Θέματα μεταπτυχιακών & διπλωματικών εργασιών για το ακ. έτος 2023- 2024

Π. Βασιλειαδης – 2023-02-02

Τα θέματα διπλωματικών και μεταπτυχιακών εργασιών συνήθως επεκτείνουν κάποια από τα υπάρχοντα εργαλεία που αναπτύσσουμε στην ομάδα μου – λίγο πιο σπάνια, ξεκινούν νέα εργαλεία.

https://github.com/DAINTINESS-Group

https://www.youtube.com/playlist?list=PL3G-N7ZzyiDfpsMLCQcm_L9KEVLDkC454

Το παρόν κείμενο περιγράφει τα προτεινόμενα θέματα και, για να διευκολύνει την κατανόηση, δίνει και μια γενική εικόνα για τα εργαλεία που αναπτύσσουμε.

Αναγκαστικά μια εργασία πρέπει να ολοκληρωθεί αυστηρά εντός ενός έτους από την ανάληψή της. Απαιτούμενα προσόντα είναι η πολύ καλή γνώση Java & σχεδίασης ΟΟ λογισμικού. Στις περισσότερες διπλωματικές χρειάζεται να μπορείτε να συνδυάσετε frameworks – e.g., Apache Spark, και ενίοτε να πρέπει να δουλέψετε με το συνδυασμό κώδικα και βάσεων δεδομένων. Επειδή θα ρωτήσετε: πουθενά δε γράφει Spring & Spring Boot. Θα δούμε όπου κολλάει να μπει web-based front-end πώς μπορεί να κολλήσει στο back-end της διπλωματικής.

Γενικά προχωράμε ως ομάδα με agile working methods, ήτοι **weekly runs & group meetings**. Κείμενο και κώδικας γράφονται εναλλάξ. Παντού δουλεύουμε github & Eclipse.

Contents

Σημειογραφία: ό,τι θα δείτε με …

- ~~Strikethrough~~, είναι έτοιμο.

- Υπογραμμισμένο με τελείες, είναι μέρος του συστήματος υπό ανάπτυξη/προς βελτίωση.

- **Bold** είναι για τις εκφωνήσεις εργασιών στο παρόν κείμενο.

- αχνά γκρι γράμματα είναι μέρος του σχεδίου, αλλά όχι για τώρα.

1 Data Science

1.1 Πυθία: πες μου όλα όσα μπορείς για ένα data set

ΠΕΡΙΛΗΨΗ: Το σύστημα Pythia κάνει ένα αυτόματο profiling ενός συνόλου δεδομένων και παραγωγή του σχετικού report https://github.com/DAINTINESS-Group/Pythia

ΣΥΝΤΟΜΗ ΠΕΡΙΓΡΑΦΗ: Ένα τυπικό data set δεν είναι παρά ένα αρχείο κειμένου, με γραμμές, όπου κάθε γραμμή είναι μια εγγραφή, τα πεδία της οποίας χωρίζονται με κάποιο διαχωριστικό (p.x., tabs, comma, pipe, …). Θέλουμε, αφού εγγράψουμε ένα data set, την 100% αυτόματη παραγωγή ενός στατιστικού προφίλ για το data set, καθώς και την υποστήριξη διαδραστικών αιτημάτων profiling.

ΕΠΙΠΕΔΟ: *Από τις απαιτήσεις που παρατίθενται στη συνέχεια με bold θα προκύψουν 1 ~~αρκετές~~ Διπλωματικές/MSc ~~(αλλά όχι όλες μαζί ταυτοχρόνως)~~.*

ΠΛΑΤΦΟΡΜΑ ΕΡΓΑΣΙΑΣ: Java

ΚΕΝΤΡΙΚΗ ΙΔΕΑ: Το σύστημα Pythia, που κάνει αυτόματα ένα data profile για ένα αρχείο με δεδομένα, αναπτύσσεται από το εργαστήριο εδώ και λίγο καιρό. Οι διπλωματικές που προτείνονται εδώ συνεχίζουν δουλειά που φτιάχτηκε σε προηγούμενες διπλωματικές.

Αρχικά:

- ~~Ο χρήστης θα πρέπει να εγγράψει ένα data set και με τη βοήθεια του συστήματος να δηλώσει τα πεδία του, και τον τύπο τους (int, double, dateTime, Boolean, enum of class labels, …).~~ Θα ονομάζουμε labeled πεδία, αυτά για τα οποία υπάρχει ένα συγκεκριμένο, πεπερασμένο σύνολο τιμών (π.χ., οι μήνες ή οι μέρες της εβδομάδας – σαν enum δλδ). Επίσης, αν έχει κάποιο ειδικό ενδιαφέρον για κάποια πεδία ως labeled, ο αναλυτής θα πρέπει να το δηλώσει.

- **Ημι-αυτόματη εκτίμηση από το σύστημα για τον τύπο των στηλών. Το σύστημα κάνει ένα μικρό sample από γραμμές του αρχείου (π.χ., τις 20 πρώτες, ή 100 τυχαίες γραμμές), και για κάθε στήλη, για κάθε τύπο δεδομένων, βγάζει ένα σκορ για το κατά πόσο η στήλη πληροί το πρότυπο (π.χ., μόνο αριθμητικοί χαρακτήρες, αριθμ. χαρακτήρες με ή χωρίς υποδιαστολή, τυπικές εκφράσεις τύπου yyyy/mm/dd ή ddd/M/yy κοκ με όρια τιμών σε κάθε υπο-συμβολοσειρά, κοκ). Στο τέλος, το σύστημα αναφέρει στον αναλυτή την εκτίμηση για τον πλειοψηφούντα τύπο ανά πεδίο και αυτός αποφασίζει.**

Στη συνέχεια, το σύστημα, απολύτως αυτόματα, θα πρέπει να κάνει τα παρακάτω (τα οποία παρέχονται έτοιμα από το Spark)

- ~~Προφίλ για τις κολώνες (δλδ., ιστόγραμμα τιμών, min, max, median, mean, stdev, …)~~
- ~~Υπολογισμός της συσχέτισης στηλών (correlation)~~
- ~~Αν ο χρήστης έχει δηλώσει κανόνες labeling για μια στήλη, χαρακτηρισμός των τιμών αυτής της στήλης και παραγωγή της νέας labeled στήλης~~
- ~~Αυτόματο τρέξιμο dominance patterns && παραγωγή των σχετικών highlights~~
- Clustering των εγγραφών του data set και αποτίμηση της ποιότητας του clustering
- **Επέκταση του προφίλ του αρχείου (& αναφορά στα statistical_report.* αρχεία) με τα χαρακτηριστικά του όλου dataset: αριθμός γραμμών, file path, file size, datetime of profiling.**

- Επέκταση του προφίλ των στηλών (& αναφορά στα statistical_report.* αρχεία) με τα εξής χαρακτηριστικά για κάθε στήλη: αρ. null τιμών, αρ. διακριτών τιμών, mode, ισοϋψές ιστόγραμμα με τη μορφή quartiles (https://en.wikipedia.org/wiki/Quartile) (για μια απλή εκδοχή της ανακάλυψης regular expressions in databases, see also https://drive.google.com/drive/folders/1WK0Lht1OMopgYcZ8JHBFQQRfa-5KYKeQ)

Για κάθε labeled πεδίο (ή/και για κάθε πεδίο με λιγότερες από 10 τιμές στο πεδίο τιμών του):

- ~~Αυτόματο τρέξιμο decision trees για κάθε labeled field στη βάση όλων των πεδίων του data set. Αν ο χρήστης έχει δηλώσει ότι για θέλει μόνο κάποια πεδία να εμπλακούν στην παραγωγή ενός συγκεκριμένου decision tree, τότε εμπλέκονται μόνο αυτά.~~

Για κάθε αριθμητικό πεδίο (και αν υπάρχουν παραπάνω από ένα τέτοια πεδία)

- ~~Multiple linear regression για τα αριθμητικά attributes του dataset, και αποτίμηση της ποιότητας της υπολογισθείσας regression (επίσης μπορεί κανείς να σκεφτεί και την απλή linear regression με μόνο ένα input πεδίο, καθώς και τα σχετικά scatterplots) [βλ. και https://www.youtube.com/watch?v=29rjWClT_3U + follow-up links για βελτιστοποιήσεις]~~

Interactive, what-if analysis

- Hypothesis testing / causality. Αν υποθέσουμε ότι το αρχείο έχει κατασκευάσει ήδη ένα ή περισσότερα decision tree για κάποιο labeled feature, να μπορούμε διαδραστικά πλέον (όχι αυτόματα) να…
    a. Αίτιο-Αιτιατό: Δοθείσης μια υποθετικής state από τιμές για διάφορα πεδία, πόσο πιθανό είναι να προκύψει ένα target label ?
    b. Αιτιατό-Αίτιο (αντίστροφος στόχος): Δοθέντος ενός target label ποια είναι τα πιο σημαντικά state από τιμές για διάφορα πεδία που οδηγούν εκεί?
    c. Ανάλυση Ευαισθησίας: Δοθείσης μια υποθετικής state από τιμές για διάφορα πεδία, και μιας αλλαγής στην τιμή σε ένα πεδίο πόσο πιο πιθανό είναι να προκύψει ένα target label ?
    d. Αιτιατό-Αίτιο με κατάσταση εκκίνησης: Δοθείσης μια υποθετικής state από τιμές για διάφορα πεδία, και ενός target label, ποιες αλλαγές πρέπει να κάνουμε για να πετύχουμε το label?

  [Ghathani et al. Αν υποθέσουμε ότι το αρχείο έχει κατασκευάσει ήδη ένα decision tree για κάποιο labeled feature, CIDR22, https://www.youtube.com/watch?v=an-oTVQzHT0]
  Πρακτικά χρειαζόμαστε ένα βοηθητικό εργαλείο, που να βγάζει από την Πυθία τα όποια στατιστικά ή δέντρα απόφασης και να τρέχει τους σχετικούς what-if algorithms.
- Tell me everything you know about a certain value of a certain attribute (e.g., what can you tell me about DepartureDate's 10/April/23?
- Automatic Uniqueness and Dependency detection (interactive due to its time cost: too slow): tell me all the candidate keys, tell me all the functional, order, inclusion, conditional, … dependencies

Ορθογώνια στα παραπάνω, όπου έχουμε δυνατότητες, το σύστημα θα πρέπει να κάνει:

- Διαδραστική παραγωγή σε βήματα, για κάθε προστιθέμενη λειτουργικότητα: επέκταση του front-end ώστε προοδευτικά ο αναλυτής να εξειδικεύει την κατανόηση του αρχείου με ερωταπαντήσεις
- Intra-operator optimizations: Βελτιστοποιήσεις σε σχέση με την επίδοση (τώρα: σε ό,τι αφορά τα dominance results)
- Derive a cost model for the execution of different tasks, on the basis of multiple experiments
- Inter-operator optimizations: μπορώ να εκμεταλλευθώ τα αποτελέσματα από προηγούμενους operators για να κάνω ταχύτερους υπολογισμούς των επόμενων (ενδεχομένως με το τίμημα της κατά προσέγγιση αποτίμησης)

Για όλα τα αυτομάτως παραχθέντα tasks το σύστημα θα πρέπει να κάνει

- Αναπαράσταση των ενδιάμεσων αποτελεσμάτων μέσω των αντίστοιχων κλάσεων && Καταγραφή ενός αναλυτικού report με όλα τα ευρήματα
- Αποτίμηση της σημαντικότητας των αποτελεσμάτων (π.χ., silhouette evaluation of clustering, value of correlation for correlation, …)
- Καταγραφή ενός συνοπτικού report / επέκταση υπαρχόντων reports με τα όλα ευρήματα που έχουν τουλάχιστον ένα ελάχιστο threshold σημαντικότητας (τώρα έχουμε ένα report ανά κατηγορία ευρημάτων)
- /* Ιδεατά, και παραγωγή bar charts, lines, scatter-plots με τα σημαντικά αποτελέσματα σε εναλλακτικά formats */

Μαζί με τα παραπάνω, η διπλωματική θα πρέπει να κάνει:

- Internal refactoring to simplify architecture
- Γραφική διαπροσωπεία (java Swing) για διαδραστική προεπισκόπηση, data type detection, dataset registration, interactive registration of tasks to run, και προβολή των αναφορών χωρίς αλλαγή του API του back-end

Μπορείτε να στήσετε την πυθία και να παίξετε με αυτή να δείτε πώς δουλεύει. Μπορείτε επίσης να παίξετε λίγο με ένα απλό εργαλείο data analytics, όπως π.χ., το Orange για να δείτε μια ιδέα για το πώς ο χρήστης διαδραστικά μπορεί να κάνει κάποια από αυτά τα tasks.

ΕΠΙΠΕΔΟ: *Διπλωματική για Μηχανικούς*

ΠΛΑΤΦΟΡΜΑ ΕΡΓΑΣΙΑΣ: Java, Apache Spark

ΠΡΟΚΛΗΣΕΙΣ και ΟΦΕΛΗ: Η δυσκολία βρίσκεται κυρίως στη σωστή και επεκτάσιμη σχεδίαση του λογισμικού. Τα οφέλη για ένα φοιτητή είναι: (α) τεχνογνωσία σε θέματα αναλυτικής δεδομένων και (β) hands-on σε ένα ευμέγεθες κομμάτι λογισμικού. Η εργασία είναι πλέον κατάλληλη για φοιτητές με ταλέντο σε προγραμματιστικά θέματα και σε θέματα αναλυτικής δεδομένων. Πρέπει να μπορείτε να ανταπεξέλθετε και στη σχεδίαση λογισμικού και στην ανάπτυξη του σχετικού συστήματος.

1.2 Checking for Uniqueness Constraints

Βαριάντα της Πυθίας αλλά για πιο βαρύ profiing: μπορούμε να βρούμε όλα τα candidate keys (minimal combinations of columns with unique values) αυτόματα?

ΕΠΙΠΕΔΟ: *Διπλωματική για Μηχανικούς (μία ανά αλγόριθμο)*

ΠΛΑΤΦΟΡΜΑ ΕΡΓΑΣΙΑΣ: Java, Apache Spark, MySQL

Ας υποθέσουμε ότι κάποιος σας δίνει ένα αρχείο με δεδομένα, ή κάποιον πίνακα σε μία βάση δεδομένων και θέλουμε να δούμε ποιος συνδυασμός από στήλες δίνει unique τιμές – ήτοι, candidate keys. Η βιβλιογραφία έχει πολλούς αλγορίθμους για το θέμα και θέλουμε ένα εργαλείο που, επαναχρησιμοποιώντας λίγα θεμελιώδη κομμάτια από την Πυθία (readers, writes, dataset profiles, …) θα καταλήξει να έχει μια παλέτα από υλοποιημένους αλγορίθμους σε ένα νέο εργαλείο.

Κάθε διπλωματική θα πρέπει να φτιάξει (σωστά και πλήρως όμως) ένα (1) αλγόριθμο της βιβλιογραφίας. Οι αλγόριθμοι που μας ενδιαφέρουν κατ' αρχήν σε επίπεδο διπλωματικών είναι οι Gordian, HCA, DUCC. Πιο απαιτητικοί οι SWAN, HyUCC, HPIValid. Όλοι οι αλγόριθμοι βρίσκονται στον φάκελο UCC στο https://drive.google.com/drive/folders/1WK0Lht1OMopgYcZ8JHBFQQRfa-5KYKeQ

Η υλοποίηση θα πρέπει να συνοδεύεται από τους σωστούς ελέγχους, και τη σχετική πειραματική αξιολόγηση – ουσιαστικά μια επανάληψη του πρωτότυπου άρθρου, χωρίς όμως την υλοποίηση των ανταγωνιστών αλγορίθμων. Όπως προαναφέρθηκε, θα θέλαμε όλοι οι αλγόριθμοι θα θέλαμε στο τέλος να μπουν σε ένα (1) εργαλείο, οπότε θα ξεκινήσουμε από μια κοινή αρχή με τον κώδικα, που είναι οι θεμελιώδεις δομές και υπηρεσίες της Πυθίας.

ΠΡΟΚΛΗΣΕΙΣ ΚΑΙ ΟΦΕΛΗ. Η δυσκολία είναι στην κατανόηση και υλοποίηση των επί μέρους δομών δεδομένων που απαιτεί ο κάθε αλγόριθμος.

Τα οφέλη για ένα φοιτητή είναι: έκθεση στην υλοποίηση αλγοριθμικά-έντονου κώδικα σε θέματα διαχείρισης δεδομένων, εκπαίδευση στην υλοποίηση, έλεγχο και πειραματική αποτίμηση ενός τέτοιου λογισμικού και τεχνογνωσία σε θέματα αναλυτικής δεδομένων.

Η εργασία είναι πλέον κατάλληλη για φοιτητές με ταλέντο και ενδιαφέρον σε ερευνητικά και αλγοριθμικά θέματα.

2 Εργαλεία Αντικειμενοστρεφούς Ανάπτυξης Λογισμικού

2.1 Code Pilot for Automated Test Generation

Για όσους είναι afficionados της Τεχνολογίας Λογισμικού, σειρά από διαδοχικές διπλωματικές (εκτός κι αν είμαστε πολύ τυχεροί και το εγχείρημα είναι πιο εύκολο απ' ό,τι νομίζω). Η πρώτη:

- **Μελέτη βιβλιογραφίας των μεθόδων αυτόματης παραγωγής tests on the basis of directives, annotations, … Θα ξεκινήσουμε με την ανασκόπηση των πολύ πρόσφατων άρθρων της βιβλιογραφίας, και ενός σχετικού εργαλείου (βλ. άρθρα και links στο https://drive.google.com/drive/folders/1j2ZzDJr2vL_z08kDBwMFN9_WNOrXDasa)**
- **Automated Junit test generation (happy && esp., rainy day) for a given method / class / project on the basis of (a) the method signature, and, (b) directives on the checks that must take place (e.g., for a certain parameter, require it to be nonnull, within a certain range, etc).**

Αφού έχει γίνει το παραπάνω:

- Automated defensive code generation. Given the src, and a similar prescription with the test generation, as well as what to do with the response (throw exceptions, return specific values, assert the immutability of certain parameters, …), generate defensive code, inside the src, to ensure the robustness of the src.

Αφού έχει γίνει το παραπάνω:

- Γενίκευση και σε γλώσσες όπως η C++ (μετά τα παραπάνω)
- Αντί για black-box testing on the basis of the method's signature, try also white-box testing, with the goal to attain high coverage (λέμε τώρα, αυτό είναι σαφώς πιο πολύπλοκο και μάλλον για διδακτορικό παρά για διπλωματική)

ΕΠΙΠΕΔΟ: *Από τις απαιτήσεις θα προκύψουν 2-3 Διπλωματικές/MSc (διαδοχικά, μία τη φορά).*

ΠΛΑΤΦΟΡΜΑ ΕΡΓΑΣΙΑΣ: Java

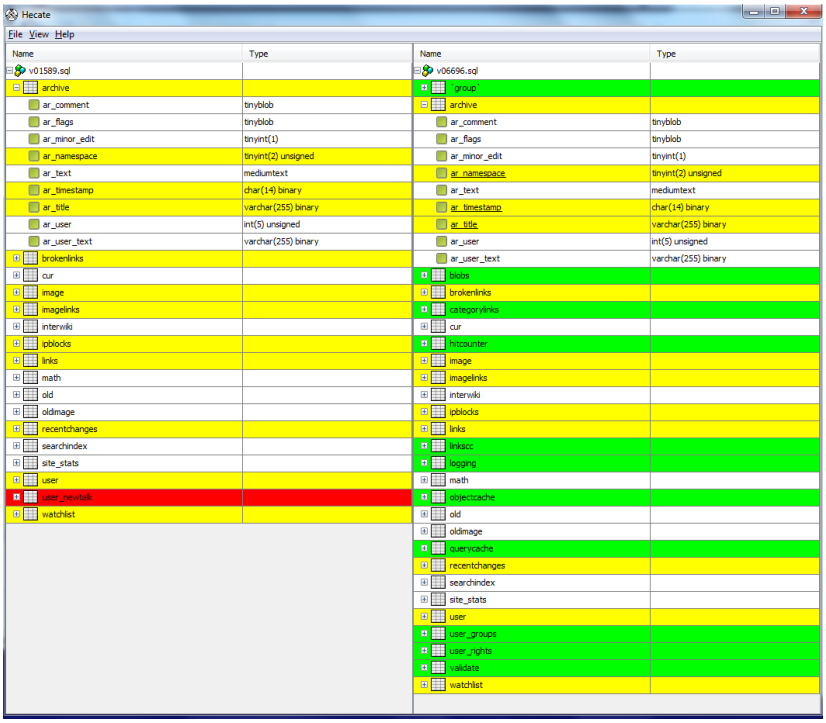
ΠΡΟΚΛΗΣΕΙΣ ΚΑΙ ΟΦΕΛΗ. Η δυσκολία είναι στην επεξεργασία του κώδικα και την σχετική αναπαράσταση. Μπορούμε να επαναχρησιμοποιήσουμε έτοιμο τον parser από το OOAD. Τα οφέλη για ένα φοιτητή είναι: (α) τεχνογνωσία σε ζητήματα σχεδίασης, ελέγχου, και hands-on σε ένα ευμέγεθες κομμάτι λογισμικού, και (β), εμπλοκή στους χώρους του static source code analysis and testing. Η εργασία είναι πλέον κατάλληλη για φοιτητές με ταλέντο σε και θέματα τεχνολογίας λογισμικού.

/* Java parser: https://github.com/javaparser/javaparser -- see also the parser package of Object Oriented Architecture Diagrammer */

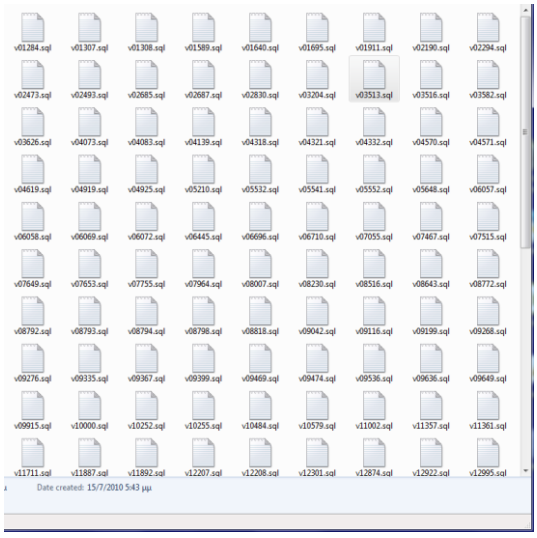
3 Μελέτη της Εξέλιξης Βάσεων Δεδομένων

Μια βάση δεδομένων, από τη στιγμή που θα δημιουργηθεί, αλλάζει εσωτερική δομή με το πέρασμα του χρόνου: νέοι πίνακες δημιουργούνται, παλαιοί καταστρέφονται, πεδία διαγράφονται, μετονομάζονται κλπ. Η διαδικασία αυτή ονομάζεται «εξέλιξη του σχήματος της βάσης δεδομένων» (schema evolution).

Το εργαλείο **Hecate** [https://github.com/DAINTINESS-Group/Hecate] μπορεί να συγκρίνει δύο σχήματα και να βρει τις διαφορές τους (κίτρινο: updated attributes, red: deletions, green: insertions).

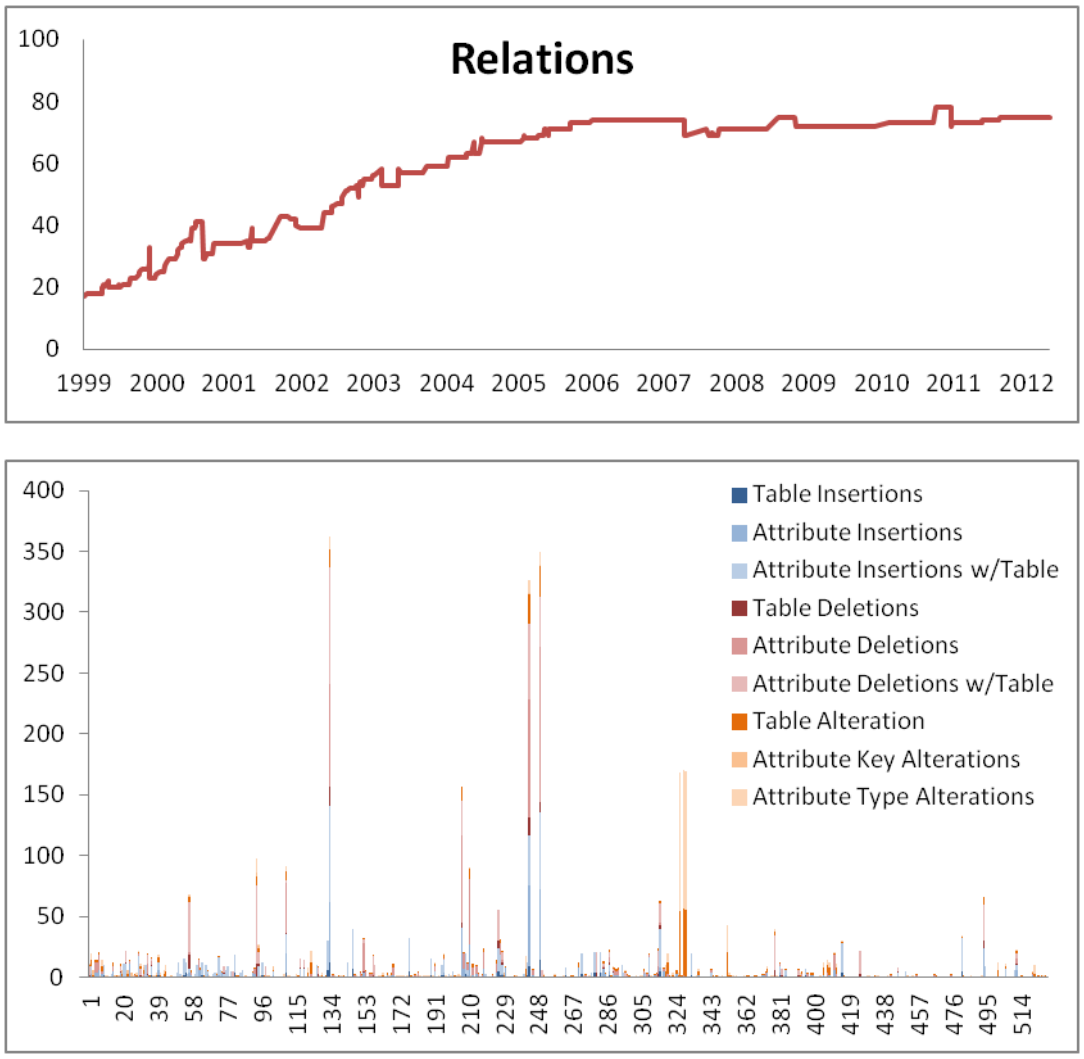


Επιπλέον, υπάρχουν αρκετές συλλογές από εκδόσεις του σχήματος της ιδίας βάσης (παρακάτω ένα screenshot από τη βάση της Wikimedia).



Η Εκάτη μπορεί να ταξινομήσει τις επί μέρους εκδοχές του σχήματος και να τις συγκρίνει διαδοχικά.

Έχουμε ήδη χρησιμοποιήσει την Εκάτη για να επεξεργαστούμε την εξέλιξη σχήματος διαφόρων βάσεων δεδομένων ανοιχτού λογισμικού, όπως για παράδειγμα, της βάσης της Wikimedia (της βάσης δεδομένων πίσω από τη Wikipedia), της βάσης του Atlas Trigger (του εργαλείου που διαχειρίζεται τα δεδομένα από το πείραμα Atlas για την ανεύρεση του μποζονίου του Χιγκς), της Ensembl (του εργαλείου για τη διαχείριση των δεδομένων του ανθρώπινου γονιδιώματος) και πολλών CMS's (opencart, coppermine, phpBB, typo3, …). Έχουμε επίσης συλλέξει την ιστορία από πολλά συστήματα ανοιχτού κώδικα που περιλαμβάνουν βάσεις δεδομένων και καταγράφουν και τις εκδοχές τους σε δημόσια αποθετήρια (κυρίως github, αλλά και svn) αλλά δεν την έχουμε επεξεργαστεί ακόμα.

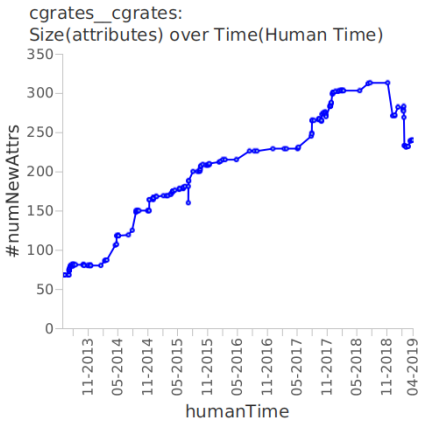


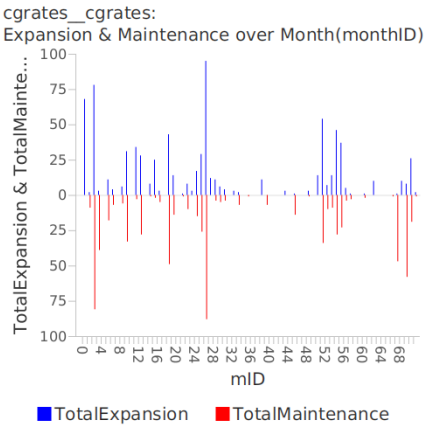
Στο παραπάνω σχήμα βλέπετε (α) το πώς εξελίχθηκε το μέγεθος του σχήματος της βάσης στο χρόνο και (β) τον παλμό των αλλαγών (το πώς διαρθρώθηκαν οι αλλαγές σε κάθε monitored version) για τη βάση Ensembl.

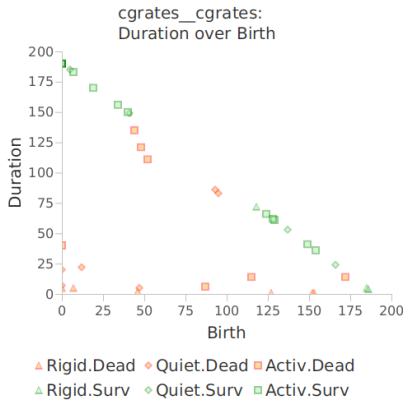
Το εργαλείο ROSES από τη Μ. Ζέρβα είναι ένα εργαλείο βασισμένο σε μια βάση δεδομένων, όπου έχουμε περάσει την εξαχθείσα πληροφορία, για να μπορούμε να απομονώνουμε εύκολα υποσύνολα πινάκων που μας ενδιαφέρουν και να οπτικοποιούμε γραφικές παραστάσεις. Το εργαλείο MUSES από τον Α. Παππά μας επιτρέπει να εξάγουμε πρότυπα συχνών υποακολουθιών από τα δεδομένα μας.

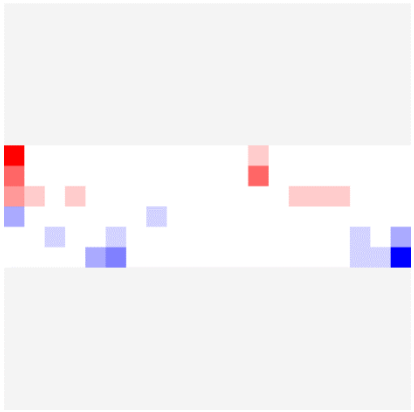
Το εργαλείο Heraclitus Fire [https://github.com/pvassil/HeraclitusFire] χρησιμοποιείται για να συμπληρώσει τα αρχικώς εξαχθέντα αποτελέσματα της Εκάτης με επιπλέον στατιστικά -- ενδεικτικά:

- Ανάλυση των αλλαγών σαν (α) επέκταση (προσθέσεις πινάκων και πεδίων) και (β) συντήρηση (διαγραφές πινάκων και πεδίων, αλλαγές τύπων, κλπ) και περαιτέρω ανάλυση σε άλλες υποκατηγορίες και μετρικές
- Συγκεντρωτικά στατιστικά για τις αλλαγές ανά commit και αλλαγές μήνα
- Έλεγχο κάποιων πρωτόλειων προτύπων
- Γραφικές παραστάσεις







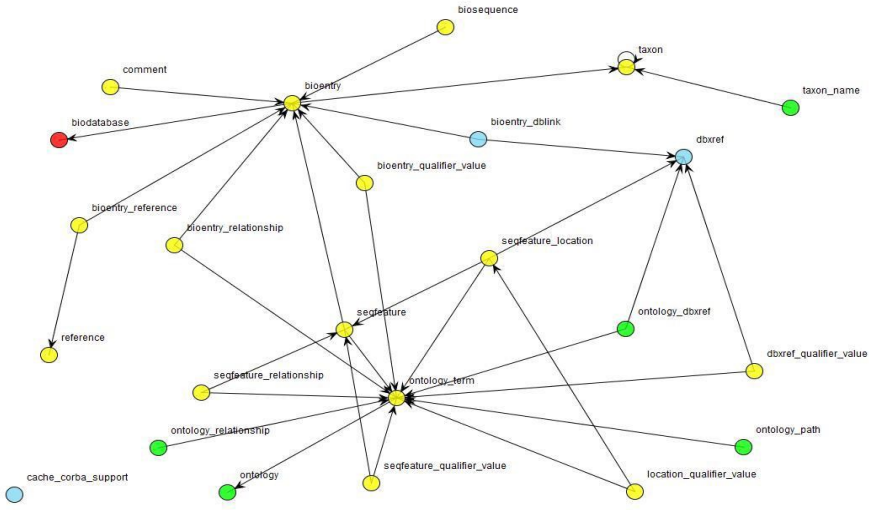


Αυτή τη στιγμή, έχουμε συλλέξει τις ζωές για 195 σχήματα από Free Open Source Software systems. Δείτε το άρθρο στο ICDE'21 στο https://www.cs.uoi.gr/~pvassil/publications/2021_ICDE/]

Ο Ηράκλειτος είναι το κεντρικό εργαλείο με το οποίο μελετάμε το πώς εξελίσσονται τα σχήματα που έχουμε συλλέξει.

Το εργαλείο «**Πλουτάρχου Βίοι Παράλληλοι**» [https://github.com/DAINTINESS-Group/PlutarchParallelLives] είναι ένα εργαλείο που βγήκε από σειρά Διπλωματικών και MSc, το οποίο απεικονίζει την εξέλιξη των πινάκων μιας βάσης δεδομένων σε παράλληλες γραμμές. Κάθε version αναπαριστάται από 3 κολώνες για εισαγωγές, διαγραφές και ενημερώσεις πινάκων. Οι γεννήσεις πινάκων και πεδίων φαίνονται με πράσινο και οι διαγραφές με κόκκινο χρώμα.

Το εργαλείο «**Παρμενίδεια Αλήθεια**» [https://github.com/DAINTINESS-Group/ParmenidianTruth] είναι ένα εργαλείο από τον Μ. Κολοζώφ που αναπαριστά το σχήμα μιας βάσης δεδομένων με ένα διαχρονικό γράφημα και φροντίζει να οπτικοποιεί κάθε version και τις εκδοχές της σε ένα slide μιας Powerpoint παρουσίασης (πρακτικά φτιάχνει μια ταινία για το πώς αλλάζει το σχήμα της βάση δεδομένων).



Η έρευνα στην περιοχή αυτή είναι θεμελιώδους φύσεως και αφορά στο να κατανοήσουμε την ύπαρξη προτύπων (ή ακόμα καλύτερα νόμων) για το πώς εξελίσσονται οι βάσεις δεδομένων με την πάροδο του χρόνου.

http://www.cs.uoi.gr/~pvassil/projects/schemaBiographies/

3.1 Πλούταρχου Βίοι Παράλληλοι

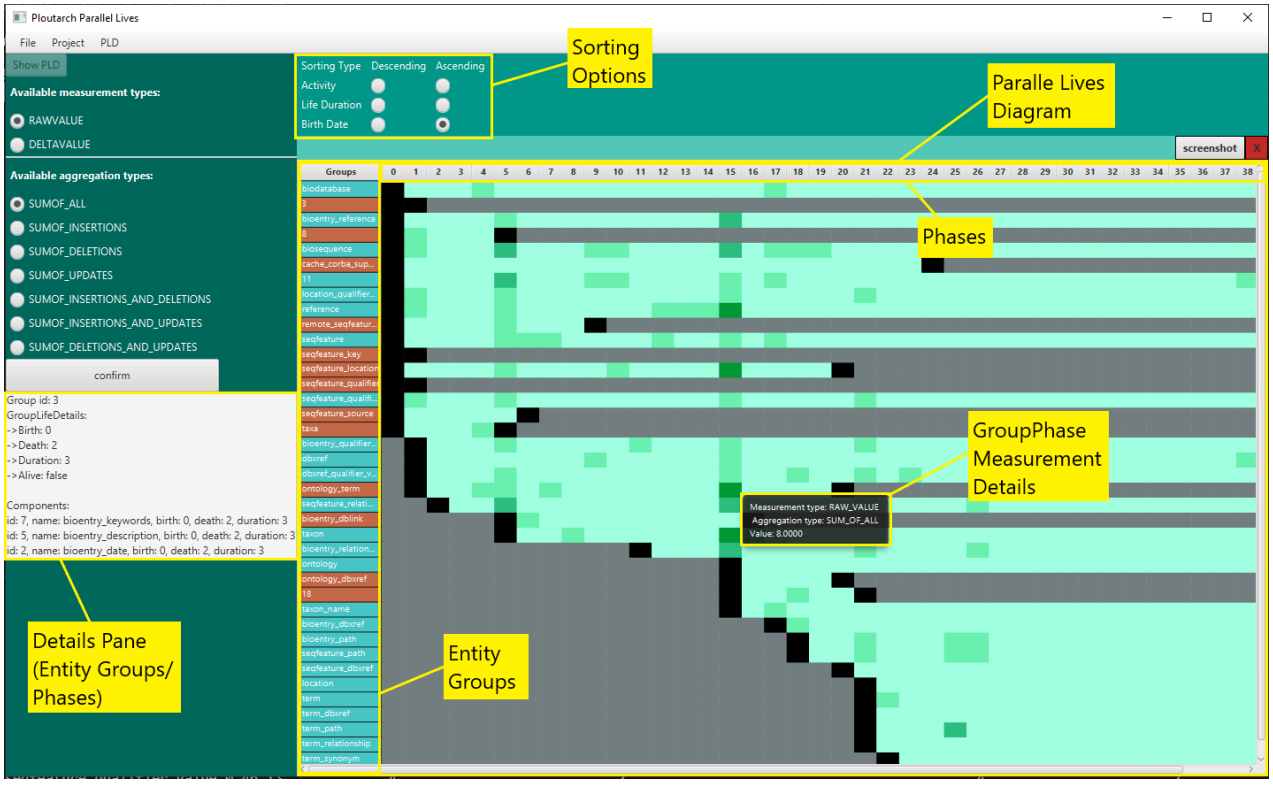
ΠΕΡΙΛΗΨΗ: Επέκταση και αναμόρφωση του εργαλείου Πλουτάρχου Βίοι Παράλληλοι

ΣΥΝΤΟΜΗ ΠΕΡΙΓΡΑΦΗ: Το εργαλείο Πλουτάρχου Βίοι Παράλληλοι απεικονίζει οπτικά την εξέλιξη ενός σχεσιακού σχήματος. https://github.com/DAINTINESS-Group/PlutarchParallelLives καθώς και η πρώτη δημοσίευση: https://www.cs.uoi.gr/~pvassil/publications/2023_ERForum/

ΕΠΙΠΕΔΟ: *Από τις απαιτήσεις που παρατίθενται στη συνέχεια θα προκύψουν 2-3 Διπλωματικές/MSc (αλλά όχι όλες μαζί ταυτοχρόνως).*

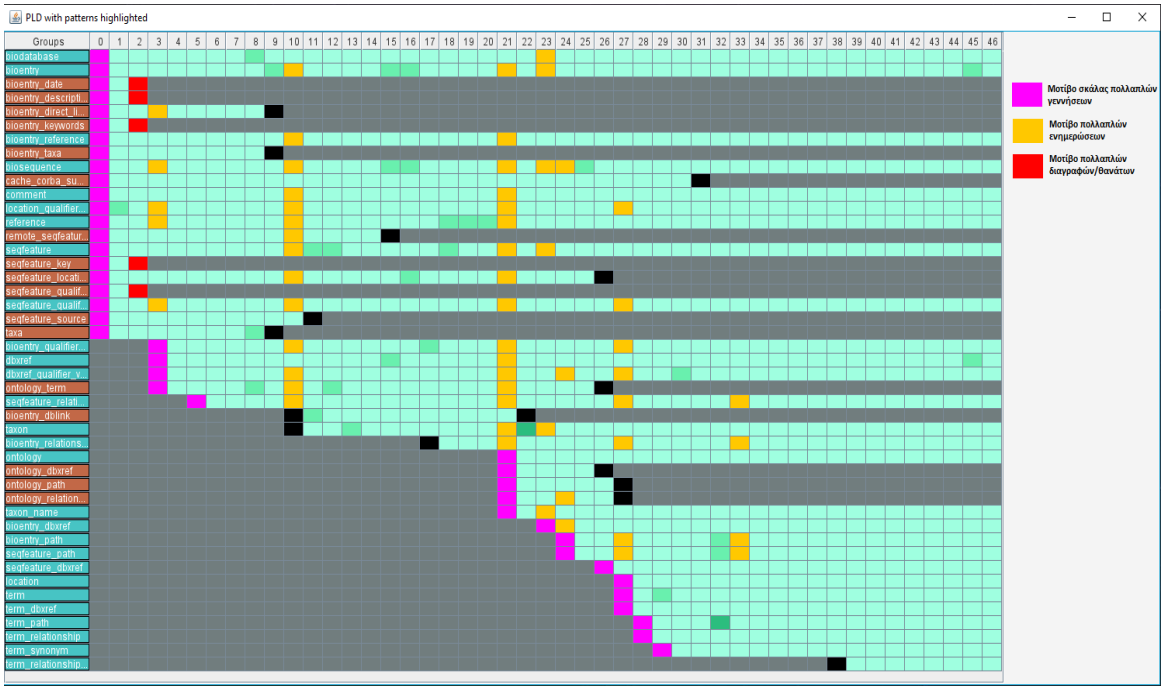
ΠΛΑΤΦΟΡΜΑ ΕΡΓΑΣΙΑΣ: Java

ΚΕΝΤΡΙΚΗ ΙΔΕΑ: Η ιδέα είναι ότι έχουμε μια κοινότητα ομοειδών οντοτήτων που εξελίσσεται στο χρόνο (π.χ., οι μετοχές ενός χρηματιστηρίου, οι πίνακες μιας βάσης δεδομένων) και θέλουμε να δούμε την εξέλιξη αυτή. Στον Πλούταρχο, το «Διάγραμμα Παράλληλων Ζωών» -- Parallel Lives Diagram (PLD) -- χρησιμοποιείται για να αναπαραστήσουμε οπτικά την εξέλιξη μίας ομάδας οντοτήτων ως έναν 2Δ πίνακα, όπου κάθε οντότητα αντιστοιχεί σε μία γραμμή, κάθε χρονική στιγμή αντιστοιχεί σε μία στήλη και η εντάσεις των χρωμάτων κάθε κελιού αναπαριστούν το μέγεθος της δραστηριότητας για τον συνδυασμό οντότητας και χρονικής στιγμής (με μαύρο οι γεννήσεις και οι διαγραφές οντοτήτων, με γκρι η «μη-ζωή» της οντότητας, πριν τη γέννηση ή μετά τη διαγραφή της).

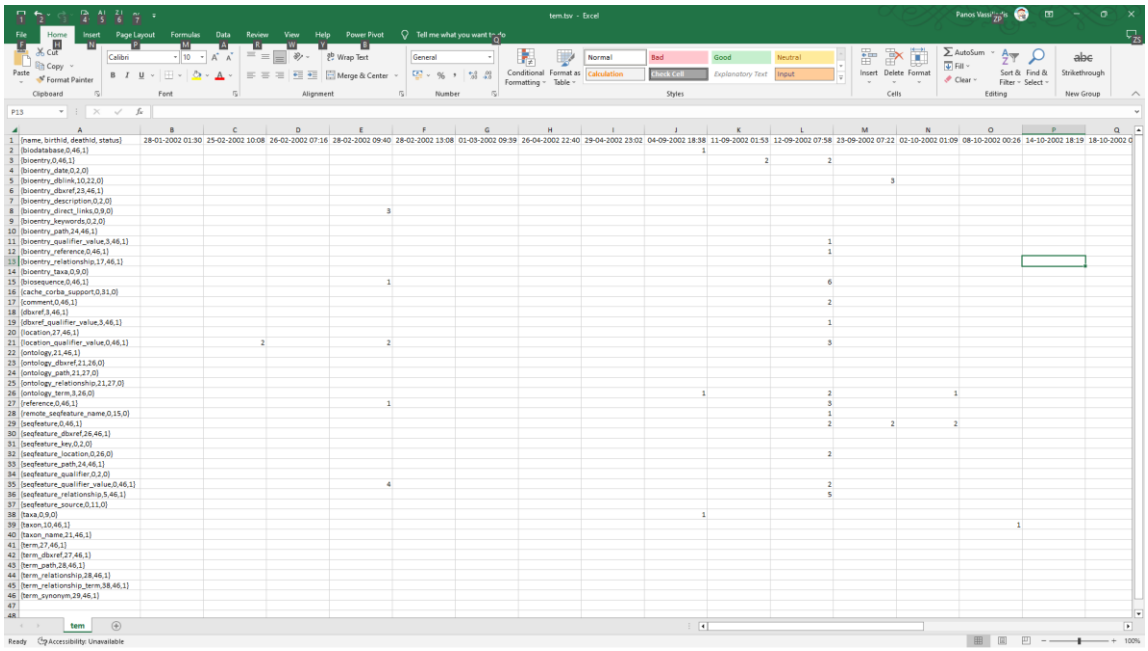


Για να δούμε ένα διάγραμμα, ενδεχομένως, στην αρχική φόρτωσή του, να χρειαστεί να κάνουμε clustering στις χρονικές στιγμές για να μειώσουμε τον αριθμό των στηλών – και ομοίως, να ομαδοποιήσουμε πίνακες με παρόμοια ζωή, για τη μείωση των γραμμών, ώστε το συνολικό διάγραμμα να χωρά στην οθόνη.

Επίσης το εργαλείο βρίσκει «πρότυπα» (patterns) αλλαγής. Τα πρότυπα καταγράφουν σημαντικές στιγμές ή φάσεις της ζωής της κοινότητας, όπου συμβαίνουν αλλαγές μαζικά: μαζικές γεννήσεις (μια στήλη (χρονική στιγμή, δλδ) με πολλές γεννήσεις), μαζικές ενημερώσεις (μια στήλη με πολλά κελιά να έχουν μεγάλο αρ. αλλαγών), μαζικές διαγραφές οντοτήτων, «σκάλα» συνεχόμενων γεννήσεων (συνεχόμενες στήλες με γεννήσεις). Η οπτική αναπαράσταση χρωματίζει με απαλό πράσινο τα κελιά χωρίς αλλαγές, με έντονο πράσινο την ύπαρξη αλλαγών, και με μωβ, κόκκινο και κίτρινο τα κελιά που συμμετέχουν σε κάποιο πρότυπο.



Εσωτερικά ο Πλούταρχος κρατά στα σχετικά αντικείμενα όλες τις πληροφορίες. Η αποτύπωσή της ζωής της κοινότητας ως κείμενο, το οποίο και συχνά αποκαλούμε *ενδιάμεση αναπαράσταση,* γίνεται σε ένα tab-delimited αρχείο κειμένου (στο επόμενο σχήμα το έχουμε ανοίξει με Excel για να φαίνονται πιο καθαρά οι γραμμές και οι στήλες). Τα κελιά με αριθμούς περιέχουν το μέγεθος της αλλαγής που υπέστη μια οντότητα σε μια συγκεκριμένη χρονική στιγμή.



Στη συνέχεια, παρατίθεται ένας μακρύς κατάλογος με επεκτάσεις και αναμορφώσεις που μπορούμε να κάνουμε στον Πλούταρχο σε διάφορες Διπλωματικές/MSc.

Σε όλες τις διπλωματικές θα χρειαστεί να πειραματιστούμε με διαφορετικά είδη δεδομένων (π.χ., schema evolution, covid cases, stock market, δεδομένα που θα βρείτε εσείς) για να δούμε ότι οι αλγόριθμοι έχουν νόημα γενικότερα.

Α. Επέκταση/Αναμόρφωση. Η επέκταση/αναμόρφωση του εργαλείου περιλαμβάνει:

1. Καθαρισμός του εργαλείου από νεκρό κώδικα, αναμόρφωση όποιων οσμών παρατηρήσουμε, και βελτίωση της τεκμηρίωσης.
2. Συγγραφή ενός user guide & ενός developer's guide για το εργαλείο
3. Μπορούμε να βελτιώσουμε τη γραφική διαπροσωπεία στη βάση μιας άλγεβρας τελεστών (π.χ., zoom-in / out, show details of all involved entities, de-cluster time/entities) και καλών προγραμματιστικών τεχνικών (βλ. παρακάτω *)?
4. Μπορούμε να περιγράψουμε με κείμενο (και εικόνα, ενδεχομένως, ως infographic) τη ζωή μιας κοινότητας ομοειδών οντοτήτων (βλ. παρακάτω **)?
5. Μπορούμε να παράξουμε απολύτως αυτόματα ένα συνολικό report από τα ευρήματα του εργαλείου με πίνακες, εικόνες, και ενδεχομένως κείμενο σαν μια κομψή data story (π.χ., σε markdown?)
6. Μπορούμε να προσθέσουμε πρότυπα? Ενδεικτικά: identification of calmness periods, progressive cooling, κ.α.

* Προς το παρόν, το zoom-in/-out είναι πρωτόλειο στη βάση ενός png. Απλές τεχνικές zoom με το swing δεν έχουν υλοποιηθεί, ενώ μια απενεργοποιημένη υλοποίηση με JavaFX παρουσιάζει τεχνικά προβλήματα.

** Για να πούμε μια ιστορία δεδομένων για ένα PLD χρειάζεται να πούμε (α) ποιες οντότητες συμμετέχουν στην κοινότητα, (β) ποιο το χρονικό πλαίσιο, (γ) πότε/πώς (χονδρικά) γεννιούνται οι οντότητες, (δ) πότε συνέβησαν σημαντικά γεγονότα (πρότυπα) στη ζωή της κοινότητας, (ε) υπάρχουν φάσεις (είτε ανευρεθείσες μέσω προτύπων, είτε ως χρόνος ανάμεσα σε άλλα πρότυπα), και ποιες είναι?

Β. Αναλυτική Δεδομένων. Τα ζητούμενα από Διπλωματικές που θα πλαισιώσουν ή επεκτείνουν το εργαλείο, χρησιμοποιώντας την ενδιάμεση αναπαράσταση με πράξεις data analytics, είναι:

1. Μπορούμε να μετρήσουμε πόσο όμοια είναι δύο PLD? Για το σκοπό αυτό θα χρειαστεί να ορίσουμε μια μετρική απόστασης δύο PLD (που όπως όλες οι μετρικές θα μετράει πόσα «βήματα» αλλαγών πρέπει να κάνουμε για να μετασχηματίσουμε ένα PLD σε ένα άλλο)
2. Μπορούμε να μετασχηματίσουμε την περιγραφή της ζωής ενός συνόλου ομοειδών οντοτήτων από ένα PLD σε μία γραμμική περιγραφή, στη βάση προτύπων? Δηλαδή, αν περιγράψουμε τη ζωή μιας κοινότητας σαν μια σειρά φάσεων (π.χ., μαζικές γεννήσεις, σχετική ηρεμία, μαζική συντήρηση, σκάλα γεννήσεων, απόλυτη ηρεμία, …), έχουμε φτιάξει μια περιγραφή της κοινής ζωής σαν μια συμβολοσειρά (πχ., MB;RC;MU;BL;TC). Μετά μπορούμε να ορίσουμε και μετρικές απόστασης στη βάση των περιγραφών προτύπων.
3. Μπορούμε (α) να συσταδοποίησουμε (cluster) σύνολα από PLD και (β) να βρούμε το κέντρο κάθε συστάδας? Αν έχουμε τη μετρική απόστασης, η συσταδοποίηση με οποιοδήποτε παραδοσιακό αλγόριθμο συσταδοποίησης είναι απλή (και υπάρχουν πλείστες έτοιμες υλοποιήσεις)

ΠΡΟΚΛΗΣΕΙΣ και ΟΦΕΛΗ: Η δυσκολία βρίσκεται αφενός στον ορισμό του ακριβώς θα υλοποιηθεί και αφετέρου στη σωστή και επεκτάσιμη σχεδίαση των όποιων επεκτάσεων. Τα οφέλη για ένα φοιτητή είναι: (α) τεχνογνωσία σε ζητήματα σχεδίασης, ελέγχου, και hands-on σε ένα ευμέγεθες κομμάτι λογισμικού, και (β), εμπλοκή στο χώρο του προγραμματισμού διαδραστικών γραφικών διαπροσωπειών.

Η εργασία είναι πλέον κατάλληλη για φοιτητές με ταλέντο σε προγραμματιστικά θέματα και σε θέματα αναλυτικής επεξεργασίας δεδομένων. Πρέπει να μπορείτε να ανταπεξέλθετε και στη σχεδίαση λογισμικού και στην ανάπτυξη των επεκτάσεων του συστήματος.