

Θέματα μεταπτυχιακών & διπλωματικών εργασιών για το ακ. έτος 2024- 2025

Π. ΒΑΣΙΛΕΙΑΔΗΣ – 2023-10-07

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-N7ZzyDfpsMLCQcm_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.

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.

Contents

1	Data Science	3
1.1	Data Cleaners	3
1.2	Data Publisher	4
2	Text to Graph.....	6
2.1	From Legal Regulations to Checklists	6
3	Μελέτη της Εξέλιξης Βάσεων Δεδομένων.....	8
3.1	Πλούταρχου Βίοι Παράλληλοι.....	12
4	Timeseries Management.....	16
4.1	An Integrated Library for Data Analytics for Time-Series	16

1 Data Science

1.1 Data Cleaners

Για το υπάρχον εργαλείο **Data Cleaners** [<https://github.com/DAINTINESS-Group/DataCleaners>] θέλουμε να επεκτείνουμε τις δυνατότητές του ως εξής:

1. Παραγωγή **offline reports** προσθέτοντας τις αναφορές τύπου md (που λείπουν).

2. Βελτιωμένη αλληλεπίδραση με τον χρήστη μέσω ενός **απλού αλλά ολοκληρωμένου GUI**

- Καταχώρηση δεδομένων (α) χειροκίνητη, (β) [αν είναι δυνατόν] ημιαυτόματα
- Βελτιωμένη διαδραστική, οπτικοποιημένη αναφορά:
 - Σφαλμάτων ανά είδος ελέγχου
 - Σφαλμάτων ανά γραμμή του αρχείου

3. **Πρόσθεση μετασχηματισμών**, είτε στο υπάρχον εργαλείο, είτε χωριστά, π.χ.,

- Απάλειψη / ανωνυμοποίηση ευαίσθητων στηλών
- Υπολογισμός νέων στηλών
- Μετατροπή των ημερομηνιών σε μια συγκεκριμένη μορφή
- Μετατροπή των κενών τιμών, σε μια προεπιλεγμένη τιμή
- Γενικά: δίνοντας μια συνθήκη, π.χ., μια κανονική έκφραση, σε μια στήλη, αντικαταστήστε τις τιμές του αναλόγως (π.χ. εάν αριθμοί float προέρχονται από ένα csv της ΕΕ και θέλουμε ένα αρχείο csv ΗΠΑ, αντικαταστήστε όλα τα "," με ".")

Η εκτέλεση της ακολουθίας των μετασχηματισμών καταγράφεται κατάλληλα σε ένα δομημένο log file. Επίσης κρατιούνται και οι ενδιαμέσες εκδόσεις των αρχείων, όπως αυτά μετασχηματίζονται.

Η επέκταση του λογισμικού με την προσθήκη μετασχηματισμών μπορεί να είναι

- i) επέκταση της τρέχουσας DataCleanerFacade κεντρικής μηχανής, ή,
- ii) μια δεύτερη μηχανή (σε αυτή την αρχιτεκτονική, η υπάρχουσα engine είναι ο ελεγκτής ποιότητας και η νέα είναι ο μετασχηματιστής/καθαριστής)

Μπορείτε να στήσετε το Data Cleaners και να παίξετε με αυτό 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 (ίσως και για 2)*

ΠΛΑΤΦΟΡΜΑ ΕΡΓΑΣΙΑΣ: Java, Apache Spark

ΠΡΟΚΛΗΣΕΙΣ και ΟΦΕΛΗ: Η δυσκολία βρίσκεται κυρίως στη σωστή και επεκτάσιμη σχεδίαση του λογισμικού (ως συνήθως: back-end façade, strategy for connectors/algorithms, front-end controller). Τα οφέλη για ένα φοιτητή είναι: (α) τεχνογνωσία σε θέματα ποιότητας δεδομένων και (β) hands-on σε ένα ευμέγεθες κομμάτι λογισμικού. Πρέπει να μπορείτε να ανταπεξέλθετε και στη σχεδίαση λογισμικού και στην ανάπτυξη του σχετικού συστήματος.

1.2 Data Publisher

Έχουμε μεγάλο αριθμό συνόλων δεδομένων (που ανακτήθηκαν από πολλά μέρη, καθώς και που δημιουργήθηκαν από τα εργαλεία μας) και θέλουμε να μπορούμε να τα δημοσιεύσουμε.

Χρειαζόμαστε ένα εργαλείο για να μας βοηθήσει με

(α) τη διασφάλιση ποιότητας δεδομένων (πιθανότατα καλώντας/χρησιμοποιώντας το εργαλείο καθαρισμού δεδομένων DataCleaners)

(β) τη δημοσίευση των πληροφοριών σχετικά με αυτά τα σύνολα δεδομένων.

Η ΕΕ έχει δημοσιεύσει ορισμένες οδηγίες / κατευθυντήριες γραμμές για την ποιότητα των δεδομένων και το άνοιγμα των δημοσιευμένων συνόλων δεδομένων, π.χ. <https://op.europa.eu/webpub/op/data-quality-guidelines/en/> και https://data.europa.eu/sites/default/files/european_data_portal_-_open_data_goldbook.pdf

Εξίσου σημαντικές είναι και οι οδηγίες FAIR (<https://www.nature.com/articles/sdata201618>) που απαιτούν τα δεδομένα να είναι Findable, Accessible, Interoperable, Reusable.

Θέλουμε ένα εργαλείο που:

- ως είσοδος του δίνεται ένα συγκεκριμένο σύνολο δεδομένων
- εφαρμόζει τις προαναφερθείσες οδηγίες (ή άλλες που τυχόν θα βρούμε) μέσα από ένα σύνολο εργασιών/οθονών/... που εκτελούνται από τον αναλυτή
- παράγει αρχεία μεταδεδομένων που χαρακτηρίζουν το σύνολο δεδομένων + αναφορές για την κατάσταση των απαιτούμενων δοκιμών

Το αποτέλεσμα είναι (α) ένα σύνολο αρχείων μεταδεδομένων που δημιουργήθηκε για το σύνολο δεδομένων και (β) μια αναφορά για το τι πήγε καλά, λάθος, δεν συνέβη καθόλου, δεν ισχύει, ...

Το εργαλείο πρέπει να έχει έναν εξαιρετικά διαδραστικό χαρακτήρα, καθώς και τη δυνατότητα σταδιακής εργασίας με το σύνολο δεδομένων (κάνω μερικές δοκιμές σήμερα, αποθηκεύω την κατάσταση, κάνω άλλο ένα σύνολο δοκιμών αύριο, αποθηκεύω ξανά τη νέα κατάσταση, ...)

Ζητούμενα:

- να υποστηρίζουμε όσο το δυνατόν πιο παραμετρικά ένα σύνολο κανόνων (ώστε να μπορούμε σταδιακά να προσθέσουμε περισσότερους κανόνες στο back-end και στο front-end)
- δομημένη και δημοσιοποιήσιμη τελική έκθεση(εις;) κ.λπ., που (α) μπορεί να είναι μέρος ενός καταλόγου δομής συνόλων δεδομένων (π.χ. ένα υπολογιστικό φύλλο) και, (β) μπορεί να δημοσιευθεί στον ιστό
- εάν η τροποποίηση δεδομένων για το σύνολο δεδομένων αποδειχθεί απαραίτητη (π.χ. μεταφράστε ημερομηνίες σε συγκεκριμένη μορφή, αντικαταστήστε τις κενές συμβολοσειρές με "NULL", ...), το εργαλείο παρακολουθεί την προέλευση των μετασχηματισμών και την ακολουθία των εκδόσεων αρχείων που παράγονται ενδιάμεσα κατά την παραγωγή του τελικού, δημοσιευμένου συνόλου δεδομένων.

ΕΠΙΠΕΔΟ: *Διπλωματική για Μηχανικούς*

ΠΛΑΤΦΟΡΜΑ ΕΡΓΑΣΙΑΣ: Java (Apache Spark αν χρειαστεί)

ΠΡΟΚΛΗΣΕΙΣ και ΟΦΕΛΗ: Η δυσκολία βρίσκεται κυρίως στη σωστή και επεκτάσιμη σχεδίαση του λογισμικού (ως συνήθως: back-end façade, strategy for connectors/algorithms, front-end controller). Τα οφέλη για ένα φοιτητή είναι: (α) τεχνογνωσία σε θέματα δημοσίευσης δεδομένων και (β) hands-on σε ένα ευμέγεθες κομμάτι λογισμικού. Πρέπει να μπορείτε να ανταπεξέλθετε και στη σχεδίαση λογισμικού και στην ανάπτυξη του σχετικού συστήματος.

2 Text to Graph

2.1 From Legal Regulations to Checklists

Θέλουμε να κατασκευάσουμε ένα κομμάτι λογισμικού που μετατρέπει κείμενα νομοθεσίας σε checklist ελέγχων.

Input: αρχείο νόμου/κανονισμού (π.χ. νόμος AI, GDPR, Νόμος Διακυβέρνησης Δεδομένων, ...)

Output: Παραγωγή (πιθανότατα με τη μορφή ιστότοπου) των εξής παραδοτέων:

- (α) μια δομημένη αναπαράσταση του input ως κεφάλαια, ενότητες, ...
- (β) ένα ευρετήριο σημαντικών όρων (π.χ. σημαντικές οντότητες που εμπλέκονται στο νόμο) (πιθανώς με ημιαυτόματο τρόπο)
- (γ) το πιο σημαντικό, ένα ημιαυτόματα (man-in-the-loop) παραγόμενο σύνολο checklists

Το κύριο παραδοτέο είναι ένα σύνολο checklists, μία ανά use case του input.

Ιδιαίτερη έμφαση θα δοθεί στην κωδικοποίηση του νόμου AI. Ιδεατά το εργαλείο θα δοκιμαστεί και έναντι των Data and Data Governance acts. Επίσης πιθανή εφαρμογή σε: GDPR

Πώς να το κάνουμε:

(i) ανάστροφη μηχανική υπαρχουσών checklists όπως <https://gdpr.eu/checklist/> και <https://artificialintelligenceact.eu/assessment/eu-ai-act-compliance-checker/> σχετικά με τον τρόπο με τον οποίο συσχετίζουν τους ελέγχους με τις κύριες οντότητες και ενότητες του κανονισμού.

(ii) πιθανότατα, να μετατρέψουμε το κείμενο κάθε σχετικής ενότητας σε ένα γράφημα με ετικέτες με οντότητες και ιδιότητες, ενέργειες, ελέγχους ως κόμβους. Κάθε πρόταση γίνεται υπογράφημα και κάθε παράγραφος γίνεται μια ακολουθία υπογραφημάτων. Συνήθως οι άνθρωποι χρησιμοποιούν τον Stanford parser [<https://nlp.stanford.edu/software/lex-parser.shtml>] για να καταλάβουν τη δομή μιας πρότασης. Υπάρχει και αρκετή έτοιμη δουλειά: <https://github.com/search?q=Text+to+knowledge+graph+&type=repositories>

(iii) ημι-αυτόματη μετατροπή κάθε ακολουθίας υπογραφημάτων σε ένα σύνολο ελέγχων.

ΕΠΙΠΕΔΟ: *Διπλωματική για Μηχανικούς [κατά παράβαση της συνήθους τακτικής, μπορεί να το σκεφτούμε και για 2 άτομα]*

ΠΛΑΤΦΟΡΜΑ ΕΡΓΑΣΙΑΣ: Java, Stanford Parser, ίσως Apache Spark GraphX

ΠΡΟΚΛΗΣΕΙΣ και ΟΦΕΛΗ: Η δυσκολία βρίσκεται κυρίως στο αλγοριθμικό κομμάτι, που θα έχει αρκετό try-n-error μέχρι την οριστικοποίηση της επεξεργασίας, και ως συνήθως στη σωστή και επεκτάσιμη σχεδίαση του λογισμικού. Τα οφέλη για τους φοιτητές είναι: (α) τεχνογνωσία σε θέματα επεξεργασίας κειμένου, (β) (σπάνια) τριβή με την ευρωπαϊκή νομοθεσία δεδομένων, (γ) hands-on σε ένα ευμέγεθες κομμάτι λογισμικού. Πρέπει να μπορείτε να ανταπεξέλθετε και στη σχεδίαση λογισμικού και στην ανάπτυξη του σχετικού συστήματος.

Δείτε ένα παράδειγμα στην επόμενη σελίδα

Παράδειγμα

Input text: The following AI practices shall be prohibited:

- (a) the placing on the market, the putting into service or the use of an AI system that deploys subliminal techniques beyond a person's consciousness *or purposefully manipulative or deceptive techniques, with the objective, or the effect of materially distorting the behaviour of a person or a group of persons by appreciably impairing their ability to make an informed decision, thereby causing them to take a decision that they would not have otherwise taken* in a manner that causes or is *reasonably* likely to cause that person, another person or group of persons significant harm;

Interim representation: node -- labeled edge -- node

AnyActor -- shall be prohibited to place on the market -- AI System

AnyActor -- shall be prohibited to put in service -- AI System

AnyActor -- shall be prohibited to use -- AI System

AI system -- has -- OffendingDeceiverSystemProperty

OffendingDeceiverSystemProperty -- deploys -- subliminal techniques beyond a person's consciousness

OffendingDeceiverSystemProperty -- deploys -- purposefully manipulative or deceptive techniques, with the objective, or the effect of materially distorting the behaviour of a person or a group of persons by appreciably impairing their ability to make an informed decision, thereby causing them to take a decision that they would not have otherwise taken in a manner that causes or is reasonably likely to cause that person, another person or group of persons significant harm;

The last two can be remerged into one, or some other structure can be devised -- but anyway the point here is that we try to convert the text to (a) actors/entities/... and (b) their relationships. We can do a better modeling than that, by other types of labels, or extra nodes, e.g., making the graph a tree with a root node "Prohibition".

Checklist

Is the AI system in a state where it

- deploys -- subliminal techniques beyond a person's consciousness
- deploys -- purposefully manipulative or deceptive techniques, with the objective, or the effect of materially distorting the behaviour of a person or a group of persons by appreciably impairing their ability to make an informed decision, thereby causing them to take a decision that they would not have otherwise taken in a manner that causes or is reasonably likely to cause that person, another person or group of persons significant harm;

If yes to any of the above, is the AI system:

- placed on the market?
- put into service?
- used?

by any possible actor?

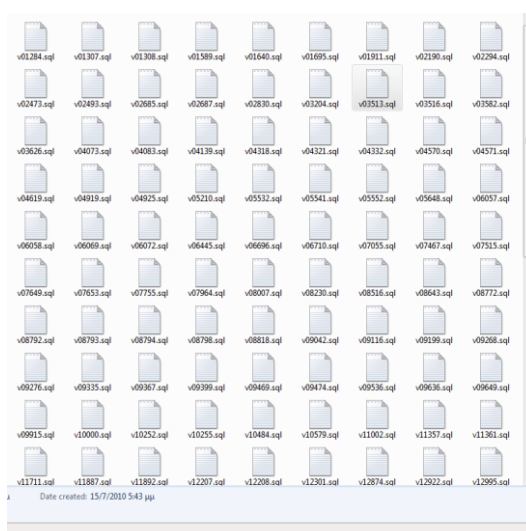
3 Μελέτη της Εξέλιξης Βάσεων Δεδομένων

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).

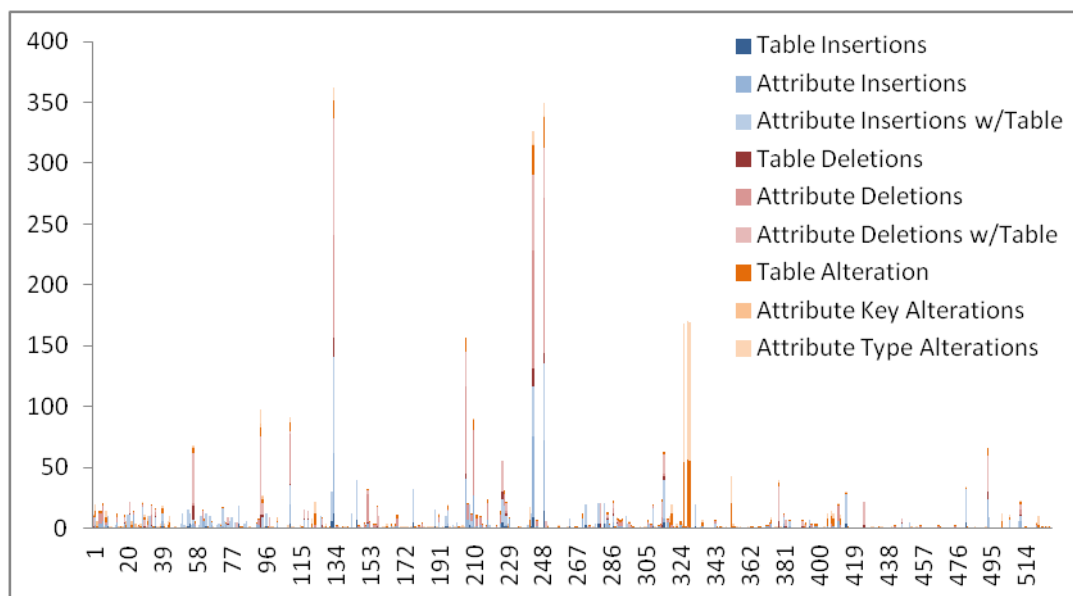
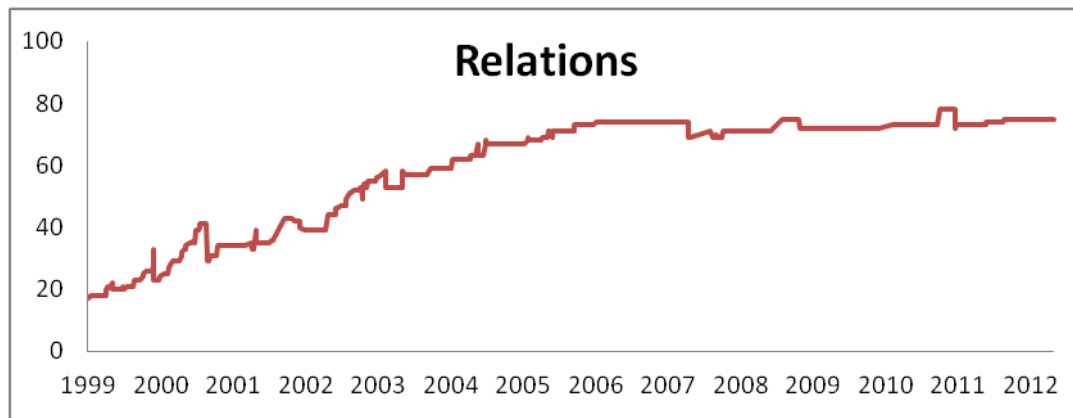
File View		File View	
Name	Type	Name	Type
v01589.sqf		v06696.sqf	
archive		grabup	
ar_comment	tryblob	archive	
ar_flags	tryblob	ar_comment	tryblob
ar_minor_edit	tryint(1)	ar_flags	tryblob
ar_namespace	tryint(2) unsigned	ar_minor_edit	tryint(1)
ar_text	mediumtext	ar_namespace	tryint(2) unsigned
ar_timestamp	char(14) binary	ar_text	mediumtext
ar_title	varchar(255) binary	ar_timestamp	char(14) binary
ar_user	int(5) unsigned	ar_title	varchar(255) binary
ar_user_text	varchar(255) binary	ar_user	int(5) unsigned
brokenlinks		ar_user_text	varchar(255) binary
cur		bbbs	
image		brokenlinks	
imagelinks		categorylinks	
interviki		cur	
ipblocks		intcounter	
links		image	
math		imagelinks	
old		interviki	
oldimage		ipblocks	
recentchanges		links	
searchindex		linkacc	
site_stats		logging	
user		math	
user_newtalk		objectcache	
watchlist		old	
		oldimage	
		querycache	
		recentchanges	
		searchindex	
		site_stats	
		user	
		user_groups	
		user_rights	
		validate	
		watchlist	

Επιπλέον, υπάρχουν αρκετές συλλογές από εκδόσεις του σχήματος της ίδιας 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, corpermine, 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.

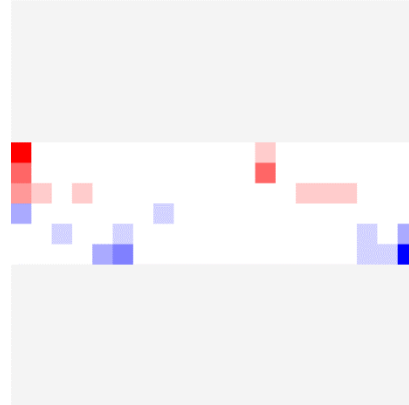
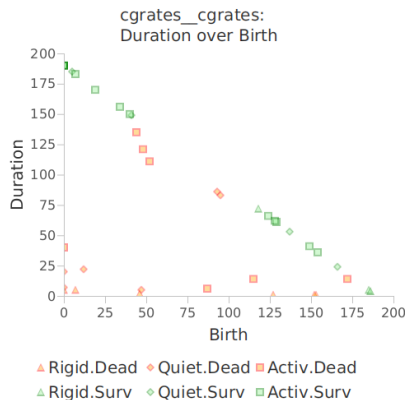
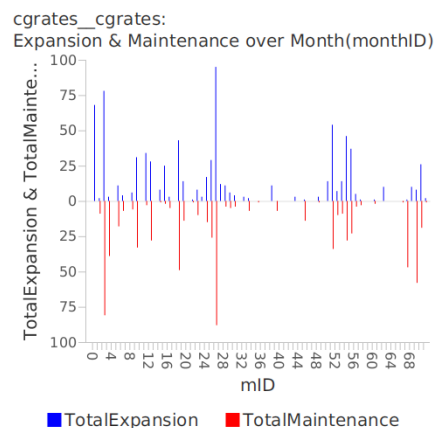
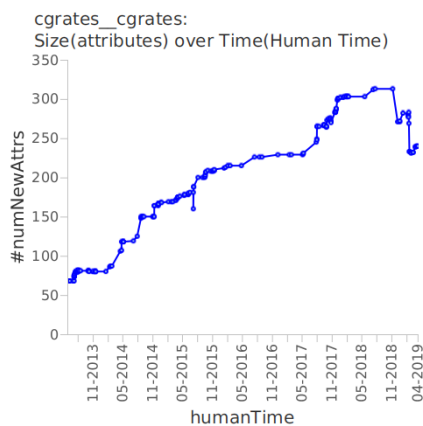


Στο παραπάνω σχήμα βλέπετε (α) το πώς εξελίχθηκε το μέγεθος του σχήματος της βάσης στο χρόνο και (β) τον παλμό των αλλαγών (το πώς διαρθρώθηκαν οι αλλαγές σε κάθε monitored version) για τη βάση Ensembl.

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 (προσθήσεις 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.

3.1 Πλούταρχου Βίοι Παράλληλοι

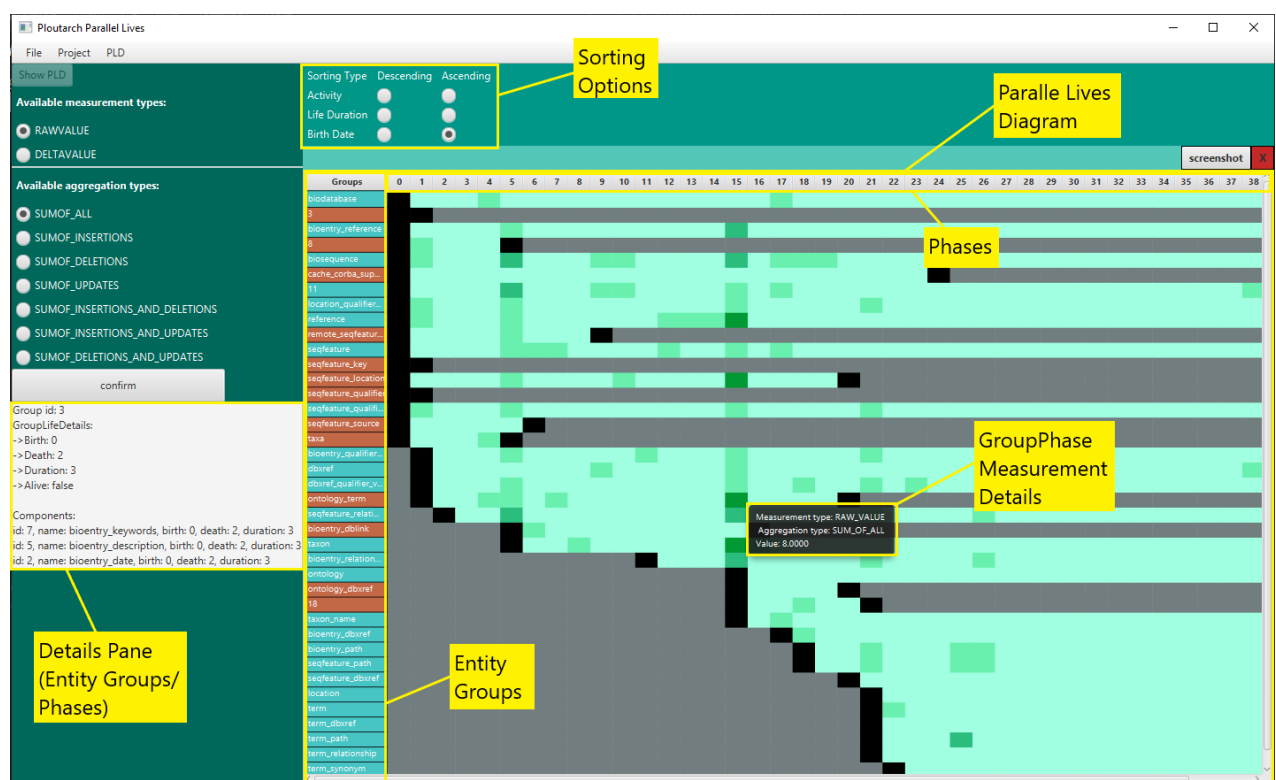
ΠΕΡΙΛΗΨΗ: Επέκταση και αναμόρφωση του εργαλείου Πλούταρχου Βίοι Παράλληλοι

ΣΥΝΤΟΜΗ ΠΕΡΙΓΡΑΦΗ: Το εργαλείο Πλούταρχου 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).

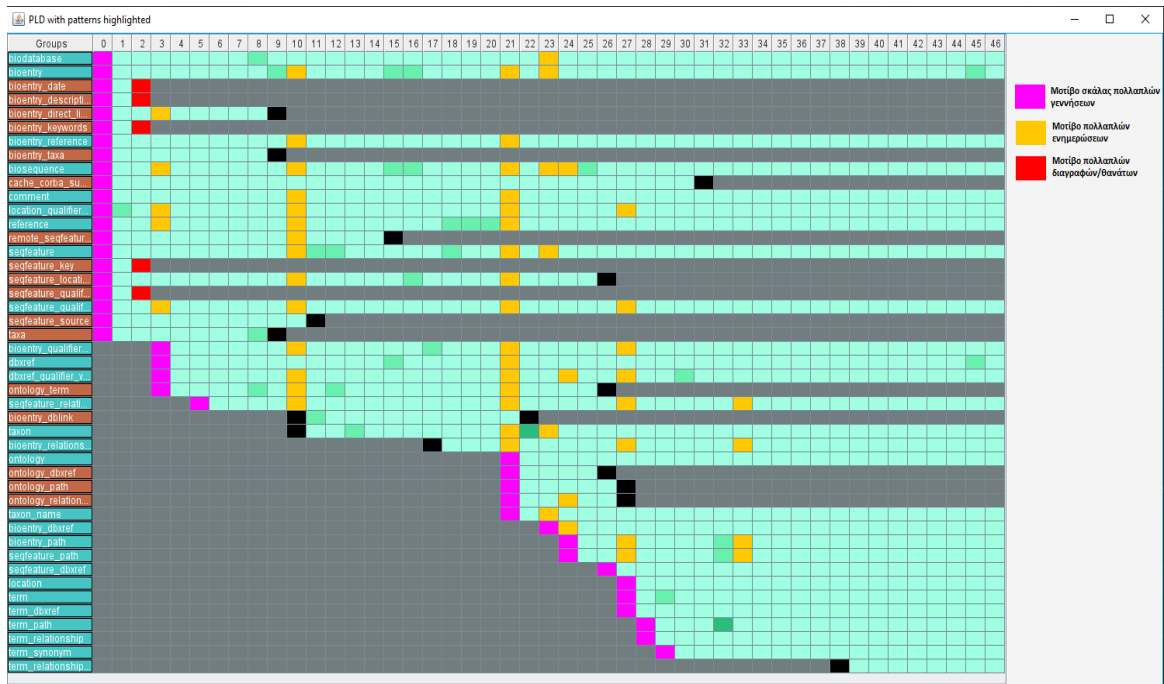
ΠΛΑΤΦΟΡΜΑ ΕΡΓΑΣΙΑΣ: Java

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 2D 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).

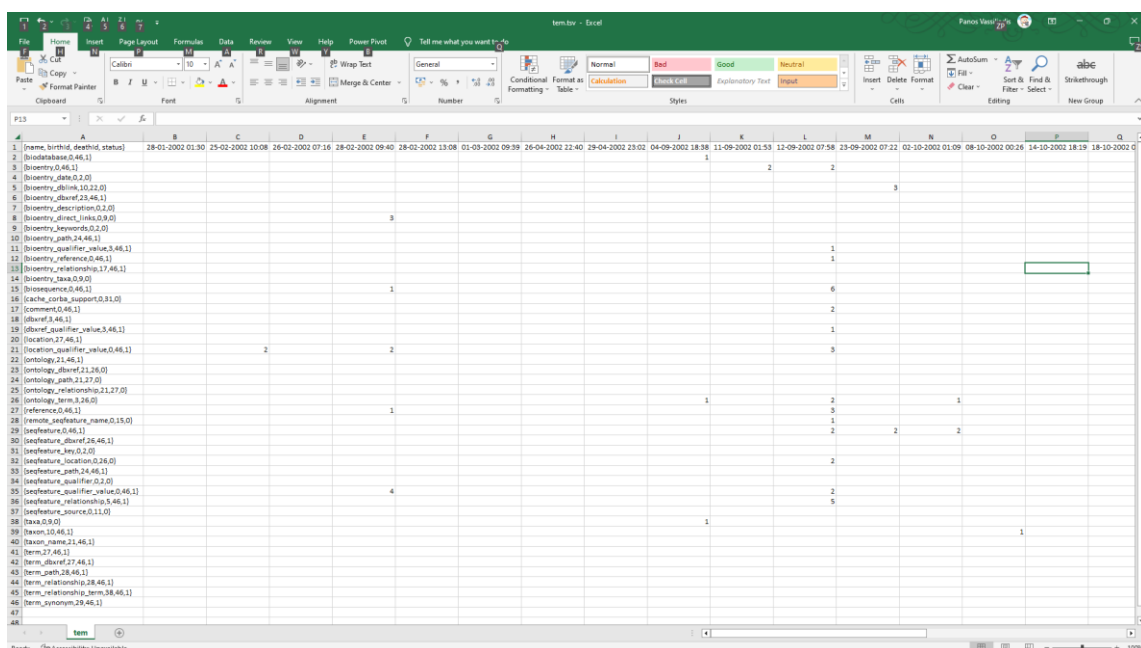


Για να δούμε ένα διάγραμμα, ενδεχομένως, στην αρχική φόρτωσή του, να χρειαστεί να κάνουμε clustering στις χρονικές στιγμές για να μειώσουμε τον αριθμό των στηλών – και ομοίως, να ομαδοποιήσουμε πίνακες με παρόμοια ζωή, για τη μείωση των γραμμών, ώστε το συνολικό διάγραμμα να χωρά στην οθόνη.

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.

A. 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 rng. 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?

B. 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 συσταδοποιήσουμε (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.

4 Timeseries Management

Μια χρονοσειρά είναι μια λίστα από τιμές (που ευλόγως μπορούμε να υποθέσουμε ότι είναι πραγματικοί αριθμοί). Κάθε μία τιμή έχει λοιπόν την θέση της στη λίστα και ενδεχομένως και άλλες ιδιότητες, όπως π.χ., κάποιο χρονόσημο. Δείτε για παράδειγμα μια χρονοσειρά με δεδομένα εξαγωγών.



4.1 An Integrated Library for Data Analytics for Time-Series

ΠΕΡΙΛΗΨΗ: Το ζητούμενο είναι, μέσα από μια **σειρά Διπλωματικών Εργασιών**, να φτιαχτεί ένα σύστημα διαχείρισης χρονοσειρών με ένα σύνολο βασικών λειτουργιών και γραφικών απεικονίσεων.

ΣΥΝΤΟΜΗ ΠΕΡΙΓΡΑΦΗ: Οι χρονοσειρές μπορούν να χρησιμοποιηθούν για διάφορους σκοπούς. Επίσης, μπορούν να αναπαρασταθούν εσωτερικά με διαφορετικούς τρόπους. Ο γενικός στόχος για μια σειρά από συμπληρωματικές διπλωματικές είναι κατασκευασθεί ένα modular σύστημα διαχείρισης χρονοσειρών με τα εξής χαρακτηριστικά:

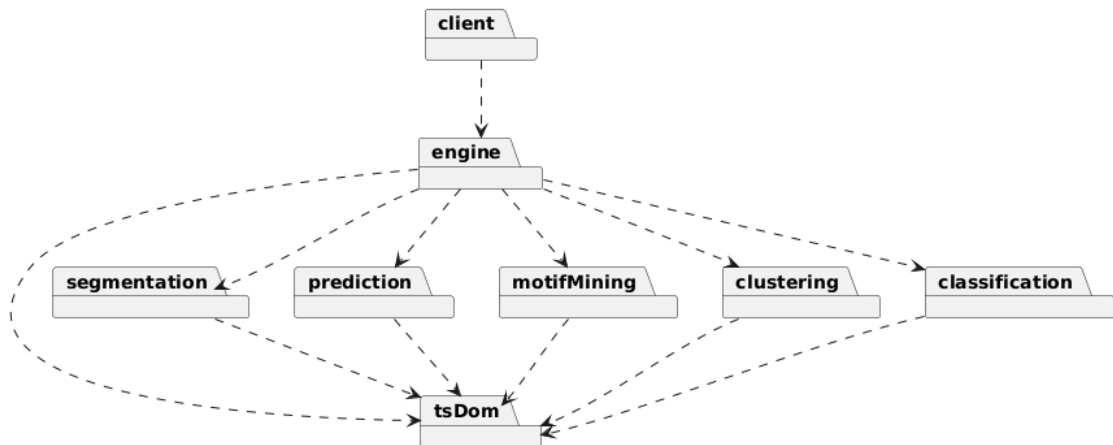
- Integrated data model for the time series
- Front-end, separate from the back-end
- Modular expansion of different algorithms (saxification, segmentation, prediction, motif extraction, outlier detection, ...)
- Handling of both multi- and single- measure time-series

Java frameworks at Github είναι διάφορα (search for “java + time-series” or “java + timeseries”):

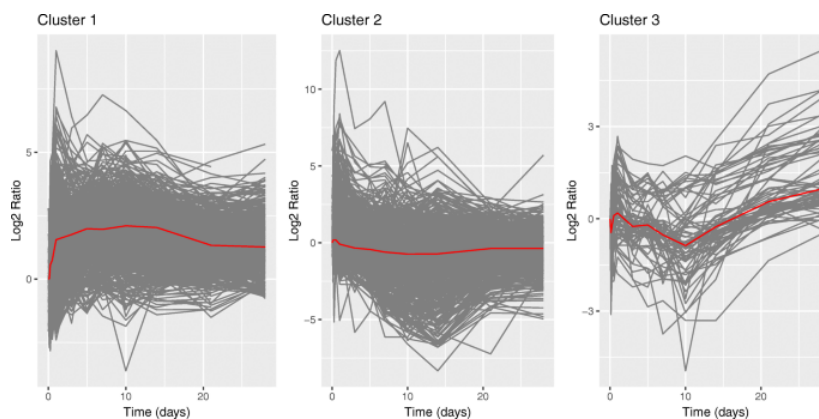
- <https://www.philippe-fournier-viger.com/spmf/index.php?link=algorithms.php> (section "Time series mining", mining algos)
- <https://github.com/elki-project/elki>
- <https://github.com/signaflo/java-timeseries>
- <https://github.com/time-series-machine-learning/tsml-java>
- <https://github.com/jiMotif> με τα SAX & iSAX projects
- <https://github.com/patrickzib/SFA>
- <https://github.com/Workday/timeseries-forecast>
- ... καθώς και μια διπλωματική από το 2020

Το ζητούμενο: να αξιολογήσουμε τι δομές δεδομένων και τι λειτουργίες προσφέρουν τα διάφορα frameworks, μια στη βάση αυτής της αρχικής αξιολόγησης, σε ποιο υπάρχον framework που θα

χρησιμοποιηθεί ως βάση για το σκοπό του συστήματος θα ενσωματώσουμε λειτουργικότητα από τα υπόλοιπα συστήματα.



Επιμέρους δουλειές με σημασία: διακριτοποίηση (via SAX), segmentation in phases (multiple techniques exist), prediction (multiple techniques exist), motif discovery (e.g., the work of A. Mueen, E. Keogh), clustering, classification.



Παράδειγμα clustering και οπτικοποίησης του κέντρου του cluster

ΕΠΙΠΕΔΟ: *Διπλωματικές για Μηχανικούς [κατά παράβαση της συνήθους τακτικής, μπορεί να το σκεφτούμε και για 2 άτομα]*

ΠΛΑΤΦΟΡΜΑ ΕΡΓΑΣΙΑΣ: Java

ΠΡΟΚΛΗΣΕΙΣ και ΟΦΕΛΗ: Τα οφέλη για ένα φοιτητή είναι: (α) εμπλοκή σε ένα χώρο με προοπτικές, αυτόν του time series management, που φαίνεται να έχει ιδιαίτερες προοπτικές στο μέλλον, (β) τεχνογνωσία σε θέματα εξόρυξης δεδομένων και ενσωμάτωσης αλγορίθμων εξόρυξης δεδομένων σε κώδικα, και (γ) hands-on σε ένα ευμέγεθες κομμάτι λογισμικού.

Η εργασία είναι πλέον κατάλληλη για φοιτητές με ταλέντο και σε θέματα διαχείρισης δεδομένων και σε προγραμματιστικά θέματα. Πρέπει να σας αρέσει πολύ η διαχείριση δεδομένων, η εξόρυξη δεδομένων και η ανάπτυξη λογισμικού.