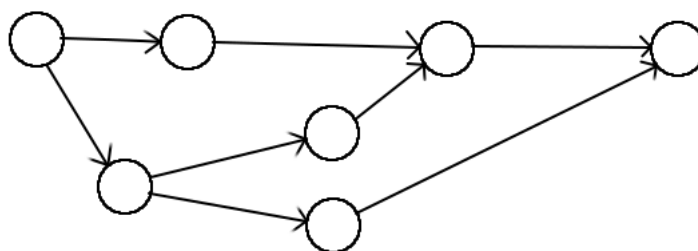
Ιστοριογραμμές

Σε αυτό το κεφάλαιο θα παρουσιαστεί αναλυτικά η έννοια της ιστοριογραμμής μέσα από τη μελέτη του ορισμού της, πώς αυτή λειτουργεί κατά πόσο είναι εύκολη η χρήση της καθώς και τις εφαρμογές που βρίσκει στους διάφορους τομείς της επιστήμης. Οι εφαρμογές των ιστοριογραμμών είναι ποικίλες και εκτείνονται σε πληθώρα τομέων. Στην προκειμένη εργασία θα μας απασχολήσει η εφαρμογή τους στην τεχνολογία και στις επιστήμες, κυρίως κατά τις ερευνητικές διαδικασίες και ειδικότερα στον τομέα των Ανθρωπιστικών Επιστημών. Είναι σημαντικό να σημειωθεί ότι η έννοια της ιστοριογραμμής διευρύνει τα όρια της θεωρίας γραφημάτων. Όμως, στα πλαίσια της παρούσας διπλωματικής, θα περιορίσουμε την έννοια αυτή σε απλούς γραφοθεωρητικούς όρους για να αναπτύξει ο αναγνώστης μια πιο σφαιρική άποψη επί του θέματος χωρίς πολλές κουραστικές λεπτομέρειες.

4.1 Ορισμός

Ιστοριογραμμή, συνήθως, ονομάζεται μια ακολουθία γεγονότων που συνδέονται με χρονικές ή αιτιοκρατικές σχέσεις. Τα γεγονότα και οι σχέσεις μεταξύ τους μπορούν να οπτικοποιηθούν μέσω κατευθυνόμενων, ακυκλικών γραφημάτων όπου τα μεν αντιπροσωπεύονται από κόμβους και οι δε από ακμές. Η γραφική τους αναπαράσταση πραγματοποιείται και πάλι με βάση τη χρονική σειρά, με ακολουθία από το παλαιότερο στο πιο πρόσφατο, ή τη σχέση αιτίας-αποτελέσματος, με τα αίτια πάντα να προηγούνται των αποτελεσμάτων κατά την τοποθέτηση στο γράφημα [11].

Ενας άλλος συναφής ορισμός είναι και ο εξής: Η ιστοριογραμμή είναι μια αφαιρε-



Σχήμα 4.1: Ένα απλό παράδειγμα του ορισμού της ιστοριογραμμής.

τική οπτικοποίηση πολύπλοκων αφηγήσεων ή ιστοριών που αφορούν συσχετίσεις μεταξύ ανθρώπων, αντικειμένων ή εννοιών. Η οπτικοποίηση αυτή προβάλλει ένα σύνολο χαρακτήρων σε ένα σύνολο επίπεδων καμπύλων, καθώς και τις ακολουθίες συναντήσεων των χαρακτήρων αυτών σε περιοχές όπου οι αντίστοιχες καμπύλες πλησιάζουν η μία την άλλη. Αυτού του είδους η οπτικοποίηση φαίνεται να έχει εφευρεθεί από τον Randall Munroe [10].

Με τους όρους “ιστορίες”, “χαρακτήρες”, “γεγονότα” και “σχέσεις” μπορεί να περιγραφεί μία ευρύτατη κλάση δεδομένων, ακόμα και αφηρημένων. Το γεγονός αυτό επιτρέπει μια πληθώρα γενικεύσεων που μπορούν να εφαρμοσθούν κατά περίπτωση για την οπτικοποίηση μεγάλου αριθμού δεδομένων με σκοπό την ευκολότερη εποπτεία τους. Ωστόσο, ένας γενικός ορισμός αυτών των εννοιών, θα διευκόλυνε την θεωρητική θεμελίωση αυτού του είδους των απεικονίσεων και κατ’ επέκταση θα αναδείκνυε νέες πιο αποδοτικές μεθόδους οπτικοποίησης γραφημάτων.

Τα κύρια χαρακτηριστικά μίας ιστοριογραμμής πρέπει να ταυτίζονται με αυτά της ιστορίας που οπτικοποιεί και να περιορίζεται σε αυτά. Αυτό είναι απαραίτητο για να διατηρείται η ιδιότητα ότι κάθε δύο διαφορετικές μεταξύ τους ιστορίες υποχρεωτικά πρέπει να έχουν διαφορετικές οπτικοποιήσεις. Είναι προφανές, λοιπόν, ότι πρέπει να ορισθεί με κάποια αυστηρότητα η έννοια της ιστορίας, βάσει κάποιων θεμελιωδών χαρακτηριστικών και ιδιοτήτων της.

Ιστορία ονομάζεται το ζεύγος ενός συνόλου από χαρακτήρες (φανταστικούς και μη) και μίας ακολουθίας από γεγονότα που με τη σειρά τους ορίζονται ως αλληλεπιδράσεις μεταξύ των χαρακτήρων που έχουν κάποια, νοητή ή μη, χρονική έκταση. Οι αλληλεπιδράσεις αυτές μπορούν να θεωρηθούν ως συναντήσεις ή και ως επικοινωνία μεταξύ ατόμων από ένα υποσύνολο των χαρακτήρων. Ο ορισμός αυτός διαθέτει πολλά κοινά με εκείνον του γραφήματος: το σύνολο των χαρακτήρων μπορεί να ταυτιστεί με ένα σύνολο των κόμβων και το σύνολο των αλληλεπιδράσεων των χαρακτήρων διαθέτει ιδιότητες

που θυμίζουν αυτές ενός συνόλου των ακμών κάποιου γραφήματος. Η σημαντικότερη διαφορά έγκειται στο γεγονός ότι το σύνολο των γεγονότων εξελίσσεται χρονικά, δηλαδή οι συσχετίσεις μεταξύ των χαρακτήρων αλλάζουν με το πέρασμα χρόνου. Αυτό το χαρακτηριστικό είναι που καθιστά την ιστοριογραμμή μία πολύ ενδιαφέρουσα γενίκευση της έννοιας του γραφήματος.

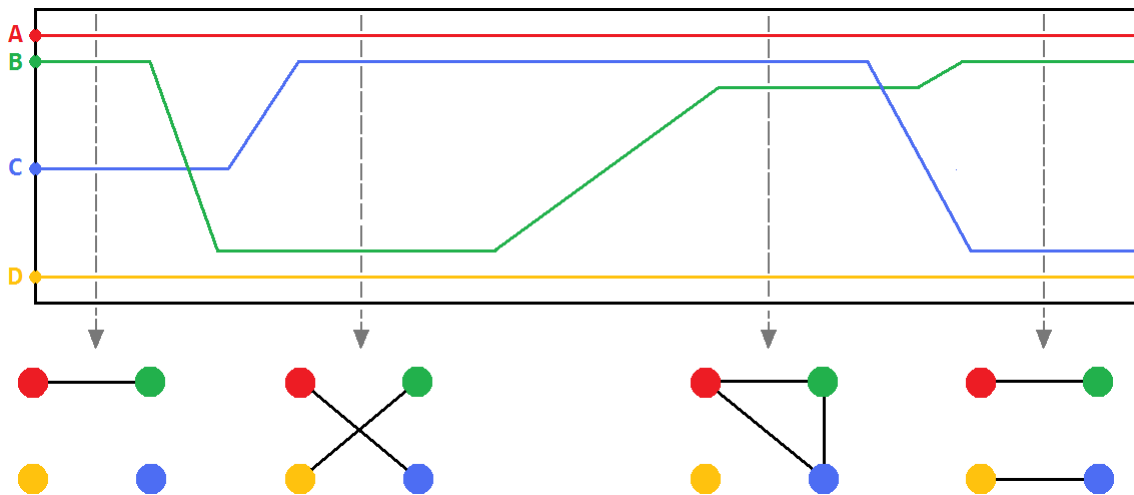
Τα γραφήματα ιστοριογραμμές δεν περιορίζονται μόνο σε περιπτώσεις όπου υπάρχει ανάγκη να μελετηθούν δεδομένα που αφορούν γεγονότα και άτομα. Μπορεί να γίνει μια πιο αφηρημένη προσέγγιση σε αυτά τα γραφήματα. Εν γένει τα γραφήματα αυτά κάθε αυτά είναι στατικές οντότητες, δηλαδή είναι δύσκολο, αλγοριθμικά και εποπτικά, να προβληθούν οι διάφορες μεταβολές πάνω στο ίδιο το γράφημα. Αντίθετα ένα γράφημα ιστοριογραμμή αποτελεί μία γενίκευση της έννοιας του γραφήματος έτσι ώστε να συμπεριλαμβάνει και την έννοια της μεταβολής-εξέλιξης. Όπως φαίνεται και στο σχήμα 4.2, υποθέτοντας ότι ο οριζόντιος άξονας είναι ο χρόνος και ότι στο κατακόρυφο άξονα φαίνονται οι συσχετίσεις μεταξύ των κόμβων, οι νοητές εγκάρσιες, κατά τη φορά του χρόνου, τομές μιας ιστοριογραμμής μπορούν να αποδοθούν ως ένα απλό γράφημα, όχι απαραίτητα συνεκτικό, το οποίο αποτελείται από κλίκες που αντιπροσωπεύουν τα σύνολα των επικοινωνούντων κόμβων. Ο παραπάνω τρόπος μελέτης των ιστοριογραμμών επιτρέπει την μεταπήδηση από τη μία απεικόνιση στην άλλη με σκοπό τη μελέτη της γενικής εικόνας εξέλιξης ενός γραφήματος και την στιγμιαία διαμόρφωσή του.

Στο σχήμα 4.2 παρουσιάζεται ένα παράδειγμα απεικόνισης ιστοριογραμμής για τέσσερις χαρακτήρες A , B , C και D . Γίνεται φανερό, μεταξύ άλλων ότι κατά μήκος όλου του γραφήματος υπάρχουν διάφορες αποστάσεις μεταξύ των παράλληλων τμημάτων των γραμμών. Υπάρχει εν γένει μία σταθερά που συνοδεύει κάθε απεικόνιση ιστοριογραμμής και αυτή αντιπροσωπεύει την ελάχιστη απόσταση μεταξύ παράλληλων τμημάτων του γραφήματος. Αυτή η απόσταση υποδεικνύει συνάντηση ή επικοινωνία μεταξύ των χαρακτήρων που αντιπροσωπεύονται από τα τμήματα αυτά. Δεν επιτρέπεται δύο παράλληλα τμήματα να διατηρούν απόσταση μικρότερη από την ελάχιστη, αλλά αντίθετα, επιτρέπεται να διατηρούν μεγαλύτερη απόσταση ώστε να υποδεικνύεται η απουσία ή η διάλυση κάποιας συνάντησης.

Η συνάντηση θεωρείται μεταβατική διμελής σχέση, δηλαδή αν για παράδειγμα μια ιστορία περιέχει τους χαρακτήρες A , B και C και στην ίδια χρονική στιγμή συμβαίνουν οι συναντήσεις (A, B) και (B, C) τότε συμβαίνει και η συνάντηση (A, C) την ίδια ακριβώς χρονική στιγμή. Επομένως δεν υπάρχει πρόβλημα αμφισημίας όταν απεικονίζονται τρία ή περισσότερα παράλληλα τμήματα να απέχουν, ανά δύο διαδοχικά, την ελάχιστη απόσταση.

Πρέπει να σημειωθεί ότι οι τομές μεταξύ των γραμμών του γραφήματος δεν έχουν κάποιο ιδιαίτερο νόημα και γενικά πρέπει να περιορίζονται ώστε η απεικόνιση να είναι το δυνατόν πιο ευανάγνωστη. Λόγω αυτού, προκύπτει και το πρόβλημα των τομών στις απεικονίσεις των ιστοριογραμμών (crossing minimization in storylines) και είναι παρόμοιο με το αντίστοιχο πρόβλημα που περιγράφηκε σε προηγούμενο κεφάλαιο.

Στο σχήμα 4.2, θεωρούμε ως άξονα του χρόνου τον οριζόντιο, και η αύξηση γίνεται από αριστερά προς τα δεξιά. Σε τέσσερις χρονικές στιγμές (γκρίζα διακεκομμένα βέλη), αντιστοιχούν απλά γραφήματα τα οποία αναπαριστούν τους συσχετισμούς μεταξύ των χαρακτήρων την εκάστοτε στιγμή. Γίνεται, έτσι φανερό, η μεταβλητότητα του αντίστοιχου απλού γραφήματος στο χρόνο.



Σχήμα 4.2: Παράδειγμα απεικόνισης ιστοριογραμμής.

Όπως αναφέρθηκε και προηγουμένως, ένα σημαντικό πρόβλημα που προκύπτει κατά την απεικόνιση των ιστοριογραμμών είναι η παρουσία τομών μεταξύ των γραμμών, οι οποίες δεν φέρουν κάποιο ιδιαίτερο νόημα και στην περίπτωση που το πλήθος τους είναι μεγάλο λαμβάνεται μία δυσανάγνωστη οπτικοποίηση. Επομένως είναι απαραίτητο να περιοριστούν, αν όχι όλες, ένα μεγάλο ποσοστό αυτών των τομών.

Εχει γραφτεί αρκετή βιβλιογραφία πάνω σε αυτό το πρόβλημα καθώς και σε προβλήματα στα οποία μπορεί να αναχθεί, όπως για παράδειγμα το πρόβλημα των γραμμών του Μετρό (Metro-line problem) όπου απαιτείται η πλήρης απουσία τομών. Ωστόσο, αποδεικνύεται ότι το παρόν πρόβλημα είναι γενικότερο και αρκετά πιο περίπλοκο αφού δεν υπάρχουν οι περιορισμοί στην τοπολογία των ιστοριογραμμών που είναι δεδομένοι στις γραμμές ενός μετρό [9]. Παρόλα αυτά, ευριστικές λύσεις με αρκετά καλά αποτελέσματα έχουν προταθεί και βελτιώνονται διαρκώς όπως η τεχνική των S. Liu και λοιπών [7] αλλά και οι τεχνικές των Y. Tanahashi και λοιπών [8].

4.2 Εφαρμογές

Από τους ορισμούς, οι ιστοριογραμμές μπορούν να παρουσιάσουν γεγονότα που αφορούν ένα σύνολο οντοτήτων εξοικονομώντας πρωτίστως χρόνο αλλά και χώρο, αφού ο ενδιαφερόμενος μπορεί να εντοπίσει το προς εξέταση γεγονός με μία γρήγορη σάρωση του γραφήματος της ιστοριογραμμής. Αυτός εξάλλου είναι και ο απώτερος σκοπός τους. Μέσω της οπτικοποίησης και της σφαιρικής θεώρησης των δεδομένων επιτυγχάνεται η ολοκλήρωση του ερευνητικού στόχου σε πολύ μικρότερο χρόνο, αφού στην περίπτωση αυτή οι ενέργειες, θα μπορούσαμε να πούμε, εκτελούνται σχεδόν αυτόματα. Η βασική δουλειά του ερευνητή είναι να εισάγει με προσοχή τα δεδομένα που έχει συλλέξει και τις υπόλοιπες διεργασίες τις αναλαμβάνουν τα προγράμματα οπτικοποίησης ιστοριογραμμών. Μέσα σε λίγα δευτερόλεπτα ταξινομούν, τοποθετούν με χρονολογική σειρά και οπτικοποιούν τα δεδομένα που τους παρέχει ο ερευνητής, προσφέροντας στον τελευταίο την δυνατότητα να επικεντρωθεί σε αυτό που πραγματικά τον ενδιαφέρει, δηλαδή την καταγραφή και σύγκριση των αποτελεσμάτων, ούτως ώστε να καταλήξει στα συμπεράσματα που χρειάζεται.

Οι ιστοριογραμμές βρίσκουν κυρίως εφαρμογές στην ιστορία, ή σε λογοτεχνικά και θεατρικά έργα, όπου οι αλληλεπιδράσεις μεταξύ των χαρακτήρων παίζουν καθοριστικό ρόλο για την πλοκή των γεγονότων. Σε τέτοιες περιπτώσεις οι ιστοριογραμμές μπορούν να εμπλουτιστούν με επιπλέον στοιχεία τα οποία θα παρουσιάζουν στον μελετητή πληροφορίες για τις βαθύτερες αιτίες των γεγονότων που διαδραματίζονται, όπως ο γεωγραφικός χώρος όπου θα βρεθούν ή οι πληροφορίες που θα ανταλλάξουν κάποιοι χαρακτήρες. Ωστόσο, γίνεται σαφές ότι η χρήση τους δεν περιορίζεται εκεί, αλλά υπάρχουν τομείς στους οποίους η εποπτεία των πληροφοριών και των σχέσεων που μοιράζεται ένα σύνολο ατόμων μέσα σε συγκεκριμένα χρονικά πλαίσια, κρίνεται υψηλής προτεραιότητας όπως για παράδειγμα στον τομέα της εγκληματολογίας όπου η εφαρμογή των ιστοριογραμμών θα αποτελούσε μια πιο αποδοτική μέθοδο διαλεύκανσης δύσκολων υποθέσεων.

Το διαδίκτυο βρίθει από εφαρμογές οπτικοποίησης ιστοριογραμμών. Κύριο χαρακτηριστικό των περισσότερων από αυτών είναι η προσπάθεια επίτευξης μίας διάταξης (layout) όπου ελαχιστοποιούνται οι τομές των ακμών και βελτιώνεται η οπτική παρουσίαση του γραφήματος. Ωστόσο οι υπάρχουσες μελέτες επί του θέματος είναι ελάχιστες και άρα κρίνεται αναγκαία η δημιουργία πιο εύχρηστων εφαρμογών και η έρευνα για ταχύτερους αλγορίθμους οπτικοποίησης ιστοριογραμμών. Η χρήση των τεχνολογιών που αναφέρθηκαν σε προηγούμενο κεφάλαιο είναι το κλειδί για καλύτερα αποτελέσματα στο μέλλον.

4.3 StorylineViewer

Στα πλαίσια της παρούσας διπλωματικής αναπτύχθηκε ένα απλό πρόγραμμα για την απεικόνιση ιστοριογραμμών. Είναι γραμμένο στη γλώσσα προγραμματισμού JavaScript και κάνει χρήση αντικειμένων και μεθόδων της βιβλιοθήκης Raphaël. Η φιλοσοφία του είναι να οπτικοποιήσει γρήγορα τα δεδομένα που του δίνονται ώστε ο χρήστης να λάβει μία πρώτη γενική εικόνα των πληροφοριών που προκύπτουν από την εφαρμογή διαφόρων αλγορίθμων οπτικοποίησης ιστοριογραμμών. Ο πλήρης κώδικάς του δίνεται στο παράρτημα Α της παρούσας εργασίας. Η χρήση του είναι αρκετά απλή. Αρχικά εισάγεται το JavaScript αρχείο σε HTML κώδικα με τον ίδιο τρόπο που εισάγονται και οι βιβλιοθήκες της JavaScript που αναφέρθηκαν σε προηγούμενο κεφάλαιο. Είναι σημαντικό να σημειωθεί ότι απαιτείται και η εισαγωγή της βιβλιοθήκης Raphaël στο ίδιο αρχείο για τη σωστή λειτουργία των μεθόδων του StorylineViewer.

Παρακάτω δίνεται κώδικας HTML μέσω του οποίου καλείται ο StorylineViewer. Στον κώδικα αυτόν, αφού εισάγουμε τα απαραίτητα στοιχεία, δημιουργούμε ένα αντικείμενο τύπου *div* καλώντας το αντίστοιχο tag, `<div>`, το οποίο δεσμεύει τον χώρο όπου θα τοποθετηθεί ένα αντικείμενο τύπου *paper* της βιβλιοθήκης Raphaël. Στη συνέχεια, στο *paper* θα σχεδιαστεί η ιστοριογραμμή, βάσει των δεδομένων που δίνονται. Τα δεδομένα στο παρόν παράδειγμα περιέχονται σε ένα ξεχωριστό αρχείο JavaScript (*data.js*) το οποίο επίσης εισάγεται στον κώδικα HTML. Η μορφή των δεδομένων εξηγείται παρακάτω.

```
1 <!DOCTYPE html>
2
3 <html>
4   <head>
5     <title> Storyline Viewer </title>
6     <meta charset="UTF-8"/>
7
8     <script src="raphael.min.js"></script>
9     <script src='StorylineViewer.js'></script>
10    <script src='data.js'></script>
11    <script>
12      window.onload = function() {
13        var viewer = new StorylineViewer('div', 700, 300,
14          actors, colors, meetings);
15        viewer.setBackground();
16        viewer.setActorMatrix();
17        viewer.visualize();
18      };
19    </script>
```

```

19     </head>
20
21     <body>
22         <div id='div'></div>
23     </body>
24
25 </html>

```

Κώδικας 4.1: Χρήση του προγράμματος StorylineViewer

Όπως συνηθίζεται, ο κώδικας του προγράμματος καλείται μετά το πέρας της φόρτωσης του υπόλοιπου κώδικα της ιστοσελίδας. Αυτό επιτυγχάνεται μέσω του παρακάτω κώδικα:

```

1     window.onload = function() {
2         ...
3     };

```

Στο εσωτερικό της συνάρτησης του παραπάνω κώδικα, αρχικά, δημιουργείται ένα νέο αντικείμενο τύπου *StorylineViewer* με πεδία ένα αλφαριθμητικό, το *id* attribute του *div* tag, δύο ακεραίους που αντιπροσωπεύουν το μήκος και το πλάτος του *paper* αντίστοιχα, δύο διανύσματα που περιέχουν τα ονόματα και τα επιθυμητά χρώματα (με τον δεκαεξαδικό τους κωδικό) των χαρακτήρων της ιστοριογραμμής αντίστοιχα και τέλος έναν πίνακα που περιέχει τις συναντήσεις των χαρακτήρων κάθε χρονική στιγμή. Επειτα καλείται η μέθοδος *setBackground(...)* του αντικειμένου *StorylineViewer* η οποία χρωματίζει την επιφάνεια σχεδιασμού και προβάλλει ένα πλέγμα τέτοιο ώστε οι οριζόντιες γραμμές απέχουν την ελάχιστη επιτρεπόμενη απόσταση της ιστοριογραμμής και οι κατακόρυφες υποδηλώνουν τις χρονικές στιγμές. Σκοπός της παραπάνω μεθόδου είναι να κάνει πιο ευανάγνωστο το σχεδιασμό της ιστοριογραμμής. Τέλος, καλούνται οι μέθοδοι *setActorMatrix()* και *visualize()* του αντικειμένου *StorylineViewer* οι οποίες υπολογίζουν τις θέσεις των γραμμών της ιστοριογραμμής και τις προβάλλουν αντίστοιχα.

```

1     var actors = ['Alice', 'Bob', 'Carol', 'Dave'];
2     var colors = ['#b30000', '#00b300', '#0000b3', '#00b3b2'];
3     var meetings = [ [ [1], [3] ],
4                     [ [1], [3] ],
5                     [ [0, 1, 3] ],
6                     [ [0, 1, 3] ],
7                     [ [0, 1, 3] ],
8                     [ [0], [1] ],
9                     [ [0], [1], [2, 3] ],
10                    [ [0], [1], [2, 3] ],
11                    [ [0, 1], [2, 3] ],

```

```

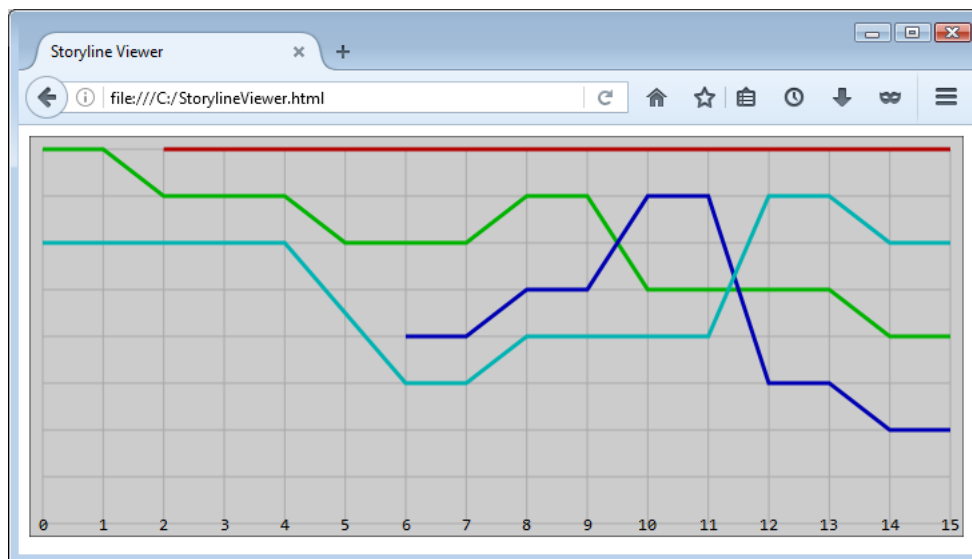
12     [ [0,1], [2,3]     ],
13     [ [0,2], [1,3]     ],
14     [ [0,2], [1,3]     ],
15     [ [0,3], [1], [2]   ],
16     [ [0,3], [1], [2]   ],
17     [ [0], [3], [1], [2] ],
18     [ [0], [3], [1], [2] ], ];

```

Κώδικας 4.2: Τα δεδομένα που εισάγονται στον StorylineViewer

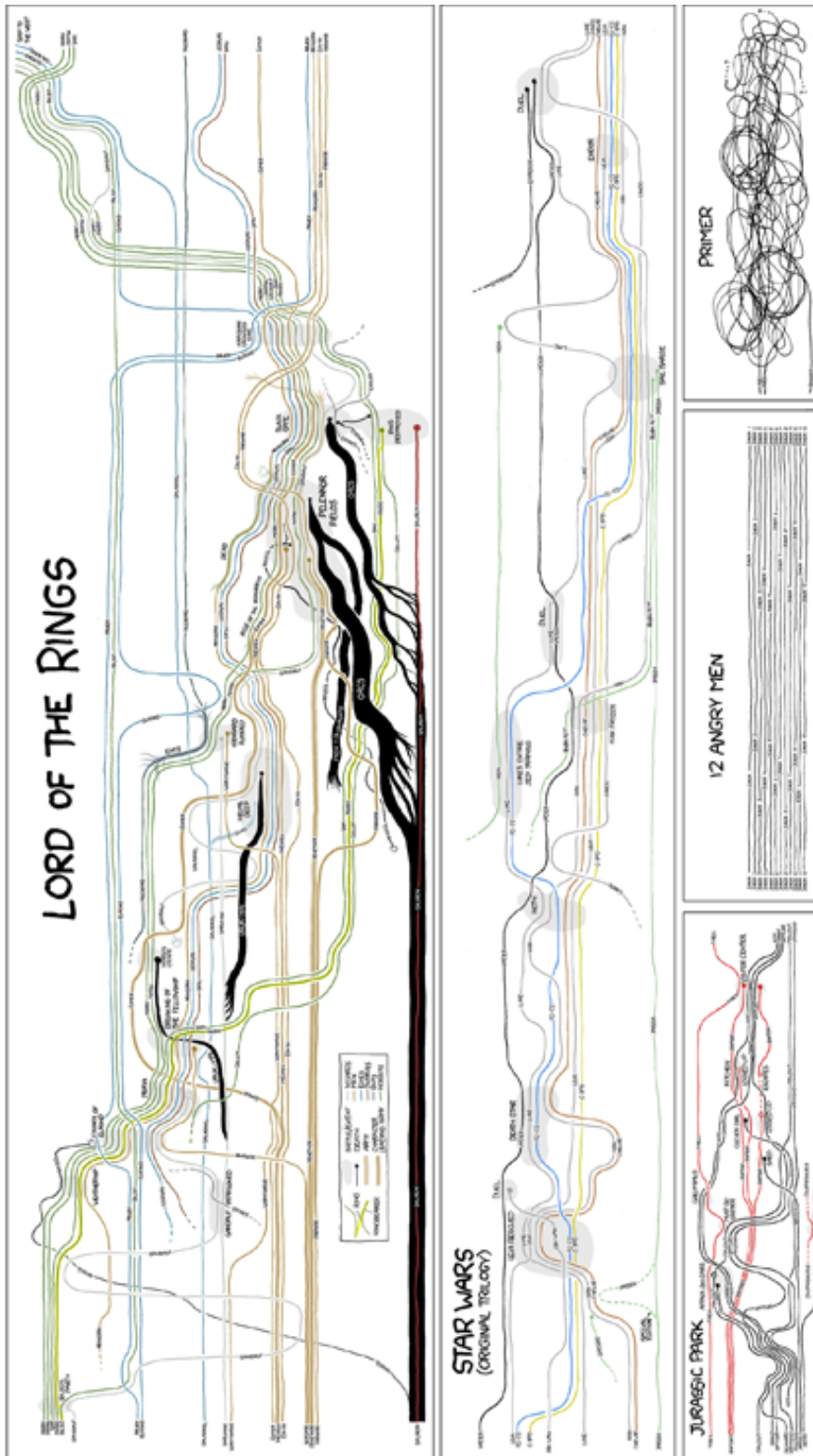
Στον κώδικα 4.2 περιέχονται τα δεδομένα που εισάγονται στο πρόγραμμα για την προβολή της ιστοριογραμμής. Ενδιαφέρον παρουσιάζει ο πίνακας των συναντήσεων (meetings) όπου καταγράφονται αναλυτικά για κάθε χρονική στιγμή οι κλίκες των χαρακτήρων που βρίσκονται σε επικοινωνία κάθε χρονική στιγμή. Ο πίνακας είναι τριδιάστατος και η πρώτη διάσταση αντιπροσωπεύει τη χρονική στιγμή, η δεύτερη τις κλίκες και η τρίτη τους χαρακτήρες που ανήκουν σε κάθε κλίκα επικοινωνίας. Με τη μέθοδο *setActorMatrix()* ο πίνακας αυτός διατρέχεται και δημιουργείται ένας νέος όπου αποθηκεύονται τα σημεία του *paper* από όπου θα διέρχεται η γραμμή κάθε χαρακτήρα ξεχωριστά και τελικά θα σχεδιάζεται μέσω της μεθόδου *visualize()*.

Το αποτέλεσμα που επιστρέφει το πρόγραμμα για τα παραπάνω δεδομένα δίνεται από την εικόνα 4.3.



Σχήμα 4.3: Αποτέλεσμα του StorylineViewer

THESE CHARTS SHOW MOVIE CHARACTER INTERACTIONS. THE HORIZONTAL AXIS IS TIME. THE VERTICAL GROUPING OF THE LINES INDICATES WHICH CHARACTERS ARE TOGETHER AT A GIVEN TIME.



Σχήμα 4.4: Πρώτη χρήση απεικόνισης ιστοριογραμμής από τον Randall Munroe.

Κεφάλαιο 5

Γραφήματα Παραλλαγών

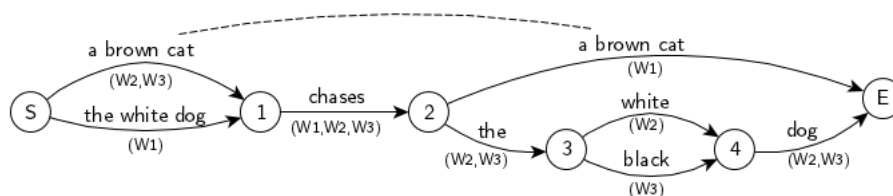
Η έννοια των γραφημάτων παραλλαγών εισήχθη για πρώτη φορά από τους D. Schmidt και R. Colomb 2009 [12] ως μία μέθοδος απεικόνισης παραλλαγών μεταξύ διαφορετικών εκδόσεων ενός κειμένου. Αυτού του είδους τα γραφήματα αποτελούν εύχρηστα εργαλεία ανάλυσης κειμένων και οι εφαρμογές τους ποικίλουν. Πέρα από την οπτικοποίηση των διαφορών μεταξύ εκδόσεων ενός κειμένου μπορούν να χρησιμοποιηθούν και ως μέσο ελέγχου λογοκλοπής, ευρετήριο συνώνυμων λέξεων, εργαλείο απλοποίησης κειμένων για άτομα με μαθησιακές δυσκολίες και πολλά άλλα.

5.1 Ορισμός

Ενα γράφημα παραλλαγών είναι ένα μέσο αναπαράστασης των παραλλαγών ενός κειμένου χρήσει ενός κατευθυνόμενου, ακυκλικού γραφήματος. Κάθε τέτοιο γράφημα διαθέτει έναν αρχικό (S) και έναν τελικό κόμβο (E). Κάθε παραλλαγή του κειμένου αναπαριστάται από ένα μοναδικό μονοπάτι από τον S στον E το οποίο περιλαμβάνει την ακολουθία των λέξεων που παράγει το κείμενο αυτό.

Ενα βασικό γνώρισμα αυτού του είδους γραφημάτων είναι ότι κάθε ακμή διαθέτει δύο πληροφορίες: (1) μια κοινή ακολουθία διαδοχικών λέξεων και (2) τις ταυτότητες των εκδόσεων στις οποίες αυτή απαντάται από κοινού στα ίδια σημεία. Επιπλέον οι κόμβοι αποτελούν σημεία ενοποίησης και σημεία διαχωρισμού των ακμών ανάλογα με το αν οι εκδόσεις που αντιστοιχούν οι ακμές αυτές συμπίπτουν ή διαφοροποιούνται αντίστοιχα.

Το γράφημα του σχήματος 5.1 παρουσιάζει όλα τα χαρακτηριστικά της δομής ενός



Σχήμα 5.1: Ένα απλό παράδειγμα γραφήματος παραλλαγών.

γραφήματος παραλλαγών. Εύκολα διακρίνονται η πορεία των κειμένων W_1 , W_2 και W_3 (ξεκινώντας από τον κόμβο S και καταλήγοντας στον κόμβο E) καθώς και σε ποια σημεία αυτά ταυτίζονται ή διαφοροποιούνται.

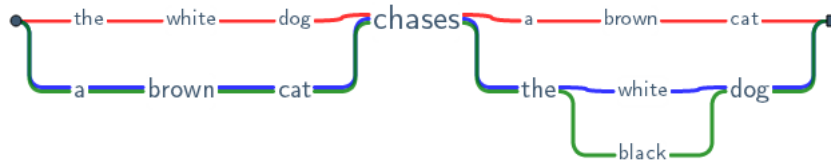
5.2 Εφαρμογές

Είναι γνωστό ότι ένα πεδίο έρευνας των Ανθρωπιστικών Επιστημών είναι και η σύγκριση του περιεχομένου παλαιών αλλά και σύγχρονων κειμένων (Textual Criticism). Ένας από τους σκοπούς αυτού του τομέα είναι η αναδόμηση της πρωτότυπης έκδοσης ενός κειμένου για την καλύτερη κατανόηση της εξέλιξής του μέσα στο χρόνο. Για τον σκοπό αυτό ο ερευνητής εξετάζει και καταγράφει τις ομοιότητες και τις διαφορές μεταξύ ενός αριθμού υποδειγμάτων. Αυτή η πρακτική του τομέα είναι γνωστή ως *σύγκριση* [14]. Η χρήση, λοιπόν, ψηφιακών εφαρμογών οπτικοποίησης κειμενικών παραλλαγών επιταχύνει την εξέταση μεγάλου αριθμού υποδειγμάτων, καθιστώντας έτσι τον ερευνητή ικανό να επικεντρωθεί στην μελέτη των αναδομημένων πρωτοτύπων. Μια τέτοια εφαρμογή αποτελεί και το *CollateX Software*.

Το *CollateX* είναι ένα λογισμικό ανοικτού κώδικα το οποίο λαμβάνοντας τις διαφορετικές εκδόσεις δημιουργεί ένα γράφημα παραλλαγών με δομή όπως αυτή του ορισμού και οπτικοποίηση παρόμοια με αυτή του 5.1. Ωστόσο, σε αντίθεση με άλλες εφαρμογές, κατά την ανάπτυξη του *CollateX* δόθηκε ιδιαίτερη έμφαση στην βελτιστοποίηση των μεθόδων παράθεσης (*alignment*) των στοιχείων του γραφήματος, πετυχαίνοντας έτσι την καλύτερη δυνατή οπτικοποίησή του. Πιο συγκεκριμένα η εφαρμογή προσπαθεί να ικανοποιήσει με όσο το δυνατόν αποδοτικότερο τρόπο τρία βασικά αιτήματα για τον υπολογισμό της παράθεσης των στοιχείων ενός γραφήματος παραλλαγών: (1) να ελέγχει για πιθανές μεταθέσεις τμημάτων κειμένου ανάμεσα στις διαφορετικές εκδόσεις, (2) να υποστηρίζει το ευέλικτο ταίριασμα μεταξύ δειγμάτων που θεωρούνται συνώνυμα και τέλος (3) να επιστρέφει το ίδιο αποτέλεσμα ανεξάρτητα από τη σειρά των συγκρινόμενων εκδόσεων [13]. Ωστόσο η οπτικοποίηση που παρέχει το *CollateX* περιορίζεται στην

αυστηρή δομή των γραφημάτων παραλλαγών με αποτέλεσμα και καθίσταται συχνά μη ικανοποιητική για μεγάλους αριθμούς δειγμάτων.

Το πρόβλημα αυτό επιλύει δραστικά το *TRAViz project*. Το *TRAViz* (Text Re-use Alignment Visualization) είναι μια βιβλιοθήκη της *JavaScript*, βασισμένη στην βιβλιοθήκη *Raphaël*, που έχει ως στόχο να οπτικοποιήσει τις μεταβολές μεταξύ διαφορετικών εκδόσεων ιστορικών και μοντέρνων κειμένων [14]. Δεν διαθέτει μεθόδους συγκρίσιμες με αυτές του *CollateX*, όμως πλεονεκτεί στον τρόπο με τον οποίο οπτικοποιεί τα γραφήματα παραλλαγών. Αυτό το πετυχαίνει υιοθετώντας έναν νέο ορισμό για τα γραφήματα παραλλαγών: οι κόμβοι πλέον αντιπροσωπεύουν τμήματα κειμένου ενώ οι ακμές τις ταυτότητες των διαφόρων εκδόσεων. Όταν δύο ακμές διέρχονται από τον ίδιο κόμβο σημαίνει ότι δύο εκδόσεις ταυτίζονται σε εκείνο ακριβώς το τμήμα κειμένου που αντιπροσωπεύει ο κόμβος αυτός. Επιπλέον σε αυτή την περίπτωση το μέγεθος του κόμβου αυξάνεται για να δοθεί και οπτικά ο αριθμός των δειγμάτων που ταυτίζονται στο σημείο αυτό. Το *TRAViz* δίνει την δυνατότητα να αναπαριστώνται οι κόμβοι στο γράφημα από το τμήμα κειμένου που αντιπροσωπεύουν καθώς και οι ταυτότητες των διαφόρων δειγμάτων να παρουσιάζονται όχι ως ετικέτες αλλά ως χρωματισμοί των ακμών. Το αποτέλεσμα της εφαρμογής του *TRAViz* στο παράδειγμα της εικόνας 5.1 δείνει την εικόνα 5.2.



Σχήμα 5.2: Παράδειγμα οπτικοποίησης από το *TRAViz*.

Κεφάλαιο 6

Μελλοντικές Εφαρμογές

Στα προηγούμενα κεφάλαια περιγράφηκαν λεπτομερώς οι εφαρμογές οπτικοποίησης ιστοριογραμμών και γραφημάτων παραλλαγών στον τομέα των Ανθρωπιστικών Επιστημών, αποδεικνύοντας πόσο χρήσιμοι μπορούν να φανούν οι παραπάνω τύποι γραφημάτων στην επιστημονική έρευνα. Στο παρόν κεφάλαιο θα εξετάσουμε τις περαιτέρω εφαρμογές που δύνανται να βρουν οι οπτικοποιήσεις γραφημάτων σε άλλους τομείς της επιστήμης, ακόμα και της καθημερινής ζωής, με στόχο πάντα την προαγωγή της έρευνας και τη διευκόλυνση του ανθρώπου. Στο διαδίκτυο, για παράδειγμα, χρησιμοποιούνται κατά κόρον τα γραφήματα, όταν πρόκειται για αναζήτηση και παρουσίαση πληροφοριών ή και εικόνων. Χάρη σε αυτά τα γραφήματα, παρέχεται η δυνατότητα στο χρήστη να αναζητήσει σε ελάχιστο χρόνο συναφείς μεταξύ τους εικόνες για ένα συγκεκριμένο γεγονός, ανάλογα με το χρόνο που έχουν «ανεβεί» στο διαδίκτυο. Έτσι η οπτικοποίηση αυτών των γραφημάτων δίνει την δυνατότητα βελτίωσης των ζητούμενων αποτελεσμάτων από τους χρήστες.

Επιπρόσθετοι επιστημονικοί τομείς τους οποίους θα ήταν εφικτό να εξυπηρετήσουν οι οπτικοποιήσεις γραφημάτων είναι αυτοί της Αστρονομίας και της Κοσμολογίας. Αντικείμενο τους είναι να ερευνούν οτιδήποτε σχετίζεται με τα ουράνια σώματα και τα φαινόμενα που προκαλούνται από τη μεταξύ τους αλληλεπίδραση, όπως για παράδειγμα στο φαινόμενο της σύγκρουσης δύο γαλαξιών. Για την μελέτη τέτοιων καταστάσεων απαιτούνται ισχυρά προγράμματα προσομοίωσης που βασιζόμενα σε τεχνολογίες οπτικοποίησης θα δίνουν ακριβέστερα αποτελέσματα σε μικρότερο χρονικό διάστημα.

Εξίσου χρήσιμα αποτελέσματα μπορούν να παρέχουν τα γραφήματα και στην Ιατρική. Αυτό διότι η γραφική απεικόνιση προσφέρεται για την καταγραφή των συμπτωμάτων διαφόρων ασθενειών, καθιστώντας έτσι δυνατή την εύρεση των κοινών συμπτωμά-

των μεταξύ τους. Ακόμα, με τον τρόπο αυτό συντομεύει τη διάγνωση, αφού ο θεράπων ιατρός μπορεί απλώς να αναζητήσει ένα ή παραπάνω συμπτώματα και τα γραφήματα να τον οδηγήσουν στις πιθανές παθήσεις που αυτά υποδηλώνουν. Τέλος, θα ήταν χρήσιμα για την καταγραφή της πορείας μίας ασθένειας, των εξάρσεων και υφέσεων της, της αποτελεσματικότητας συγκεκριμένων φαρμάκων στον ασθενή μέσα από την παρακολούθηση της εξέλιξης της υγείας του. Αρα η οπτικοποίηση γραφημάτων μπορεί να προωθήσει την βελτίωση των μεθόδων διάγνωσης αυξάνοντας έτσι τα ποσοστά επιτυχίας των θεραπειών.

Μία άλλη πιθανή εφαρμογή των οπτικοποιημένων γραφημάτων θα ήταν στην καθημερινή ζωή, στον τομέα της εκπαίδευσης, προκειμένου να επιτευχθεί η ευκολότερη απομνημόνευση και κατ' επέκταση η καλύτερη εμπέδωση του διδασκόμενου αντικειμένου. Το μάθημα που θα μπορούσε να εκμεταλλευτεί με τον βέλτιστο τρόπο τις δυνατότητες των γραφημάτων αυτών είναι η Ιστορία. Ο εκπαιδευτικός, χρησιμοποιώντας πάντα απλουστευμένες μορφές των γραφημάτων, έχει την ευκαιρία να τοποθετήσει εικόνες ή σημεία-κλειδιά του μαθήματος της ημέρας με χρονολογική σειρά, με παράλληλη επισήμανση των σχέσεων ανάμεσα σε πρόσωπα και γεγονότα. Οι ειδικοί παιδαγωγοί, καθώς και οι γονείς παιδιών με ειδικές ανάγκες μπορούν να επωφεληθούν εξίσου από τις γραφικές οπτικοποιήσεις, διαμορφώνοντας, πάλι με τη χρήση εικόνων, το καθημερινό πρόγραμμα των παιδιών με ακρίβεια, ούτως ώστε να είναι πιο απτό, πιο πρακτικό, να το κατανοούν πιο άμεσα και να το τηρούν πιο εύκολα. Επίσης, μπορούν να χρησιμεύσουν στη δημιουργία ασκήσεων και εργασιών με κλίμακα δυσκολίας: ο μαθητής ξεκινά από τις πιο απλές και καθώς τις λύνει, προχωρά αυτόματα στις επόμενες, ανωτέρου επιπέδου, όπως έχουν ορισθεί και τοποθετηθεί στο πρόγραμμα από τον εκπαιδευτικό. Έτσι η όλη μαθησιακή διαδικασία αποκτά παιγνιώδη χαρακτήρα, άρα γίνεται πιο ευχάριστη για τους μαθητές, ενώ παράλληλα εξοικονομείται χρόνος και κόπος από την πλευρά του εκπαιδευτικού, ο οποίος επιπλέον μπορεί να έχει τις εργασίες/ασκήσεις μαζί του ανά πάσα στιγμή, εφόσον είναι σε ηλεκτρονική μορφή.

Τέλος η λειτουργία των μέσων κοινωνικής δικτύωσης διέπεται σε μεγάλο βαθμό από κανόνες της θεωρίας γραφημάτων. Σε αυτά τα μέσα προβάλλονται με χρονολογική σειρά οι δραστηριότητες των χρηστών τους, δίνοντάς τους συγχρόνως τη δυνατότητα να περιηγηθούν σε αυτές και να τις επεξεργαστούν αναδρομικά. Έτσι η μελέτη ενός τέτοιου δικτύου σχέσεων μεταξύ ατόμων και ανταλλαγής πληροφορίας απαιτεί μεθόδους οπτικοποίησης γραφημάτων.

Κεφάλαιο 7

Συμπεράσματα

Στην παρούσα εργασία, παρουσιάστηκαν οι τρέχουσες τεχνολογίες οπτικοποίησης που βασίζονται στη γλώσσα προγραμματισμού *JavaScript* και σε μια ιδιαίτερα χρηστική βιβλιοθήκη της, την *Raphaël*. Δόθηκε έμφαση σε δύο είδη γραφημάτων με πολλές εφαρμογές, τις ιστοριογραμμές και τα γραφήματα παραλλαγών, που απαιτούν αποδοτικές μεθόδους οπτικοποίησης και δόθηκαν παραδείγματα εφαρμογών που εκμεταλλεύονται αυτές τις τεχνολογίες για να πετύχουν αυτόν τον στόχο. Κυρίως εξετάστηκαν οι δυνατότητες των προγραμμάτων *CollateX* και *TRAViz* τα οποία εισάγουν καινοτόμες ιδέες στην οπτικοποίηση των γραφημάτων παραλλαγών. Επιπλέον στα πλαίσια της παρούσας διπλωματικής εργασίας δημιουργήθηκε μια εφαρμογή με σκοπό την προβολή απλών γραφημάτων ιστοριογραμμών, χρήση της γλώσσας *JavaScript*. Είναι φανερό ότι η εργασία αυτή αποτελεί μία συγκεντρωτική αναφορά των τρεχουσών τεχνολογιών και μελετών επί του θέματος των οπτικοποιήσεων. Σε καμία περίπτωση η περιεχόμενες έννοιες και μέθοδοι δεν πρέπει να περιορίζουν την αναζήτηση διαφορετικών ιδεών και τρόπων. Αντιθέτως, επιβάλλεται να μελετηθούν αυτές οι τεχνολογίες και να βελτιωθούν.

Παράρτημα Α'

Κώδικας Εφαρμογής StorylineViewer

```
1 function StorylineViewer(id, width, height, actors, colors, meetings) {
2   this.p = new Raphael(id, width, height);
3   this.actors = actors;
4   this.colors = colors;
5   this.meetings = meetings;
6
7   this.x0 = 10;
8   this.y0 = 10;
9   this.dt = (this.p.width-2*this.x0)/(this.meetings.length-1);
10  this.maxDist = (this.p.height-2*this.y0)/(this.actors.length);
11  this.minDist = this.maxDist/2;
12
13  this.setBackground = function(backgroundColor='#cccccc', showGrid=
14    true, gridColor='#aaaaaa') {
15    this.p.rect(0, 0, this.p.width, this.p.height)
16      .attr({
17        'fill': backgroundColor
18      });
19    if(showGrid) {
20      for(var r=0; r<=Math.floor((this.p.height-2*this.y0)/this.
21        minDist); r++) {
22        this.p.path('M '+this.x0+' '+r*this.minDist+this.y0)
23          +' L '+this.p.width-this.x0+' '+r*this.minDist+
24            this.y0)
25          .attr({
26            'stroke': gridColor
27          });
28      }
29    }
30  }
```

```
25     for(var c=0; c<=Math.floor((this.p.width-2*this.x0)/this.dt
26         ); c++) {
27         this.p.path('M '+ (c*this.dt+this.x0)+' '+ (this.y0)+' L
28             '(c*this.dt+this.x0)+' '+ (this.p.height-this.y0))
29             .attr({
30                 'stroke': gridColor
31             });
32         this.p.text(c*this.dt+this.x0, this.p.height-this.y0, c
33             )
34             .attr({
35                 'font': '10pt Consolas'
36             });
37     }
38     this.actorMatrix = [];
39
40     this.setActorMatrix = function() {
41         for(var i=0; i<this.meetings.length; i++) {
42             var tempArray = [];
43             for(var j=0; j<this.actors.length; j++) {
44                 tempArray.push(-1);
45             }
46             this.actorMatrix.push(tempArray);
47         }
48
49         var tempPoint = new Point(this.x0, this.y0);
50         for(var i=0; i<this.meetings.length; i++) {
51             for(var j=0; j<this.meetings[i].length; j++) {
52                 for(var k=0; k<this.meetings[i][j].length; k++) {
53                     this.actorMatrix[i][this.meetings[i][j][k]] = new
54                         Point(tempPoint.x, tempPoint.y);
55                     tempPoint = new Point(tempPoint.x, tempPoint.y+this
56                         .minDist);
57                 }
58                 tempPoint = new Point(tempPoint.x, tempPoint.y+this.
59                     minDist);
60             }
61             tempPoint = new Point(tempPoint.x+this.dt, this.y0);
62         }
63     };
64
65     this.visualize = function() {
```



```
64     var info = this.p.text(0, 20, '').attr({'font': '16pt Consolas'  
65     });  
66     for(var col=0; col<this.actorMatrix[0].length; col++) {  
67         var path = 'M ';  
68         for(var row=0; row<this.actorMatrix.length; row++) {  
69             path += ( this.actorMatrix[row][col]==-1 ? '' : this.  
70                 actorMatrix[row][col].toString()+ 'L ' );  
71         }  
72         this.p.path(path)  
73         .attr({  
74             'stroke': (this.colors[col]),  
75             'stroke-width': 3  
76         })  
77         .data('id', this.actors[col])  
78         .data('color', this.colors[col])  
79         .tooltip(info)  
80         .mouseover(function() {  
81             this.animate({'stroke-width': 7}, 500, 'elastic');  
82             info  
83                 .attr({  
84                     'text': this.data('id'),  
85                     'fill': this.data('color')  
86                 });  
87         })  
88         .mouseout(function() {  
89             this.animate({'stroke-width': 3}, 500, 'elastic');  
90             info.attr({  
91                 'text': ''  
92             });  
93         });  
94     }  
95 }  
96 function Point(x, y) {  
97     this.x = x;  
98     this.y = y;  
99 }  
100 this.toString = function() {  
101     return ((this.x)+' '+(this.y)+' ');  
102 };  
103 }  
104 }  
105 Raphael.el.tooltip = function (tp) {  
106     this.tp = tp;
```

```
107     this.tp.ox = 0;
108     this.tp.oy = 0;
109     this.tp.hide();
110     this.hover(
111         function(event) {
112             this.mousemove( function(event) {
113                 this.tp.translate(event.clientX-this.tp.ox, event.
114                     clientY-this.tp.oy);
115                 this.tp.ox = event.clientX;
116                 this.tp.oy = event.clientY;
117             });
118             this.tp.show().toFront();
119         },
120         function(event) {
121             this.tp.hide();
122             this.unmousemove();
123         }
124     );
125     return this;
126 };
```

Κώδικας Α'.1: Ο κώδικας της εφαρμογής Storyline Viewer.

Βιβλιογραφία

- [1] R. J. Wilson
Graph Theory, ISBN: 0-582-24993-7
- [2] G. Di Battista, P. Eades, R. Tamassia, I. G. Tollis
Graph Drawing: Algorithms for the Visualization of Graphs, ISBN: 0-1330-1615-3
- [3] Vladimir P. Korzhik, Bojan Mohar
Minimal obstructions for 1-immersions and hardness of 1-planarity testing (2009)
- [4] M. R. Garey, D. S. Johnson
Crossing number is NP-complete (1983)
- [5] Michael R. Garey, David S. Johnson
Computers and Intractability: A Guide to the Theory of NP-Completeness (1979), ISBN 0-7167-1045-5
- [6] <https://angularjs.org/>, <https://docs.angularjs.org/guide>
AngularJS Developer Guide
- [7] Shixia Liu, Yingcai Wu, Enxun Wei, Mengchen Liu, Yang Liu
StoryFlow: Tracking the evolution of stories
- [8] Yuzuru Tanahashi, Kwan-Liu Ma
Design considerations for optimizing storyline visualizations
- [9] M. Gronemann, M. Jünger, F. Liers, Francesco Mambelli
Crossing Minimization in Storyline Visualization
- [10] Thomas C. van Dijk, M. Fink, N. Fischer, F. Lipp, P. Markfelder, A. Ravsky, S. Suri, A. Wolff
Block Crossing in Storyline Visualizations (2016)

- [11] K. Gunhee, E. P. Xing
Reconstructing Storyline Graphs for Image Recommendation from Web Community Photos
- [12] D. Schmidt, R. Colomb
A Data Structure For Representing Multi-Version Texts Online
- [13] R. H. Dekker, G. Middell
Computer-Supported Collation with CollateX
- [14] S. Jänicke, A. Geßner, G. Franzini, M. Terras, S. Mahony, G. Scheuermann
TRAVis: A Visualization for Variant Graphs