ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΕΦΑΡΜΟΣΜΕΝΩΝ ΜΑΘΗΜΑΤΙΚΩΝ ΚΑΙ ΦΥΣΙΚΩΝ ΕΠΙΣΤΗΜΩΝ

ΤΟΜΕΑΣ ΜΗΧΑΝΙΚΗΣ

Υπολογιστική ανάλυση και μοντελοποίηση της συμπεριφοράς καταγραφής συμβάντων ιστοσελίδων του διαδικτύου με τεχνικές εξόρυξης δεδομένων (Data Mining).

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ΒΑΤΙΚΙΩΤΗΣ ΦΩΤΙΟΣ

Τριμελής Επιτροπή:

Κ. Σιέττος, Αν. Καθηγητής (Επιβλέπων)
Γ. Ματσόπουλος, Αν. Καθηγητής ΣΗΜΜΥ
Χ. Κυρανούδης, Καθηγητής ΕΜΠ

Αθήνα, Μάρτιος 2015

Βατικιώτης Φώτιος Μάρτιος 2015

Ευχαριστίες

Θα ήθελα αρχικά να ευχαριστήσω όλους όσους με βοήθησαν να έχω μια αξέχαστη εμπειρία. Η εμπειρία αυτή ήταν να μετακομίσω στην Γκρενόμπλ της Γαλλίας για μια πρακτική άσκηση διάρκειας δέκα μηνών για ένα project το οποίο τελικά έγινε η διπλωματική μου εργασία. Η πρακτική άσκηση έγινε σε συνεργασία με το Πολυτεχνείο της Βαρκελώνης.

Θα ήθελα να ευχαριστήσω τους Thibault Paramentier, Frederique Segond για την εμπιστοσύνη που μου δείξανε. Τους Elmehdi Damou και Isaac Garcia για την πολύτιμη βοήθειά τους σε θέματα προγραμματισμού και τεχνικά θέματα που είναι απαραίτητα στην Εξόρυξη Δεδομένων. Επίσης, θα ήθελα να ευχαριστήσω θερμά την διδάκτορα Βασιλική Σφύρλα από την Viseo και τον Josep Carmona Vargas καθηγητή του Πολυτεχνείου της Βαρκελώνης για την εξαιρετική συνεργασία μας. Και οι δύο ήταν πάντοτε διαθέσιμοι να μου προσφέρουν τις γνώσεις και την εμπειρία τους για την κατανόηση και την εμβάθυνση στον τομέα της Εξόρυξης Διαδικασιών.

Έπειτα θα ήθελα να ευχαριστήσω τους καθηγητές της σχολής ΣΕΜΦΕ του Εθνικού Μετσόβιου Πολυτεχνείου που με καθοδήγησαν και μου προσέφεραν απαραίτητες γνώσεις για την μετέπειτα πορεία μου. Ευχαριστώ τον κ. Κωνσταντίνο Σιέττο, πρώτον για την επίβλεψη αυτής της διπλωματικής και δεύτερων για τη συστατική επιστολή που μου προσέφερε χωρίς ενδοιασμούς όταν εγώ του το ζήτησα. Ευχαριστώ θερμά τον κ. Ηλία Ζουμπούλη, καθηγητή ΕΜΠ, υπεύθυνο για τις πρακτικές εξωτερικού, που μου έδωσε αυτή την πολύτιμη ευκαιρία να φύγω στο εξωτερικό και να ζήσω μόνος μου γνωρίζοντας δεκάδες διαφορετικούς ανθρώπους και κουλτούρες.

Σε αυτό το σημείο θα ήθελα να ευχαριστήσω ανθρώπους εκτός του στενού ακαδημαϊκού κύκλου οι οποίοι υπήρξαν σημαντικοί πόλοι στην ζωή μου. Θα ήθελα αρχικά να ευχαριστήσω την σχολική μου παρέα που παρόλο που είμαστε μακριά ήταν και είμαι σίγουρος, πως θα είναι πάντα δίπλα μου. Έπειτα θα ήθελα να ευχαριστήσω τους φίλους και τις φίλες που απέκτησα κατά την διάρκεια των φοιτητικών μου χρόνων και να τους θυμίσω τα ατελείωτα βράδια διαβάσματος που κάναμε βοηθώντας και δίνοντας κουράγιο ο ένας στον άλλο. Βέβαια, το μεγαλύτερο ευχαριστώ το οφείλω στους γονείς μου και τον αδελφό μου για την στήριξη, την εμπιστοσύνη, τις συμβουλές, τα ιδανικά και την αγάπη που μου δείχνουν όλα αυτά τα χρόνια.

Η παρούσα εργασία είναι αφιερωμένη στην οικογένεια μου και στον κ. Παναγιώτη Καραπαναγιώτη.

Περίληψη

Οι τομείς της Εξόρυξης Δεδομένων και της Εξόρυξης Διαδικασιών αποτελούν από τους πιο δραστήριους και καινοτόμους τομείς των τελευταίων ετών. Σκοπός τους είναι η εξεύρεση πληροφοριών, προτύπων και δικτύων από μεγάλες βάσεις δεδομένων με την χρήση αλγορίθμων. Στόχος τους είναι η πληροφορία που θα εξαχθεί, τα πρότυπα που θα προκύψουν και τα δίκτυα που θα παραχθούν να έχουν δομή κατανοητή προς τον άνθρωπο για να τον βοηθήσουν στην λήψη αποφάσεων.

Στην διπλωματική εργασία αρχικά, παρουσιάζονται τα αρχεία καταγραφής συμβάντων που αποτελούν την αφετηρία για τους δυο αυτούς τομείς. Στη συνέχεια, αναλύονται κάποιες από τις γλώσσες μοντελοποίησης των δικτύων που χρησιμοποιούν οι αλγόριθμοι Εξόρυξης Διαδικασιών, ο αλγόριθμος άλφα και το μαθηματικό τους υπόβαθρο. Οι αλγόριθμοι που χρησιμοποιούνται όμως, δεν είναι το ίδιο αποτελεσματικοί. Ο έλεγχος συσχέτισης του αρχείου καταγραφής συμβάντων με το μοντέλο, μας δίνει την δυνατότητα να αναγνωρίσουμε τους αποτελεσματικότερους αλγορίθμους με τεχνικές όπως είναι, το “παιχνίδι” αναπαραγωγής των tokens και η σύγκριση των αποτυπωμάτων. Αυτές οι τεχνικές έχουν αναλυθεί όπως και οι τέσσερεις διαστάσεις των μοντέλων που αποτελούν μέτρο σύγκρισης της ποιότητάς τους.

Τέλος, παρουσιάζεται ένας τρόπος δημιουργίας αρχείων καταγραφής συμβάντων από μια διαδικτυακή πλατφόρμα, η ανάλυση τους όπως επίσης και κάποιες άλλες εφαρμογές Εξόρυξης Διαδικασιών σε μεγάλα δεδομένα.

Abstract

Nowadays, Data Mining and Process Mining are two of the most active and innovative research fields. Their objective is the extraction of knowledge, norms and networks from large databases by using algorithms in order to provide guidance on decision points.

The starting point of the whole process is the event logs which are presented in the first chapters. We continue with analyzing the mathematical backgrounds of some process modeling languages which are used by Process Mining algorithms as well as with the well-known alpha algorithm. The algorithms that are used by Process Mining are not equally effective. Conformance checking techniques determine the most efficient algorithms by using methods like the token-based replay and the comparison of causal footprints. These techniques have been analyzed as well as the four model quality dimensions that are a yardstick of process models.

Finally, a way to create event logs from an online platform is presented, our approach as well as some other Process Mining applications on large data.

Περιεχόμενα

1. Εισαγωγή

Η διπλωματική αυτή αποτελεί μέρος μιας μελέτης που έγινε κατά τη διάρκεια μιας πρακτικής άσκησης δέκα μηνών. Η εργασία αυτή έγινε στην εταιρεία Viseo R&D που εδρεύει στην Γκρενόμπλ της Γαλλίας σε συνεργασία με το Εθνικό Μετσόβιο Πολυτεχνείο (ΕΜΠ) και το Πολυτεχνείο της Βαρκελώνης (UPC).

Η Viseo (www.viseo.net) χρονολογείται από το 2000 και ειδικεύεται σε λύσεις διαχείρισης, συμβουλές διαχείρισης καθώς και στην παροχή βοήθειας εταιρειών για τα συστήματα πληροφοριών τους. Η επιχειρηματική μονάδα R&D της Viseo, αποσκοπεί στην διερεύνηση νέων αγορών και στην πρόβλεψη των αναγκών των πελατών της αξιοποιώντας τις τεχνικές και της επιχειρηματικές εμπειρίες της ομάδας της. Η μονάδα αυτή επικεντρώνεται σε τρία ερευνητικά θέματα: Ανάλυση Δεδομένων (Data Analysis), Τεχνολογίες Λογισμικών (Software Engineering) και Natural User Interface. Η συνεργασία αυτή έγινε με τον Josep Carmona Vargas καθηγητή του UPC που ειδικεύεται στην Εξόρυξη Διαδικασιών και την Βασιλική Σφύρλα διδάκτορα που εργάζεται στην Viseo.

Η βασική ιδέα της μελέτης είναι η Εξόρυξη Δεδομένων από μια διαδικτυακή πλατφόρμα και η ανάλυση των δεδομένων με τεχνικές Εξόρυξης Διαδικασιών με απώτερο στόχο τη βελτίωση της πλατφόρμας. Είναι σημαντικό να τονίσουμε πως τα δεδομένα καταγράφονται μόνο για ερευνητικούς σκοπούς, για να κατανοήσουμε από αυτά τι χρειάζονται οι χρήστες πραγματικά από το πρόγραμμα, και σε καμία περίπτωση για λόγους παρακολούθησης. Είναι δεδομένο άλλωστε πως ζούμε στην εποχή της πληροφόρησης και της πληροφορίας είναι όμως αναγκαίο οι πληροφορίες να χρησιμοποιούνται για καλό σκοπό και να μας εξυπηρετούν.

Τα τελευταία χρόνια η τεχνολογία αναπτύσσεται ραγδαία. Ο περιβόητος νόμος του Moore (1965) άλλωστε έχει περιγράψει τον διπλασιασμό των τρανζίστορ κάθε 18 μήνες που έχει σαν άμεσο αποτέλεσμα την “έκρηξη” της ποσότητας των αποθηκευμένων δεδομένων. Κατά την διάρκεια των τελευταίων πενήντα ετών ο αριθμός των τρανζίστορ σε ολοκληρωμένα κυκλώματα όντως διπλασιάζεται κάθε δυο χρόνια. Επίσης, οι χωρητικότητες των δίσκων, οι επιδόσεις των υπολογιστών ανά μονάδα κόστους όπως και ο αριθμός των pixels ανά δολάριο έχουν αυξηθεί με έναν παρόμοιο ρυθμό. Αυτές οι τεχνολογικές εξελίξεις, επιβάλουν στις επιχειρήσεις και τους οργανισμούς να εξαρτώνται όλο και περισσότερο από τα ηλεκτρονικά συστήματα και τις πηγές πληροφόρησης. Μελέτη άλλωστε έχει δείξει [1] την θεαματική ανάπτυξη των αποθηκευμένων δεδομένων αφού μέσα σε επτά μήνες οι αποθηκευμένες πληροφορίες στο διαδίκτυο ξεπέρασαν τα 34 Zettabytes. Για να καταλάβουμε το ποσό της πληροφορίας, αν θα έπρεπε να στοιβάξουμε 35 Zettabytes από CD, η στοίβα μας θα έφτανε στη μισή απόσταση έως τον Άρη [2].

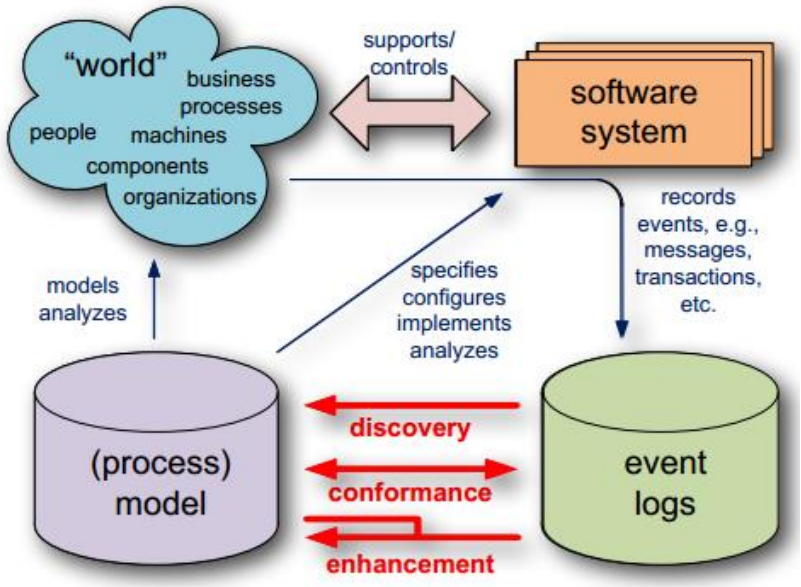
Όπως μπορούμε να καταλάβουμε λοιπόν είναι αναγκαίο να αναπτυχθούν τεχνικές για την Εξόρυξη και την μελέτη αυτών των πληροφοριών που σε πολλές περιπτώσεις είναι

σημαντικές και παραμένουν ανεκμετάλλευτες. Η Εξόρυξη Διαδικασιών είναι ένας καινοτόμος τομέας που επικεντρώνεται στην μελέτη δεδομένων για την εξεύρεση μοντέλων που αποτυπώνουν τον ακριβή τρόπο με τον οποίο εκτελούνται οι ενέργειες σε μια επιχείρηση ή έναν οργανισμό.

2. Εξόρυξη Διαδικασιών

Σήμερα, η πλειοψηφία των οργανισμών και των online επιχειρήσεων χρησιμοποιούν πληροφοριακά συστήματα για την υποστήριξη της εκτέλεσης των επιχειρηματικών διαδικασιών τους [3]. Παραδείγματα τέτοιων πληροφοριακών συστημάτων που υποστηρίζουν επιχειρησιακές διαδικασίες είναι τα *Συστήματα Διαχείρισης Ροών-Εργασιών (Work-flow Management Systems (WMS)) [4], Συστήματα Διαχείρισης Πελατειακών Σχέσεων (Customer Relationship Management (CRM) systems), Συστήματα Διαχείρισης Επιχειρησιακών Πόρων (Enterprise Resource Planning (ERP) systems)* και ούτω καθεξής.

Αυτά τα πληροφοριακά συστήματα μπορεί να περιέχουν ένα σαφές μοντέλο το οποίο ακολουθούν, ή μπορεί απλά να καταγράφουν ενέργειες οι οποίες έχουν γίνει χωρίς να παρέχουν καμία υποστήριξη για την εκτέλεση αυτών των ενεργειών. Σε κάθε περίπτωση, αυτά τα πληροφοριακά συστήματα υποστηρίζουν δυνατότητες καταγραφής. Αυτά τα αρχεία καταγραφής που παράγονται, συνήθως περιέχουν δεδομένα σχετικά με τις ενέργειες που έχουν εκτελεστεί στην οργάνωση, τις ώρες κατά τις οποίες εκτελέστηκαν, τα πρόσωπα ή τα συστήματα που εκτελούν τις ενέργειες όπως και άλλου είδους δεδομένα σχετικά με τις ενέργειες αυτές. Αυτά τα αρχεία αποτελούν το σημείο εκκίνησης για την *Εξόρυξη Διαδικασιών (Process Mining)* και καλούνται συνήθως *αρχεία καταγραφής συμβάντων (event logs)*.



Σχήμα 1: Παρουσίαση των βασικών τύπων της Εξόρυξης Διαδικασιών. [2]

Η Εξόρυξη Διαδικασιών στοχεύει στην αυτόματη πληροφόρηση από ένα αρχείο καταγραφής συμβάντων. Αυτού του είδους η πληροφόρηση μπορεί να χρησιμοποιηθεί για την ανάπτυξη νέων συστημάτων που υποστηρίζουν την εκτέλεση των επιχειρησιακών διαδικασιών ή ως εργαλείο ανατροφοδότησης που βοηθά στην επεξεργασία, την ανάλυση

και τη βελτίωση των ήδη θεσπισμένων επιχειρησιακών διαδικασιών. Το κύριο πλεονέκτημα των τεχνικών Εξόρυξης Διαδικασιών είναι πως οι πληροφορίες συγκεντρώνονται αντικειμενικά. Με άλλα λόγια, η Εξόρυξη Διαδικασιών είναι χρήσιμη διότι συγκεντρώνονται πληροφορίες σχετικά με το τι συμβαίνει στην πραγματικότητα, σύμφωνα με ένα αρχείο καταγραφής συμβάντων ενός οργανισμού ή μίας επιχείρησης και όχι αυτό που οι άνθρωποι πιστεύουν ότι συμβαίνει.

Διακρίνονται τρεις βασικοί τύποι Εξόρυξης Διαδικασιών [5] όπως φαίνεται στο Σχήμα 1:

- **Ανακάλυψη (Discovery):** Στόχος είναι η παραγωγή ενός πρωταρχικού μοντέλου μιας και δεν υπάρχει ένα πρωταρχικό μοντέλο. Το πρωταρχικό μοντέλο δημιουργείται από ένα αρχείο καταγραφής συμβάντων χωρίς τη χρήση οποιαδήποτε άλλης πληροφορίας. Υπάρχουν δεκάδες τεχνικές για την εξαγωγή ενός μοντέλου διεργασίας από ένα αρχείο καταγραφής συμβάντων. Για παράδειγμα ο κλασικός α-αλγόριθμος (a-algorithm), όπως θα δούμε και αναλυτικότερα σε επόμενο κεφάλαιο, έχει την δυνατότητα να ανακαλύψει ένα Petri-net με τον προσδιορισμό των βασικών διαδικαστικών μοτίβων από ένα αρχείο καταγραφής συμβάντων. Η μέθοδος ανακάλυψης χρησιμοποιείται συχνά ως ένα σημείο εκκίνησης για όλους τους τύπους ανάλυσης.

- **Συσχέτιση (Conformance):** Για να φτάσουμε σε αυτό τον τύπο Εξόρυξης Διαδικασιών, θα πρέπει να υπάρχει ένα πρωταρχικό μοντέλο. Το υπάρχον μοντέλο συγκρίνεται με ένα αρχείο καταγραφής συμβάντων της ίδιας διαδικασίας. Η σύγκριση αυτή, δείχνει που η πραγματική διαδικασία διαφέρει από την μοντελοποιημένη διαδικασία. Επιπλέον, είναι δυνατόν να ποσοτικοποιηθεί το επίπεδο συσχέτισης και μπορούν να διαγνωστούν πιθανές διαφορές. Ο έλεγχος συσχέτισης μπορεί να χρησιμοποιηθεί για να ελεγχθεί κατά πόσο η πραγματικότητα, όπως καταγράφεται στο αρχείο καταγραφής συμβάντων συμφωνεί με το πρωταρχικό μοντέλο και το αντίστροφο.

- **Βελτίωση (Enhancement):** Υπάρχει ένα πρωταρχικό μοντέλο. Το μοντέλο αυτό έχει επεκταθεί με μια νέα πτυχή ή προοπτική, ο στόχος δεν είναι να ελεγχθεί η συσχέτιση αλλά να εμπλουτιστεί το μοντέλο. Για παράδειγμα, χρησιμοποιώντας χρονικές σημάνσεις στο αρχείο καταγραφής συμβάντων είναι δυνατόν να επεκταθεί το μοντέλο μας ώστε να βρεθούν τα σημεία όπου χάνεται πολύς χρόνος, να βρεθούν οι συχνότητες, τα επίπεδα υπηρεσιών κ.α.

Όπως μπορούμε να καταλάβουμε και οι τρείς τύποι Εξόρυξης Διαδικασιών έχουν ένα κοινό χαρακτηριστικό, ότι υποθέτουν την ύπαρξη κάποιου αρχείου καταγραφής συμβάντων. Το φαινόμενο όμως που οι πιο πολλές από τις καθημερινές μας δραστηριότητες σιωπηρά καταγράφονται σε στοιχεία καταγραφής συμβάντων και βάσεις

δεδομένων όλο και αυξάνονται με ταχείς ρυθμούς. Για αυτό το λόγο υπάρχει πραγματική ανάγκη για τεχνικές όπως η Εξόρυξη Διαδικασιών, η οποία μπορεί να βοηθήσει στην ανάλυση των πληροφοριών που σχετίζονται με τις διεργασίες που καταγράφονται από τα αρχεία καταγραφής συμβάντων. Το πεδίο Εξόρυξη Διαδικασιών έχει αναπτύξει ένα μεγάλο αριθμό τεχνικών που μπορούν να εφαρμοστούν οι οποίες παράγουν τα *μοντέλα διαδικασιών (process models).* Έτσι λοιπόν, δεν υπάρχει καμία ανάγκη να σκεφτούμε το μοντέλο διαδικασιών, επειδή οι αλγόριθμοι οι οποίοι έχουν αναπτυχθεί για την Εξόρυξη Διαδικασιών μπορούν να δημιουργήσουν ένα μοντέλο διαδικασίας με έναν αυτόματο και μη επιβλέψιμο τρόπο. Επιπλέον, οι αλγόριθμοι αυτοί μπορούν να αναλύσουν μεγάλες ποσότητες δεδομένων σε πολύ σύντομο χρονικό διάστημα, έτσι ώστε η διαδικασία να μπορεί να πραγματοποιηθεί διαδραστικά και να μπορεί να γίνει μέρος της καθημερινής ρουτίνας για τη διάγνωση διαδικασιών.

3. Αρχεία καταγραφής συμβάντων (event logs)

Τα αρχεία καταγραφής συμβάντων περιγράφουν το ιστορικό εκτελέσεων μιας διαδικασίας, όπως αυτό καταγράφεται στις ροές γεγονότων. Μπορούν να ερμηνευθούν ως παρατηρήσεις αυτής της διαδικασίας όπως ακριβώς αυτή έχει εκτελεστεί. Οι διαδικασίες ορίζονται συνήθως με ένα σκόπιμα ασαφή τρόπο ή δεν ορίζονται καθόλου. Έτσι λοιπόν τα αρχεία καταγραφής συμβάντων είναι η πιο συγκεκριμένη και πραγματική (και συχνά η μόνη) πηγή πληροφοριών η οποία περιγράφει τη ροή διαδικασιών σε μια επιχείρηση ή έναν οργανισμό. Ταυτόχρονα, τα αρχεία καταγραφής συμβάντων είναι το σημείο εκκίνησης της Εξόρυξης Διαδικασιών, διότι προκειμένου να δημιουργηθεί το μοντέλο μας συλλέγουμε πληροφορίες σχετικές με την διαδικασία όπως αυτή έχει συμβεί. Τα αρχεία καταγραφής συμβάντων είναι κυρίως δεδομένα τα οποία μπορούν να εξαχθούν από τις βάσεις δεδομένων και η μορφή τους διαφέρει από το ένα αρχείο καταγραφής συμβάντων στο άλλο.

Όπως αναφέραμε και προηγουμένως τα αρχεία καταγραφής συμβάντων παρέχουν πληροφορίες σχετικά με μια διαδικασία η οποία περιλαμβάνει γεγονότα. Έτσι κάθε φορά που συναντούμε ένα αρχείο καταγραφής συμβάντων περιμένουμε να δούμε μια μήτρα όπου η κάθε γραμμή περιέχει πληροφορίες ενός γεγονότος. Ένα γεγονός όμως είναι μια πολύ γενική έννοια αλλά τα διαθέσιμα στοιχεία είναι συνήθως περισσότερα από αρκετά για να περιγράψουν αυτό το γεγονός. Σε αυτό το κεφάλαιο θα κάνουμε μια εισαγωγή στις θεμελιώδεις έννοιες και τα θεωρήματα τα οποία είναι σχετικά με τα αρχεία καταγραφής συμβάντων. Στην πρώτη ενότητα παρουσιάζονται οι προϋποθέσεις και η γενική δομή των αρχείων καταγραφής συμβάντων. Στη συνέχεια, δίνεται ένα παράδειγμα ενός αρχείου καταγραφής συμβάντων και το παραγόμενο μοντέλο (Petri-net) το οποίο παράγεται από αυτό το αρχείο.

3.1. Δομή και προϋποθέσεις των αρχείων καταγραφής συμβάντων.

Υποθέτουμε ότι ένα αρχείο καταγραφής συμβάντων περιέχει δεδομένα σχετικά με μια ενιαία διαδικασία και επιπλέον ότι κάθε συμβάν θα πρέπει να αναφέρεται σε μια μόνο περίπτωση. Υποθέτουμε επίσης, πως τα γεγονότα μπορεί να σχετίζονται με κάποια δραστηριότητα. Αυτές οι υποθέσεις είναι απόλυτα φυσιολογικές στο πλαίσιο της Εξόρυξης Διαδικασιών. Όλα τα κύρια σύμβολα της διαδικασίας μοντελοποίησης καθορίζουν μια διαδικασία σαν μια συλλογή από δραστηριότητες. Ως εκ τούτου, οι ελάχιστες απαιτήσεις για τα αρχεία καταγραφής συμβάντων και της Εξόρυξης Διαδικασιών είναι:

1. **Case(περίπτωση)**

   Αυτό το μέγεθος εκφράζει τις περιπτώσεις κατά τις οποίες έχουν εκτελεστεί οι δραστηριότητες. Είναι πολύ σημαντικό να αναφερθεί ότι χωρίς τις περιπτώσεις, δεν είναι δυνατόν να προσδιοριστεί, πότε οι διαδικασίες αρχίζουν και πότε τελειώνουν.

2. **Event(γεγονός)**

   Τα γεγονότα εκφράζουν τις δραστηριότητες που έχουν εκτελεστεί. Όταν ένα γεγονός εκτελείται για μια συγκεκριμένη περίπτωση, αναφερόμαστε σε αυτό σαν μια δραστηριότητα.

3. **Τα γεγονότα να είναι ταξινομημένα**

   Η τελευταία απαίτηση της Εξόρυξης Διαδικασιών είναι ότι τα γεγονότα θα πρέπει να είναι ταξινομημένα. Γνωρίζουμε ότι σε ένα αρχείο καταγραφής συμβάντων αν κάποιο συμβάν προηγείται ενός άλλου στην μήτρα τότε το πρώτο είναι αυτό που συνέβη πριν από το δεύτερο. Έτσι λοιπόν το αρχείο καταγραφής συμβάντων πρέπει να είναι ταξινομημένο από την άποψη του χρόνου. Συνήθως τα περισσότερα από τα αρχεία καταγραφής συμβάντων περιέχουν πρόσθετες πληροφορίες ανά περίπτωση οι οποίες ονομάζονται *χαρακτηριστικά (attributes)* των γεγονότων, ένα από αυτά είναι και η χρονική σήμανση των γεγονότων. Με αυτό τον τρόπο μπορούμε να καταλάβουμε τη σειρά των ενεργειών.

Οι πρόσθετες πληροφορίες ανά ενέργεια είναι πολύ χρήσιμες όταν κάνουμε την ανάλυση, π.χ. για την ανάλυση σχετικά με τον χρόνο αναμονής μεταξύ δύο δραστηριοτήτων. Συνήθως, αναφέρονται πληροφορίες σχετικά με τους πόρους, δηλαδή, τα πρόσωπα τα οποία εκτελούν τις δραστηριότητες, τα κόστη που σχετίζονται με τα γεγονότα κ.α. Για να είμαστε σε θέση να αναφερόμαστε στα αρχεία καταγραφής συμβάντων και να προσδιορίσουμε με ακρίβεια τις απαιτήσεις είναι αναγκαίο να επισημοποιήσουμε τις διάφορες έννοιες, έτσι λοιπόν.

Ορισμός 3.1 (Γεγονότα (Events), Χαρακτηριστικά (Attributes)): Έστω Ε το σύνολο όλων των πιθανών γεγονότων. Τα γεγονότα μπορεί να χαρακτηρίζονται από διάφορα χαρακτηριστικά, για παράδειγμα ένα γεγονός μπορεί να έχει μια χρονική σήμανση, να εκτελείται από ένα συγκεκριμένο πρόσωπο, να έχει αντίστοιχο κόστος, κλπ. Έστω λοιπόν, AN ένα σύνολο από ονόματα χαρακτηριστικών. Για κάθε γεγονός $e \in E$ και για κάθε όνομα $n \in AN$: $\#_n(e)$ είναι η τιμή του χαρακτηριστικού n για το γεγονός e. Εάν το γεγονός e δεν έχει ένα χαρακτηριστικό n, τότε $\#_n(e) = \perp$ (μηδενική αξία).

Για ευκολία συγκρατούμε τα παρακάτω τυπικά χαρακτηριστικά:

- $\#_{activity}(e)$ είναι η δραστηριότητα που σχετίζεται με το γεγονός e.
- $\#_{time}(e)$ είναι η χρονική σήμανση του γεγονότος e.
- $\#_{resource}(e)$ είναι ο πόρος που σχετιζόταν με ένα γεγονός e.
- $\#_{trans}(e)$ είναι το είδος της πράξης που σχετίζεται με ένα γεγονός e, διάφορα παραδείγματα είναι schedule, start, complete, και suspend.

Έτσι λοιπόν για να κλείσουμε αυτήν την ενότητα, εκτός από τις επιπρόσθετες πληροφορίες που αναφέρονται ως χαρακτηριστικά χωρίς τις περιπτώσεις, τα γεγονότα και την ταξινόμηση των γεγονότων δεν είμαστε σε θέση να προσδιορίσουμε ένα *μοντέλο διαδικασίας* το οποίο βασίζεται στις εκτελεσμένες πληροφορίες. Τα αρχεία καταγραφής συμβάντων μπορούν να βρεθούν σε πολλές μορφές, για παράδειγμα txt files, κείμενο, csv files κλπ. Ένας τεχνικός περιορισμός λοιπόν, είναι πως τα εργαλεία που χρησιμοποιούνται για την Εξόρυξη Διαδικασιών δέχονται μόνο XML αρχεία και θα πρέπει να βρεθεί ένας τρόπος για την μετατροπή των αρχείων από κάθε τύπο αρχείου σε XML αρχείο. Αυτό όμως δεν αποτελεί εμπόδιο αφού υπάρχουν τρόποι επεξεργασίας των δεδομένων όπου θα δούμε λεπτομερέστερα και σε επόμενο κεφάλαιο. Στο επόμενο κεφάλαιο θα αναλύσουμε ένα παράδειγμα για την παραγωγή ενός μοντέλου από ένα αρχείο καταγραφής συμβάντων και θα αναφερθούμε σε κάποιους βασικούς κανόνες των Petri-nets.

Παράδειγμα

Case identifier	Task identifier
Case 1	Task A
Case 2	Task A
Case 3	Task A
Case 3	Task B
Case 1	Task B
Case 1	Task C
Case 2	Task C
Case 4	Task A
Case 2	Task B
Case 2	Task D
Case 5	Task A
Case 4	Task C
Case 1	Task D
Case 3	Task C
Case 3	Task D
Case 4	Task B
Case 5	Task E
Case 5	Task D
Case 4	Task D

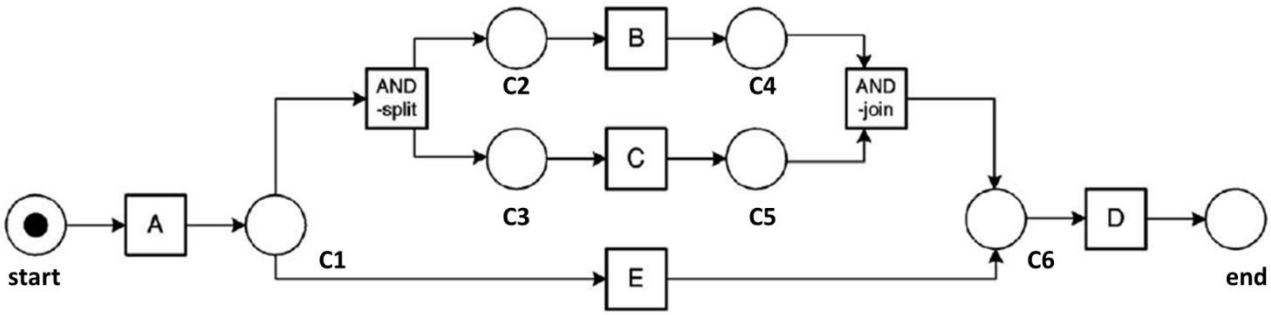
Πίνακας 1: Αρχείο καταγραφής συμβάντων με χρονικά κατανεμημένες ενέργειες. [6]

Έστω το αρχείο καταγραφής συμβάντων όπου φαίνεται στον Πίνακα 1. Αυτό το αρχείο περιέχει πληροφορίες για 5 διαφορετικές περιπτώσεις (Cases). Σύμφωνα με τις περιπτώσεις οι εργασίες (Tasks) εκτελούνται ως εξής:

- Case 1: A → B → C → D
- Case 2: A → C → B → D
- Case 3: A → B → C → D
- Case 4: A → C → B → D
- Case 5: A → E → D

Μπορούμε να παρατηρήσουμε πως όλες οι περιπτώσεις ξεκινούν με την εκτέλεση του A και κλείνουν με την εκτέλεση του D. Περαιτέρω, παρατηρούμε ότι εάν εκτελεστεί η εργασία B τότε η εργασία C εκτελείτε στην ίδια περίπτωση και το αντίθετο. Τέλος, παρατηρούμε πως σε περίπτωση όπου εκτελεστεί η εργασία E οι εργασίες B και C δεν εμφανίζονται καθόλου.

Από τις πληροφορίες που παρέχονται από το αρχείο καταγραφής συμβάντων που φαίνεται στον Πίνακα 1 μπορούμε να καταλήξουμε στο μοντέλο που φαίνεται στο Σχήμα 2. Το μοντέλο που παρουσιάζεται εδώ είναι ένα Petri-net. Τα Petri-nets είναι μοντέλα τα οποία αναπαριστούν γραφικά την εκτέλεση των περιπτώσεων οι οποίες είναι καταγεγραμμένες σε κάποιο αρχείο καταγραφής συμβάντων. Για την καλύτερη και ακριβέστερη αναπαράσταση των αρχείων καταγραφής συμβάντων τα Petri-nets ακολουθούν αυστηρούς κανόνες.



Σχήμα 2: Μοντέλο που περιγράφει τις ενέργειες που εκτελούνται στον Πίνακα 1. [6]

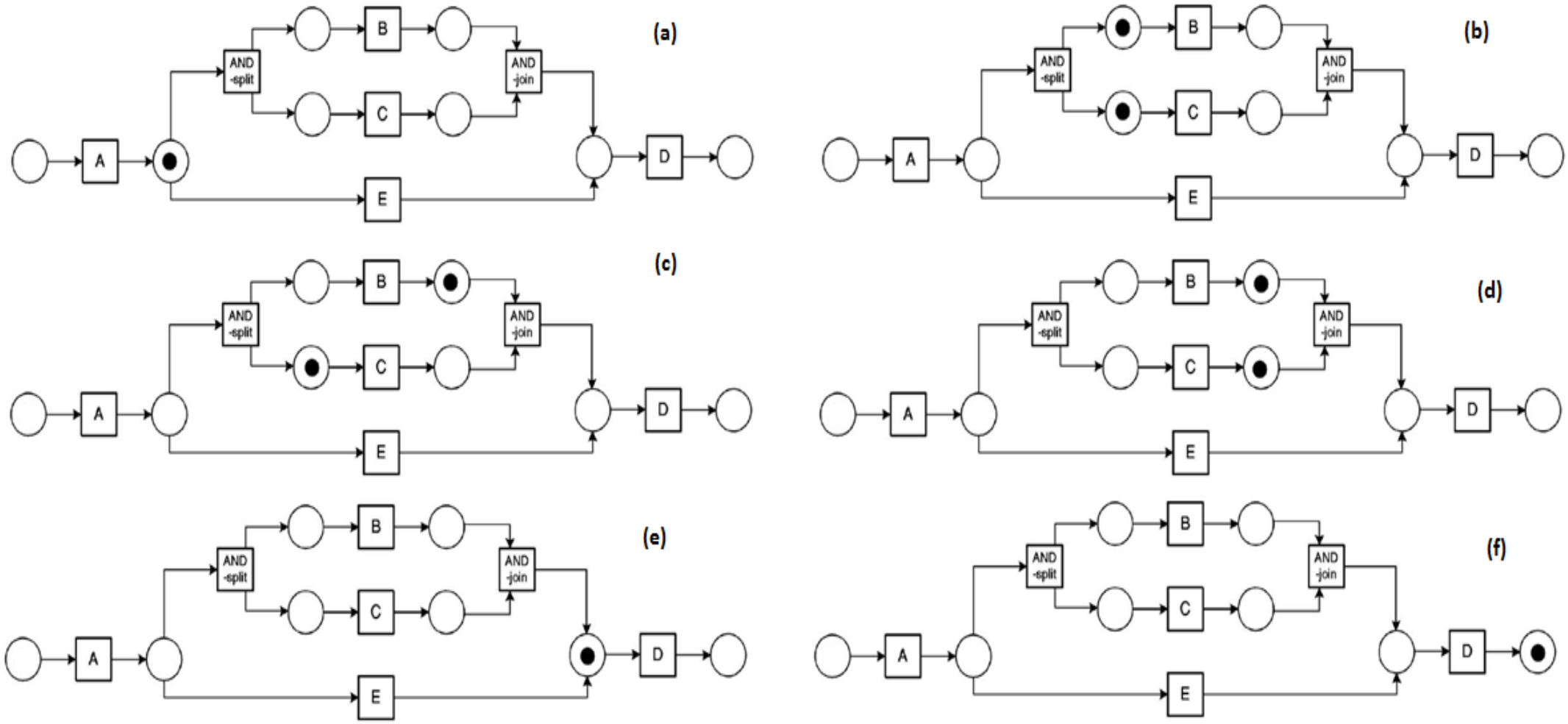
Έτσι λοιπόν, ένα Petri-net είναι ένα δίκτυο το οποίο αποτελείται από *τόπους (places)* και *μεταβάσεις (transitions).* Στο διάγραμμα οι κύκλοι αντιπροσωπεύουν τους τόπος ενώ τα τετράγωνα τις μεταβάσεις. Οι τόποι και οι μεταβάσεις ενώνονται μεταξύ τους με βέλη (arcs) και παρατηρούμε πως οι μεταβάσεις είναι το μέσον ώστε να κινηθούμε από τον ένα τόπο στον άλλο. Η κίνηση αυτή όμως είναι αδύνατη αφού το δίκτυό μας είναι στατικό, για αυτό το λόγο λοιπόν, μέσα στο Petri-net υπάρχουν αυτά που τα ονομάζουμε tokens τα οποία έχουν την δυνατότητα να κινούνται από τον ένα τόπο στον άλλο. Σε ένα Petri-net θα μπορούσε να κινείται την ίδια στιγμή ένα ή και περισσότερα tokens. Οι τόποι έχουν την ιδιότητα να συγκρατούν tokens, οι μεταβάσεις από την άλλη έχουν την ιδιότητα να καταναλώνουν tokens για να μπορέσουν να πυροδοτηθούν και μετά την πυροδότησή τους να παράγουν tokens τα οποία κινούνται στον επόμενο τόπο. Λέμε ότι *μια μετάβαση είναι*

ενεργοποιημένη (transition is enabled) αν όλοι οι τόποι που περιλαμβάνουν τις εισροές της μετάβασης κατέχουν ένα τουλάχιστον token. Μια ενεργοποιημένη μετάβαση μπορεί να πυροδοτηθεί καταναλώνοντας ένα token από κάθε τόπο εισροών και παράγωντας ένα token για κάθε τόπο εκροών.

Στο παράδειγμά μας παρατηρούμε πως το δέντρο μας ξεκινά πάντα με την εκτέλεση της εργασίας A, συνεχίζει με την επιλογή είτε με την επιλογή του B ή του C όπου είναι παράλληλα συνδεδεμένα είτε με την εκτέλεση του E ως μια ενιαία εργασία και κλείνει πάντοτε με την εκτέλεση του D. Για την εκτέλεση των B και C παράλληλα, έχουμε προσθέσει δύο *μη παρατηρήσιμες εργασίες (non-observable tasks)* τις AND-split και AND-join. Οι μη παρατηρήσιμες εργασίες δεν υπάρχουν στο αρχείο καταγραφής ροής εργασιών αλλά έχουν προστεθεί στο μοντέλο μας για λόγους δρομολόγησης των 'tokens'.

Εάν θέλουμε να αναπαραστήσουμε την περίπτωση 1 (Case 1) του αρχείου καταγραφής συμβάντων A → B → C → D θα ξεκινήσουμε με την τοποθέτηση ενός token στον αρχικό τόπο Σχήμα 2. Η μετάβαση A θα καταναλώσει το token που τοποθετήθηκε στον αρχικό τόπο, θα πυροδοτηθεί για να εκτελεστεί το A της περίπτωσης 1 (Case 1) και εν συνεχεία θα παράξει ένα νέο token που θα κινηθεί με την σειρά του στον επόμενο τόπο Σχήμα 3 (a). Στη συνέχεια, η μετάβαση AND-split καταναλώνει ένα token δεν εκτελεί καμία εργασία αφού αναφέραμε και προηγούμενος πως είναι μη παρατηρήσιμη εργασία και θα παράξει δύο tokens που θα συγκρατηθούν από τους δύο τόπους εκροών Σχήμα 3 (b). Η μετάβαση B θα καταναλώσει ένα token για την εκτέλεση του B και έτσι θα έχουμε, A → B Σχήμα 3 (c). Η μετάβαση C στη συνέχεια, καταναλώνει ένα token παράγωντας άλλο ένα που τοποθετείται στον αντίστοιχο τόπο εκροών. Μέχρι στιγμής έχουμε εκτελέσει A → B → C και έχουμε δύο tokens στους τόπους εισροών της AND-join μη παρατηρήσιμης εργασίας Σχήμα 3 (d), η οποία καταναλώνει δυο tokens και παράγει ένα Σχήμα 3 (e). Τέλος, η μετάβαση D καταναλώνει ένα token για την εκτέλεση της εργασίας D και ολοκλήρωσης εκτέλεσης της περίπτωσης 1 Σχήμα 3 (f).

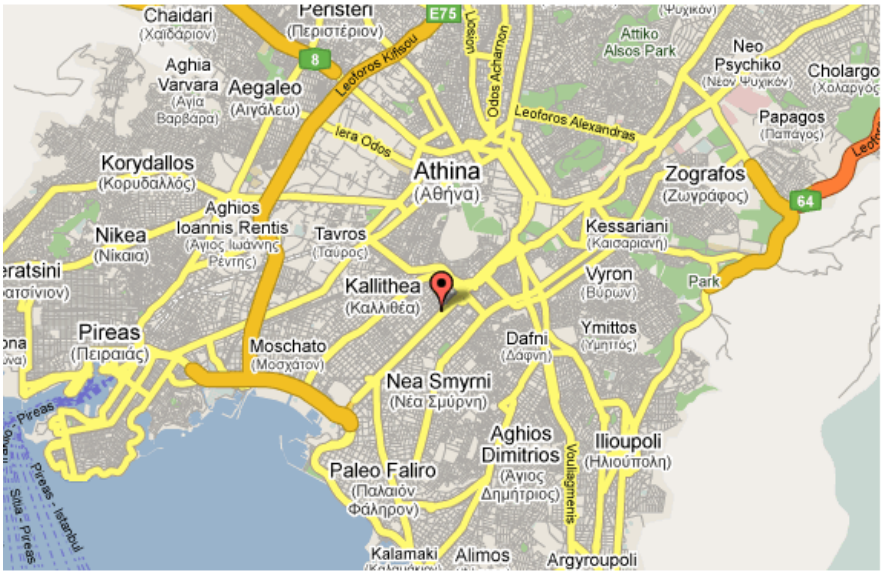
Σε αυτό το κεφάλαιο κάναμε μια εισαγωγή στα Petri-nets και στον *κανόνα πυροδότησής (firing rule)* ώστε να δούμε πως κινούνται τα tokens πάνω στο στατικό μοντέλο. Στα επόμενα κεφάλαια θα αναλύσουμε περισσότερο τα Petri-nets όπως και άλλους τύπους μοντελοποίησης που χρησιμοποιούνται σήμερα. Επίσης, θα μελετήσουμε έναν αλγόριθμο ο οποίος παράγει αυτόματα Petri-nets από αρχεία καταγραφής συμβάντων.



Σχήμα 3: Αναπαραγωγή ίχνους A → B → C → D στο Petri-net.

4. Μοντέλα Διαδικασιών (Process Models)

Υπάρχουν πολλές και διαφορετικές γλώσσες μοντελοποίησης διαδικασιών. Οι περισσότερες από αυτές, παρέχουν μια γραφική σημειογραφία με κόμβους διάσπασης και με κόμβους ένωσης, ειδικότερα AND, XOR κλπ. Όλες αυτές οι γλώσσες έχουν πολύ σημαντικές διαφορές αλλά όμως όλες υπάρχουν για τον ίδιο λόγο, για να συντάξουν και να εξηγήσουν τη δρομολόγηση των περιπτώσεων των επιχειρηματικών διαδικασιών. Αυτά τα μοντέλα φαίνεται πως είναι μια καλή προσέγγιση, μιας και παρατηρούνται συχνά στη βιομηχανία. Ωστόσο, η εμπειρία μας μέσω της Εξόρυξης Διαδικασιών αποκάλυψε πολλές αδυναμίες σχετικά με την κλασική προσέγγιση. Τα λεγόμενα *"σπαγγέτι μοντέλα" (spaghetti-like models)*, που θα δούμε αναλυτικότερα σε επόμενο κεφάλαιο, αποδεικνύουν ότι τα παραγόμενα μοντέλα είναι πολλές φορές δύσκολο να γίνουν κατανοητά και επίσης υστερούν από θέμα εκφραστικότητας. Για αυτό το λόγο, τα μοντέλα μας θα πρέπει να αντιμετωπίζονται όπως οι γεωγραφικοί χάρτες αφού ο τομέας της χαρτογράφησης έρχεται αντιμέτωπος με παρόμοιες προκλήσεις, δηλαδή την απλούστευση εξαιρετικά πολύπλοκων και αδόμητων τοπολογιών. Οι δραστηριότητες σε μια διαδικασία μπορεί να συσχετιστούν με τις θέσεις σε μια τοπολογία (π.χ. πόλεις ή διασταυρώσεις) και οι συσχετίσεις των δραστηριοτήτων με κυκλοφοριακές συνδέσεις (π.χ. αυτοκινητόδρομοι σιδηρόδρομοι). Όταν ρίξουμε μια προσεκτική ματιά σε χάρτες, μπορούμε να εξάγουμε μια σειρά από πολύτιμες έννοιες από αυτούς.



Σχήμα 4: Οι μέθοδοι χαρτογράφησης χρησιμοποιούνται και στην μοντελοποίηση διαδικασιών.

- **Συσσωμάτωση (Aggregation):** Για τον περιορισμό των πληροφοριών που εμφανίζονται, οι χάρτες συχνά συγκεντρώνουν χαμηλού επιπέδου πληροφορίες σε μια συστάδα. Ένα παράδειγμα είναι οι πόλεις σε οδικούς χάρτες, όπου τα σπίτια οι δρόμοι συνδυάζονται μέσα στην κλειστότητα της πόλης και φαίνονται μόνο οι μεγάλοι αυτοκινητόδρομοι (Σχήμα 4).
- **Έμφαση (Emphasis):** Οι σημαντικότερες πληροφορίες τονίζονται με πολλά οπτικά μέσα όπως το χρώμα, την αντίθεση και το μέγεθος. Για παράδειγμα οι χάρτες δίνουν έμφαση στους πιο μεγάλους και σημαντικούς δρόμους, εμφανίζοντάς τους παχύτερους και με πιο έντονο χρώμα (Σχήμα 4).
- **Προσαρμογή (Customization):** Δεν υπάρχει ένας ενιαίος χάρτης για όλο τον κόσμο και όλα τα μέρη. Οι χάρτες είναι εξειδικευμένοι σε ένα συγκεκριμένο τοπικό πλαίσιο, έχουν ένα συγκεκριμένο επίπεδο λεπτομέρειας (χάρτες πόλεων και χάρτες εθνικών οδών), καθώς και έναν ειδικό σκοπό.

Αυτές οι έννοιες είναι καθολικές, κατανοητές και χρησιμοποιούνται στις περισσότερες γλώσσες μοντελοποίησης διαδικασιών. Στις επόμενες ενότητες θα μελετήσουμε μερικές γλώσσες μοντελοποίησης που χρησιμοποιούνται για την Εξόρυξη Διαδικασιών.

4.1. Petri-nets

Τα Petri-nets, αποτελούν την παλαιότερη και καλύτερα διευρυμένη γλώσσα μοντελοποίησης. Η γραφική τους σχεδίαση είναι έξυπνη και απλή και επίσης υπάρχουν πολλές εκτελέσιμες τεχνικές που μπορούν να χρησιμοποιηθούν για την ανάλυσή τους. Σε προηγούμενο κεφάλαιο έχουμε παρουσιάσει συνοπτικά ένα Petri-net (Σχήμα 2) όπως επίσης και τον κανόνα πυροδότησης στα οποία θα εμβαθύνουμε και τώρα. Έτσι λοιπόν, ένα Petri-net είναι ένα διμερές γράφημα που αποτελείται από τόπους και μεταβάσεις. Η δομή του δικτύου είναι στατική, αλλά, διέπεται από τον κανόνα πυροδότησης, που επιτρέπει στα tokens να ρέουν μέσα από το δίκτυο. Η κατάσταση ενός Petri-net καθορίζεται από την κατανομή των tokens πάνω στο διάγραμμα και αυτό ονομάζεται ως η *σήμανση (marking)* του. Έτσι,

> **Ορισμός 4.1 (Petri-net):** Ένα *Petri-net* είναι ένα σύνολο $\mathrm{N} = (\mathrm{P}, \mathrm{T}, \mathrm{F})$ όπου, P ένα πεπερασμένο σύνολο *τόπων*, T ένα πεπερασμένο σύνολο μεταβάσεων έτσι ώστε $P \cap T \neq \emptyset$ και $F \subseteq (P \times \cup) \cup (T \times P)$ να είναι ένα σύνολο από κατευθυνόμενα βέλη, που ονομάζεται *σχέση ροής (flow relation). Ένα "μαρκαρισμένο"* Petri-net είναι ένα ζεύγος (N, M) όπου, $N = (P, T, F)$ είναι το Petri-net και $M \in \mathbb{B}(P)$ είναι ένα *πολυσύνολο* του P που υποδηλώνει τη σήμανση του δικτύου. Το σύνολο όλων των "μαρκαρισμένων" Petri-net, συμβολίζεται με $\mathcal{N}$.

Ένα *πολυσύνολο (multi-set)*,είναι ένα σύνολο όπου το κάθε στοιχείο του μπορεί να συμβεί πολλές φορές. Για παράδειγμα το,$[a^2, b, c^3]$ είναι το πολυσύνολο με έξι στοιχεία:

δύο a, ένα b και τρία c. Τα επόμενα πολυσύνολα είναι παρόμοια: $[b, c, c, c, a, a]$ $και$ $[c^3, a, a, b]$ [2]. Το μόνο που έχει σημασία είναι ο αριθμός εμφανίσεως της κάθε αξίας. Πιο επίσημα, $\mathbb{B}(D) = D \to \mathbb{N}$ είναι το σύνολο από πολυσύνολα σε μια πεπερασμένη περιοχή D, δηλαδή, $X \in \mathbb{B}(D)$ is a multi-set, όπου κάθε $d \in D, X(d)$ υποδηλώνει τον αριθμό των φορών όπου το d περιλαμβάνεται στο πολυσύνολο. Για παράδειγμα αν $[a^{11}, b^3, c^8]$, τότε $X(a) = 11$ και $X(d) = 0$.

Το άθροισμα δύο πολυσύνολων $(X \uplus Y)$, η διαφορά$(X \backslash Y)$, η παρουσία ενός στοιχείου$(x \in X)$, καθώς και η έννοια του υποσυνόλου $(X \leq Y)$ ορίζονται με άμεσο τρόπο και έχουν την δυνατότητα να χειρίζονται ένα μίγμα από σύνολα και πολυσύνολα. Για παράδειγμα, $[a^2, b^3, c^6] \uplus [b, c] = [a^2, b^4, c^7]$ και $[a, b] \leq [a, b^3, c]$.

Για να επισημοποιήσουμε τον κανόνα πυροδότησης, θα παρουσιάσουμε τον συμβολισμό για τους *κόμβους εισόδου και εξόδου (input and output nodes).*Έστω, $N = (P, T, F)$ ένα Petri-net. Τα στοιχεία της ένωσης $P \cup T$ ονομάζονται *κόμβοι (nodes)*. Ένας κόμβος x θεωρείται ως ένας *κόμβος εισόδου (input node)* ενός άλλου κόμβου y αν και μόνο αν υπάρχει ένα κατευθυνόμενο βέλος από x το στο y, δηλαδή, $(x, y) \in F$και για κάθε $x \in P \cup T, {}^{N}_{\bullet}x = \{y | (x, y) \in F\}$. Ένας κόμβος x θεωρείται ως ένας *κόμβος εξόδου (output node)* ενός άλλου κόμβου y αν και μόνο $(y, x) \in F$ και για κάθε $x \in P \cup T$, $x^{N}_{\bullet} = \{y | (x, y) \in F\}$. Το υπερσύνολο N μπορεί να παραληφθεί εάν είναι εμφανές από τα περιεχόμενα. Έτσι λοιπόν όπως μπορούμε να παρατηρήσουμε στο Σχήμα 2 υπάρχουν ετικέτες τόσο στους κόμβους όσο και στις μεταβάσεις. Το διάγραμμά μας αποτελείται από 8 τόπους και 7 μεταβάσεις. Ο τόπος C1 έχει έναν κόμβο εισόδου και δύο κόμβους εξόδου, έτσι $\bullet C1 = \{A\}$ $και$ $C1 \bullet = \{AND - split, E\}$. Οι τόποι C2, C3, C4 και C5 έχουν ένα κόμβο εισόδου και ένα κόμβο εξόδου, έτσι $\bullet C2 = \{AND - slpit\}$ $και$ $C2 \bullet = \{B\}$, $\bullet C3 = \{AND - slpit\}$ $και$ $C3 \bullet = \{C\}$, $\bullet C4 = \{B\}$ $και$ $C4 \bullet = \{AND - join\}$, $\bullet C5 = \{C\}$ $και$ $C5 \bullet = \{AND - join\}$. Τέλος, ο τόπος C6 έχει δύο κόμβους εισόδου και έναν τόπο εξόδου, έτσι $\bullet C6 = \{AND - join, E\}$ $και$ $C6 \bullet = \{D\}$.

Όπως έχουμε αναφέρει και προηγουμένως, η μαύρη τελεία που είναι τοποθετημένη στον κόμβο εισόδου του A αντιπροσωπεύει ένα token όπου δηλώνει την αρχική σήμανση. Η δυναμική συμπεριφορά αυτών των tokens ορίζεται από τον κανόνα πυροδότησης.

> **Ορισμός 4.2 (Κανόνας πυροδότησης (Firing rule)):** Έστω $(N = (P, T, F), M)$, $M \in \mathbb{B}(D)$ ένα "μαρκαρισμένο" Petri-net. Η μετάβαση $t \in T$ είναι *ενεργοποιημένη (enabled),* συμβολισμός $(N, M)[t\rangle$, αν και μόνο αν $\bullet t \leq M$. Ο κανόνας πυροδότησης $[\ \rangle \subseteq \mathcal{N} \times T \times \mathcal{N}$ είναι η μικρότερη συσχέτιση που ικανοποιείται για κάθε $t \in T, (N, M)[t\rangle \Rightarrow (N, M)[t\rangle(N, (M \backslash \bullet t) \uplus t \bullet)$.

Το $(N, M)[t\rangle$ υποδηλώνει ότι το t ενεργοποιείται στην σήμανση M, π.χ. στο Petri-net που παρουσιάζεται στο Σχήμα 2 αρχικά έχουμε, $(N, [start])[A\rangle$. Το $(N, M)[\tau\rangle(N, M')$ υποδηλώνει ότι πυροδοτώντας αυτή την ενεργοποιημένη μετάβαση, δίνει σαν αποτέλεσμα

την σήμανση M'. Έτσι λοιπόν για παράδειγμα, $(N,[start])[A\rangle(N,[C1])$, $(N,[C1])[AND-split\rangle(N,[C2,C3])$ και $(N,[C4,C5])[AND-join\rangle(N,[C6])$.

Όπως μπορούμε να καταλάβουμε, τα μοντέλα Petri-nets έχουν μια πολύ αυστηρή θεωρητική βάση και μπορούν να αποτυπώσουν τους συσχετισμούς που παρατηρούνται στα αρχεία καταγραφής συμβάντων αποτελεσματικά. Ακόμα, υπάρχει μια μεγάλη ποικιλία από πολύ ισχυρές τεχνικές ανάλυσης και εργαλείων για την μελέτη και την ανάλυσή τους. Είναι προφανές, ότι αυτό το συνοπτικό μοντέλο παρουσιάζει προβλήματα στην καταγραφή πτυχών που αφορούν περισσότερα δεδομένα όπως για παράδειγμα, δεδομένα που σχετίζονται με χρόνο. Ως εκ τούτου, έχουν προταθεί διάφοροι τύποι υψηλού επιπέδου Petri-nets ώστε να προστεθούν περισσότερες πληροφορίες σχετικές με τα αποθηκευμένα δεδομένα. Τα *Petri-nets με χρώματα (Colored Petri-nets CPNs),* αποτελούν ένα από τα πιο ευρέως χρησιμοποιούμενα δίκτυα τα οποία μπορούν να καταγράψουν πληροφορίες που αφορούν τον χρόνο όπως και άλλου είδους δεδομένα. Τα tokens ουσιαστικά, σε ένα χρωματιστό Petri-net, φέρουν μια τιμή δεδομένων η οποία εκφράζεται ως χρώμα των tokens και περιγράφει τις ιδιότητες του αντικειμένου. Η χρονική σήμανση υποδεικνύει το ελάχιστο χρονικό σημείο κατά το οποίο μπορεί να καταναλωθεί το token και οι μεταβάσεις μπορούν να ορίσουν την καθυστέρηση για την παραγωγή των tokens μετά από την κατανάλωσή τους. Με αυτόν τον τρόπο λοιπόν, οι χρόνοι αναμονής μπορούν να μοντελοποιηθούν.

4.2. Σημειογραφία μοντελοποίησης επιχειρηματικών διαδικασιών (Business process modeling notation BPMN)

Τα BPMN [7] είναι τα αρχικά για Business process modeling notation και αποτελεί μια γλώσσα μοντελοποίησης διαδικασιών. Η BPMN κυκλοφόρησε ως γλώσσα μοντελοποίησης διαδικασιών το 2004 και πρωταρχικός της στόχος είναι να παρέχει έναν συμβολισμό που είναι εύκολα κατανοητός από όλους τους επιχειρηματικούς χρήστες και τους αναλυτές των επιχειρήσεων που δημιουργούν τα αρχικά σχέδια των διαδικασιών των επιχειρήσεων. Μέσω των τεχνικών αυτών, προγραμματιστές που είναι υπεύθυνοι για την εφαρμογή των τεχνολογιών που θα εκτελούν τις διαδικασίες που καταγράφονται, έχουν την δυνατότητα να διαχειρίζονται και να παρακολουθούν τις διαδικασίες αυτές, τους ανθρώπους τους οποίους τις εκτελούν, όπως επίσης έχουν την δυνατότητα να γνωρίζουν άμεσα σε ποιο σημείο βρίσκεται την στιγμή που θέλουν η διαδικασία η οποία μελετούν. Έτσι λοιπόν, η BPMN δημιουργεί μια γέφυρα για το χάσμα μεταξύ του σχεδιασμού των επιχειρηματικών διαδικασιών και της εφαρμογής αυτών.

Η BPMN λοιπόν, καθορίζει ένα *επιχειρηματικό διάγραμμα (Business process diagram)*, το οποίο βασίζεται σε μια τεχνική διαγραμμάτων ροής προσαρμοσμένη για τη

δημιουργία γραφικών μοντέλων που εκφράζουν τις διαδικασίες των επιχειρηματικών δραστηριοτήτων. Ένα τέτοιο μοντέλο, αποτελείται από ένα δίκτυο γραφικών αντικειμένων (δραστηριοτήτων), και συστημάτων ελέγχου ροής που ορίζουν την σειρά της εκτέλεσής τους. Η BPMN αποτελεί μια μέθοδο γνώστη προς τους περισσότερους αναλυτές των επιχειρήσεων και τα σχήματα που χρησιμοποιούνται αναγνωρίζονται από τους περισσότερους μοντελιστές. Για παράδειγμα, τα ορθογώνια αντιπροσωπεύουν τις δραστηριότητες και τα διαμάντια τις αποφάσεις. Θα πρέπει να τονιστεί ότι ένας από τους τρόπους για την ανάπτυξη της BPMN είναι η δημιουργία μοντέλων επιχειρηματικών διαδικασιών και την ίδια στιγμή ο χειρισμός της πολυπλοκότητας που ενυπάρχει στις επιχειρηματικές διαδικασίες. Για να επιτευχθούν αυτές οι δύο αντικρουόμενες απαιτήσεις, οι γραφικές πτυχές έχουν χωριστεί σε συγκεκριμένες κατηγορίες. Με αυτόν τον τρόπο λοιπόν έχουμε ένα μικρό σύνολο κατηγοριών ώστε οι αναγνώστες των μοντέλων να μπορούν εύκολα να αναγνωρίσουν τα βασικά στοιχεία και να κατανοήσουν το διάγραμμα. Εκτός των βασικών αυτών κατηγοριών, μπορούμε να προσθέσουμε επιπλέον πληροφορίες για την υποστήριξη των πολύπλοκων επιχειρηματικών δραστηριοτήτων χωρίς να αλλάζει δραματικά η βασική εμφάνιση και η αίσθηση του αρχικού διαγράμματος. Έτσι οι τέσσερις βασικές κατηγορίες των στοιχείων είναι οι εξής:

- Αντικείμενα Ροής (Flow Objects)
- Αντικείμενα Σύνδεσης (Connecting Objects)
- Πισίνες-διαδρομές (Swimlanes)
- Τα είδωλα (Artifacts)

Αντικείμενα Ροής:

Ένα διάγραμμα επιχειρηματικών διαδικασιών έχει τρία βασικά στοιχεία τα οποία είναι τα αντικείμενα ροής, έτσι οι μοντελιστές να μην χρειάζεται να μάθουν και να αναγνωρίζουν μεγάλο αριθμό διαφορετικών σχημάτων. Τα αντικείμενα ροής είναι:

- **Γεγονότα (Events):** Τα γεγονότα αντιπροσωπεύονται από έναν κύκλο κατά τη διάρκεια της επιχειρηματικής διαδικασίας. Τα γεγονότα αυτά επηρεάζουν τη ροή και συνήθως έχουν μια αιτία ή κάποια επίπτωση. Τα γεγονότα αυτά λοιπόν είναι ανοικτοί κύκλοι που επιτρέπουν την τοποθέτηση δεικτών εσωτερικά ώστε να δείχνουν τα εναύσματα ή τα αποτελέσματα των γεγονότων αυτών. Έχουμε τρείς διαφορετικούς τύπους γεγονότων όπου ο κάθε τύπος εξαρτάται από το που εμφανίζεται το γεγονός στο διάγραμμά μας. Έτσι λοιπόν, έχουμε τα αρχικά, τα μεσαία και τα γεγονότα που κλείνουν τα διαγράμματα.

- **Δραστηριότητες (Activities):** Οι δραστηριότητες αντιπροσωπεύονται από ένα ορθογώνιο και αποτελεί ένα γενικό όρο για μια εργασία που εκτελεί μια επιχείρηση ή κάποια ενέργεια που εκτελεί ένας χρήστης

ή ένας εργαζόμενος. Υπάρχουν δύο είδη δραστηριοτήτων όπου μπορεί να είναι είτε εργασίες (tasks) είτε υπο-διαδικασίες (Sub-processes). Οι υπο-διαδικασίες διακρίνονται από ένα μικρό προσθετικό σύμβολο που φέρουν στο κάτω μέρος του ορθογωνίου.

- **Πύλες (Gateways):** Οι πύλες αντιπροσωπεύονται από το γνωστό σχήμα διαμαντιού και χρησιμοποιούνται για τον έλεγχο της απόκλισης και της σύγκλισης της ροής ακολουθίας. Έτσι οι πύλες καθορίζουν τις παραδοσιακές αποφάσεις, τις διακλαδώσεις, καθώς και την ένταξη των μονοπατιών στο διάγραμμα.

Αντικείμενα Σύνδεσης:

Τα αντικείμενα ροής συνδέονται μεταξύ τους για να δημιουργήσουν τη βασική σκελετική δομή του μοντέλου διεργασιών μας. Υπάρχουν τρία είδη αντικειμένων σύνδεσης που παρέχουν αυτή τη λειτουργία. Αυτές οι συνδέσεις είναι:

- **Ροή ακολουθίας (Sequence Flow):** Μία ροή ακολουθίας αντιπροσωπεύεται από μια συνεχή γραμμή με ένα βέλος στο τέλος της και χρησιμοποιείται για να δείξει τη σειρά με την οποία οι δραστηριότητες θα πρέπει να εκτελούνται σε μια διαδικασία.

- **Ροή μηνυμάτων (Message Flow):** Η ροή μηνυμάτων αντιπροσωπεύεται από μια διακεκομμένη γραμμή με ένα ανοικτό βέλος στο τέλος και χρησιμοποιείται για να δείξει τη ροή των μηνυμάτων μεταξύ δύο διαφορετικών συμμετεχόντων που στέλνουν και λαμβάνουν μηνύματα ώστε να εκτελέσουν κάποια διαδικασία. Για παράδειγμα δυο διαφορετικές πισίνες σε ένα επιχειρηματικό διάγραμμα θα μπορούσαν να αντιπροσωπεύουν τους δυο συμμετέχοντες.

- **Σύνδεσμοι (Association):** Ένας σύνδεσμος αντιπροσωπεύεται από μια διακεκομμένη γραμμή με μια γραμμή βέλους και χρησιμοποιείται για να συνδέσει δεδομένα, κείμενο, και άλλα είδωλα με αντικείμενα ροής. Οι σύνδεσμοι χρησιμοποιούνται για να δείχνουν εισερχόμενες και εξερχόμενες δραστηριότητες στο μοντέλο.

Πισίνες-διαδρομές:

Οι πισίνες-διαδρομές είναι ένας χρήσιμος μηχανισμός που οργανώνει τις δραστηριότητες σε ξεχωριστές οπτικές κατηγορίες προκειμένου να απεικονίζονται οι διαφορετικές λειτουργικές ικανότητες ή ευθύνες. Η BPMN υποστηρίζει πισίνες-διαδρομές με δυο διαφορετικές κατασκευές, τα δυο είδη αυτών των αντικειμένων είναι:

- **Πισίνες (Pools):** Η κάθε πισίνα συνήθως, αντιπροσωπεύει κάθε συμμετέχοντα σε μια διαδικασία. Αποτελεί, επίσης, ένα γραφικό δοχείο για την διαμέριση του συνόλου των δραστηριοτήτων, στο πλαίσιο των B2B καταστάσεων.

- **Διαδρομές (Lanes):** Μια διαδρομή αποτελεί ένα υπο-χώρισμα εντός της πισίνας το οποίο μπορεί να είναι είτε κάθετο ή οριζόντιο. Οι διαδρομές χρησιμοποιούνται για να ταξινομήσουν τις δραστηριότητες μεσα σε μια πισίνα σύμφωνα με τη λειτουργία ή το ρόλο τους.

Είδωλα:

Τα είδωλα μας επιτρέπουν να εκπροσωπήσουμε τα οπτικά αντικείμενα που δεν ανήκουν στην πραγματική διαδικασία. Αντικείμενα μπορούν να αποτελέσουν στοιχεία ή σημειώσεις που περιγράφουν την διαδικασία, ή μπορούν να χρησιμοποιηθούν για την οργάνωση της διαδικασίας.

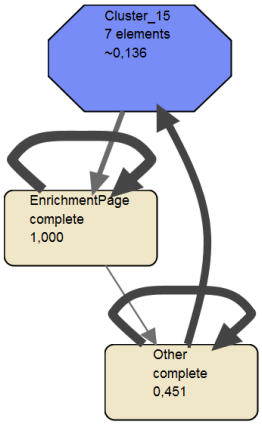
4.3. Ασαφή μοντέλα (Fuzzy Models)

Μπορούμε να φανταστούμε ότι η Εξόρυξη Διαδικασιών μπορεί να εφαρμοστεί σε οποιαδήποτε ”διάσταση” στην οποία υπάρχουν διαδικασίες. Οι διαδικασίες μπορεί να είναι περίπλοκες με πολλαπλές περιπτώσεις και σχεδόν άπειρα μονοπάτια επιλογών. Ο τρόπος με τον οποίο θα απεικονίσουμε τα δεδομένα μας σε μορφή χάρτη θα πρέπει να είναι κατανοητός. Μέσα από αυτούς τους χάρτες άλλωστε εμείς θέλουμε να πάρουμε πληροφορία ακόμα και αν δεν θέλουμε να ακολουθήσουμε αυστηρούς κανόνες απεικόνισης. Τα ασαφή μοντέλα (Fuzzy Models) [8, 9] έρχονται να λύσουν αυτό το πρόβλημα. Τα ασαφή μοντέλα αποτελούν τον καταλληλότερο τρόπο απεικονίσεως περίπλοκων διαδικασιών. Με τον όρο περιπλοκή διαδικασία θέλουμε να πούμε διαδικασίες που έχουν μεγάλο αριθμό κλάσεων ενεργειών. Φανταστείτε ένα δέντρο στο οποίο θα πρέπει να απεικονίσουμε 300 διαφορετικούς τόπους και άλλες τόσες μεταβάσεις στην καλύτερη των περιπτώσεων. Άλλες τόσες μεταβάσεις στην καλύτερη περίπτωση επειδή, αυτό το ίχνος μπορεί να αναπαραχθεί με 300! διαφορετικούς τρόπους. Επίσης, μια διαδικασία μπορεί να μην εκτελεί όλα τα βήματα αλλά να εκτελεί κάποια μέρη αυτών των διαδικασιών. Οπότε τα πιθανά σενάρια αυξάνονται ραγδαία και τώρα έχουμε να αντιμετωπίσουμε κάτι πολύ περισσότερο από 2!3!4!…300! πιθανά σενάρια. Αυτός ο

αριθμός τείνει στο άπειρο, προκύπτουν δηλαδή άπειρα πιθανά σενάρια. Τώρα μπορούμε να αντιληφθούμε τις δυσκολίες αποτύπωσης και κατανόησης περίπλοκων διαδικασιών. Είναι προφανές ότι δεν υπάρχει αλγόριθμος που να μπορεί να αποτυπώσει ένα τέτοιο πρόβλημα άπειρων περιπτώσεων. Όμως το πρόβλημα της αποτύπωσης ενός μοντέλου παραμένει για περίπλοκες διαδικασίες που περιέχουν και 20 διαφορετικές κλάσεις ενεργειών. Πως θα αποτυπώσουμε ένα τέτοιο περίπλοκο μοντέλο και πως μπορούμε να το επεξεργαστούμε για να το φέρουμε σε μια μορφή κατανοητή για τον άνθρωπο;

Για την κατανόηση αυτών των μοντέλων υπάρχει ένας αλγόριθμος που χρησιμοποιεί μια πλατφόρμα για την Εξόρυξη Διαδικασιών το ProM. Ο αλγόριθμος αυτός ονομάζεται Fuzzy miner (ασαφής εξορύκτης) και παράγει ασαφή μοντέλα. Τα ασαφή μοντέλα, δεν ακολουθούν τους αυστηρούς κανόνες που ακολουθούν μοντέλα διαδικασιών όπως είναι τα Petri-nets ή τα BPMN. Στα ασαφή μοντέλα δεν έχουμε τη δυνατότητα να διακρίνουμε αν τα γεγονότα είναι παράλληλα δεν υπάρχουν δηλαδή τόποι και μεταβάσεις όπως έχουν τα Petri-nets ούτε πύλες που έχουν τα BPMN. Τα ασαφή μοντέλα αντί να "μάχονται" για την ακρίβεια, δεχόμαστε ότι σκόπιμα απορρίπτουν κομμάτι πληροφορίας για την οπτικοποίηση διαδικασιών. Επιπλέον, η προσέγγιση αυτή ανέχεται μια υπεραπλούστευση για χάρη της σαφήνειας, δηλαδή, το μοντέλο μπορεί να περιλαμβάνει πληροφορίες που μπορεί να έρχονται σε αντίθεση με το αρχείο καταγραφής συμβάντων προκειμένου να αυξηθεί η απλότητα και η κατανόησή του. Οι στόχοι αυτοί δεν μπορούν να ικανοποιηθούν από τα ήδη παρουσιασμένα μοντέλα όπως τα Petri-nets, τα BPMN μοντέλα. Τέλος, τα ασαφή μοντέλα δεν έχουν την δυνατότητα να μεταφραστούν σε άλλα είδη μοντέλων ενώ τα άλλα είδη μοντέλων μπορούν να μεταφραστούν σε ασαφή μοντέλα. Σε αυτό το κεφάλαιο θα παρουσιάσουμε τα ασαφή μοντέλα όπως και κάποιες από τις δυνατότητες που μας παρέχουν για την μελέτη και την ανάλυση πολύπλοκων διαδικασιών που περιέχουν ένα μεγάλο αριθμό ενεργειών.

Τα ασαφή μοντέλα χρησιμοποιούν μεθόδους από τη χαρτογράφηση για τη αποτύπωση των διαδικασιών. Έτσι τα ασαφή μοντέλα αποτελούνται μόνο από τους τόπους που συνήθως συμβολίζονται με ορθογώνια παραλληλόγραμμα, και από τα βέλη που τους ενώνουν. Το κύριο πλεονέκτημα αυτών των μοντέλων είναι ότι η αναπαράστασή τους βασίζεται στη συχνότητα. Στη συχνότητα όχι μόνο των ενεργειών αλλά και στην συχνότητα όπου κάποια ενέργεια ακολουθεί της άλλης. Έτσι τα βέλη τα οποία αποτελούν τις "λεωφόρους του χάρτη μας", συμβολίζονται με πιο έντονο τρόπο από ότι τα μονοπάτια που χρησιμοποιούνται σε μικρότερο βαθμό. Ο αλγόριθμος Fuzzy miner χρησιμοποιεί και άλλες μεθόδους για να ρίξει την πληροφορία που απεικονίζει το μοντέλο μας και να μας βοηθήσει να κατανοήσουμε το βασικό κομμάτι της διαδικασίας. Με την έννοια βασικό κομμάτι της διαδικασίας εννοούμε το μέρος της διαδικασίας εκείνο που έχει την μεγαλύτερη συχνότητα. Θα μελετήσουμε σε αυτό το σημείο τους τρόπους με τους οποίους ο αλγόριθμος αυτός αποτυπώνει τα δεδομένα και τους τρόπους με τους οποίους ρίχνουμε το επίπεδο της πληροφορίας για την κατανόηση του μοντέλου.



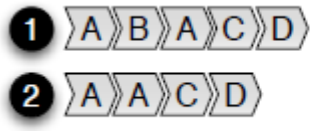
Σχήμα 5: Κομμάτι ασαφούς μοντέλου που παράχθηκε από τον αλγόριθμο Fuzzy-miner.

Στο Σχήμα 4 απεικονίζεται ένα κομμάτι από ένα ασαφές μοντέλο το οποίο παράχθηκε από τον αλγόριθμο Fuzzy miner. Παρατηρούμε ότι υπάρχουν δύο είδη κόμβων, οι τετράγωνοι κίτρινοι κόμβοι και οι οχτάγωνοι μπλε κόμβοι. Οι κίτρινοι κόμβοι αντιπροσωπεύουν τις κλάσεις των γεγονότων και αναγράφεται πάνω τους το όνομα της κλάσης και το *επίπεδο της συχνότητας (frequency significance).* Τα μπλε οκτάγωνα αποτελούν τους *κόμβους των ομαδοποιημένων ενεργειών (cluster nodes)* και πάνω τους αναγράφεται το μέσο επίπεδο της συχνότητας των ομαδοποιημένων ενεργειών. Το επίπεδο συχνότητας αποτελεί ένα μέτρο που συνδέει την συχνότητα εμφάνισης μίας ενέργειας στο αρχείο συμβάντων με τις συχνότητες των υπόλοιπων ενεργειών. Η ενέργεια η οποία εμφανίζεται στο αρχείο τις περισσότερες φορές, δηλαδή αυτή με την μεγαλύτερη συχνότητα παίρνει την τιμή ένα. Όλες οι υπόλοιπες υπολογίζονται βασιζόμενες σε αυτή την ενέργεια με το μεγαλύτερο επίπεδο συχνότητας.

Έστω τα ίχνη που φαίνονται στο Σχήμα 5, $< A, B, A, C, D >$ και $< A, A, C, D >$. Οι αντίστοιχες συχνότητες και τα επίπεδα συχνότητας φαίνονται στον πίνακα στο ίδιο σχήμα. Έτσι, το επίπεδο συχνότητας της ενέργειας A που έχει και την μεγαλύτερη συχνότητα είναι ίσο με ένα. Τα επίπεδα συχνότητας των υπόλοιπων ενεργειών υπολογίζονται από τη διαίρεση της συχνότητας της ενέργειας προς τη μέγιστη συχνότητα των συχνοτήτων των

κλάσεων των ενεργειών. Έτσι λοιπόν για τα γεγονότα C και D που έχουν τις ίδιες συχνότητες τα αντίστοιχα επίπεδα συχνότητας είναι ίσα με $\frac{2}{4} = 0.5$. Όμοια για το B υπολογίζουμε το επίπεδο συχνότητας ίσο με $\frac{1}{4} = 0.25$. Ο αλγόριθμος Fuzzy miner για την ομαδοποίηση των ενεργειών βασίζεται στα επίπεδα συχνοτήτων. Μετά την παραγωγή του μοντέλου ο αλγόριθμος μας δίνει τη δυνατότητα να ρίξουμε το επίπεδο της πληροφορίας με την βοήθεια δύο μπάρων. Η πρώτη μπάρα μας βοηθάει στην ομαδοποίηση των ενεργειών ανάλογα με το επίπεδο συχνότητας τους και το επίπεδο της μπάρας. Έτσι αν ανεβάσουμε την μπάρα ομαδοποιούνται σε ένα κόμβο οι ενέργειες με τα μικρότερα επίπεδα συχνοτήτων. Όσο ανεβάζουμε την μπάρα τόσες περισσότερες ενέργειες ομαδοποιούνται. Η δεύτερη μπάρα μας βοηθάει να περιορίσουμε τον αριθμό των διαφορετικών μονοπατιών που ακολουθεί η διαδικασία μας. Αν ανεβάσουμε την μπάρα

στο μέγιστο δυνατό σημείο δηλαδή, το επίπεδο της πληροφορίας θα μειωθεί αφού στο μοντέλο θα φαίνεται μόνο το πιο συχνό μονοπάτι το οποίο ακολουθείται από το αρχείο μας. Ο αλγόριθμος αυτός έχει πολλές ακόμα δυνατότητες όπως την δυνατότητα *εμψύχωσης (Animate)* του μοντέλου που παρουσιάζονται στο [10].



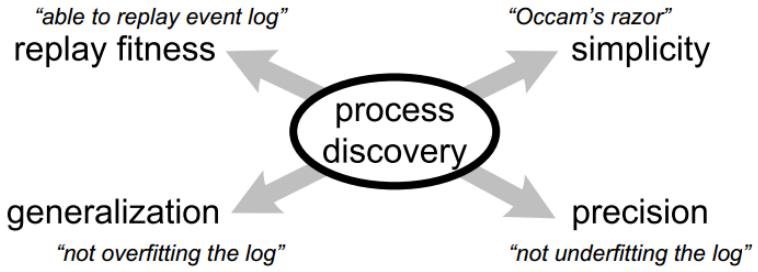
Event class	A	B	C	D
Frequency	4	1	2	2
Significance	1.0	0.25	0.5	0.5

Σχήμα 6: Στα αριστερά φαίνονται τα ίχνη ενός αρχείου δεδομένων και δεξιά οι συχνότητες και τα επίπεδα συχνοτήτων. [10]

Ο αλγόριθμος του ProM Fuzzy miner έχει πολλές δυνατότητες και μας λύνει τα χέρια σε περιπτώσεις που έχουμε να κάνουμε με περίπλοκες διαδικασίες. Χρησιμοποιήσαμε και εμείς τον συγκεκριμένο αλγόριθμο όταν είχαμε να κάνουμε με δεδομένα από μία διαδικτυακή πλατφόρμα που θα παρουσιάσουμε παρακάτω στις εφαρμογές. Στο επόμενο υποκεφάλαιο θα μελετήσουμε τις διαστάσεις ποιότητας των μοντέλων που μας βοηθούν να καταλάβουμε κατά πόσο τα μοντέλα που παράγουμε αναπαριστούν τα ίχνη που έχουμε καταγράψει. Επίσης οι διαστάσεις αυτές μας βοηθούν να ελέγξουμε κατά πόσο οι αλγόριθμοι οι οποίοι χρησιμοποιούμε είναι ακριβείς ή οχι.

4.4. Διαστάσεις ποιότητας των μοντέλων

Υπάρχουν δεκάδες διαφορετικά είδη μοντελοποίησης διαδικασιών και κάθε είδος μπορεί να έχει περισσότερους από έναν αλγόριθμους. Είναι ανάγκη λοιπόν να επιλέξουμε για την αναπαράσταση των δεδομένων μας το καταλληλότερο είδος και επιπλέον τον καταλληλότερο αλγόριθμο. Έτσι για την επίλυση αυτών των δυσκολιών έχουν οριστεί οι διαστάσεις ποιότητας των μοντέλων. Ο προσδιορισμός της ποιότητας των αποτελεσμάτων της Εξόρυξης Διαδικασιών όμως, αποτελεί μια δύσκολη διαδικασία που μπορεί να χαρακτηριστεί από πολλές διαστάσεις. Οι κύριες από αυτές τις διαστάσεις είναι τέσσερις και αυτές είναι [11, 12]: *καταλληλότητα (fitness), απλότητα (simplicity), ακρίβεια (precision) και γενίκευση (generalization)*. Σε αυτό το κεφάλαιο θα αναφερθούμε σε αυτές τις τέσσερεις διαστάσεις και θα παρουσιάσουμε κάποια παραδείγματα που εξηγούν τις διαφορές των διαστάσεων αυτών.



Σχήμα 7: Οι τέσσερεις διαστάσεις ποιότητας που ισορροπούν την ποιότητα του μοντέλου. [2]

Τα τέσσερα αυτά κριτήρια μπορούμε να τα θεωρήσουμε σαν τέσσερεις δυνάμεις που εφαρμόζονται στο μοντέλο μας όπως φαίνεται και στο Σχήμα 6 και ένα μοντέλο θεωρείται καλό όταν υπάρχει μια ισορροπία αυτών των τεσσάρων δυνάμεων έτσι λοιπόν:

Καταλληλότητα (fitness): Η καταλληλότητα που σε πολλές δημοσιεύσεις αναφέρεται και ως επανάληψη καταλληλότητας (replay fitness) [13, 14], εκφράζει κατά πόσο το μοντέλο μας αναπαριστά την συμπεριφορά η οποία είναι καταγεγραμμένη από το αρχείο καταγραφής συμβάντων. Ένα μοντέλο θεωρείται κατάλληλο όταν επιτρέπει και αναπαριστά την συμπεριφορά που είναι καταγεγραμμένη από το αντίστοιχο αρχείο καταγραφής που το αναπαρήγαγε. Ένα μοντέλο έχει την μέγιστη καταλληλότητα αν και μόνο αν έχει την δυνατότητα να αναπαράγει όλα τα ίχνη που υπάρχουν στο αρχείο καταγραφής συμβάντων από την αρχή μέχρι το τέλος του. Για να υπολογίσουμε το επίπεδο καταλληλότητας ενός μοντέλου θα πρέπει να ορίσουμε κάποιες μεταβλητές όπως για παράδειγμα ποιο είναι το κόστος σε περίπτωση που κάποιο ίχνος δεν αναπαριστάται από μοντέλο μας κ.α. Έτσι η καταλληλότητα υπολογίζεται:

$$Q_f = 1 - \frac{Cost\ for\ aligning\ model\ and\ event\ log}{Minimal\ cost\ to\ align\ event\ log\ on\ model\ with\ no\ synchronous\ moves}$$

Σε κάποιες περιπτώσεις, γεγονότα παραλείπονται ή δραστηριότητες εισάγονται χωρίς να υπάρχουν στο αρχείο μας. Ο αριθμητής του κλάσματος ουσιαστικά εκφράζει τα κόστη για την παράλειψη ή την εισαγωγή των δραστηριοτήτων. Ο παρονομαστής εκφράζει το ελάχιστο κόστος τις χείριστης περίπτωσης όταν δηλαδή το μοντέλο μας δεν συμφωνεί σε καμία περίπτωση με το αρχείο. Η τιμή της καταλληλότητας του μοντέλου κινείται από 0 έως το 1 με την καλύτερη πιθανή περίπτωση $Q_f = 1$. Ο υπολογισμός παράλειψης ή εισαγωγής των δραστηριοτήτων αποτελεί μία χρονοβόρα διαδικασία και αυτό γίνεται αντιληπτό από την πρώτη εφαρμογή. Ωστόσο, αποτελεί τον πιο ισχυρό τρόπο για την συσχέτιση μοντέλου διαδικασιών και αρχείου καταγραφής συμβάντων.

Απλότητα (simplicity): Η απλότητα βασικά ποσοτικοποιεί την πολυπλοκότητα του μοντέλου. Η απλότητα υπολογίζεται συγκρίνοντας το μέγεθος του μοντέλου σε σχέση με τον αριθμό των δραστηριοτήτων στο αρχείο καταγραφής. Η απλότητα επίσης αποτελεί έναν από του κύριους παράγοντες για την κατανόηση της πολυπλοκότητας και την

εισαγωγή σφαλμάτων στα μοντέλα. Εάν κάθε δραστηριότητα αναπαριστάται ακριβώς μια φορά στο μοντέλο μας, αυτό το μοντέλο θεωρείται πως είναι το απλούστερο δυνατό. Η απλότητα λοιπόν υπολογίζεται από την παρακάτω παράσταση:

$$Q_s = 1 - \frac{\#duplicate\ activities + \#missing\ activities}{\#nodes\ in\ process\ tree + \#event\ classes\ in\ the\ event\ log}$$

Η επανάληψη των δραστηριοτήτων (duplicate activities) μετράται με την απαρίθμηση του αριθμού των φορών που η δραστηριότητα επαναλαμβάνεται στο μοντέλο μας. Επίσης, μια δραστηριότητα μια δραστηριότητα απουσιάζει (missing activities) από το μοντέλο διεργασίας αν και μόνο αν υπάρχει στο αρχείο καταγραφής και δεν υπάρχει στο μοντέλο. Υπολογίζουμε το άθροισμα αυτών των αριθμών και εν συνεχεία, υπολογίζουμε το πηλίκο του αθροίσματός τους ως προς το άθροισμα του συνολικού αριθμού των κόμβων στο μοντέλο διαδικασίας συν τον συνολικό αριθμό των διαφορετικών κλάσεων των δραστηριοτήτων που εμφανίζονται στο αρχείο καταγραφής συμβάντων. Η τιμή της απλότητας του μοντέλου κινείται από 0 έως 1 με καλύτερη πιθανή περίπτωση $Q_s = 1$.

Ακρίβεια (precision): Η ακρίβεια εκφράζει το ποσό της επιπλέον δραστηριότητας που επιτρέπει το μοντέλο μας σε σχέση με το αρχείο καταγραφής συμβάντων. Το μέτρο αυτό έχει παρουσιαστεί και εμπνευστεί από μια μελέτη που έχει γίνει στο [15] που υπολογίζουμε τα λεγόμενα *διαφεύγοντα άκρα (escaping edges)*, δηλαδή ενέργειες που είναι δυνατές στο μοντέλο μας αλλά δεν έχουν καταγραφεί ποτέ από το αρχείο μας. Αν δεν υπάρχουν διαφεύγοντα άκρα, τότε το μοντέλο μας θεωρείται ακριβές. Έτσι η ακρίβεια υπολογίζεται από την παρακάτω σχέση:

$$Q_p = 1 - \frac{\sum_{visited\ markings} \#visits\ * \frac{\#outgoing\ edges - \#used\ edges}{\#outgoing\ edges}}{\#total\ marking\ visits\ over\ all\ markings}$$

Γενίκευση (generalization): Η διάσταση της γενίκευσης εκτιμά κατά πόσο το μοντέλο διαδικασιών περιγράφει στο σύνολό του το σύστημα και όχι μόνο ότι έχει καταγραφεί από το αρχείο συμβάντων. Αν όλα τα μέρη του μοντέλου χρησιμοποιούνται συχνά, το μοντέλο μας είναι πολύ πιθανό να είναι γενικό. Αν όμως, ορισμένα τμήματα του μοντέλου χρησιμοποιούνται σπάνια είναι πιθανό πως το σύστημα επιτρέπει στην πραγματικότητα περισσότερη συμπεριφορά από όσο χρειάζεται. Για αυτούς τους λόγους, το μέτρο της γενίκευσης βασίζεται στο πόσο συχνά χρησιμοποιούνται μέρη του μοντέλου όταν αναπαράγουμε το αρχείο καταγραφής συμβάντων πάνω του. Έτσι λοιπόν, αν κάποια μέρη του μοντέλου μας επισκέπτονται σπάνια τότε είμαστε βέβαιοι πως η γενίκευση του μοντέλου είναι κακή. Ως εκ τούτου, η γενίκευση υπολογίζεται ως εξής:

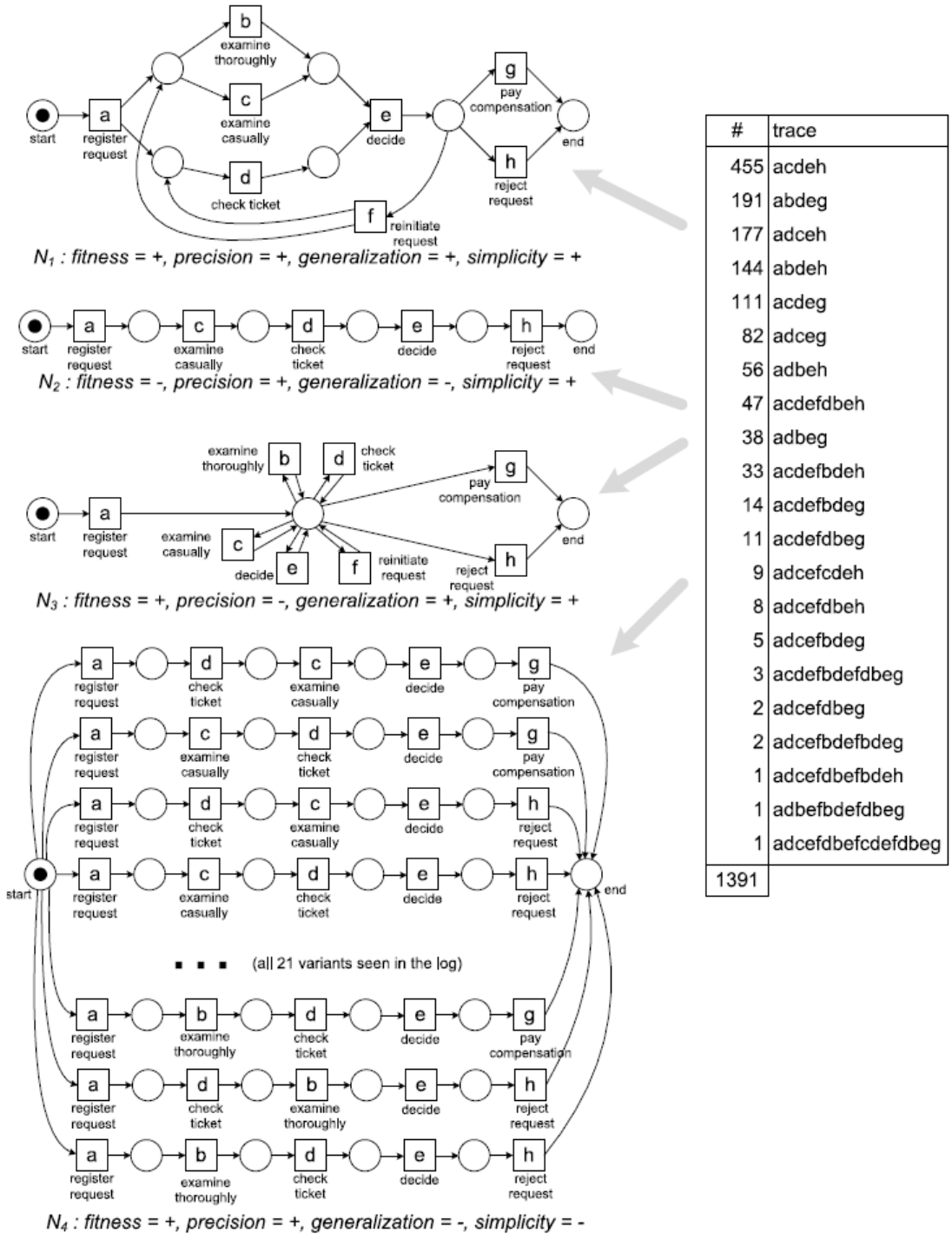
$$Q_g = 1 - \frac{\sum_{nodes} (\sqrt{\#executions})^{-1}}{\#nodes\ in\ model}$$

Η τετραγωνική ρίζα λαμβάνεται από τον αριθμό των εκτελέσεων επειδή η επίδραση της ύπαρξης δέκα εκτελέσεων αντί μίας θεωρείται ως μια πιο αισθητή βελτίωση. Από κάθε μια από τις αξίες η δύναμη του μείον ένα ομαλοποιεί το σύνολο σε μια τιμή από μηδέν μέχρι ένα. Στη συνέχεια οι τιμές αυτές αθροίζονται και διαιρούνται με το συνολικό αριθμό των κόμβων του μοντέλου για να πάρουμε το μέσο όρο για όλο το μοντέλο. Η γενίκευση ορίζεται πιο δύσκολα από όλες τις διαστάσεις ποιότητας των μοντέλων. Όπως και οι υπόλοιπες διαστάσεις ποιότητας που παρουσιάσαμε προηγουμένως οι τιμές της γενίκευσης κινούνται από μηδέν έως ένα με βέλτιστη περίπτωση την $Q_g = 1$.

Ας δούμε λοιπόν ένα παράδειγμα για να κατανοήσουμε καλύτερα τις τέσσερεις διαστάσεις ποιότητας των μοντέλων. Στο Σχήμα 7 παρουσιάζονται 4 μοντέλα διαδικασιών N_1, N_2, N_3 και N_4 και δεξιά τους ένα αρχείο καταγραφής συμβάντων με τα ίχνη του και τις αντίστοιχες συχνότητες. Όλα τα μοντέλα που φαίνονται στο σχήμα έχουν παραχθεί για να εξηγήσουν την συμπεριφορά που έχει καταγραφεί στο συγκεκριμένο αρχείο καταγραφής συμβάντων. Το πρώτο μοντέλο, N_1, θεωρείται το καλύτερο των τεσσάρων. Αυτό συμβαίνει επειδή μπορεί να αναπαραστήσει όλα τα ίχνη τα οποία είναι καταγεγραμμένα στο αρχείο καταγραφής άρα έχει καλό επίπεδο καταλληλόλητας, χωρίς να επιτρέπει μεγάλο επίπεδο επιπλέον δραστηριότητας εκτός από αυτή που υπάρχει στο αρχείο άρα έχει και καλό επίπεδο ακρίβειας. Επιπλέον, το επίπεδο της γενίκευσης είναι καλό επειδή μετά την εκτέλεση της διαδικασίας $e\,(decide)$ έχει την δυνατότητα είτε να συνεχίσει για να καταλήξει στην τερματική μετάβαση ή να εκτελέσει την διαδικασία $f\ (reinitiate\ request)$ που αποτελεί ένας βρόγχος επανάληψης του μοντέλου και να ξαναεκτελέσει κάποιες από τις διαδικασίες. Να σημειωθεί σε αυτό το σημείο ότι ο βρόγχος επανάληψης του μοντέλου μπορεί να εκτελεστεί όσες φορές επιθυμούμε. Τέλος αποτελεί ένα απλό μοντέλο οπότε το επίπεδο της απλότητας είναι καλό. Κάποιοι από τους αλγόριθμους για την παραγωγή μοντέλων όμως δεν θα μας δώσουν πάντοτε το επιθυμητό και το καλύτερο αποτέλεσμα. Είναι πιθανόν κάποιος αλγόριθμος να μας δώσει το μοντέλο N_2. Το N_2 ουσιαστικά αναπαριστά την πρώτη γραμμή του αρχείου καταγραφής συμβάντων και παράγεται επειδή ο αλγόριθμος που επιλέξαμε παράγει εκείνο το ίχνος του αρχείου μας που έχει την μεγαλύτερη συχνότητα. Το μοντέλο αυτό έχει πολύ κακό επίπεδο καταλληλότητας μίας και υπάρχουν πολλά διαφορετικά ίχνη τα οποία δεν μπορούμε να τα αναπαραστήσουμε στο συγκεκριμένο μοντέλο. Από την άλλη, το επίπεδο της ακρίβειας είναι πολύ καλό επειδή αναπαριστά ακριβώς το ίχνος το όποιο του εισάγαμε. Από τα τέσσερα μοντέλα, το μοντέλο N_2 είναι το χειρότερο σε επίπεδο γενίκευσης και καταλληλότητας αφού σε καμία περίπτωση δεν έχει την δυνατότητα να αναπαραστήσει και να γενικεύσει σε καμία άλλη δραστηριότητα εκτός του ενός ίχνους. Τέλος, σε επίπεδο απλότητας το μοντέλο αυτό είναι καλό. Στη συνέχεια έχουμε το μοντέλο N_3 το συγκεκριμένο μοντέλο ονομάζεται *μοντέλο λουλούδι (flower model).* Ας αναλύσουμε λίγο την συμπεριφορά του μοντέλου αυτού. Αρχικά, το μοντέλο εκτελεί πάντα πρώτα την ενέργεια $a\,(register\ request)$. Μετά, το token μας μετακινείται στη μεσαία μετάβαση που μοιάζει με λουλούδι. Από αυτό το σημείο έχουμε την δυνατότητα να εκτελέσουμε οποιαδήποτε ενέργεια επιστρέφοντας στη

μεσαία μετάβαση που βρισκόμασταν προηγουμένως. Αυτό το μοντέλο μπορεί να εκτελέσει πολλά και διαφορετικά ίχνη και είναι το πιο γενικό από όλα τα μοντέλα και ταυτόχρονα δεν είναι ακριβές. Επίσης, το επίπεδο καταλληλότητας είναι πολύ καλό αφού μπορεί να αναπαράγει όλα τα ίχνη που υπάρχουν στο αρχείο συμβάντων. Τέλος, είναι ένα απλό μοντέλο οπότε το επίπεδο απλότητας είναι καλό.Τέλος το μοντέλο N_4, έχει παραχθεί από έναν αλγόριθμο που ουσιαστικά παράγει παράλληλα το κάθε ίχνος το οποίο έχει καταγραφεί. Το πρόβλημα αυτού του μοντέλου είναι πως δεν μπορούμε να το γενικεύσουμε αφού αν προσθέσουμε ένα νέο ίχνος θα πρέπει να προσθέσουμε ένα νέο μονοπάτι στο μοντέλο μας. Εκτός αυτού, το μοντέλο αυτό δεν αποτελεί ένα απλό μοντέλο τα επίπεδα απλότητας δηλαδή είναι πολύ χαμηλά.

Είναι αναγκαίο λοιπόν να χρησιμοποιούμε αλγόριθμους οι οποίοι καλύπτουν σε μεγάλο ποσοστό τις τέσσερεις διαστάσεις ποιότητας των μοντέλων. Η ποικιλία των αλγόριθμων για την διαδικασία της ανακάλυψης της Εξόρυξης Διαδικασιών είναι μεγάλη και μπορούμε να ανακαλύψουμε με αυτούς πολλά διαφορετικά είδη μοντέλων όπως τα παρουσιάσαμε και σε προηγούμενο κεφάλαιο. Στο επόμενο κεφάλαιο θα έχουμε την ευκαιρία να παρουσιάσουμε έναν από αυτούς, τον αλγόριθμο άλφα. Ο αλγόριθμος άλφα, δημιουργήθηκε πριν από έντεκα χρόνια και είναι ένας από τους πρώτους αλγόριθμους που παράχθηκε για την διαδικασία της ανακάλυψης. Θα δούμε λοιπόν τον αλγόριθμο, το πως αυτός λειτουργεί και κάποιους από τους περιορισμούς του αλγόριθμου αυτού.



#	trace
455	acdeh
191	abdeg
177	adceh
144	abdeh
111	acdeg
82	adceg
56	adbeh
47	acdefdbeh
38	adbeg
33	acdefbdeh
14	acdefbdeg
11	acdefdbeg
9	adcefcdeh
8	adcefdbeh
5	adcefbdeg
3	acdefbdefdbeg
2	adcefdbeg
2	adcefbdefbdeg
1	adcefdbefbdeh
1	adbefbdefdbeg
1	adcefdbefcdefdbeg
1391	

Σχήμα 8: Τα διαφορετικά μοντέλα N_1, N_2, N_3 και N_4 και το αρχείο καταγραφής συμβάντων από το οποίο έχουν παραχθεί. [2]

5. Εξόρυξη Διαδικασιών, Ανακάλυψη

Η ανακάλυψη των διαδικασιών αποτελεί το πιο δύσκολο κομμάτι της Εξόρυξης διαδικασιών. Κατασκευάζουμε ένα μοντέλο διαδικασιών βασιζόμενοι σε ένα αρχείο καταγραφής συμβάντων. Σε αυτό το κεφάλαιο θα παρουσιάσουμε τη διαδικασία της ανακάλυψης μοντέλων με την βοήθεια ενός από τους αλγορίθμους που χρησιμοποιούνται για αυτή τη διαδικασία, τον *αλγόριθμο άλφα(alpha algorithm)* που παράγει Petri-nets. Ο αλγόριθμος άλφα θεωρείται απλοϊκός αλλά περιγράφει με ωραίο τρόπο τις γενικές ιδέες που χρησιμοποιούν πολλοί από τους αλγόριθμους Εξόρυξης Διαδικασιών και μας βοηθάει να καταλάβουμε την έννοια της ανακάλυψης των διαδικασιών.

Πριν την παρουσίαση του αλγόριθμου άλφα, θα πρέπει να κάνουμε μια εισαγωγή παρουσιάζοντας τις *σχέσεις ταξινόμησης βάσει αρχείου (log-based ordering relations).* Όπως θα δούμε αργότερα, ο αλγόριθμος άλφα σαρώνει το αρχείο καταγραφής συμβάντων για συγκεκριμένα πρότυπα και καταφέρνει να δημιουργήσει ένα Petri-net βασιζόμενος σε αυτά τα μοτίβα. Για παράδειγμα, αν μια δραστηριότητα a ακολουθείται από μια δραστηριότητα b και η δραστηριότητα b δεν ακολουθείται πότε από την δραστηριότητα a, τότε θεωρείται ότι υπάρχει μια *συναφής συσχέτιση (causal dependency)* μεταξύ a και b. Για να εκφράσουμε αυτή τη συσχέτιση, το αντίστοιχο Petri-net θα πρέπει να έχει ένα τόπο ο οποίος ενώνει το a με το b. Διακρίνουμε τέσσερις σχέσεις ταξινόμησης βάση αρχείου οι οποίες στοχεύουν στην καταγραφή σχετικών προτύπων σε ένα αρχείο καταγραφής.

Ορισμός 5.1 (Log-based ordering relations): Έστω L ένα αρχείο καταγραφής συμβάντων στο $\mathcal{A}$, δηλαδή $L \in \mathbb{B}(\mathcal{A}^*)$. Έστω $a, b \in \mathcal{A}$:

- $a >_L b$ αν και μόνο αν υπάρχει ένα ίχνος $\sigma = \langle t_1, t_2, t_3, \dots, t_n \rangle$ και $i \in \{1, \dots, n-1\}$ τέτοιο ώστε $\sigma \in L$ και $t_i = a$ και $t_{i+1} = b$.
- $a \rightarrow_L b$ αν και μόνο αν $a >_L b$ και $b \not> _L a$.
- $a\#_L$ αν και μόνο αν $a \not> _L b$ και $b \not> _L a$.
- $a \parallel_L b$ αν και μόνο αν $a >_L b$ και $b >_L a$.

Ας υποθέσουμε, για παράδειγμα, πως έχουμε το ακόλουθο αρχείο καταγραφής συμβάντων $L = [\langle a, b, c, d \rangle, \langle a, c, b, d \rangle, \langle a, e, d \rangle]$. Για αυτό το αρχείο καταγραφής συμβάντων οι σχέσεις ταξινόμησης βάση αρχείου που μπορούν να βρεθούν είναι οι ακόλουθες:

$> L = \{(a, b), (a, c), (a, e), (b, c), (b, d), (c, b), (c, d), (e, d)\}$

$\rightarrow L = \{(a, b), (a, c), (a, e), (b, d), (c, d), (e, d)\}$

$\#L = \{(a, a), (a, d), (b, b), (b, e), (c, c), (c, e), (d, a), (d, d), (e, e), (b, e), (e, c)\}$

$\parallel L = \{(b, c), (c, b)\}$

Για κάθε αρχείο καταγραφής συμβάντων L στο $\mathcal{A}$ και $x, y \in \mathcal{A}: x \rightarrow_L y, y \rightarrow_L x, x \#_L y, or\ x \|_L y,$ μπορούμε να παρατηρήσουμε πως ακριβώς μία από αυτές τις σχέσεις ισχύει για κάθε ζεύγος των δραστηριοτήτων. Ως εκ τούτου, το αποτύπωμα (footprint) ενός αρχείου καταγραφής συμβάντων μπορεί να οργανωθεί σε μια μήτρα όπως φαίνεται στον Πίνακα 2.

	a	b	c	d	e
a	#	→	→	#	→
b	←	#	‖	→	#
c	←	‖	#	→	#
d	#	←	←	#	←
e	←	#	#	→	#

Πίνακας 2: Αποτύπωμα του αρχείου συμβάντων $L = [\langle a, b, c, d\rangle, \langle a, c, b, d\rangle, \langle a, e, d\rangle]$. *[2]*

Τα αποτυπώματα μας παρέχουν τη δυνατότητα να καταλαβαίνουμε τις σχέσεις που έχουν οι ενέργειες που εμφανίζονται στα ίχνη ενός αρχείου καταγραφής συμβάντων χωρίς να χρειάζεται να μελετήσουμε όλο το αρχείο. Ο αλγόριθμος άλφα επίσης βασίζεται στις σχέσεις που εμφανίζονται μεταξύ των δραστηριοτήτων για να κατασκευάσει τα μοντέλα που περιγράφουν τις διαδικασίες στο σύνολό τους.

5.1. Αλγόριθμος άλφα

Ορισμός 5.2 (αλγόριθμος άλφα (α-algorithm)): Έστω L ένα αρχείο καταγραφής συμβάντων και $T \subseteq \mathcal{A}$. Το $\alpha(L)$ ορίζεται ως εξής:

1) $\mathrm{T_L} = \{\mathrm{t} \in \mathrm{T} \mid \exists_{\sigma \in \mathrm{L}} \mathrm{t} \in \sigma\}$
2) $T_I = \{t \in T \mid \exists_{\sigma \in \mathrm{L}}\ \mathrm{t} = \mathrm{first}(\sigma)\}$
3) $T_O = \{t \in T \mid \exists_{\sigma \in \mathrm{L}}\ \mathrm{t} = last(\sigma)\}$
4) $X_L = \{(A, B) \mid A \subseteq T_L \wedge A \neq \emptyset \wedge B \subseteq T_L \wedge B \neq \emptyset \wedge \forall_{\alpha \in A} \forall_{b \in B}\ \alpha \rightarrow L\ b \wedge \forall_{\alpha 1, \alpha 2 \in A}\ \alpha 1\ \#_L\ a2 \wedge \forall_{b1, b2 \in B}\ b1\ \#_L\ b2\}$
5) $Y_L = \{(A, B) \in X_L \mid \forall_{(A', B') \in X_L} A \subseteq A' \wedge B \subseteq B' \Rightarrow (A, B) = (A', B')\}$
6) $P_L = \{p_{(A,B)} \mid (A, B) \in Y_L\} \cup \{i_L, o_L\}$
7) $F_L = \{(\alpha, p_{(A,B)}) \mid (A, B) \in Y_L \wedge \alpha \in A\} \cup \{(p_{(A,B)}, b) \mid (A, B) \in Y_L \wedge b \in B\} \cup \{(i_L, t) \mid t \in T_I\} \cup \{(t, o_L) \mid t \in T_O\}$
8) $a(L) = (P_L, T_L, F_L)$

Έστω L ένα αρχείο καταγραφής συμβάντων από ένα σύνολο T, όπου T το σύνολο των δραστηριοτήτων. Στο πρώτο βήμα του αλγόριθμου, ελέγχουμε ποιες δραστηριότητες εμφανίζονται στο αρχείο καταγραφής συμβάντων (T_L). Αυτές οι δραστηριότητες θα αντιστοιχούν στις μεταβάσεις του μοντέλου που θα παραχθεί. T_I είναι το σύνολο των αρχικών δραστηριοτήτων και T_O είναι το σύνολο των τελικών δραστηριοτήτων. Οι δραστηριότητες δηλαδή, που εμφανίζονται είτε πρώτες είτε τελευταίες στο αρχείο καταγραφής συμβάντων. Τα βήματα τέσσερα και πέντε αποτελούν τον πυρήνα του αλγόριθμου άλφα. Η πρόκληση είναι να καθοριστούν οι τόποι του μοντέλου μας όπως και οι συνδέσεις τους. Στόχος μας είναι να κατασκευάσουμε τους τόπους που τους συμβολίζουμε με $p_{(A,B)}$ έτσι ώστε το Α να είναι το σύνολο των μεταβάσεων εισόδου ($\bullet p_{(A,B)} = A$) και το Β να είναι το σύνολο των μεταβάσεων εξόδου ($p_{(A,B)} \bullet = B$) των $p_{(A,B)}$. Για να είμαστε και πιο συγκεκριμένοι, στο βήμα τέσσερα προσπαθούμε να βρούμε τα ζεύγη (A, B) των συνόλων των δραστηριοτήτων τέτοια ώστε κάθε στοιχείο $a \in A$ και κάθε στοιχείο $b \in B$ να έχουν μια συναφή εξάρτηση (δηλαδή $a \rightarrow_L b$) και όλα τα στοιχεία του A να είναι ανεξάρτητα ($a_1 \#_L a_2$) όπως και όλα τα στοιχεία του B να είναι ανεξάρτητα ($b_1 \#_L b_2$). Στο πέμπτο βήμα, κατασκευάζουμε ένα σύνολο στοιχείων Y_L το οποίο διατηρεί μόνο τα μέγιστα στοιχεία του συνόλου X_L. Στο βήμα έξι, το P_L είναι ένα σύνολο από τόπους, και έτσι κρατάμε το σύνολο των τόπων που βρήκαμε στο βήμα πέντε και προσθέτουμε όλες τις μεταβάσεις εισόδου και εξόδου. Στο βήμα επτά τέλος, προσθέτουμε τα βέλη, καθορίζουμε κυρίως τη σχέση ροής ενώνοντας κάθε τόπο $p_{(A,B)}$ με κάθε στοιχείο του $a \in A$, *πηγαίες μεταβάσεις (source transitions)*, με κάθε στοιχείο του $b \in B$, *μεταβάσεις στόχοι (target transitions).* Επιπλέον, ο αλγόριθμος άλφα παράγει ένα βέλος από κάθε πηγαίο τόπο (source place) i_L σε κάθε αρχική μετάβαση $t \in T_i$ και ένα βέλος από κάθε τελική μετάβαση $t \in T_O$ στον τόπο O_L. Το βήμα οκτώ είναι το αποτέλεσμα όπου είναι ένα Petri-net $a(L)$, και P_L είναι οι τόποι, T_L οι μεταβάσεις και F_L τα βέλη, που περιγράφει την δραστηριότητα που είναι καταγεγραμμένη στο αντίστοιχο αρχείο καταγραφής συμβάντων.

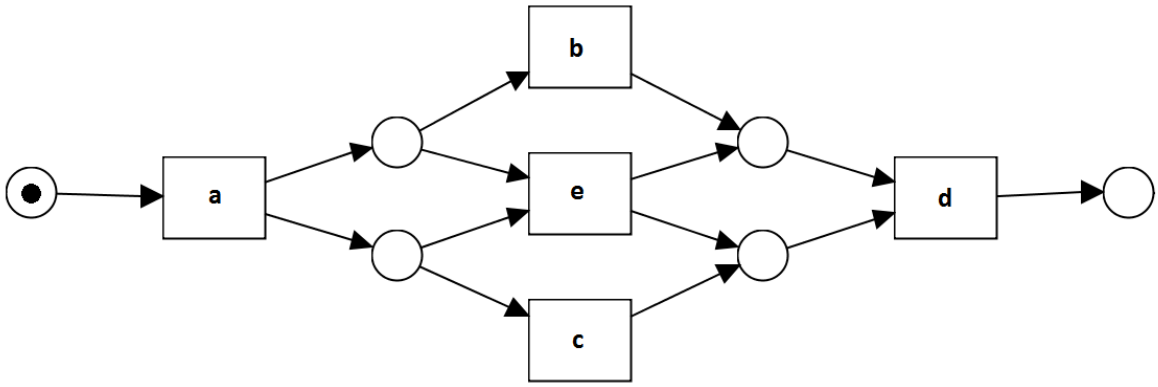
5.2. Παράδειγμα, αλγορίθμου άλφα.

Θεωρούμε το αρχείο καταγραφής συμβάντων από το προηγούμενο κεφάλαιο $L = [\langle a, b, c, d\rangle, \langle a, c, b, d\rangle, \langle a, e, d\rangle]$. Είναι προφανές ότι, $A = \{a\}$ και $B = \{b, e\}$, επειδή το a είναι άμεσα συσχετισμένο με τα b, c και e, όμως το a και το c δεν είναι ανεξάρτητα (βήμα τέσσερα). Ακόμα, το $A' = \{a\}$ και το $B' = \{b\}$ πληρούν τις ίδιες απαιτήσεις που μόλις αναφέρθηκαν. Το X_L αποτελεί το σύνολο όλων των ζευγών που πληρούν τις απαιτήσεις που μόλις αναφέραμε. Έτσι:

$X_L = \{(\{a\},\{b\}),\ (\{a\},\{c\}),\ (\{a\},\{e\}),\ (\{a\},\{b,e\}),\ (\{a\},\{c,e\}),\ (\{b\},\{d\}),\ (\{c\},\{d\}),$
$(\{e\},\{d\}),\ (\{b,e\},\{d\}),\ (\{c,e\},\{d\})\}$

Στο επόμενο βήμα εισάγουμε έναν τόπο για όλα τα "μέγιστα ζεύγη" του συνόλου X_L, να σημειωθεί πως για κάθε ζεύγος $(A, B) \in X_L$, ένα μη κενό σύνολο $A' \subseteq A$, και ένα μη κενό σύνολο $B' \subseteq B$, υπονοείται ότι $(A', B') \in X_L$. Στο βήμα πέντε όλα τα μη-μέγιστα ζεύγη αφαιρούνται και έτσι έχουμε:

$Y_L = \{(\{a\}, \{b, e\}), (\{a\}, \{c, e\}), (\{b, e\}, \{d\}), (\{c, e\}, \{d\})\}$



Σχήμα 9: Petri-net που παράχθηκε από το $L = [\langle a, b, c, d\rangle, \langle a, c, b, d\rangle, \langle a, e, d\rangle]$ με την βοήθεια του αλγόριθμου άλφα.

Κάθε στοιχείο του $(A, B) \in Y_L$ αντιστοιχεί σε ένα τόπο $p_{(A,B)}$ ενώνοντας τις μεταβάσεις A στις μεταβάσεις B. Ακόμα, P_L επίσης περιλαμβάνει ένα μοναδικό τόπο εισόδου i_L και ένα μοναδικό τόπο place o_L (βήμα 6). Στο βήμα επτά, τα βέλη του μοντέλου μας δημιουργούνται. Όλες οι αρχικές μεταβάσεις του T_I έχουν το i_L σαν ένα τόπο εισόδου και όλες οι τελικές μεταβάσεις T_O έχουν το o_L σαν τόπο εξόδου. Όλοι οι τόποι $p_{(A,B)}$ έχουν το A ως κόμβο εισόδου και το B σαν κόμβο εξόδου. Το αποτέλεσμα είναι ένα Petri-net $a(L) = (P_L, T_L, F_L)$ (Σχήμα 8) το οποίο εκφράζει την συμπεριφορά που είναι καταγεγραμμένη στο αρχείο καταγραφής συμβάντων L.

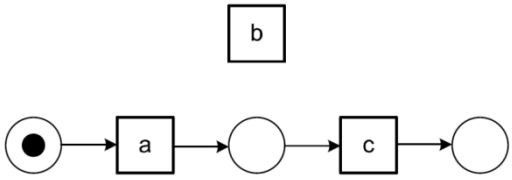
5.3. Περιορισμοί του αλγόριθμου άλφα

Αν και ο αλγόριθμος άλφα έχει τη δυνατότητα να ανακαλύπτει μοντέλα από μεγάλα αρχεία καταγραφής συμβάντων, ο αλγόριθμος αυτός κρύβει και πολλά μειονεκτήματα. Υπάρχουν πολλά διαφορετικά μοντέλα τα οποία έχουν ακριβώς την ίδια συμπεριφορά, δηλαδή δυο μοντέλα μπορεί να έχουν διαφορετική δομή αλλά στην πραγματικότητα να αναπαριστούν το ίδιο ίχνος. Αυτό συμβαίνει επειδή υπάρχουν διάφορα είδη αλγορίθμων που χρησιμοποιούν τις ίδιες γλώσσες μοντελοποίησης. Έτσι μπορεί να παραχθούν δύο

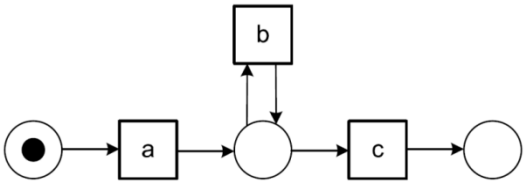
διαφορετικά Petri-nets, για παράδειγμα ένα χρησιμοποιώντας τον αλγόριθμο άλφα και ένα χρησιμοποιώντας τον αλγόριθμο ILP, από το ίδιο αρχείο καταγραφής συμβάντων. Είναι πιθανό δηλαδή ένα από τα μοντέλα που θα παραχθούν να είναι άσκοπα περίπλοκο με αποτέλεσμα να αναπαριστά περισσότερη συμπεριφορά χωρίς αυτό να είναι αναγκαίο. Αυτό συμβαίνει επειδή από ένα αρχείο καταγραφής συμβάντων μπορούν να παραχθούν πολλά και διαφορετικά μοντέλα, ανάλογα με τους αλγόριθμους που χρησιμοποιούμε, που να εξηγούν την συμπεριφορά του. Οι τέσσερεις διαστάσεις ποιότητας μας βοηθούν να καταλάβουμε ποιο από αυτά τα μοντέλα είναι το καταλληλότερο και ακριβέστερο για την κάθε περίπτωση.

Ο αυθεντικός άλφα αλγόριθμος, όπως παρουσιάστηκε στο προηγούμενο κεφάλαιο, εμφανίζει προβλήματα όταν έχει να απεικονίσει μοντέλα τα οποία περιέχουν μικρούς βρόγχους, δηλαδή βρόγχους μήκους ένα ή δύο. Όσων αφορά τον βρόγχο μήκους ένα, το αποτέλεσμα του άλφα αλγόριθμου για το αρχείο καταγραφής συμβάντων L_1 παρουσιάζεται στο Σχήμα 10.

$$L_1 = [\langle a, c\rangle, \langle a, b, c\rangle, \langle a, b, b, c\rangle, \langle a, b, b, b, b, c\rangle]$$



Σχήμα 11: Αποτέλεσμα αλγόριθμου άλφα για το L_1, βρόγχος μήκους 1. [2]

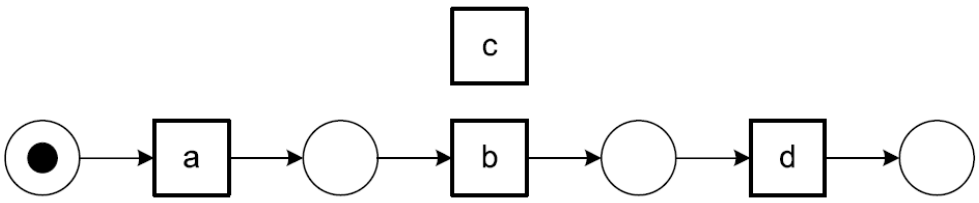


Σχήμα 10: Αποτέλεσμα εξελιγμένου αλγόριθμου άλφα για το L_1. [2]

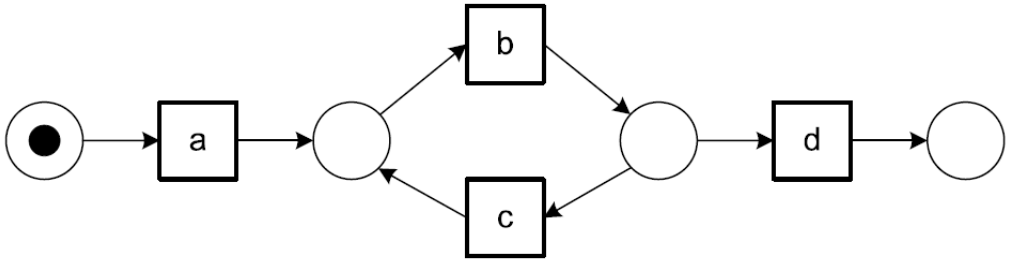
Το μοντέλο δεν αποτελεί ένα έγκυρο Petri-net μιας και η μετάβαση b είναι αποκομμένη από το σύνολο των μεταβάσεων και των τόπων. Το συγκεκριμένο μοντέλο επιτρέπει την εκτέλεση του b πριν το a και μετά το c, που δεν συμβαδίζει με το αρχείο καταγραφής συμβάντων. Αυτό το πρόβλημα μπορεί να αντιμετωπιστεί εύκολα όπως παρουσιάζεται και στο [14] με την χρήση του εξελιγμένου αλγόριθμου άλφα α^+, που παράγει το μοντέλο που παρουσιάζεται στο Σχήμα 11. Αξίζει να σημειωθεί πως οι μικροί βρόγχοι πρέπει να φαίνονται στο μοντέλο μιας και η ύπαρξή τους μας ενημερώνει πως υπάρχει μια αντιληπτικότητα στην διαδικασία μας και συνεπώς κάποια καθυστέρηση.

Η δυσκολία με τους βρόγχους μήκους δυο αναπαριστάται από το Petri-net που φαίνεται στο Σχήμα 12 που αποτελεί ένα μοντέλο το οποίο παράχθηκε από το αρχείο καταγραφής συμβάντων L_2.

$$L_2 = [\langle a, b, d\rangle, \langle a, b, c, b, d\rangle, \langle a, b, c, b, c, b, d\rangle]$$



Σχήμα 12: Αποτέλεσμα αλγόριθμου άλφα για το L_2, βρόγχος μήκους 2. [2]



Σχήμα 13: Αποτέλεσμα εξελιγμένου αλγόριθμου άλφα για το L_2. [2]

Από το συγκεκριμένο αρχείο καταγραφής συμβάντων δημιουργούνται οι ακόλουθες log-based ordering relations: $a \rightarrow_{L_2} b, b \rightarrow_{L_2} d$, και $b \parallel_{L_2} c$. Έτσι λοιπόν, ο αλγόριθμος υποθέτει λανθασμένα πως τα b και c είναι παράλληλα επειδή ακολουθούν το ένα το άλλο. Το μοντέλο αυτό δεν αποτελεί ένα έγκυρο Petri-net μιας και η μετάβαση c είναι αποκομμένη από το σύνολο των μεταβάσεων και των τόπων και η πορεία η οποία είναι καταγεγραμμένη στο αρχείο καταγραφής συμβάντων δεν ακολουθείται από το μοντέλο. Χρησιμοποιώντας τον βελτιωμένο αλγόριθμο που παρουσιάζεται στο [14], ο βελτιωμένος αλγόριθμος άλφα παράγει το Petri-net που φαίνεται στο Σχήμα 13.

Υπάρχουν πολλοί τρόποι για να βελτιώσουμε τον βασικό άλφα αλγόριθμο ώστε να μπορεί να αντιμετωπίσει βρόγχους επανάληψης. Η εξελιγμένη μορφή του που περιγράφεται στο [14] είναι μια από τις εναλλακτικές λύσεις όπου ο συγκεκριμένος αλγόριθμος αποτελείται από δύο φάσεις, την φάση επεξεργασίας και την φάση μετά την επεξεργασία. Η φάση επεξεργασίας, αντιμετωπίζει τους βρόγχους μήκους δύο ενώ η φάση μετά την επεξεργασία εισάγει βρόγχους μήκους ένα.

Ο βασικός αλγόριθμος, δεν έχει κανένα πρόβλημα με την εξόρυξη βρόγχων μήκους τρία ή μεγαλύτερων. Ένας ακόμα περιορισμός του αλγόριθμου άλφα είναι ότι δεν

λαμβάνονται υπόψη οι συχνότητες. Ως εκ τούτου, ο αλγόριθμος είναι πολύ ευαίσθητος στο θόρυβο και τις ατέλειες. Ο αλγόριθμος άλφα, είναι σε θέση να ανακαλύψει μια μεγάλη κατηγορία μοντέλων, όμως ο οχτάγραμμος αλγόριθμος έχει κάποιους περιορισμούς όταν πρόκειται για συγκεκριμένα μοτίβα διαδικασιών. Όπως φαίνεται στα [6,14] ο αλγόριθμος άλφα εγγυάται για την παραγωγή ενός σωστού μοντέλου διαδικασιών με την προϋπόθεση ότι η υποκείμενη διαδικασία δεν περιέχει διπλότυπες διαδικασίες (δυο μεταβάσεις με τις ίδιες ετικέτες) και αθόρυβες μεταβάσεις (δραστηριότητες που δεν καταγράφονται στο αρχείο καταγραφής συμβάντων).

6. Εξόρυξη Διαδικασιών, έλεγχος συσχέτισης

Μετά την σε βάθος ανάλυση της διαδικασίας ανακάλυψης που είδαμε στο προηγούμενο κεφάλαιο, σειρά έχει ο έλεγχος συσχέτισης [13]. Το κεφάλαιο αυτό, εξετάζει την κατάσταση στην οποία τόσο το μοντέλο διαδικασίας και ένα αρχείο καταγραφής συμβάντων θεωρούνται δεδομένα. Το μοντέλο μπορεί να έχει κατασκευαστεί με το χέρι ή μπορεί να έχει ανακαλυφθεί με την χρήση κάποιου αλγόριθμου, όπως του αλγόριθμου άλφα. Ο έλεγχος συσχέτισης αποτελεί ουσιαστικά ένα τρόπο για να ελέγξουμε κατά πόσο το μοντέλο μας αποκλίνει από το αρχείο καταγραφής συμβάντων και το αντίστροφο. Ο απώτερος στόχος είναι να βρεθούν οι ομοιότητες και οι διαφορές μεταξύ της μοντελοποιημένης και της καταγεγραμμένης συμπεριφοράς. Το αρχείο καταγραφής συμβάντων μπορεί να αναπαραχθεί πάνω στο μοντέλο διαδικασιών για την εξεύρεση ανεπιθύμητων αποκλίσεων που υποδηλώνουν ανεπάρκεια ή αναποτελεσματικότητα. Επίσης, οι τεχνικές ελέγχου συσχέτισης μπορούν να χρησιμοποιηθούν για την μέτρηση της απόδοσης των αλγορίθμων που χρησιμοποιούνται για την εξεύρεση μοντέλων διαδικασιών και για την διόρθωση μοντέλων που δεν απεικονίζουν την πραγματικότητα.

Σε προηγούμενο κεφάλαιο παρουσιάσαμε τις τέσσερις διαστάσεις ποιότητας των μοντέλων, την καταλληλότητα, την απλότητα, την ακρίβεια και τη γενίκευση. Χρησιμοποιήσαμε τέσσερα διαφορετικά μοντέλα (Σχήμα 8), για την καλύτερη κατανόηση των διαστάσεων ποιότητας. Για κάθε ένα από αυτά τα μοντέλα δώσαμε μία υποκειμενική εξήγηση χωρίς να παρουσιάσουμε αριθμητικές μεθόδους για κάθε ένα από τα μοντέλα N_1, N_2, N_3, N_4. Η επιλογή των συγκεκριμένων μοντέλων έγινε εσκεμμένα μιας και κάθε ένα από αυτά αποτελούν ακραίες περιπτώσεις και κάθε ένα μια από τις τέσσερεις διαστάσεις ποιότητας για το κάθε μοντέλο εκτιμώνται χωρίς δυσκολία. Αντιθέτως, σε μια πιο ρεαλιστική περίπτωση είναι πολύ δυσκολότερο να εκτιμηθεί η ποιότητα των μοντέλων. Σε αυτό το κεφάλαιο θα παρουσιάσουμε κάποιους από τους τρόπους για τον ακριβή υπολογισμό της καταλληλότητας. Η καταλληλότητα, αποτελεί τη διάσταση ποιότητας του μοντέλου που “ελέγχει” κατά πόσο το μοντέλο μας αναπαριστά την συμπεριφορά που έχει καταγραφεί σε ένα αρχείο καταγραφής συμβάντων. Έτσι λοιπόν μπορούμε να καταλάβουμε πως από τις τέσσερεις διαστάσεις η καταλληλότητα είναι η πιο κοντινή στον έλεγχο συσχέτισης. Σε αυτό το κεφάλαιο λοιπόν, θα παρουσιάσουμε δύο τεχνικές που αποτελούν τις πιο διαδεδομένες τεχνικές για τον έλεγχο συσχέτισης. Η πρώτη, ονομάζεται *αναπαραγωγή των tokens (token replay)* ενώ η δεύτερη *σύγκριση αποτυπωμάτων (comparing footprints).*

6.1. Αναπαραγωγή των tokens

Σχετικά με τα μοντέλα που παρουσιάσαμε στο κεφάλαιο Διαστάσεις ποιότητας των μοντέλων, μια απλοϊκή προσέγγιση του ελέγχου συσχέτισης θα ήταν ο υπολογισμός του ποσοστού που τα ίχνη αναπαριστώνται από το μοντέλο μας. Έτσι λοιπόν, όλα τα μοντέλα εκτός του μοντέλου N_2 έχουν τη μέγιστη δυνατή καταλληλότητα που είναι ίση με ένα. Για να υπολογίσουμε το επίπεδο καταλληλότητας του μοντέλου N_2 αρκεί να υπολογίσουμε το πηλίκο 455 περιπτώσεων που αναπαριστώνται από το μοντέλο προς τον συνολικό αριθμό των περιπτώσεών μας, δηλαδή $\frac{455}{1391} = 0.3271$. Αυτή η απλοϊκή προσέγγιση όμως δεν εφαρμόζεται σε περιπτώσεις πιο ρεαλιστικών καταστάσεων και θα πρέπει να βρεθούν άλλες μέθοδοι. Μια από αυτές τις μεθόδους αποτελεί και η αναπαραγωγή των tokens που θα αναλύσουμε σε αυτό το κεφάλαιο.

Η μέθοδος αυτή βασίζεται στο "παιχνίδι" εκτέλεσης των διαδικασιών από το αρχείο καταγραφής συμβάντων στο μοντέλο μας. Κατά την διάρκεια που εκτελούμε τα ίχνη πάνω στο μοντέλο καταγράφουμε τις διαφορές τις οποίες παρουσιάζονται μεταξύ μοντέλου και αρχείου συμβάντων. Για να μπορέσουμε να εφαρμόσουμε αυτή τη μέθοδο είναι αναγκαίο να γνωρίζουμε τον κανόνα πυροδότησης που είδαμε και σε προηγούμενο κεφάλαιο. Έτσι λοιπόν μέσω αυτής της διαδικασίας καταφέρνουμε να απαριθμήσουμε τις διαφορές που παρουσιάζονται στην μοντελοποιημένη και την καταγεγραμμένη συμπεριφορά. Η διαδικασία έχει ως εξής. Παίρνουμε το κάθε ίχνος του αρχείου μας και το αναπαράγουμε στο μοντέλο. Κατά την διάρκεια της διαδικασίας αυτής καταγράφουμε τον αριθμό των *καταναλωμένων token (consumed tokens)*, των *παραγόμενων token (produced tokens)*, των *token που λείπουν (missing tokens)* και των *token που απομένουν (remaining tokens)*. Όπως έχουμε αναφέρει στο κεφάλαιο των προϋποθέσεων των αρχείων καταγραφής, οι μεταβάσεις έχουν την ιδιότητα να καταναλώνουν και να παράγουν tokens στο διάγραμμα. Τα tokens τα οποία λείπουν είναι τα tokens τα οποία καταναλώνονται, για να μπορέσουμε να φτάσουμε στον τερματικό κόμβο, χωρίς να υπάρχουν στο μοντέλο. Είναι δηλαδή οι περιπτώσεις όπου κάποια από τις μεταβάσεις δεν μπορεί να πυροδοτηθεί επειδή ο τόπος εισόδου της αντίστοιχης μετάβασης δεν περιέχει token και για αυτό το λόγο δεν μπορεί να πυροδοτηθεί. Για να μπορέσουμε να πυροδοτήσουμε αυτή τη μετάβαση αναγκαζόμαστε να προσθέσουμε ένα token στο μοντέλο μας για να συνεχίσουμε την διαδικασία. Τα tokens τα οποία απομένουν είναι τα tokens τα οποία μένουν στο μοντέλο μας στο τέλος της διαδικασίας, δηλαδή τα tokens που δεν καταναλώθηκαν. Για τον συμβολισμό θα χρησιμοποιήσουμε το p για τα παραγόμενα tokens, το c για τα καταναλωμένα tokens, το m για τα tokens που λείπουν και το r για τα tokens που απομένουν.

Όσο αναπαράγουμε ένα ίχνος σε οποιοδήποτε σημείο της διαδικασίας και αν βρισκόμαστε, ένας τόπος περιέχει $p + m - c$ αριθμό token, που είναι ένας μη αρνητικός αριθμός. Από αυτή τη σχέση καταλήγουμε σε μια σειρά από άλλες σχέσεις οι οποίες ισχύουν για κάθε χρονική περίοδο που εκτελούμε την διαδικασία καταγραφής των token.

Οι σχέσεις οι οποίες ισχύουν είναι: $p + m \geq c \geq m$ και $r = p + m - c$. Αυτές οι σχέσεις ισχύουν τόσο για έναν συγκεκριμένο τόπο όσο και για όλο το μοντέλο μας. Τέλος, υπάρχει ένα ιδιαίτερο βήμα κατά την αρχή και το τέλος της διαδικασίας. Στην αρχή της διαδικασίας το περιβάλλον πρέπει να παράξει ένα token έτσι στην αρχή κάθε διαδικασίας θα έχουμε πάντοτε $p = 1$. Ενώ στο τέλος της διαδικασίας το περιβάλλον πρέπει να καταναλώσει ένα token από τον τερματικό κόμβο και έτσι θα έχουμε πάντα σαν τελικό βήμα $c' = c + 1$.

Μετά την καταμέτρηση και την καταγραφή των μεταβλητών αυτών η καταλληλότητα του μοντέλου υπολογίζεται από τον ακόλουθο τύπο:

$$fitness(\sigma, N) = \frac{1}{2}\left(1 - \frac{m}{c}\right) + (1 - \frac{r}{p})$$

Η παραπάνω σχέση ισχύει για ένα και μοναδικό ίχνος . Η ίδια διαδικασία μπορεί να χρησιμοποιηθεί όταν έχουμε να κάνουμε με περισσότερα από ένα ίχνη. Ακολουθούμε την ίδια διαδικασία υπολογίζοντας τα καταναλωμένα, τα παραγόμενων, τα token που λείπουν και τα token που απομένουν υπολογίζουμε τα αντίστοιχα αθροίσματα και υπολογίζουμε την καταλληλότητα του μοντέλου. Έστω λοιπόν $p_{N,\sigma}$ ο αριθμός των παραγόμενων token όταν αναπαράγουμε το σ στο N. Όμοια έχουμε $c_{N,\sigma}$, $m_{N,\sigma}$, $r_{N,\sigma}$ τις αντίστοιχες ποσότητες. Μπορούμε λοιπόν να υπολογίσουμε το επίπεδο καταλληλότητας του αρχείου καταγραφής συμβάντων L σε ένα μοντέλο N από την παρακάτω σχέση:

$$fitness(L, N) = \frac{1}{2}\left(1 - \frac{\sum_{\sigma \in L} L(\sigma) \times m_{N,\sigma}}{\sum_{\sigma \in L} L(\sigma) \times c_{N,\sigma}}\right) + \frac{1}{2}\left(1 - \frac{\sum_{\sigma \in L} L(\sigma) \times r_{N,\sigma}}{\sum_{\sigma \in L} L(\sigma) \times p_{N,\sigma}}\right)$$

6.2. Σύγκριση αποτυπωμάτων

Στο προηγούμενο κεφάλαιο που αναλύσαμε την διαδικασία ανακάλυψης, ορίσαμε και τον όρο αποτύπωμα ενός αρχείου καταγραφής συμβάντων, Πίνακας 2. Η βασική ιδέα αυτής της τεχνικής βασίζεται στα αποτυπώματα. Τα αποτυπώματα χρησιμοποιούν τις σχέσεις ταξινόμησης όπως την ακολουθία ενός γεγονότος από ένα άλλο, την παράλληλη συσχέτιση γεγονότων και την μη ακολουθία γεγονότων. Μπορούμε να παράξουμε αποτυπώματα όχι μόνο για τα αρχεία καταγραφής συμβάντων αλλά και για τα μοντέλα, αφού τα μοντέλα παράγουν ίχνη. Έτσι μέσω μιας στατικής ανάλυσης έχουμε την δυνατότητα να αναγνωρίσουμε αυτές τις συσχετίσεις και να παράγουμε τα αντίστοιχα αποτυπώματα. Η διαδικασία που ακολουθούμε με αυτή τη τεχνική είναι να παράγουμε δύο αποτυπώματα ένα για το αρχείο καταγραφής συμβάντων και ένα για το μοντέλο μας. Αφού παράξουμε τα δύο αυτά αποτυπώματα αθροίζουμε τον συνολικό αριθμό των διαφορών που παρουσιάζονται μεταξύ των αποτυπωμάτων αυτών. Ο υπολογισμός του επιπέδου καταλληλότητας γίνεται υπολογίζοντας το πηλίκο του συνόλου των διαφορών των δύο

αποτυπωμάτων προς το συνολικό αριθμό των κλάσεων των ενεργειών στο τετράγωνο, δηλαδή τον αριθμό των θέσεων που έχει η μήτρα των αποτυπωμάτων.

7. Εργαλειοθήκη Εξόρυξης διαδικασιών

7.1. ProM

Όπως έχουμε προαναφέρει, η Εξόρυξη Διαδικασιών χρησιμοποιεί τις διαθέσιμες πληροφορίες από τα αρχεία καταγραφής συμβάντων ώστε να δείξει τη ροή των διαδικασιών σε μορφή ενός γραφικού μοντέλου. Το μοντέλο αντιπροσωπεύει τις εκτελούμενες διεργασίες βασιζόμενο στη σειρά που αυτές εμφανίζονται στα αρχεία καταγραφής συμβάντων. Το Πολυτεχνείο του Αϊντχόβεν (Eindhoven University of Technology TU/e), παρέχει ένα σύνολο εργαλείων που χρησιμοποιούνται για την εφαρμογή τεχνικών Εξόρυξης Διαδικασιών. Για να επιτευχθεί η Εξόρυξη Διαδικασιών μέσω αυτών των εργαλείων, υπάρχει μια σειρά βημάτων τα οποία θα πρέπει να ακολουθηθούν. Αρχικά, είναι απαραίτητη η παραγωγή ή η εξεύρεση αρχείων καταγραφής συμβάντων, κατά δεύτερον, τα αρχεία αυτά θα πρέπει να επεξεργαστούν και να μετατραπούν στη κατάλληλη μορφή (xml ή mxml) για εργαλεία Εξόρυξης Διαδικασιών. Τέλος, εισάγουμε τα αρχεία στα εργαλεία όπως το ProM [16, 17] προκειμένου να επιτευχθεί η ανάλυση.

Η πρώτη έκδοση του ProM κυκλοφόρησε το 2004 και το 2005 παρουσιάστηκε στο συνέδριο με θέμα Petri-nets. Στα χρόνια που ακολούθησαν, το ProM έχει επεκταθεί δραματικά και δεκάδες ερευνητές έχουν αναπτύξει νέες μεθόδους για αυτό το εργαλείο. Το ProM είναι ένα εργαλείο το οποίο παρέχεται ελεύθερα και οι χρήστες μπορούν να αναπτύξουν και να προσθέσουν νέες τεχνικές και εργαλεία για την Εξόρυξη Διαδικασιών. Η έκδοση η οποία παρουσιάστηκε το 2005 περιείχε 29 μεθόδους ενώ η σημερινή έκδοση περιέχει περισσότερες από 120 μεθόδους. Μπορούμε λοιπόν να καταλάβουμε ότι η εξέλιξη της επιστήμης της Εξόρυξης Διαδικασιών τα τελευταία χρόνια είναι ραγδαία και αποτελεί ένα από τα πιο καυτά θέματα. Θα παρουσιάσουμε την κατηγοριοποίηση των μεθόδων που χρησιμοποιούνται από το ProM:

- Μέθοδοι εισαγωγής (Import plug-ins): Οι μέθοδοι εισαγωγής χρησιμοποιούνται για να εισάγουν τα αρχεία καταγραφής συμβάντων.
- Μέθοδοι φίλτρων (Filter plug-ins): Οι μέθοδοι φίλτρων, χρησιμοποιούνται για την εξαγωγή περιττών πληροφοριών που μπορεί να αποτελούν θόρυβος για την εξαγωγή του μοντέλου διαδικασιών. Μπορούν επίσης να χρησιμοποιηθούν για να καθαριστεί το μητρώο αφαιρώντας περιπτώσεις που είναι ελλιπείς.
- Μέθοδοι εξόρυξης (Mining plug-ins): Σε αυτή την κατηγορία κατατάσσουμε όλους τους αλγόριθμους που χρησιμοποιούνται για την εξαγωγή των μοντέλων μας είτε αυτά είναι Petri-nets, μοντέλα BPMN, ασαφή μοντέλα κλπ. Σε αυτή τη κατηγορία επίσης ανήκει και ο αλγόριθμος άλφα που έχουμε αναλύσει σε προηγούμενο κεφάλαιο.

- Μέθοδοι εξαγωγής (Export plug-ins): Οι μέθοδοι εξαγωγής χρησιμοποιούνται για την αποθήκευση των μοντέλων και των αποτελεσμάτων που έχουν δημιουργηθεί από τα αρχεία καταγραφής συμβάντων.

Έτσι λοιπόν το ProM μας παρέχει την δυνατότητα να εισάγουμε σε αυτό δεδομένα τα οποία μπορούμε να επεξεργαστούμε με τις διάφορες μεθόδους ώστε να μπορέσουμε να παράξουμε μοντέλα τα οποία εκφράζουν το ιστορικό εκτέλεσης διαδικασιών σε έναν οργανισμό ή μια επιχείρηση. Επίσης τα μοντέλα αυτά έχουν την δυνατότητα να εμπεριέχουν πληροφορία σχετική με την ακριβή ημερομηνία και ώρα εκτέλεσης της κάθε ενέργειας όπως και τους πόρους τους οποίους έχουν εκτελέσει τις ενέργειες αυτές. Μέσω αυτών των πληροφοριών μπορούμε να έχουμε μια γενικότερη εικόνα για το πώς οι ενέργειες εκτελούνται την διάρκεια που χρειάζεται η κάθε ενέργεια για να εκτελεστεί όπως και το ποιος την εκτέλεσε. Έπειτα, έχουμε την δυνατότητα να μελετήσουμε τα παραγμένα μοντέλα και να βρούμε μέσω αυτών πιθανά σημεία καθυστέρησης ή δυσλειτουργίας της όλης διαδικασίας ώστε να την βελτιώσουμε στο σύνολό της. Επίσης μέσω της Εξόρυξης Διαδικασιών έχουμε την δυνατότητα να παράξουμε διαγράμματα συσχετίσεων όπου μας πληροφορούν για τον τρόπο αλληλεπίδρασης των πόρων ώστε να μπορέσουμε να δούμε ποιοι από αυτούς αποτελούν τα κύρια πρόσωπα και πως κατανέμεται η εργασία από πόρο σε πόρο. Με αυτό τον τρόπο δημιουργούμε τα διάφορα προφίλ που μπορεί να υπάρχουν στην όλη διαδικασία όπως διευθυντής, υποδιευθυντής, υπάλληλοι κτλ. Τέλος, το ProM μας δίνει τη δυνατότητα να βελτιώσουμε τα μοντέλα μας και να τα εξάγουμε στο σκληρό μας δίσκο για να μπορούμε να τα μελετήσουμε ή ακόμα να τα επεξεργαστούμε ανά πάσα στιγμή.

7.2. Αρχεία καταγραφής συμβάντων σε μορφή xml

Πολλά από τα συστήματα πληροφοριών καταγράφουν και αποθηκεύουν ενέργειες και διαδικασίες ως δεδομένα. Έχουν αναπτυχθεί πολλοί τρόποι και μέθοδοι που μας βοηθούν να αναλύσουμε σε βάθος αυτά τα δεδομένα με απώτερο σκοπό, να κατανοήσουμε και να βελτιώσουμε τις διαδικασίες αυτές. Η Εξόρυξη Διαδικασιών αποτελεί ένας από αυτούς τους τρόπους και τα αποθηκευμένα δεδομένα των διαδικασιών λέγονται αρχεία καταγραφής συμβάντων. Η μορφή στην οποία μπορούμε να βρούμε τα αρχεία καταγραφής συμβάντων διαφέρει από το ένα αρχείο στο άλλο. Επίσης, υπάρχουν πολλές περιπτώσεις όπου τα αρχεία αυτά περιέχουν πληροφορίες οι οποίες αποτελούν θόρυβο για την εξόρυξη ενός γραφικού μοντέλου μιας και οι διαδικασίες στην πραγματικότητα είναι πολύπλοκες και οι πληροφορίες οι όποιες καταγράφονται είναι παραπάνω από αρκετές από αυτές που οι αλγόριθμοι που χρησιμοποιούν και απαιτούν για την Εξόρυξη Διαδικασιών. Σε αυτό το κεφάλαιο θα παρουσιάσουμε την δομή των αρχείων καταγραφής συμβάντων σε μορφή xml ώστε να γίνει ευκολότερα κατανοητό το επόμενο κεφάλαιο όπου θα παρουσιάσουμε

το εργαλείο το οποίο έχει τη δυνατότητα να μετατρέπει ένα αρχείο καταγραφής συμβάντων από txt σε xml μορφή.

Για λόγους κατανόησης αποφασίσαμε να χρησιμοποιήσουμε ένα παράδειγμα για την παρουσίαση των αρχείων καταγραφής συμβάντων σε xml μορφή. Έστω λοιπόν, μια ιστοσελίδα στην οποία οι χρήστες μπορούν να ανεβάζουν και να μοιράζονται φωτογραφίες με άλλους χρήστες της ιστοσελίδας και έστω ότι υπάρχει ένας μηχανισμός ο οποίος έχει τη δυνατότητα να καταγράφει και να αποθηκεύει τις ενέργειες οι οποίες γίνονται σε αυτή την ιστοσελίδα. Πρέπει να τονίσουμε σε αυτό το σημείο πως η καταγραφή αυτή πρέπει να γίνεται με σκοπούς βελτίωσης της ιστοσελίδας και μελέτης των αναγκών των χρηστών και όχι με σκοπούς μαγνητοσκόπησης και παρακολούθησης αυτών. Έτσι λοιπόν για να γυρίσουμε πίσω στο παράδειγμά μας, έστω ότι το σύνολο των ενεργειών που μπορούν οι χρήστες σε αυτή την ιστοσελίδα να κάνουν είναι:

[Connection, UploadPhoto, CreateAlbum, LikePhoto, Disconnection]

Το αρχείο καταγραφής συμβάντων που θα αποθηκευτεί σε μορφή txt για τρείς χρήστες είναι το ακόλουθο:

User 1, 02/09/2014 11:03:04, Connection
User 1, 02/09/2014 11:04:55, CreateAlbum
User 2, 02/09/2014 11:05:00, Connection
User 1, 02/09/2014 11:05:46:, UploadPhoto
User 2, 02/09/2014 11:06:03, LikePhoto
User 1, 02/09/2014 11:06:34, UploadPhoto
User 1, 02/09/2014 11:08:10, Disconnection
User 2, 02/09/2014 11:09:14, Disconnection
User 3, 02/09/2014 11:10:00, Connection
User 3, 02/09/2014 11:11:16, CreateAlbum
User 3, 02/09/2014 11:11:55, UploadPhoto
User 3, 02/09/2014 11:12:34, LikePhoto
User 3, 02/09/2014 11:13:20, Disconnection

Παρατηρούμε πως το αρχείο καταγραφής συμβάντων, πληρεί τις ελάχιστες απαιτήσεις που είναι απαραίτητες για την Εξόρυξη Διαδικασιών, αφού τα ονόματα των χρηστών αποτελούν τις περιπτώσεις (cases), τα κλικ των χρηστών αποτελούν τις ενέργειες (events) και τέλος οι ενέργειες είναι ταξινομημένες αφού η κάθε ενέργεια συνοδεύεται από μια χρονική σήμανση (timestamp). Παρατηρούμε επίσης ότι οι περιπτώσεις, οι χρονικές σημάνσεις όπως και οι ενέργειες των χρηστών χωρίζονται μεταξύ τους με ένα κόμμα. Υπάρχουν πολλές περιπτώσεις από την άλλη μεριά όπου η κάθε πληροφορία χωρίζεται από την άλλη με άλλου είδους χαρακτήρες (π.χ. με ; όπου αυτού του είδους τα αρχεία ονομάζονται csv αρχεία) ή που οι πληροφορίες οι οποίες είναι καταγεγραμμένες

δεν είναι απαραίτητο να χρησιμοποιηθούν για την εξόρυξη ενός μοντέλου. Η δομή του συγκεκριμένου αρχείου μπορεί να θεωρηθεί ιδανική για την εφαρμογή τεχνικών Εξόρυξης Διαδικασιών όμως τα εργαλεία που χρησιμοποιούμε απαιτούν τα xml αρχεία. Αν μετατρέψουμε το παραπάνω txt αρχείο σε xml θα έχει την παρακάτω μορφή:

```
<?xml version="1.0" encoding="UTF-8"?>

<WorkflowLog xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="WorkflowLog.xsd" >

<Process id = “Internet Site”>
<ProcessInstance id="User 1">
      <AuditTrailEntry>
            <WorkflowModelElement>Start</WorkflowModelElement>
            <EventType>complete</EventType>
      </AuditTrailEntry>
      <AuditTrailEntry>
            <WorkflowModelElement>Connection</WorkflowModelElement>
            <EventType>complete</EventType>
            <Timestamp>2014-09-02T11:03:04,000</Timestamp>
      </AuditTrailEntry>
      <AuditTrailEntry>
            <WorkflowModelElement>CreateAlbum</WorkflowModelElement>
            <EventType>complete</EventType>
            <Timestamp>2014-09-02T11:04:55,000</Timestamp>
      </AuditTrailEntry>
      <AuditTrailEntry>
            <WorkflowModelElement>UploadPhoto</WorkflowModelElement>
            <EventType>complete</EventType>
            <Timestamp>2014-09-02T11:05:46,000</Timestamp>
```

```
        </AuditTrailEntry>
        <AuditTrailEntry>
                <WorkflowModelElement>UploadPhoto</WorkflowModelElement>
                <EventType>complete</EventType>
                <Timestamp>2014-09-02T11:06:34,000</Timestamp>
        </AuditTrailEntry>
        <AuditTrailEntry>
                <WorkflowModelElement>Disconnection
                </WorkflowModelElement>
                <EventType>complete</EventType>
                <Timestamp>2014-09-02T11:08:10,000</Timestamp>
        </AuditTrailEntry>
        <AuditTrailEntry>
                <WorkflowModelElement>End</WorkflowModelElement>
                <EventType>complete</EventType>
        </AuditTrailEntry>
</ProcessInstance>
<ProcessInstance id="User 2">
        <AuditTrailEntry>
                <WorkflowModelElement>Start</WorkflowModelElement>
                <EventType>complete</EventType>
        </AuditTrailEntry>
        <AuditTrailEntry>
                <WorkflowModelElement>Connection</WorkflowModelElement>
                <EventType>complete</EventType>
                <Timestamp>2014-09-02T11:05:00,000</Timestamp>
        </AuditTrailEntry>
        <AuditTrailEntry>
                <WorkflowModelElement>LikePhoto</WorkflowModelElement>
```

```
            <EventType>complete</EventType>
            <Timestamp>2014-09-02T11:06:03,000</Timestamp>
      </AuditTrailEntry>
      <AuditTrailEntry>
            <WorkflowModelElement>Disconnection
            </WorkflowModelElement>
            <EventType>complete</EventType>
            <Timestamp>2014-09-02T11:09:14,000</Timestamp>
      </AuditTrailEntry>
      <AuditTrailEntry>
            <WorkflowModelElement>End</WorkflowModelElement>
            <EventType>complete</EventType>
      </AuditTrailEntry>
</ProcessInstance>
<ProcessInstance id="User 3">
      <AuditTrailEntry>
            <WorkflowModelElement>Start</WorkflowModelElement>
            <EventType>complete</EventType>
      </AuditTrailEntry>
      <AuditTrailEntry>
            <WorkflowModelElement>Connection</WorkflowModelElement>
            <EventType>complete</EventType>
            <Timestamp>2014-09-02T11:11:10,000</Timestamp>
      </AuditTrailEntry>
      <AuditTrailEntry>
            <WorkflowModelElement>CreateAlbum</WorkflowModelElement>
            <EventType>complete</EventType>
            <Timestamp>2014-09-02T11:11:16,000</Timestamp>
      </AuditTrailEntry>
```

```
        <AuditTrailEntry>
            <WorkflowModelElement>UploadPhoto</WorkflowModelElement>
            <EventType>complete</EventType>
            <Timestamp>2014-09-02T11:11:55,000</Timestamp>
        </AuditTrailEntry>
        <AuditTrailEntry>
            <WorkflowModelElement>LikePhoto</WorkflowModelElement>
            <EventType>complete</EventType>
            <Timestamp>2014-09-02T11:12:34,000</Timestamp>
        </AuditTrailEntry>
        <AuditTrailEntry>
            <WorkflowModelElement>Disconnection
            </WorkflowModelElement>
            <EventType>complete</EventType>
            <Timestamp>2014-09-02T11:13:20,000</Timestamp>
        </AuditTrailEntry>
        <AuditTrailEntry>
            <WorkflowModelElement>End</WorkflowModelElement>
            <EventType>complete</EventType>
        </AuditTrailEntry>
    </ProcessInstance>
    </Process>
    </WorkflowLog>
```

Τώρα που έχουμε ένα παράδειγμα, μπορούμε να εξηγήσουμε λεπτομερώς την μορφή xml. Αρχικά, το xml αποτελεί συντομογραφία για *EXtensible Markup Language (επεκτάσιμη γλώσσα σήμανσης)* και είναι μια γλώσσα που ορίζεται από ένα σύνολο κανόνων για την κωδικοποίηση των εγγράφων που είναι αναγνώσιμη από τον άνθρωπο και από υπολογιστές. Ο συμβολισμός των xml αρχείων στοχεύει στην απλοϊκότητα, την γενικότητα και την ευχρηστία μέσω του διαδικτύου. Παρά το γεγονός ότι ο σχεδιασμός της

εστιάζει σε έγγραφα, η xml χρησιμοποιείται ευρέως για την αναπαράσταση αυθαίρετων δομών δεδομένων.

Για να καταλάβουμε καλύτερα το xml αρχείο, θα πρέπει να εστιάσουμε στις ετικέτες που περιβάλλονται από τα "< >", αυτές οι ετικέτες αποτελούν τη *σήμανση (markup)* και σε ένα αρχείο xml υπάρχουν αρχικές και τελικές ετικέτες σήμανσης (πχ. `<WorkflowModelElement>`, `<EventType>`, `<Timestamp>`, `<AuditTrailEntry>` κλπ.). Οι συμβολοσειρές από χαρακτήρες που δεν αποτελούν σήμανση, αποτελούν το *περιεχόμενο (content)* και είναι οι πληροφορίες που ήδη υπάρχουν στο txt αρχείο καταγραφής συμβάντων (πχ. `Complete`, `LikePhoto`, `2014-09-02T11:09:14,000` κλπ.). Μια αρχική ετικέτα σήμανσης μαζί με το περιεχόμενό της και την τελική ετικέτα σήμανσης είναι ένα *στοιχείο (element)*. Μερικές φορές, υπάρχουν στοιχεία που περιέχουν άλλα στοιχεία ενώ αυτά έχουν άλλες ετικέτες σήμανσης. Σε αυτή την περίπτωση, τα συμπεριλαμβανόμενα στοιχεία ονομάζονται *θυγατρικά στοιχεία (child elemets)* του βασικού στοιχείου που *ονομάζεται γονικό στοιχείο (parent element)*.

Έτσι λοιπόν για να κατασκευάσουμε το κατάλληλο xml αρχείο για το παράδειγμά μας, θα πρέπει αρχικά να ορίσουμε την εκδοχή που xml που θα χρησιμοποιήσουμε, εμείς χρησιμοποιούμε την 1.0, όπως και το είδος κωδικοποίησης, το οποίο στο παράδειγμά μας είναι το UTF-8. Στη συνέχεια, κατασκευάζουμε ένα στοιχείο για κάθε ένα χρήστη (`<ProcessInstance id="User">`). Τα στοιχεία αυτά αποτελούν τα κύρια γονικά στοιχεία (cases) του αρχείου μας τα οποία έχουν σαν θυγατρικά στοιχεία τα:

- The WorkflowModelElement, το οποίο αποτελεί την ετικέτα σήμανσης η οποία έχει σαν περιεχόμενο το όνομα της ενέργειας (task) του αντίστοιχου χρήστη.
- The EventType; αυτή ετικέτα σήμανσης έχει ως περιεχόμενο την αρχή ή το τέλος μιας ενέργειας και μας πληροφορεί εάν η ενέργεια έχει μόλις ξεκινήσει ή εάν έχει μόλις τελειώσει.
- The Timestamp, η οποία αποτελεί την ακριβή χρονική σήμανση της ενέργειας που έχει μόλις αρχίσει ή μόλις τελειώσει, ανάλογα με το περιεχόμενο της ετικέτας EventType. Είναι αρκετά σημαντικό να προσέξουμε τον τρόπο με τον οποίο θα γράψουμε την χρονική σήμανση.

Θα πρέπει να προσέξουμε πως κάθε φορά που εισάγουμε μια ετικέτα σήμανσης, θα πρέπει να την αντιστοιχούμε με την ανάλογη τελική ετικέτα που συμβολίζεται με τον ίδιο τρόπο με την μόνη διαφορά ότι μπροστά από το περιεχόμενο εισάγουμε τον χαρακτήρα "/". Τώρα όπου έχουμε τις βάσεις σχετικά με τη δομή των xml αρχείων μπορούμε να παρουσιάσουμε αναλυτικά το εργαλείο το οποίο μετατρέπει τα txt αρχεία καταγραφής συμβάντων σε xml αρχεία.

7.3. Txt σε xml μετατροπέας

Ένα πολύ σημαντικό μέρος της όλης διαδικασίας ήταν να δημιουργήσουμε ένα εργαλείο που μετατρέπει τα txt αρχεία σε xml, προκειμένου να έχουμε τη δυνατότητα να χρησιμοποιήσουμε τα κατάλληλα εργαλεία για την Εξόρυξη Διαδικασιών. Η δημιουργία μέσω αυτού του μετατροπέα των xml αρχείων είναι ένα πολύ βασικό βήμα αφού μέσω αυτού του εργαλείου απομονώνουμε της πληροφορίες που αποτελούν θόρυβο και κρατάμε μόνο την πληροφορία που είναι σημαντική για την Εξόρυξη Διαδικασιών. Επιπλέον, ο μετατροπέας έχει την δυνατότητα να ελέγχει τις χρονικές σημάνσεις και να κατανέμει ανάλογα με αυτές, τη σειρά των ενεργειών. Ο μετατροπέας αυτός είναι ένα πρόγραμμα το οποίο έχει γραφτεί σε Python (γλώσσα προγραμματισμού) το οποίο έχει την δυνατότητα να σαρώνει το κείμενο και να δημιουργεί πίνακες μέσα στους οποίους αποθηκεύει την πληροφορία. Το πλεονέκτημα του μετατροπέα αυτού είναι ότι αυτή η τεχνική μπορεί να εφαρμοστεί σε κάθε αρχείο καταγραφής συμβάντων σε μορφή κειμένου ώστε να δημιουργηθεί το κατάλληλο xml αρχείο.

Τα αρχεία καταγραφής συμβάντων, παράγονται κυρίως από μηχανές οι οποίες κρατάνε αρχείο για τις επιχειρησιακές διαδικασίες. Έτσι λοιπόν, μπορούμε να καταλάβουμε ότι η δομή των αρχείων αυτών ακολουθεί κάποιο κύριο κανόνα, αφού τα αρχεία αυτά παράγονται με αυτόματο τρόπο. Η πρόκληση σε αυτό το σημείο, είναι η ανακάλυψη των κανόνων αυτών και η διάκριση των σημαντικών πληροφοριών πριν την μετατροπή τους σε xml αρχεία. Όταν ανακαλυφθούν οι κανόνες και οι σημαντικές πληροφορίες, εφαρμόζουμε μεθόδους διαχωρισμού του κειμένου (πχ. Πριν από ορισμένους χαρακτήρες ή κενά) και συμπληρώνουμε τις πληροφορίες σε μήτρες. Ο μετατροπέας αυτός χρησιμοποιεί κυρίως τρείς ή τέσσερις διαφορετικές μήτρες, ανάλογα με τις πληροφορίες που παρέχονται από τα δεδομένα. Έτσι λοιπόν χρησιμοποιούμε μια μήτρα για τις χρονικές σφραγίδες, μια για τις ενέργειες και μια για τις περιπτώσεις, τα οποία αποτελούν τις βασικές μεταβλητές για την δομή των αρχείων καταγραφής συμβάντων που είναι απαραίτητα για την Εξόρυξη Διαδικασιών. Μετά την προσάρτηση των μητρών, χρησιμοποιούμε βρόγχους για να τυπώσουμε της πληροφορίες και να δημιουργήσουμε τα κατάλληλα xml αρχεία. Ο αλγόριθμος του προγράμματος παρουσιάζεται παρακάτω.

Algorithm 1 Converting txt file to xml

1: infile = open txt file for read

2: lines = save lines of infile

3: create **empty list** names

4: create **empty list** date

5: create **empty list** action

6: **for** i from 0 until length(lines):

8: c = split the lines between ‘,’

9: **append** first position of c to names

10: **append** first position of c to action

11: **append** first position of c to date

12: **end for**

13: create **empty list** test_names

14: **for** i from 1 until length(names):

15: **if** names[i] **not in** test_names:

16: append i position of names to test_names

17: **end if**

18: **end for**

19: create **empty list** date_new

20: **for** i from 1 until length(date):

21: **append** i position of date, format: YYYY-mm-ddThh:mm:ss to date_new

22: **end for**

23: g = open xml new_file for writing

24: g.write (‘<?xml version=\"1.0\" encoding=\"UTF-8\"?>\n\n<WorkflowLog xmlns:xsi=\"http://www.w3.org/2001/XMLSchema-instance\"\nxsi:noNamespaceSchemaLocation=\"WorkflowLog.xsd\" >\n<Source program=\"dummy\">\n</Source>\n<Process id=\"Companies_01-06till08-06 xml\">\n')

25: **for** i from 1 until length(test_names):

26: g.write('<ProcessInstance id="' + test_names[i]+ '">\n')

27: g.write('<AuditTrailEntry>\n<WorkflowModelElement>Start </WorkflowModelElement>\n<EventType>complete </EventType>\n</AuditTrailEntry>\n')

28: **for** j from 1 until length(action):

```
29:         if test_names[i] == names[j]

30:         g.write (''<AuditTrailEntry>\n<WorkflowModelElement>' +
                action[j] +  '</WorkflowModelElement>\n<EventType>
                complete </EventType>\n <Timestamp>' + date[j] +
                '</Timestamp>\n'  + '</AuditTrailEntry>\n'')

31:     end if

32:     end for

33:     g.write('<AuditTrailEntry>\n<WorkflowModelElement> End
            </WorkflowModelElement>\n<EventType>complete</EventType>\n
            </AuditTrailEntry>\n')

34: g.write('</ProcessInstance>\n')

35: g.write('</Process>\n</WorkflowLog>')

36: end algorithm
```

Ο αλγόριθμος αυτός όπως μπορούμε να δούμε, αρχικά ανοίγει το txt αρχείο για ανάγνωση και δημιουργεί μια μήτρα όπου το κάθε στοιχείο της αποτελεί την κάθε γραμμή του txt αρχείου καταγραφής συμβάντων. Στη συνέχεια, δημιουργεί τρεις κενές μήτρες όπου σταδιακά με την βοήθεια ενός βρόγχου επαναλήψεων που έχει μήκος ίσο με τον αριθμό γραμμών του txt αρχείου, γεμίζει με πληροφορίες απαραίτητες για την Εξόρυξη Διαδικασιών. Οι τρείς αυτές μήτρες περιέχουν δεδομένα για τα ονόματα των χρηστών, τις ενέργειές τους και τις ακριβείς ημερομηνίες και ώρες που οι ενέργειες αυτές πραγματοποιήθηκαν. Παρατηρούμε ότι για να ορίσουμε τις ακριβείς πληροφορίες διαχωρίζουμε το txt αρχείο μεταξύ των κομμάτων όπου είναι ο χαρακτήρας ο οποίος το αρχείο μας ουσιαστικά χωρίζει τις πληροφορίες. Αφού γεμίσουμε τις μήτρες με τις πληροφορίες από κάθε σειρά του αρχικού αρχείου καταγραφής συμβάντων, γεμίζουμε άλλη μια μήτρα που ονομάζεται test_names η οποία περιέχει όλα τα ονόματα των χρηστών μόνο μία φορά. Βασικά, δημιουργούμε ένα βρόγχο ελέγχου όπου σαρώνει την αρχική μήτρα με τα ονόματα των χρηστών και κάθε φορά που πέφτει πάνω σε κάποιο όνομα το οποίο δεν εμπεριέχεται στην test_names προσαρτά το όνομα αυτό στη μήτρα. Στη συνέχεια, πάλι με την βοήθεια ενός νέου βρόγχου αλλάζουμε την δομή των ετικετών ημερομηνίας και ώρας και τις προσαρμόζουμε ανάλογα με τις απαιτήσεις των xml αρχείων. Τέλος, φτάνουμε στη τύπωση των πληροφοριών και την δημιουργία του xml αρχείου. Για να γίνει αυτό, αρχικά τυπώνουμε στο νέο αρχείο τις αρχικές γραμμές σχετικά με την εκδοχή του αρχείου και τον τύπο κωδικοποίησης και έπειτα, δημιουργούμε κατασκευάζουμε με την βοήθεια της μήτρας test_names ένα στοιχείο για κάθε χρήστη το οποίο εμπεριέχει πληροφορίες σχετικες με τις ενέργειές του και τις ακριβείς χρονικές τους ταμπέλες όπως ακριβώς είδαμε και στο προηγούμενο κεφάλαιο.

Ο αλγόριθμος αυτός είναι επίσης σε θέση επίσης να *ομαδοποιεί (cluster)* τις ενέργειες σύμφωνα με ονοματικές συσχετίσεις που αποτελεί μια πολύ σημαντική ικανότητα όταν έχεις να μελετήσεις δεδομένα από τις δράσεις επιχειρησιακών διαδικασιών με παρόμοια ονόματα, όμως για λόγους κατανόησης δεν συμπεριλάβαμε αυτή την ιδιότητα στον αλγόριθμο που παρουσιάζεται παραπάνω. Για την ιδιότητα αυτή του αλγορίθμου βασιστήκαμε στις μεθοδολογίες επιχειρησιακών διαδικασιών που χρησιμοποιούνται σε δημοσίευση [18] για το ηλεκτρονικό εμπόριο, όπου χρησιμοποιούνται *τοποθεσίες ηλεκτρονικών διευθύνσεων (URLs)* ως ενέργειες, όμοια με το παράδειγμα που παρουσιάσαμε στο προηγούμενο κεφάλαιο, και με τη μέθοδο της ομαδοποίησης επιτεύχθηκε η μείωση των ενεργειών από 949 532 σε 14 λογικούς τύπους σελίδων. Και αξίζει να σημειωθεί ότι, όπως υποστηρίζουν και οι συντάκτες της δημοσίευσης αυτής με την διαδικασία ταξινόμησης δεν χάθηκαν δεδομένα. Θα παρουσιάσουμε όμως σε επόμενα κεφάλαια τρόπους για να φιλτράρουμε και να ομαδοποιούμε ενέργειες από ένα αρχείο καταγραφής συμβάντων.

Σε αυτό το κεφάλαιο παρουσιάσαμε τα αρχεία καταγραφής συμβάντων σε μορφή xml όπως και τον αλγόριθμο του μετατροπέα txt σε xml. Στα επόμενα κεφάλαια θα επικεντρωθούμε στην Εξόρυξη Διαδικασιών από πραγματικά δεδομένα τα οποία έχουμε καταφέρει να αναλύσουμε αλλά και να παράξουμε από μια πλατφόρμα που χρησιμοποιείται από εταιρίες για την πρόβλεψη των πωλήσεων των προϊόντων τους εν ονόματι Colibri. Θα δούμε λοιπόν ένα τρόπο για παραγωγή δεδομένων πως τα αποθηκεύουμε όπως επίσης και τα αποτελέσματα από την εξόρυξη μοντέλων.

8. Used case the Colibri platform

8.1. Εισαγώγη στο Colibri

Η βασική ιδέα της διπλωματικής εργασίας είναι η εφαρμογή τεχνικών Εξόρυξης Διαδικασιών σε ένα εργαλείο ή μια εφαρμογή. Για το σκοπό αυτό έχουμε επιλέξει το Colibri το οποίο είναι ένα εργαλείο πρόβλεψης. Ένα από τα πλεονεκτήματα του Colibri είναι ότι η κύρια πλοήγησή του είναι οργανωμένη σύμφωνα με τις βασικές φάσεις της πρόβλεψης πωλήσεων οι οποίες είναι:

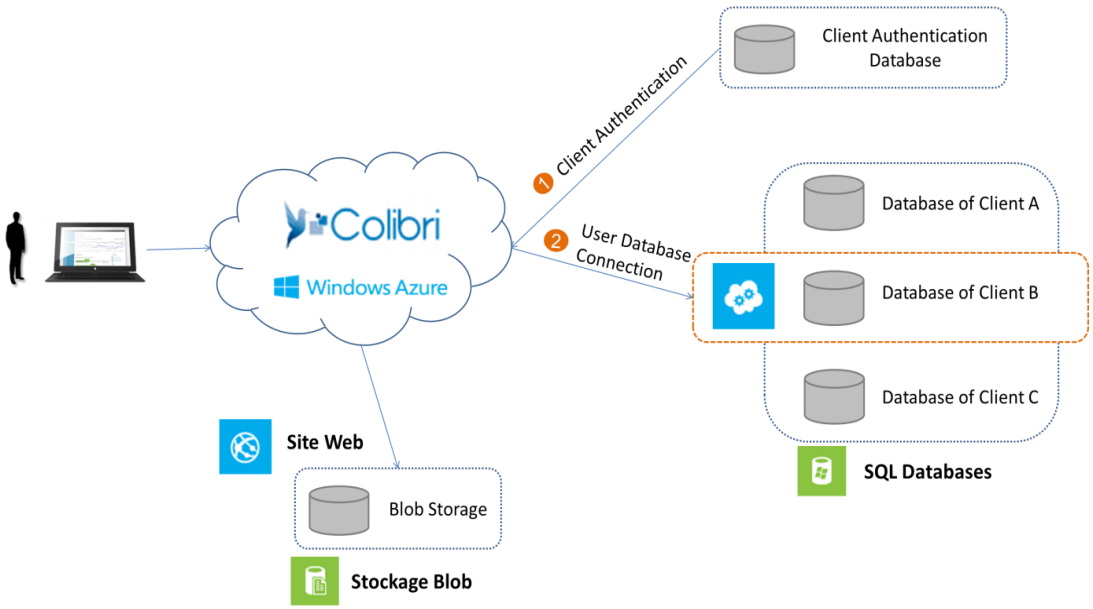
- **Παρελθοντικές πωλήσεις (History Page):** Σε αυτή τη σελίδα οι χρήστες έχουν την δυνατότητα να επανεξετάσουν τις παρελθοντικές πωλήσεις των προϊόντων τους. Επίσης, το Colibri εντοπίζει αυτόματα μη φυσιολογικές τιμές υπολογίζοντας ένα περιθώριο ομαλότητας (ένα διάστημα εμπιστοσύνης). Έτσι, οι χρήστες έχουν τη δυνατότητα να διορθώσουν το ιστορικό των δεδομένων χειροκίνητα, αυτόματα ή ζητώντας τη συμβουλή των συναδέλφων τους μέσω μηνυμάτων. Μετά την τροποποίηση των τιμών οι χρήστες μπορούν να επικυρώσουν και να αποθηκεύσουν τις αλλαγές τους.
- **Διαχείριση κύκλου ζωής των προϊόντων (New Products Page):** Αυτή η σελίδα αποτελείται από την προσθήκη πληροφοριών σχετικά με νέα προϊόντα, καθώς και στον προγραμματισμό της αντικαταστάσεως και διαγραφής άλλων προϊόντων.
- **Υπολογισμός πρόβλεψης (Forecast Page):** Η σελίδα αυτή αποτελείται από τον αυτόματο υπολογισμό μελλοντικών πωλήσεων με βάση τα δεδομένα που εξετάζονται από τις πωλήσεις του παρελθόντος. Αυτή η λειτουργία πραγματοποιείται με τη χρήση στατιστικών αλγορίθμων.
- **Εμπλουτισμός (Enrichment Page):** Η δυνατότητα εμπλουτισμού επιτρέπει την ενσωμάτωση της επιχειρηματικής γνώσης των χρηστών στην διαδικασία για μια πιο αξιόπιστη και εύχρηστη πρόβλεψη πωλήσεων. Κάθε χρήστης μπορεί να εμπλουτίσει αυτή την πρόβλεψη και να υποβάλει τα αποτελέσματά του (ανάλογα με τα δικαιώματα του στο πλαίσιο της πλατφόρμας). Εν συνεχεία ένας υπεύθυνος για τις προβλέψεις πωλήσεων μπορεί να επιλέξει μεταξύ αυτών των αποτελεσμάτων, προκειμένου να παρθεί μια τελική απόφαση.
- **Ανάλυση αξιοπιστίας πρόβλεψης (Reliability Page):** Αυτή η σελίδα αποτελείται από τη σύγκριση των διάφορων προβλέψεων, έτσι ώστε να εντοπιστούν νωρίς σημαντικά κενά.

Κύριος στόχος της εργασίας αυτής είναι να καθοριστεί και να διερευνηθεί ο τρόπος με τον οποίο οι χρήστες πλοηγούνται στο Colibri προκειμένου να βελτιώσουμε και να κάνουμε την πλατφόρμα χρησιμότερη. Η κυριότερη δυσκολία παρουσιάζεται στη μεγάλη

ευελιξία που επιτρέπει το Colibri στο σχεδιασμό και την πλοήγησή του. Η πλοήγηση λοιπόν είναι σύνθετη και υπάρχει μεγάλος αριθμός κουμπιών και μπαρών που κάνουν το έργο μας ακόμα πιο δύσκολο. Για την προσέγγισή μας θα χρησιμοποιήσουμε εργαλεία και τεχνικές της Εξόρυξης Διαδικασιών ώστε να ανακαλύψουμε τα πιο συνηθισμένα μονοπάτια που ακολουθούν οι χρήστες πάνω στην πλατφόρμα, για να προσδιορίσουμε τα διάφορα προφίλ των χρηστών όπως επίσης για να δούμε ποια παράθυρα της πλατφόρμας χρησιμοποιούν περισσότερο.

Αρχιτεκτονική του Colibri:

Για να μπορέσουμε να εξηγήσουμε τους τρόπους με τους οποίους παράξαμε και αποθηκεύσαμε τα δεδομένα μας, είναι αναγκαίο να αναφερθούμε και να αναλύσουμε την αρχιτεκτονική που κρύβεται πίσω από το Colibri. Πρώτα από όλα, το Colibri λειτουργεί με την βοήθεια του Windows Azure και ο κάθε χρήστης ανήκει σε μια από τις πλατφόρμες του Colibri ανάλογα με την έκδοση του Colibri που αυτός κατέχει. Η κάθε πλατφόρμα ανήκει σε διαφορετικό περιβάλλον και κάθε περιβάλλον έχει ένα Windows Azure container που μέσα του μπορούμε να αποθηκεύσουμε αρχεία blob. Φανταστείτε ότι υπάρχουν περίπου επτά διαφορετικά περιβάλλοντα του Colibri και το κάθε περιβάλλον έχει πάνω από δύο διαφορετικές πλατφόρμες. Οι λογαριασμοί Windows Azure χρησιμεύουν για την αποθήκευση αρχείων καταγραφής συμβάντων σε μορφή αρχείων blob. Τα blobs αποτελούν συνήθως txt αρχεία τα οποία αποθηκεύονται σε θέσεις οι οποίες είναι καταγεγραμμένες στον κώδικα και με αυτό τον τρόπο γνωρίζουμε κάθε στιγμή που αποθηκεύονται τα αρχεία μας.



Σχήμα 14: Αρχιτεκτονική και λειτουργία του Colibri.

Η λειτουργία του Colibri λοιπόν έχει ως εξής. Κάθε φορά που κάποιος χρήστης πάει να συνδεθεί στο Colibri (Σχήμα 14), αναλόγως στον περιβάλλον που αυτός ανήκει, ο αντίστοιχος λογαριασμός Windows Azure συνδέεται στη βάση δεδομένων (Authentication Database) που είναι υπεύθυνη για τις υπηρεσίες πιστοποίησης (π.χ. ονόματα χρηστών, κωδικοί χρηστών, κτλ.). Μετά τον έλεγχο των κωδικών, η ίδια βάση δεδομένων στέλνει σήμα στη διαδικτυακή εφαρμογή και ο χρήστης συνδέεται στην δική του βάση δεδομένων. Ο κάθε χρήστης έχει την δική του βάση δεδομένων όπου σε αυτή αποθηκεύονται πληροφορίες σχετικές με τις προσωπικές του ρυθμίσεις, τα προϊόντα που πουλάει η εταιρεία του, και τα προϊόντα τα οποία έχει επικυρώσει, καταγράψει και αποθηκεύσει. Κάθε περιβάλλον του Colibri έχει το δικό του λογαριασμό αποθήκευσης Windows Azure και κάθε μέρα ένα αρχείο καταγραφής συμβάντων δημιουργείται για κάθε περιβάλλον που περιέχει πληροφορίες σχετικά με την δραστηριότητα των χρηστών την προηγούμενη μέρα. Έτσι λοιπόν καταφέρνουμε να αποθηκεύσουμε χωρίς δυσκολίες μεγάλες ποσότητες δεδομένων που στη συνέχεια κατεβάζουμε με αυτόματο τρόπο δημιουργώντας ένα αρχείο το οποίο περιέχει τις κινήσεις των χρηστών ανεξάρτητα από την πλατφόρμα από την οποία αυτός ανήκει.

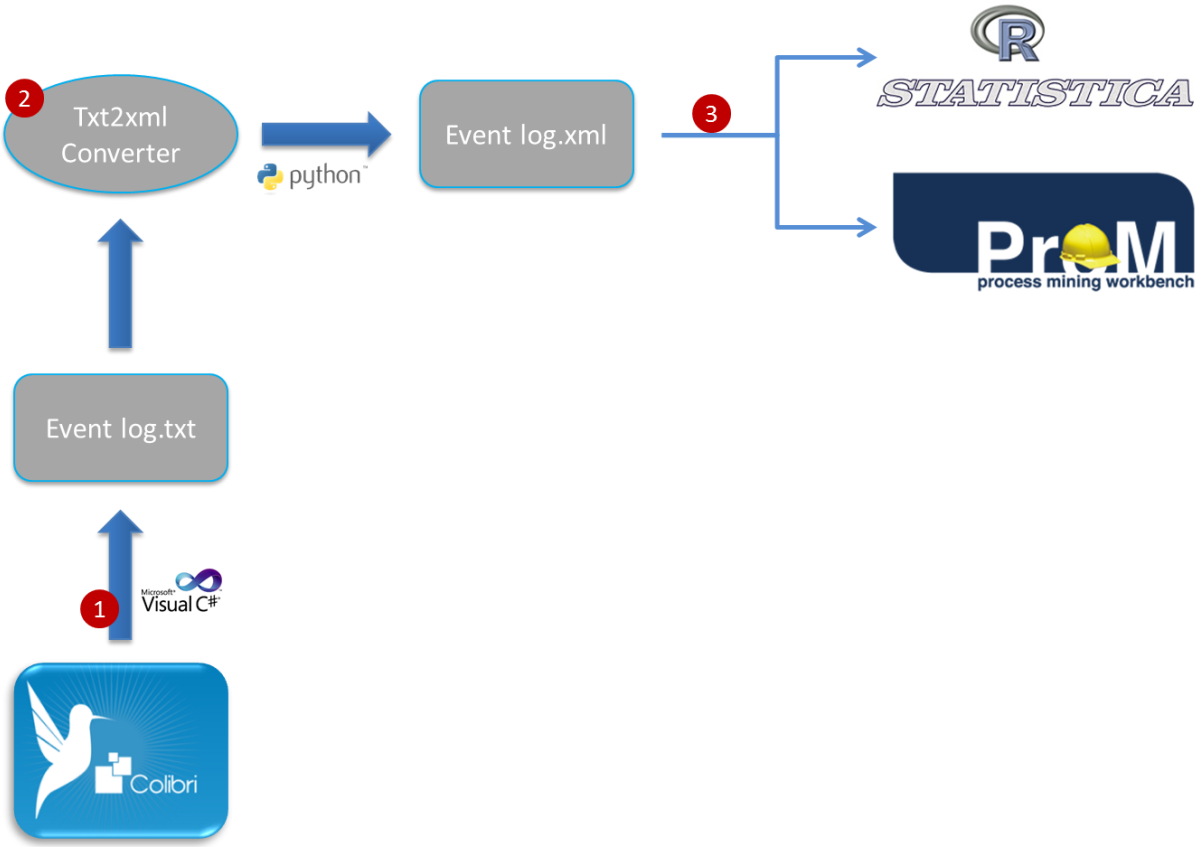
8.2. Διαδικασία

Το διαδίκτυο στις μέρες μας έχει αναπτύξει μια δυναμική φύση με αποτέλεσμα να αυξάνεται η ανάγκη για νέες μεθόδους και εργαλεία τα οποία θα αυξήσουν την αποδοτικότητα και την απόδοση των ιστοσελίδων. Οι πληροφορίες που μπορούμε να αντλήσουμε από την συμπεριφορά των χρηστών στις εκάστοτε ιστοσελίδες προσφέρουν ιδέες για τους προγραμματιστές ώστε να τις τροποποιήσουν και να τις βελτιώσουν. Τα ήδη υπάρχοντα εργαλεία για ανάλυση ιστοσελίδων όμως, δεν λαμβάνουν υπόψη την αφηρημένη συμπεριφορά των επισκεπτών τους.

Εμείς από την μεριά μας, εφαρμόσαμε τεχνικές Εξόρυξης Διαδικασιών πάνω στο Colibri και τα παραγόμενα μοντέλα παρουσιάζουν μια αφηρημένη άποψη της σχέσης μεταξύ των ιστοσελίδων. Έτσι λοιπόν μπορούμε να εστιάσουμε στις κρισιμότερες διαδρομές που ακολουθούνται από τους χρήστες, τις ιστοσελίδες που επισκέπτονται περισσότερο, κλπ. Τα μοντέλα συμπεριφοράς των χρηστών μπορούν να παρέχουν γνώσεις σχετικά με τις πιθανές βελτιώσεις και ενισχύει στη λήψη αποφάσεων για πιθανές τροποποιήσεις τους. Ειδικότερα, επικεντρωνόμαστε στην πλοήγηση των χρηστών μέσω της παραγωγής μοντέλων θεωρώντας τα κλικ των χρηστών ως ένα μια μη δομημένη διαδικασία και χρησιμοποιούμε αλγορίθμους Εξόρυξης Διαδικασιών για την ανακάλυψη της συμπεριφοράς των χρηστών του Colibri. Όπως αναφέραμε και σε προηγούμενο κεφάλαιο το Colibri είναι ένα συνεργατικό εργαλείο πρόβλεψης πωλήσεων που δημιουργήθηκε από την εταιρεία Viseo η οποία έχει μια έδρα της στην Γκρενόμπλ της Γαλλίας. Πρόκειται λοιπόν για μια μελέτη η

οποία πραγματοποιήθηκε στην Viseo, στην Γκρενόμπλ κατά την διάρκεια μιας πρακτικής άσκησης έξι μηνών.

Η δημιουργία των μοντέλων και η επίτευξη της ανάλυσης μέσω της Εξόρυξης Διαδικασιών αποτελεί το τελικό στάδιο του έργου. Για να καταφέρουμε να φτάσουμε σε αυτό το στάδιο υπάρχει μια σειρά από επιμέρους βήματα που πρέπει να ακολουθηθούν τα οποία θα μας επιτρέψουν να χρησιμοποιήσουμε τα εργαλεία και τις τεχνικές της Εξόρυξης Διαδικασιών ώστε να πάρουμε κάποια αποτελέσματα. Η δομή της μεθόδου που ακολουθήσαμε όπως και τα επιμέρους βήματα παρουσιάζονται στο Σχήμα 15. Σύμφωνα με το Σχήμα 15 τα βήματα έχουν ως εξής:



Σχήμα 15: Η διαδικασία από την Εξόρυξη Δεδομένων έως την Ανάλυση και την Εξόρυξη Διαδικασιών.

1. Το πρώτο βήμα είναι η δημιουργία και η συλλογή των αρχείων καταγραφής συμβάντων από το Colibri. Μιας και τα αρχεία καταγραφής συμβάντων αποτελούν την αφετηρία της όλης διαδικασίας και είναι απαραίτητο αυτά να περιέχουν όλες τις απαραίτητες πληροφορίες. Τα αρχεία καταγραφής συμβάντων μας πρέπει να ακολουθούν τους θεμελιώδεις κανόνες και θα πρέπει να είναι καλά οργανωμένα και δομημένα. Εδώ αξίζει να σημειωθεί πως στις περισσότερες περιπτώσεις είναι

καλύτερο να καταγράφονται περισσότερες πληροφορίες από αυτές που είναι αναγκαίες μιας και υπάρχουν πολλές μέθοδοι για το φιλτράρισμα και την επεξεργασία των αρχείων αυτών. Τα αρχεία καταγραφής συμβάντων παράγονται κατευθείαν από τον κώδικα του Colibri ο οποίος είναι γραμμένος σε C#. Για την επίτευξη του πρώτου βήματος χρησιμοποιούμε μια βιβλιοθήκη που ονομάζεται Apache log4net η οποία παρουσιάζεται αναλυτικά σε επόμενο κεφάλαιο. Η log4net λειτουργεί σε Windows Azure Storage είναι λοιπόν συμβατή με τα δικά μας δεδομένα. Τα αρχεία καταγραφής συμβάντων που παράγονται είναι σε txt μορφή και αποθηκεύονται στον χώρο αποθήκευσης του Windows Azure ως blobs.

2. Στη συνέχεια, χρησιμοποιούμε τον μετατροπέα που έχει γραφτεί σε Python, ο οποίος μετατρέπει το αρχείο καταγραφής συμβάντων από txt σε xml ή και σε csv αρχεία ανάλογα με τις ανάγκες μας. Η xml μορφή αποτελεί την κατάλληλη μορφή για το εργαλείο Εξόρυξης Διαδικασιών, το ProM, και η csv αποτελεί την κατάλληλη μορφή για το εργαλείο R το οποίο χρησιμοποιείται για στατιστικές αναλύσεις.
3. Τελικό βήμα είναι η δημιουργία μοντέλων διαδικασίας και η ανάλυση αυτών και των δεδομένων που έχουμε συλλέξει από το Colibri. Αυτό το στάδιο παρουσιάζεται λεπτομερώς σε κεφάλαια που θα δούμε αργότερα.

Εμείς λοιπόν θα πρέπει να συμπεριλάβουμε πληροφορίες στο αρχείο μας που είναι σχετικές με:

- Ονόματα χρηστών
- Το περιβάλλον του Colibri όπου ο κάθε χρήστες ανήκει
- Τις ενέργειες των χρηστών (τα κουμπιά που χρησιμοποιούν)
- Και τις χρονοετικέτες των ενεργειών αυτών

Κάθε φορά λοιπόν που κάποιος χρήστης κάνει κάποια ενέργεια πάνω στην πλατφόρμα, παράγεται αυτόματα μια γραμμή στα αρχεία μας η οποία περιέχει τις παραπάνω πληροφορίες.

8.2.1. Παραγωγή αρχείων καταγραφης συμβάντων από το Colibri

Στο κεφάλαιο αυτό παρουσιάζεται μια τεχνική για την λήψη αρχείων καταγραφής συμβάντων από το Colibri. Για την λήψη των αρχείων από τον κώδικα του Colibri, χρησιμοποιούμε μια βιβλιοθήκη που ονομάζεται Apache log4net η οποία βοηθά τους προγραμματιστές να εξάγουν αρχεία δηλώσεων σε μια ποικιλία στόχων εξόδου. Η log4net είναι ένα μέρος της δημοφιλούς log4j βιβλιοθήκης η οποία χρησιμοποιείται στην Java και

παρέχει ένα απλό μηχανισμό για την καταγραφή πληροφοριών ως δεδομένα. Οι πληροφορίες καταγράφονται μέσω ενός ή πολλαπλών *καταγραφέων (loggers)*. Οι καταγραφείς αυτοί, παρέχουν πέντε επίπεδα καταγραφής και αυτά είναι:

- Εντοπισμού σφαλμάτων (Debug)
- Πληροφοριών
- Προειδοποιήσεων
- Σφαλμάτων
- Ασφαλείας

Εμείς χρησιμοποιούμε το δεύτερο επίπεδο καταγραφέων στο οποίο και ανήκουμε αφού παράγουμε αρχεία τα οποία περιέχουν μεγάλη ποικιλία πληροφορίας. Το ερώτημα το οποίο παρουσιάζεται σε αυτό το σημείο είναι, που αποθηκεύονται τα αρχεία καταγραφής; Οι καταγεγραμμένες πληροφορίες λοιπόν πηγαίνουν σε αυτό που ονομάζεται appender. Ένας appender, είναι ο προορισμός στον οποίο αποθηκεύονται τα αρχεία μας. Υπάρχουν πολλά είδη appenders που επιτρέπουν την αποστολή των αρχείων καταγραφής συμβάντων σε βάσεις δεδομένων, ηλεκτρονικά ταχυδρομεία κλπ. Έχουμε επίσης την δυνατότητα να μην περιοριστούμε στη χρήση ενός μόνο appender μπορούμε να έχουμε όσους appenders επιθυμούμε και άρα και όσα μονοπάτια στα οποία στέλνουμε τις πληροφορίες μας.

Δομή των αρχείων καταγραφής συμβάντων του Colibri:

Κάθε φορά που ένας χρήστης εκτελεί κάποια ενέργεια στην πλατφόρμα Colibri, είναι ανάγκη να καταγράφονται ένα σύνολο από ενέργειες ώστε η ανάλυση μέσω τεχνικών Εξόρυξης Διαδικασιών να καταστεί εφικτή. Έτσι λοιπόν, κάθε φορά που γίνεται ένα κλικ στην πλατφόρμα καταγράφονται οι παρακάτω πληροφορίες:

- Όνομα χρήστη (σε μορφή e-mail)
- Η ενέργεια που εκτέλεσε ο χρήστης (κλικ στην πλατφόρμα)
- Η ακριβής μέρα και ώρα της ενέργειας αυτής
- Το περιβάλλον του Colibri στο οποίο ο χρήστης είναι καταχωρημένος

Η πρόκληση σε αυτό το σημείο είναι η εύρεση ενός τρόπου για να ορίσουμε τις ενέργειες των χρηστών, μιας και κάθε κουμπί είναι μοναδικό στην πλατφόρμα. Έτσι λοιπόν αποφασίσαμε οι ενέργειες των χρηστών να καταγράφονται με τον ίδιο τρόπο όπως και οι διευθύνσεις URL εμφανίζονται σε μια ιστοσελίδα. Για παράδειγμα αν ένας χρήστης επιλέξει το πλήκτρο General Settings το οποίο αποτελεί μια μετάβαση για τις γενικές ρυθμίσεις και είναι τοποθετημένο στην σελίδα των ρυθμίσεων (Settings page) η αντίστοιχη ενέργεια η οποία θα καταγραφεί είναι: “Settings/GeneralSettings”. Μια ακόμη δυσκολία την οποία

αντιμετωπίσαμε κατά την διάρκεια ορισμού των ενεργειών ήταν μια μεγάλη λίστα, που παρουσιάζεται σε βασικές σελίδες, η οποία περιέχει μια σειρά από προϊόντα τα οποία ο κάθε χρήστης έχει αποθηκευμένα στην βάση δεδομένων του. Για την επίλυση του προβλήματος αυτού και για να μην χάσουμε πληροφορία αποφασίσαμε κάθε φορά που κάποιος χρήστης επιλέγει ένα προϊόν από την λίστα να τυπώνουμε δύο γραμμές ενεργειών στο αρχείο καταγραφής συμβάντων. Έτσι για παράδειγμα, εάν ένας χρήστης επιλέξει ένα προϊόν από την λίστα που εμπεριέχεται στην σελίδα εμπλουτισμού (Enrichment page) θα τυπωθούν αυτόματα δύο σειρές όπου η πρώτη σειρά αποτελεί μια γενικότερη πληροφόρηση, δηλαδή τυπώνεται σαν ενέργεια Enrichment Product, ενώ η δεύτερη αποτελεί μια ειδικότερη πληροφόρηση, όπου τυπώνεται ο κωδικός τον οποίο έχει το προϊόν επιλογής στην βάση δεδομένων του χρήστη και έτσι έχουμε, Enrichment κωδικός προϊόντος. Ο πίνακας 3 αποτελεί ένα μέρος από ένα πραγματικό αρχείο καταγραφής συμβάντων του Colibri, έχουν τροποποιηθεί τα ονόματα των χρηστών για λόγους ανωνυμίας.

Timestamp	Noise	Platform	User	Action
2014-03-05 19:06:41,875	[7] INFO BusinessLogger	[127]	User 1	Connection
2014-03-05 19:06:41,953	[9] INFO BusinessLogger	[127]	User 1	Home
2014-03-05 19:07:33,977	[11] INFO BusinessLogger	[demo4]	User 2	Connection
2014-03-05 19:07:49,583	[29] INFO BusinessLogger	[demo4]	User 2	Home
2014-03-05 19:07:49,730	[9] INFO BusinessLogger	[127]	User 1	Import
2014-03-05 19:07:52,191	[40] INFO BusinessLogger	[127]	User 1	Import /Mensuel
2014-03-12 09:48:08,503	[10] INFO BusinessLogger	[dev2]	User 3	Connection
2014-03-12 09:48:08,549	[33] INFO BusinessLogger	[dev2]	User 3	Home
2014-03-12 09:48:10,187	[5] INFO BusinessLogger	[127]	User 1	History
2014-03-12 09:48:10,226	[7] INFO BusinessLogger	[demo4]	User 2	Enrichment
2014-03-12 09:52:45,370	[7] INFO BusinessLogger	[127]	User 1	Export
2014-03-12 09:52:51,946	[46] INFO BusinessLogger	[demo4]	User 2	Enrichment Product
2014-03-12 09:52:51,946	[22] INFO BusinessLogger	[demo4]	User 2	Enrichment 133
2014-03-12 09:52:55,049	[32] INFO BusinessLogger	[dev2]	User 3	Logout
2014-03-12 09:53:05,783	[6] INFO BusinessLogger	[demo4]	User 2	Logout

Πίνακας 3: Κομμάτι αρχείου καταγραφής συμβάντων από το Colibri.

Όπως μπορούμε να παρατηρήσουμε από τις χρονικές σημάνσεις, όλες οι ενέργειες είναι μοναδικές με μια εξαίρεση σε δυο ενέργειες, Enrichment Product και Enrichment 133, που είναι δυο ενέργειες που έχουν την ακριβώς ίδια χρονική σήμανση. Αυτό συμβαίνει επειδή, αυτές οι δυο ενέργειες αποτελούν το ίδιο γεγονός. Έτσι λοιπόν, πριν από την δημιουργία του μοντέλου διαδικασίας, θα ήταν λάθος να κρατήσουμε και τις δυο αυτές ενέργειες. Για αυτό το λόγο έχουμε προσθέσει ένα σύνολο από ετικέτες στο αρχείο καταγραφής συμβάντων, ώστε το φιλτράρισμα των ενεργειών να γίνει ευκολότερο, το οποίο θα αναλύσουμε σε επόμενο κεφάλαιο.

8.2.2. Αποθήκευση και Συνδυασμός αρχείων καταγραφής συμβάντων.

Για την αποθήκευση των αρχείων καταγραφής συμβάντων χρησιμοποιούμε το Microsoft Azure Storage (MAS), το οποίο είναι μια μηχανή αποθήκευσης που έχει αναπτυχθεί από την Microsoft, για την αποθήκευση και τον συνδυασμό αρχείων καταγραφής συμβάντων. Το MAS επιτρέπει την διαμόρφωση νέων σεναρίων για εφαρμογές και προγράμματα που απαιτούν επεκτάσιμο, ανθεκτικό και πολύ διαθέσιμο αποθηκευτικό χώρο για τα δεδομένα τους. Το MAS εργάζεται σε σύννεφο τεράστιας επεκτασιμότητας έτσι ώστε να μπορεί να αποθηκεύσει και να επεξεργαστεί εκατοντάδες terabytes δεδομένων για την υποστήριξη σεναρίων μεγάλων δεδομένων που απαιτούνται από την επιστημονική έρευνα, τη χρηματοοικονομική ανάλυση και τις εφαρμογές πολυμέσων.

Το Azure Storage υποστηρίζει υπηρεσίες όπως, την αποθήκευση blobs, την αποθήκευση πινάκων, αποθήκευση σειρών και αρχείων.

- Αποθήκευση blobs. Ένα blob μπορεί να είναι οποιοδήποτε είδος κειμένου ή αρχείο δυαδικών δεδομένων, όπως ένα έγγραφο, ένα αρχείο πολυμέσων ή ένα αρχείο που επιτρέπει την εγκατάσταση μιας εφαρμογής.
- Αποθήκευση πινάκων. Η αποθήκευση πινάκων επιτρέπεται με ένα χαρακτηριστικό κλειδί και επιτρέπεται η ταχεία ανάπτυξη και η γρήγορη πρόσβαση σε μεγάλες ποσότητες δεδομένων.
- Η αποθήκευση σειρών. Η αποθήκευση σειρών παρέχει μια αξιόπιστη ανταλλαγή μηνυμάτων για την επεξεργασία της ροής εργασιών και την επικοινωνία μεταξύ των στοιχείων των υπηρεσιών cloud.
- Η αποθήκευση αρχείων. Η αποθήκευση αρχείων, προσφέρει χώρο αποθήκευσης για εφαρμογές που χρησιμοποιούν τα τυποποιημένα πρωτόκολλα.

Στην περίπτωσή μας τα αρχεία καταγραφής συμβάντων είναι αρχεία κειμένου και αποθηκεύονται ως blobs στο MAS. Κάθε blob είναι οργανωμένο σε ένα *δοχείο (container)*

το οποίο παρέχει ένα χρήσιμο τρόπο για την εκχώρηση πολιτικών ασφαλείας για τις ομάδες των αντικειμένων. Ένας λογαριασμός αποθήκευσης μπορεί να περιέχει οποιοδήποτε αριθμό δοχείων και ένα δοχείο αποθήκευσης μπορεί να περιέχει οποιοδήποτε αριθμό blobs. Φανταστείτε λοιπόν, ότι έχουμε ένα δοχείο για κάθε περιβάλλον του Colibri που περιέχει ένα μεγάλο αριθμό από blobs (τα οποία είναι τα αρχεία καταγραφής συμβάντων). Κάθε μέρα, δημιουργείται ένα νέο blob αρχείο σε κάθε δοχείο το οποίο συγκρατεί τις απαραίτητες πληροφορίες για την δραστηριότητα στο Colibri της προηγούμενης μέρας. Έτσι λοιπόν, η πρόκληση σε αυτό το σημείο είναι να βρούμε ένα τρόπο να συνδυάσουμε όλα τα αρχεία καταγραφής συμβάντων σε ένα μεγάλο αρχείο ανάλογα με την χρονική περίοδο και το περιβάλλον του Colibri στο οποίο θέλουμε να εστιάσουμε την ανάλυσή μας.

Η προετοιμασία των δεδομένων συνεπάγεται τη δημιουργία ενός αρχείου κατάλληλου για το ProM και ενός αρχείου κατάλληλου για το εργαλείο στατιστικής ανάλυσης R. Για να γίνει αυτό εφικτό θα πρέπει να ακολουθηθεί μια διαδικασία: πρώτα από όλα, κατεβάζουμε τα δεδομένα από το MAS και οργανώνουμε τα αρχεία ανάλογα με το περιβάλλον του Colibri στο οποίο αυτά αναλογούν. Εν συνεχεία, θα πρέπει να επιλέξουμε τα αρχεία που χρειαζόμαστε ανάλογα με το είδος της ανάλυσης που θέλουμε να κάνουμε. Η επιλογή των αρχείων γίνεται επειδή ουσιαστικά με αυτόματο τρόπο δημιουργείται ένα μεγάλο αρχείο καταγραφής συμβάντων για να γίνει η ανάλυση. Σε γενικές γραμμές οι αναλύσεις που γίνονται εστιάζουν είτε σε συγκεκριμένα περιβάλλοντα ή σε συγκεκριμένες χρονικές περιόδους. Μετά την δημιουργία του μεγάλου αρχείου καταγραφής συμβάντων παρατηρήθηκε πως υπήρχαν πολλές διακυμάνσεις στην σωστή σειρά με την οποία οι καταγεγραμμένες διαδικασίες έχουν αποθηκευτεί. Έτσι λοιπόν, ήταν αναγκαίο να βρεθεί κάποιος μηχανισμός ο οποίος θα οργανώσει το αρχείο μας ανάλογα με τις χρονικές σφραγίδες. *Για αυτό το λόγο γράψαμε ένα νέο πρόγραμμα σε Python* το οποίο μετατρέπει τις χρονικές σφραγίδες σε συντονισμένη παγκόσμια ώρα *(Coordinated Universal Time UTC)* και συγκρίνει τις χρονικές σφραγίδες για να διατάξει το αρχείο καταγραφής συμβάντων. Έτσι λοιπόν, αφού το αρχείο μας είναι σωστά διατεταγμένο, την σκυτάλη παίρνει ο μετατροπέας που επεξεργάζεται και παράγει τα απαραίτητα αρχεία για την χρήση των εργαλείων ProM και R.

8.3. Μέθοδοι

Μια από τις μεγαλύτερες προκλήσεις του τομέα Εξόρυξης Διαδικασιών αποτελεί ο θόρυβος των δεδομένων συμβάντων. Ο θόρυβος όμως είναι δεδομένος όταν έχουμε να κάνουμε με δεδομένα τα οποία έχουν καταγραφεί από το διαδίκτυο ή από πλατφόρμες που χρησιμοποιούν χρήστες όπως ακριβώς στην περίπτωσή μας. Αυτό συμβαίνει επειδή η διαδικασία που ακολουθεί ο κάθε χρήστης διαφέρει και δεν υπάρχει ένα "καλούπι" διαδικασίας που ακολουθούν κατά κανόνα όλοι οι χρήστες. Επιπλέον, ένα ακόμα

χαρακτηριστικό είναι ότι εμφανίζονται πολλοί βρόγχοι επανάληψης, που φαίνονται σαν λούπες στο μοντέλο μας, όπως και πολλές παράλληλες εργασίες. Για παράδειγμα, δίνεται η δυνατότητα στον χρήστη από το Colibri να ελέγξει τις μελλοντικές του πωλήσεις για ένα συγκεκριμένο προϊόν για διάφορες τιμές αυτό θα δημιουργήσει στο μοντέλο μας λούπες αφού ο χρήστης για να το κάνει αυτό ανανεώνει την σελίδα. Ο χρήστης επίσης μπορεί να έχει ανοίξει πολλές σελίδες και να εργάζεται στο Colibri, πράγμα που δεν καταγράφεται στο αρχείο καταγραφής συμβάντων, με αποτέλεσμα να εκτελούνται μια σειρά από άναρχες ενέργειες που κάνει το έργο μας ακόμα πιο δύσκολο. Τέλος, διπλές δραστηριότητες που πιθανόν να εμφανίζονται στο αρχείο μπορεί να έχουν παραχθεί είτε επειδή ο χρήστης χρησιμοποίησε το κουμπί της ανανέωσης ή επειδή κλίκαρε στο ίδιο κουμπί παραπάνω από μία φορές.

Παρόλες τις δυσκολίες αυτές, πιστεύουμε πως μέσω της μελέτης πλοήγησης των χρηστών μπορούν να βελτιωθούν τόσο οι τρόποι σχεδίασης των πλατφόρμων που μελετάμε όσο και οι ίδιες οι πλατφόρμες. Στα επόμενα κεφάλαια παρουσιάζουμε τρόπους απλοποίησης δεδομένων που εμπεριέχουν θόρυβο όπως και τα αποτελέσματα τα οποία καταλήξαμε μετά την μελέτη που κάναμε στο Colibri.

8.3.1. Φιλτράρισμα (Filtering)

Τα αρχεία καταγραφής συμβάντων που παράγονται από το Colibri περιέχουν μια τεράστια ποικιλία ενεργειών αφού το Colibri περιέχει πολλά κουμπιά, μπάρες, φίλτρα αναζήτησης κ.α. Σε κάθε κουμπί της πλατφόρμας αντιστοιχεί ένας κωδικός URL που τα καθιστά μοναδικά, έτσι λοιπόν, κάθε φορά που επιλέγουμε ένα νέο κουμπί ο συνολικός αριθμός των κλάσεων ενεργειών αυξάνεται. Επιπλέον, ο κάθε χρήστης μπορεί να έχει περισσότερα από 1000 προϊόντα αποθηκευμένα στη βάση δεδομένων του. Αυτή η ελαστικότητα και η ποικιλία των ενεργειών κάνει την μελέτη και την ανάλυση όλο και πιο σύνθετη και δύσκολη.

Για αυτό το λόγο, το φιλτράρισμα του αρχείου αποτελεί μια αναγκαία διαδικασία. Οι τρόποι φιλτραρίσματος διακρίνονται σε τρείς κατηγορίες οι οποίες είναι:

- Φιλτράρισμα ενεργειών
- Φιλτράρισμα χρηστών ή πλατφόρμων
- Φιλτράρισμα χρονικής περιόδου

Εμείς κάνουμε χρήση του φιλτραρίσματος ενεργειών για την απλούστευση του αρχείου μας. Το φιλτράρισμα των χρηστών γίνεται σε περιπτώσεις που είτε θέλουμε να εξάγουμε κάποιον χρήστη για να είναι η μελέτη μας πιο αντικειμενική ή για την εστίαση της μελέτης μας σε κάποιον συγκεκριμένο χρήστη, αφού κάποιοι από τους χρήστες είναι οι προγραμματιστές που δουλεύουν για την ανάπτυξη του Colibri και το να τους

συμπεριλάβουμε στην ανάλυση δεν την καταστεί αντικειμενική. Επίσης, εστιάζουμε σε συγκεκριμένους χρήστες για την επίλυση σφαλμάτων του Colibri τα οποία καταγράφονται στο αρχείο καταγραφής συμβάντων και μπορούμε να βρούμε μετά ή πριν από ποιες σελίδες τα σφάλματα αυτά παρουσιάζονται. Το φιλτράρισμα των χρονικών περιόδων χρησιμοποιείται όταν θέλουμε να εστιάσουμε σε κάποια χρονική περίοδο και να δούμε τι ακριβώς συνέβη στην πλατφόρμα την περίοδο αυτή.

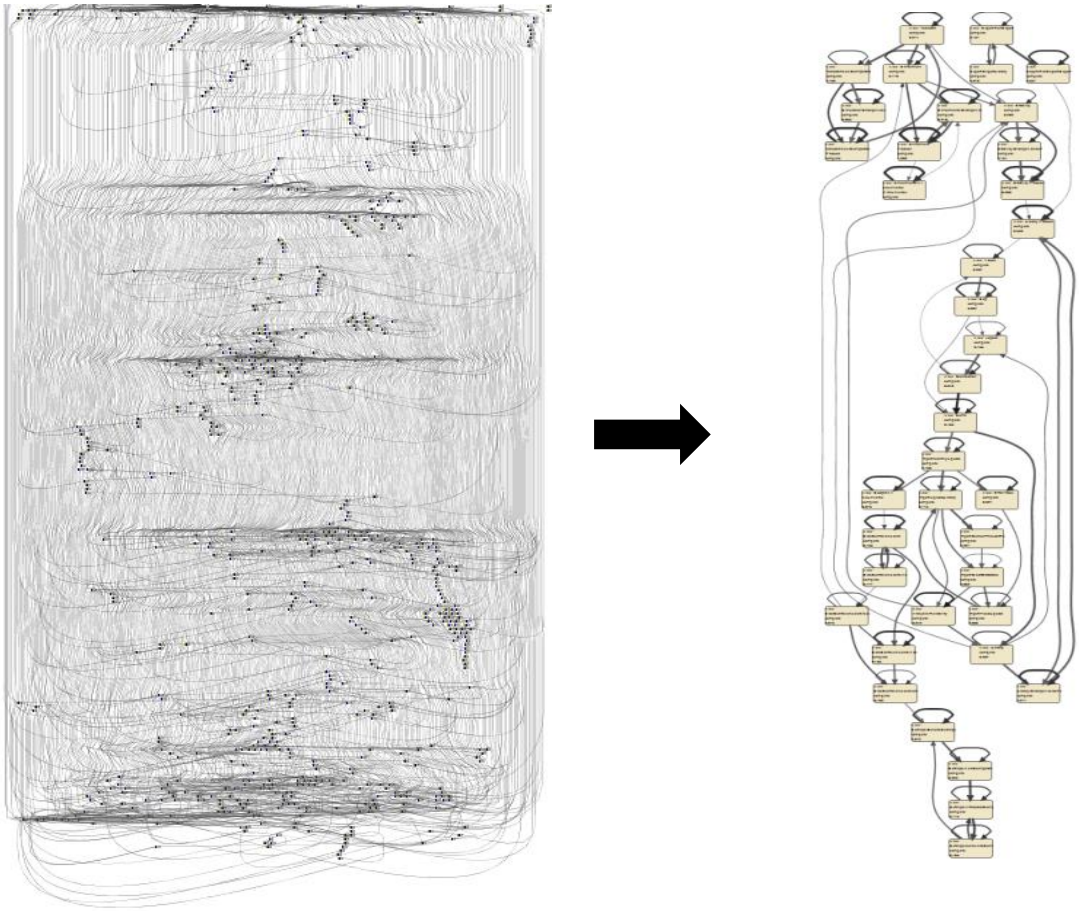
Φιλτράρισμα Ενεργειών:

Για το φιλτράρισμα ενεργειών, κάνουμε χρήση του φίλτρου Simple Heuristics που είναι μία μέθοδος που εμπεριέχεται στο εργαλείο ProM. Αυτή η μέθοδος μας δίνει την δυνατότητα να διαγράψουμε ενέργειες από το αρχείο καταγραφής συμβάντων. Έτσι λοιπόν μπορούμε είτε να διαγράψουμε τις ενέργειες που εμείς επιλέγουμε χειροκίνητα ή να διαγράψουμε τις ενέργειες που έχουν την μικρότερη συχνότητα ανάλογα με το επίπεδο (ποσοστό) φιλτραρίσματος που θα επιλέξουμε.

Για να κάνουμε την διαδικασία του φιλτραρίσματος ενεργειών ευκολότερη επιλέξαμε να κατηγοριοποιήσουμε τις ενέργειες προσθέτοντας ετικέτες στο αρχείο καταγραφής συμβάντων ανάλογα με το είδος των ενεργειών. Έτσι λοιπόν οι ετικέτες που χρησιμοποιούμε είναι οι ακόλουθες:

- View: Περιέχει πληροφορίες σχετικές με τις κύριες σελίδες του Colibri και μας δίνει την κύρια πλοήγηση των χρηστών.
- Action: Περιέχει πληροφορίες σχετικές με ενέργειες που αφορούν την επεξεργασία, τις αλλαγές των τιμών και την αποθήκευση των αλλαγών των προϊόντων.
- Context: Περιέχει πληροφορίες σχετικές με τις ταυτότητες των προϊόντων όπως αυτές αποθηκεύονται στη βάση δεδομένων του κάθε χρήστη. Περιέχει επίσης πληροφορίες σχετικές με τα επίπεδα επιλογής των μπαρών που χρησιμεύουν για την αναζήτηση των προϊόντων και άλλες ρυθμίσεις που χρησιμοποιούνται από τους χρήστες και είναι μοναδικές για τον κάθε έναν από αυτούς όπως τα φίλτρα αναζήτησης κ.α.

Έτσι λοιπόν, κάθε φορά που ένας χρήστης επιλέξει ένα προϊόν από το Colibri παράγονται αυτόματα δύο γραμμές όπως είδαμε στον Πίνακα 3. Η πρώτη γραμμή μας ενημερώνει για την πλοήγηση του χρήστη και την επιλογή ενός προϊόντος (View), ενώ η δεύτερη καταγράφει την ενέργεια όπως και τον κωδικό του προϊόντος όπως αυτό έχει αποθηκευτεί στη βάση δεδομένων του (Context). Όπως καταλαβαίνουμε το δεύτερο μέρος της πληροφορίας δεν αποτελεί σημαντικό για την δημιουργία του μοντέλου αφού εμείς θα πρέπει να καταγράψουμε απλά ότι ο χρήστης έκανε χρήση της λίστας προϊόντων και όχι ποιο προϊόν επέλεξε. Έτσι λοιπόν αυτό που μας μένει να κάνουμε είναι να χρησιμοποιήσουμε τη μέθοδο φιλτραρίσματος Simple Heuristics και να φιλτράρουμε από το αρχείο καταγραφής συμβάντων όλες τις ενέργειες που έχουν ετικέτες Context και κρατάμε όλες τις υπόλοιπες.



Σχήμα 16: Μοντέλα πλοήγησης των χρηστών του Colibri. Στο αριστερό μας χέρι το μοντέλο είναι περίπλοκο. Τέτοιου είδους μοντέλα ονομάζονται Spaghetti-like models.

Στο Σχήμα 16 παρουσιάζονται δυο μοντέλα τα οποία δημιουργήθηκαν από το Colibri. Αξίζει να σημειωθεί ότι αυτά τα μοντέλα δημιουργήθηκαν από ένα αρχείο καταγραφής συμβάντων για όλα τα περιβάλλοντα του Colibri και για τη χρονική περίοδο από τη Δευτέρα 26 Μαΐου 2014 μέχρι την Τετάρτη 15 Οκτωβρίου 2014. Στο αριστερό μοντέλο δεν έχει εφαρμοστεί κανένα είδος φίλτρου ενώ στο δεξί έχουμε αφαιρέσει τις ενέργειες Context όπως αναφέραμε παραπάνω. Όπως μπορούμε να δούμε το αριστερό μας μοντέλο είναι πολύ περίπλοκο σε σημείο να μην να μπορεί να διαβαστεί και να μελετηθεί μοντέλα τέτοιου είδους τα ονομάζουμε spaghetti-like μοντέλα. Από την άλλη μεριά στο δεξί μας χέρι, παρουσιάζεται ένα μοντέλο που είναι πολύ πιο απλό και αξίζει να σημειωθεί πως με την εφαρμογή αυτών των φίλτρων δεν χάνεται ποσό πληροφορίας για τον τρόπο πλοήγησης των χρηστών απλώς καταφέρνουμε να ρίξουμε το επίπεδο πληροφορίας σε ένα επίπεδο που τα παραγόμενα μοντέλα να μπορούν να μελετηθούν.

8.3.2. Ομαδοποίηση (Clustering)

Οι τεχνικές ομαδοποίησης μπορούν να χρησιμοποιηθούν ως ένα βήμα προ-επεξεργασίας και ο σκοπός τους είναι η χρήση των αρχείων καταγραφής συμβάντων που περιέχουν μεγάλες ποσότητες δεδομένων με ένα υψηλό επίπεδο μεταβλητότητας της καταγεγραμμένης συμπεριφοράς. Αντί να χρησιμοποιούμε τεχνικές εξόρυξης ελέγχου ροής σε μεγάλες ποσότητες δεδομένων, οι οποίες παράγουν περίπλοκα μοντέλα, χρησιμοποιούμε τεχνικές ομαδοποίησης που μας παρέχουν τη δυνατότητα να χωρίσουμε τα ίχνη μας σε ομάδες όταν αυτά παρουσιάζουν παρόμοιες μορφές συμπεριφοράς. Έτσι λοιπόν ομαδοποιούμε τα δεδομένα μας σε διάφορα συμπλέγματα ανάλογα με τα χαρακτηριστικά που επιλέγουμε εμείς να επικεντρωθούμε. Μετά την ομαδοποίηση των δεδομένων μας μπορούμε να ανακαλύψουμε απλούστερα μοντέλα διαδικασιών.

Υπάρχει ένα σύνολο από αλγόριθμους ομαδοποίησης που μπορούμε να χρησιμοποιήσουμε για την ομαδοποίηση των δεδομένων. Οι μέθοδοι ομαδοποίησης μπορούν να κατηγοριοποιηθούν σε δυο κατηγορίες: την *ιεραρχική ομαδοποίηση (hierarchical clustering)* και την *διαιρετική ομαδοποίηση (partitional clustering)*. Στην ιεραρχική ομαδοποίηση, οι περιπτώσεις μας οργανώνονται βάση μιας ιεραρχίας που περιγράφει το βαθμό της πληροφορίας και πολλοί αλγόριθμοι έχουν προταθεί για αυτή τη μέθοδο. Η διαιρετική ομαδοποίηση, από τη άλλη μεριά, απλά δημιουργεί τις ομάδες των δεδομένων και εν συνεχεία η κάθε περίπτωση εμπίπτει σε μία ομάδα. Με αυτόν τον τρόπο, λαμβάνονται λιγότερες πληροφορίες όμως η ικανότητα να εργαζόμαστε με μεγάλο αριθμό περιπτώσεων βελτιώνεται.

Όπως είδαμε και πριν, το πρόβλημα της διαιρετικής ομαδοποίησης μπορεί να θεωρηθεί και ως ένα πρόβλημα βελτιστοποίησης. Τα θέματα τα οποία παρουσιάζονται είναι πρώτα πως μπορούμε να ορίσουμε τις μεταβλητές απόφασης και τις αντικειμενικές συναρτήσεις, τα οποία δεν έχουν μια γενική απάντηση. Οι βασικοί στόχοι τις ομαδοποίησης λοιπόν είναι η ελαχιστοποίηση του αριθμού των κλάσεων των ενεργειών σε κάθε συστάδα και η μεγιστοποίηση της διαφορετικότητας μεταξύ των ομάδων (διαχωρισμός), ή σε κάποιο συνδυασμό των δύο μέτρων.

Στην περίπτωσή μας, ο κύριος στόχος της ομαδοποίησης των δεδομένων είναι να μειωθεί το επίπεδο πολυπλοκότητας των πληροφοριών ώστε στη συνέχεια το παραγόμενο μοντέλο μας να είναι απλούστερο και ευανάγνωστο. Το κύριο πρόβλημα όμως είναι ότι το αρχείο καταγραφής συμβάντων περιέχει μια τεράστια ποικιλία διαφορετικών ενεργειών. Από την άλλη πλευρά η εφαρμογή τεχνικών ομαδοποίησης αποτελεί ένα εύκολο έργο μιας και τα δεδομένα μας είναι κατηγορικά. Όπως είδαμε και προηγουμένως η δομή των μεταβλητών μας είναι όμοια με εκείνη των URL των ιστοσελίδων και καθένα από τα βασικά παράθυρά μας περιέχει άλλα υπο-παράθυρα. Για αυτό το λόγο δεν είναι αναγκαίο να χρησιμοποιήσουμε κάποιον αλγόριθμο ιεραρχικής ομαδοποίησης καθώς θα

δημιουργήσουμε τις συστάδες σύμφωνα με το αρχικό μέρος των ετικετών όπου αποτελούν και τα βασικά παράθυρα του Colibri.

Έτσι λοιπόν μετά την ομαδοποίηση των δεδομένων μας καταφέραμε να έχουμε εννέα κλάσεις ενεργειών από συνολικά 13 345 κλάσεις που είχαμε. Όπως μπορούμε να καταλάβουμε τα μοντέλα που θα παράγουμε τώρα θα είναι πολύ πιο απλά και ευανάγνωστα σε σχέση με τα spaghetti-like μοντέλα που παράγαμε προηγουμένως. Οι εννέα κλάσεις ενεργειών που έχουμε τώρα είναι οι:

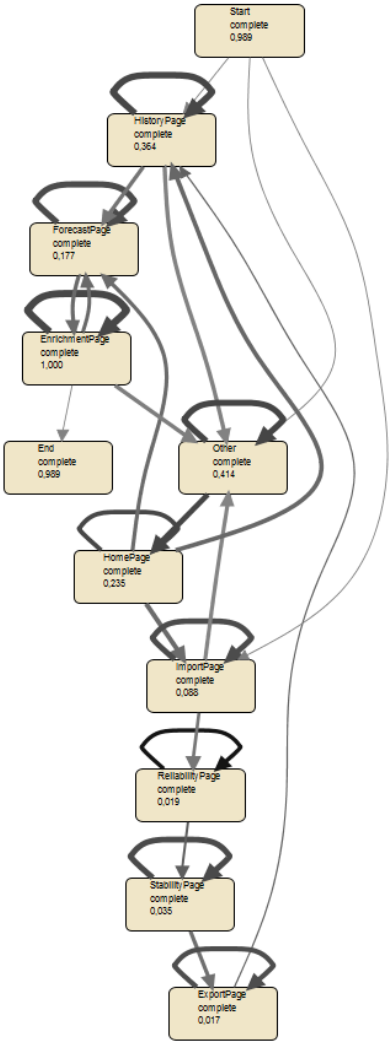
1. History Page
2. Enrichments Page
3. Forecast Page
4. Import Page
5. Export Page
6. Errors
7. Reliability Page
8. Stability Page
9. Other

8.4. Αποτελέσματα

Μέσω αυτών των τεχνικών λοιπόν καταφέραμε επιτυχώς να αποτυπώσουμε τον τρόπο με τον οποίο οι χρήστες πλοηγούνται στο Colibri. Η μελέτη μπορεί να εστιαστεί είτε σε μια συγκεκριμένη χρονική περίοδο ή σε κάποια συγκεκριμένη ομάδα χρηστών για να μην είναι σύνθετη. Εμείς όμως αποφασίσαμε να παρουσιάσουμε τα αποτελέσματα μιας μεγάλης μελέτης, έτσι το αρχείο καταγραφής συμβάντων που θα αναλύσουμε σε αυτό το σημείο παίρνει μέρος από τη Δευτέρα 26 Μαΐου 2014 στις 12:32:28 έως τη Τετάρτη 15 Οκτωβρίου 2014 στις 12:57:26. Στο αρχείο αυτό παρατηρούνται 54 305 διαφορετικές ενέργειες από συνολικά 70 διαφορετικούς χρήστες το μοντέλο το οποίο παράγεται με την βοήθεια του αλγόριθμου Fuzzy miner φαίνεται στο Σχήμα 17.

Από το σχήμα μπορούμε να αντλήσουμε πολύ σημαντικές πληροφορίες σχετικά με τον τρόπο πλοήγησης των χρηστών και των κλάσεων των ενεργειών που έχουν την μεγαλύτερη συχνότητα. Έτσι λοιπόν παρατηρούμε πως η σελίδα με τις περισσότερες επισκέψεις είναι η σελίδα εμπλουτισμού (Enrichment Page). Αμέσως μετά ακολουθεί η σελίδα παρελθοντικών πωλήσεων (History Page) κλπ. Αξίζει να σημειωθεί πως θα ήταν πολύ καλό να μπορούσαμε να δούμε τον χρόνο στον οποίο οι χρήστες χρησιμοποιούν το κάθε παράθυρο αυτή η πληροφορία όμως για να παραχθεί κρύβει από πίσω τις πολλές δυσκολίες. Ή δυσκολίες οι οποίες παρουσιάζονται έχουν να κάνουν με τον τρόπο πληροφόρησης του αρχείου καταγραφής συμβάντων για την αποσύνδεση κάποιου χρήστη από την πλατφόρμα. Ένας χρήστης είναι δυνατό να αποσυνδεθεί και να πάψει να δουλεύει

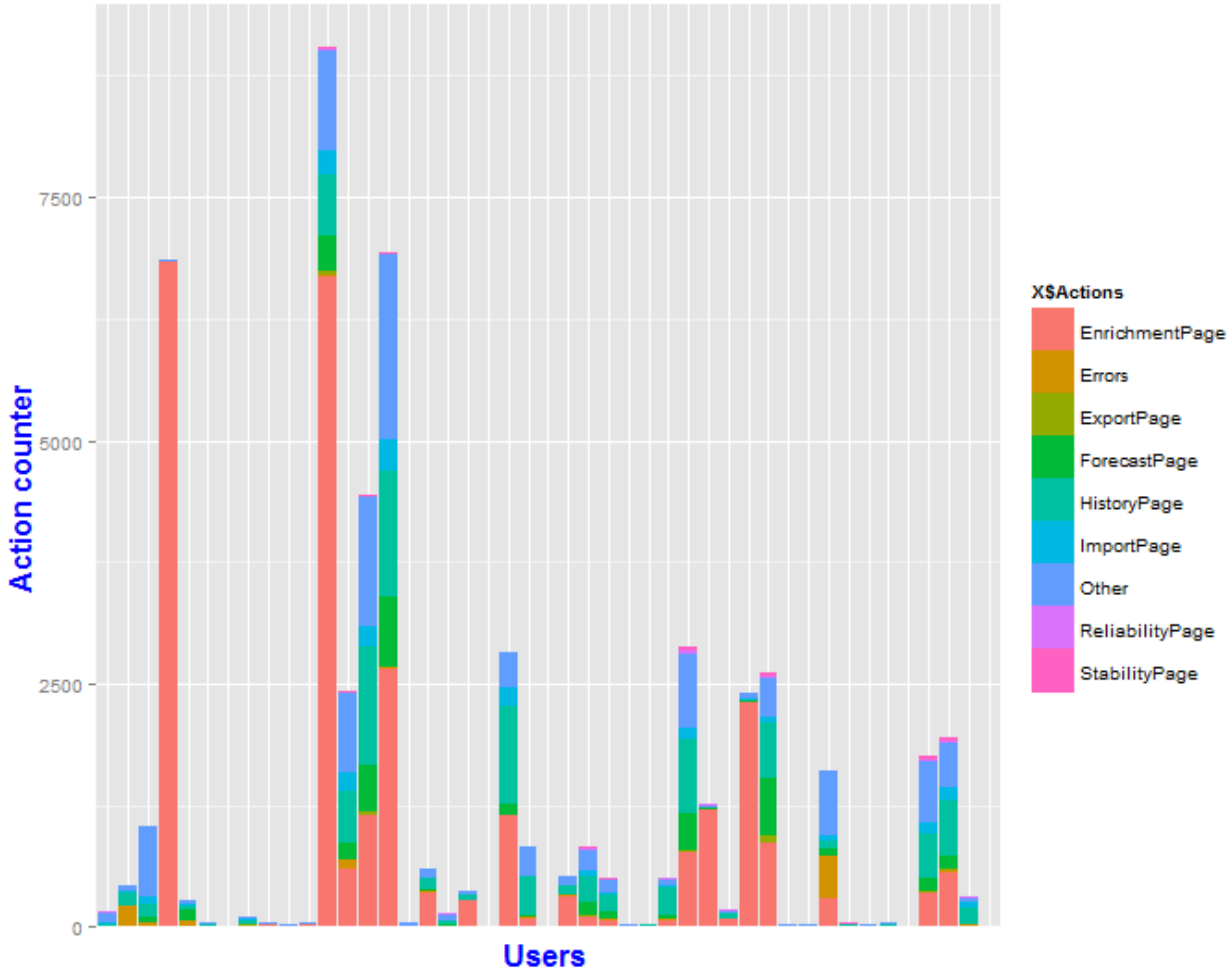
στην πλατφόρμα όχι μόνο στην περίπτωση όπου πατάει το κουμπί της αποσύνδεσης από την πλατφόρμα.



Σχήμα 17: Μοντέλο που παράχθηκε από τον αλγόριθμο Fuzzy miner με ομαδοποιημένα δεδομένα.

Είναι πιθανό για παράδειγμα να έχει την πλατφόρμα ανοιχτή και να ασχολείται για μεγάλη περίοδο με άλλα πράγματα στον υπολογιστή του. Σε αυτή την περίπτωση η μέτρηση του χρόνου παραμονής σε μια σελίδα του Colibri δεν θα ήταν αντικειμενική. Μια λύση για αυτό το πρόβλημα θα ήταν ο κάθε χρήστης να αποσυνδέεται αυτόματα μετά από ένα χρονικό διάστημα αδράνειας της πλατφόρμας για να ενημερώνεται το αρχείο μας σωστά. Μια άλλη λύση θα ήταν να ενημερώνουμε μόνο το αρχείο σχετικά με το ότι ο χρήστης είναι ανενεργός, δηλαδή αν το ο χρήστης είναι ανενεργός για n χρόνο να τυπώνεται μια ψευδής ενέργεια αποσύνδεσης του χρήστη. Σε αυτή την περίπτωση όμως θα πρέπει να τυπώνουμε άλλη μία ψευδή ετικέτα σύνδεσης του χρήστη όταν επιστρέφει και χρησιμοποιεί πάλι το Colibri. Μια άλλη περίπτωση είναι ο χρήστης αφού έχει τελειώσει με τις ενέργειες που έχει

να κάνει στην πλατφόρμα να τερματίσει την διαδικασία κλείνοντας το πρόγραμμα περιήγησης χωρίς να κάνει απαραίτητα αποσύνδεση. Σε αυτή την περίπτωση πάλι όπως καταλαβαίνετε υπάρχει πάλι πρόβλημα αφού το γεγονός ότι ο χρήστης δεν χρησιμοποιεί την πλατφόρμα πάλι δεν καταγράφεται. Η λύση σε αυτό θα ήταν να χρησιμοποιήσουμε υποδοχές (sockets) στο πρόγραμμα μας. Οι υποδοχές μας ενημερώνουν σχετικά με το κλείσιμο των προγραμμάτων περιήγησης. Έτσι λοιπόν θα μπορούσαμε να ενημερωνόμαστε με αυτό τον τρόπο και να παράγουμε τις απαραίτητες γραμμές ενεργειών στο αρχείο καταγραφής συμβάντων. Αυτή η εργασία δεν έγινε επειδή η διαδικασία παραγωγής των αρχείων καταγραφής συμβάντων είναι πολύ χρονοβόρα και θα έπρεπε να περιμένουμε άλλο τόσο χρόνο να παραχθούν τα καινούρια αρχεία καταγραφής συμβάντων για να μπορέσουμε να τα επεξεργαστούμε. Σε κάθε περίπτωση θα πρέπει να προσέξουμε τον τατρόπο με το οποίο ορίζουμε τις περιπτώσεις μας. Θα μπορούσαμε να ορίσουμε σαν κάθε περίπτωση από τη στιγμή σύνδεσης ενός χρήστη έως την στιγμή αποσύνδεσής του όπως ακριβώς εφαρμόζεται στο [19].



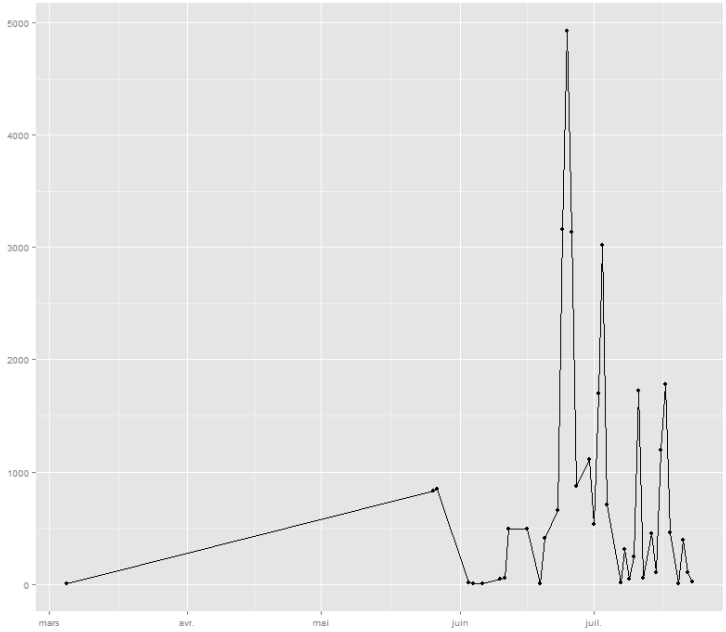
Σχήμα 18: Ιστόγραμμα συχνοτήτων των βασικών παραθύρων του Colibri.

Η πλοήγηση του κάθε χρήστη σε μια πλατφόρμα όμως δεν μπορεί να είναι τυποποιημένη ή η ίδια ανάμεσα σε διαφορετικούς χρήστες. Αυτό συμβαίνει είτε επειδή ο κάθε χρήστης έχει άλλες απαιτήσεις από την πλατφόρμα ή επειδή ο ρόλος του κάθε χρήστη στην πλατφόρμα δεν είναι ο ίδιος. Το Colibri από την μεριά του απασχολεί διάφορα είδη χρηστών όπου το κάθε είδος έχει και διαφορετικές απαιτήσεις. Έτσι τα διαφορετικά προφίλ χρηστών που απασχολεί το Colibri είναι οι νέοι χρήστες, οι μη-ειδικοί και οι ειδικοί. Οι ειδικοί με την σειρά τους χωρίζονται σε άλλα προφίλ τα οποία ανάλογα με την σελίδα με την οποία επικεντρώνονται. Έτσι τα τέσσερα διαφορετικά προφίλ ειδικών που μπορεί να συναντήσουμε στο Colibri είναι:

- Ο ανώτερος προγνώστης
- Ο συντελεστής
- Ο διαχειριστής
- Ο εξαγωγέας

Για να αναγνωρίσουμε αυτά τα διαφορετικά είδη χρηστών αποφασίσαμε να χρησιμοποιήσουμε το εργαλείο R για να τυπώσουμε ιστογράμματα που μας ενημερώνουν για τις συχνότητες χρήσης της κάθε σελίδας όπως φαίνεται στο Σχήμα 18.

Όπως βλέπουμε στο σχήμα η κάθε μπάρα στο ιστόγραμμα αντιπροσωπεύει τον κάθε χρήστη του Colibri. Η κάθε σελίδα από την μεριά της αντιπροσωπεύεται στο διάγραμμα από ένα χρώμα. Με αυτό τον τρόπο μπορούμε λοιπόν να ενημερωνόμαστε άμεσα για τα είδη των χρηστών που έχουμε και που χρησιμοποιούν την πλατφόρμα μας. Αφού είναι προφανές πως ένας χρήστης που αποτελεί ανώτερος προγνώστης κάνει χρήση των σελίδων πρόγνωσης και εμπλουτισμού.



Σχήμα 19: Δραστηριότητα της πλατφόρμας. Στον άξονα χ απεικονίζεται ο χρόνος και στον άξονα y ο αριθμός των ενεργειών όλων των χρηστών του Colibri.

Τέλος, με την βοήθεια των συχνοτήτων, καταφέραμε να βρούμε τα κουμπιά τα οποία χρησιμοποιούνται συχνότερα. Παρατηρήσαμε ότι οι λίστες των προϊόντων χρησιμοποιούνται σε πολύ μεγάλο βαθμό, σχεδόν 40% στο σύνολο όλων των δραστηριοτήτων. Έτσι αποφασίσαμε με τους προγραμματιστές να προσθέσουμε επιπλέον επιλογές σε αυτή τη λίστα για να βοηθήσουμε τους χρήστες. Οι επιλογές που προσθέσαμε αποτελούν τις βασικές επιλογές που υπάρχουν σε πολλές λίστες, όπως προϊόντα με κορυφαία επισκεψιμότητα, κατηγορίες όμοιών προϊόντων κ.α. Με την βοήθεια του R είχαμε ενημέρωση για την δραστηριότητα του Colibri στη διάρκεια του χρόνου(Σχήμα 19). Παρατηρούμε σε αυτό το διάγραμμα ότι στον άξονα x αναπαριστάται ο χρόνος ενώ στον άξονα y η συχνότητα των δραστηριοτήτων στο Colibri. Παρατηρούμε ότι το διάγραμμα αναφέρεται για δεδομένα μέχρι τις αρχές Αυγούστου και είναι εμφανές ότι η δραστηριότητα του Colibri αρχικά είχε μια σταδιακή αύξηση και στα τέλη Ιουλίου είχε μια ραγδαία. Στη συνέχεια το διάγραμμα δραστηριότητας έχει μια φθίνουσα πορεία και αυτό είναι λογικό μιας και οι χρήστες τέτοιων προγραμμάτων αυτές τις περιόδους λείπουν για διακοπές.

9. Αναλύσεις άλλων δεδομένων.

9.1. Διαδικασία εγκρίσεων καταναλωτικών δανείων

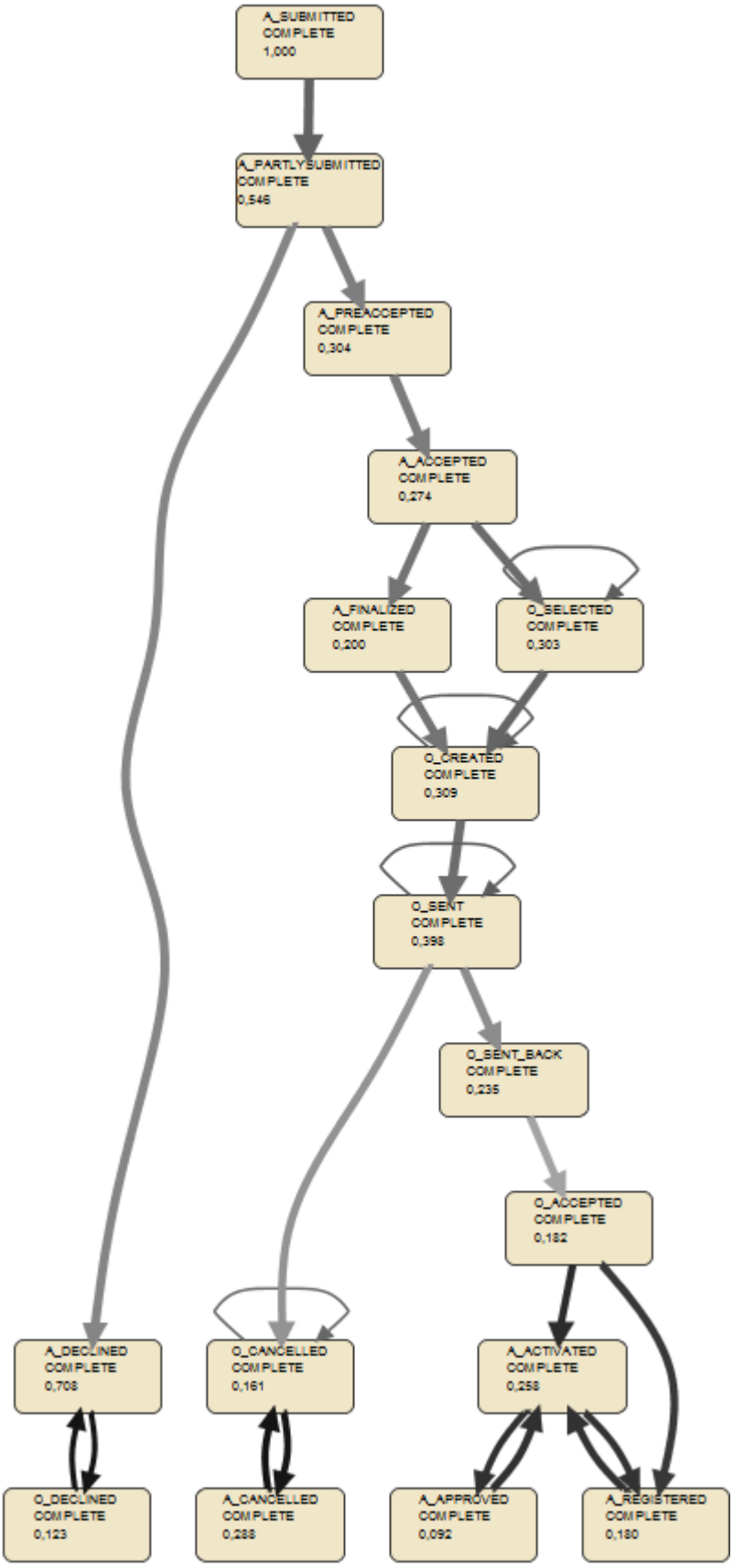
Η ακόλουθη μελέτη γίνεται σε δεδομένα (πηγή: http://www.win.tue.nl) από μία τράπεζα της Ολλανδίας σχετικά με την διαδικασία παροχής δανείων. Τα δεδομένα ανακτήθηκαν από το δεύτερο διεθνή διαγωνισμό επιχειρησιακών διαδικασιών που έγινε το 2012 (Second International Business Process Intelligence Challenge 2012). Το αρχείο περιέχει 262 200 ενέργειες και 13 087 διαφορετικές περιπτώσεις. Η ανάλυση μας έγινε με την χρήση του εργαλείου ProM. Η μελέτη αυτή έχει επικεντρωθεί αρχικά στην κατανόηση των δεδομένων, αφού τα δεδομένα αυτά δεν έχουν παραχθεί με δικές μας μεθόδους. Στην κατανόηση της διαδικασίας, στην εξεύρεση των κρίσιμων σημείων της διαδικασίας όπως και τα σημεία λήψεως αποφάσεων. Επίσης, θα επικεντρωθούμε και στις χρονοετικέτες των ενεργειών ώστε να βρούμε περιπτώσεις *συμφόρησης (bottlenecks)* της διαδικασίας σημεία δηλαδή όπου η διαδικασία καθυστερεί.

Τύποι κλάσεων	**Περιγραφή**
“A_” Ενέργειες αιτήσεων	Αναφέρονται σε καταστάσεις της αίτησης. Ο πελάτης αρχικά συμπληρώνει την αίτηση. Η τράπεζα στη συνέχεια, ολοκληρώνει την διαδικασία συμπληρώνοντας κάποιες επιπλέον πληροφορίες. Αρχική υποβολή της αίτησης: - A_SUBMITED / A_PARTLYSUBMITED Η αίτηση γίνεται αποδεκτή, αλλά απαιτούνται πρόσθετες πληροφορίες: - A_PREACCEPTED Η αίτηση γίνεται αποδεκτή: - A_ACCEPTED Η αίτηση οριστικοποιείται: - A_FINALIZED Τελική κατάσταση επιτυχημένων αιτήσεων: - A_APPROVED / A_REGISTERED / A_ACTIVATED Τελική κατάσταση αποτυχημένων αιτήσεων: - A_CANCELLED / A_DECLINED
“O_” Ενέργειες προσφοράς	Αναφέρεται σε καταστάσεις της προσφοράς της τράπεζας. Ο αιτών επιλέγει να λάβει την προσφορά: - O_SELECTED Η προσφορά καταρτίζεται και διαβιβάζεται στον/ην αιτούσα: - O_PREPARED / O_SENT Απάντηση του αιτούντα για την προσφορά: - O_SENTBACK

	Τέλος κατάστασης επιτυχούς προσφοράς: - O_ACCEPTED Τέλος κατάστασης ανεπιτυχούς προσφοράς: - O_CANCELLED / O_DECLINED
“W_” Ενέργειες εργασίας	Αναφέρεται σε καταστάσεις εργασίας που συμβαίνουν κατά τη διάρκεια της διαδικασίας έγκρισης. Δίνοντας συνέχεια στις αρχικά ελλιπείς αιτήσεις: - W_afhandelen aanvraag Ολοκλήρωση αποδεχόμενων αιτήσεων: - W_Completeren aanvraag Παρακολούθηση για την διαβίβαση ειδικών προσφορών προς τους αιτούντες: - W_Nabellen offertes Αξιολόγηση της αίτησης: - W_Valideren aanvraag Αναζήτηση πρόσθετων πληροφοριών κατά τη διάρκεια της αξιολόγησης: - W_Nabellen incomplete dossiers Διερεύνηση ύποπτων περιπτώσεων: - W_Beoordelen fraude Τροποποίηση εγκεκριμένων συμβάσεων - W_Wjzigen contractgegevens

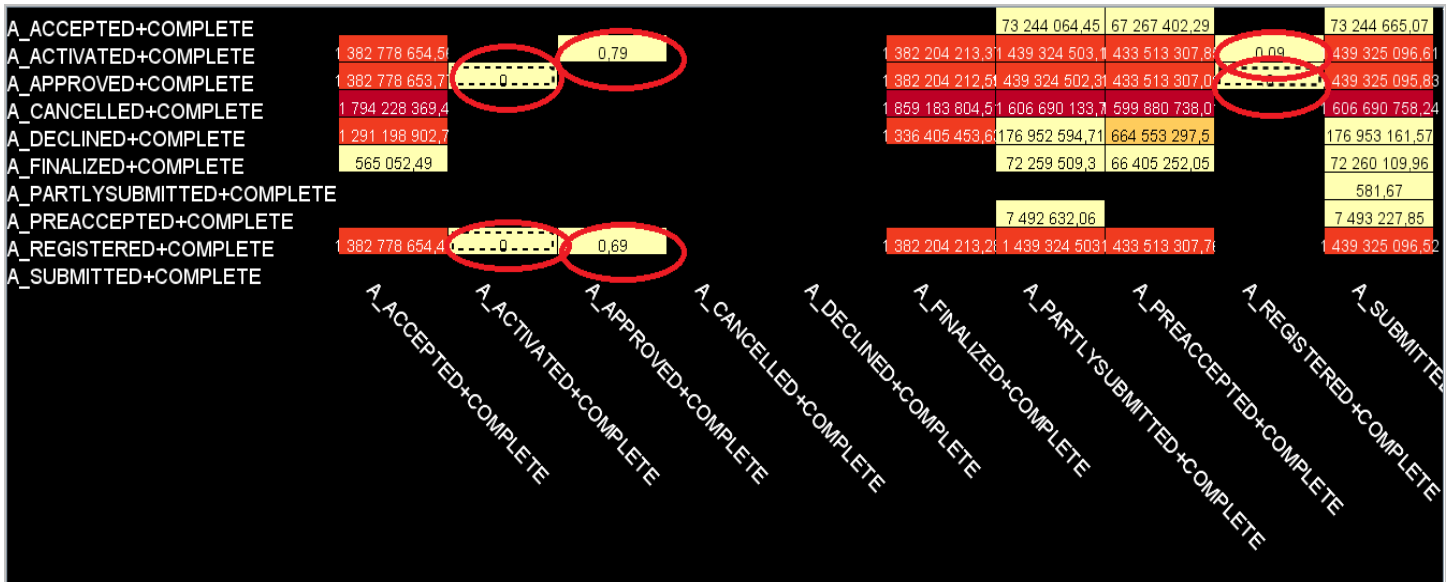
Πίνακας 4: Κατηγοριοποίηση ενεργειών του αρχείου καταγραφής συμβάντων. [20]

Έτσι λοιπόν, τα δεδομένα μας καταγράφουν τη διαδικασία υποβολής αιτήσεων δανείων, οι οποίες είναι 13 087, στην περίοδο έξι μηνών από τον Οκτώβριο του 2011 έως το Μάρτιο του 2012. Η διαδικασία ξεκινά με την υποβολή της αίτησης των πελατών και τελειώνει με την τελική απόφαση που είτε θα εγκριθεί ή θα απορριφθεί το αίτημα. Συνολικά έχουμε 36 κλάσεις ενεργειών από τις οποίες η διαδικασία πάντοτε ξεκινάμε την ενέργεια A_SUBMITED COMPLETE που μας ενημερώνει για την υποβολή της αίτησης και 13 από τις 36 κλάσεις αποτελούν τις τελικές κλάσεις ενεργειών. Οι κλάσεις ενεργειών μπορούν να χωριστούν σε τρείς βασικές κατηγορίες, όπως ακριβώς φαίνεται και στον Πίνακα 4. Μετά την μελέτη και την κατανόηση των κατηγοριών των ενεργειών, αποφασίσαμε να φιλτράρουμε τα δεδομένα μας και να κρατήσουμε τα δεδομένα μόνο από τις δύο πρώτες κατηγορίες,”A_””O_”, που περιγράφουν την διαδικασία χωρίς να χρειάζεται η τρίτη κατηγορία, “W_”. Μετά το φιλτράρισμα των δεδομένων μας, το αρχείο μας περιέχει δεκαεπτά συνολικά κλάσεις ενεργειών με μια αρχική κλάση και δώδεκα τελικές κλάσεις ενεργειών. Το μοντέλο το οποίο εξηγεί την διαδικασία φαίνεται στο Σχήμα 20. Στο μοντέλο μπορούμε να παρατηρήσουμε πως κάποιες ενέργειες είναι γραμμικές και επιπλέον ο χρόνος εκτέλεσης μεταξύ αυτών των ενεργειών είναι υπερβολικά μικρός σχετικά με το μέσο χρόνο εκτέλεσης των ενεργειών.



Σχήμα 20: Διαδικασία αιτήσεων σχετικά με την χορήγηση δανείων.

Για να είμαστε πιο συγκεκριμένοι, οι μέγιστοι χρόνοι μεταξύ των ενεργειών A_REGISTERED, A_ACTIVATED, A_APPROUVED είναι 957ms Σχήμα 21. Όπως επίσης ο μέσος χρόνος για να μεταβούμε από την αρχική ενέργεια A_SUBMITED στην αμέσως επόμενη A_PARTLYSUBMITED, που είναι και αυτές γραμμικές, είναι ίσος με 581.67ms, Σχήμα 21. Για τους δύο αυτούς λόγους λοιπόν, πρώτων ότι οι ενέργειες είναι σειριακές και δεύτερων επειδή ο χρόνος εκτέλεσής από την μία στην άλλη είναι πολύ μικρός, μπορούμε να ομαδοποιήσουμε τις ενέργειες αυτές χωρίς να χαθεί πληροφορία σχετικά με το σύνολο της διαδικασίας. Με αυτό τον τρόπο μπορούμε να καταλήξουμε σε ένα απλούστερο μοντέλο που περιγράφει την διαδικασία.

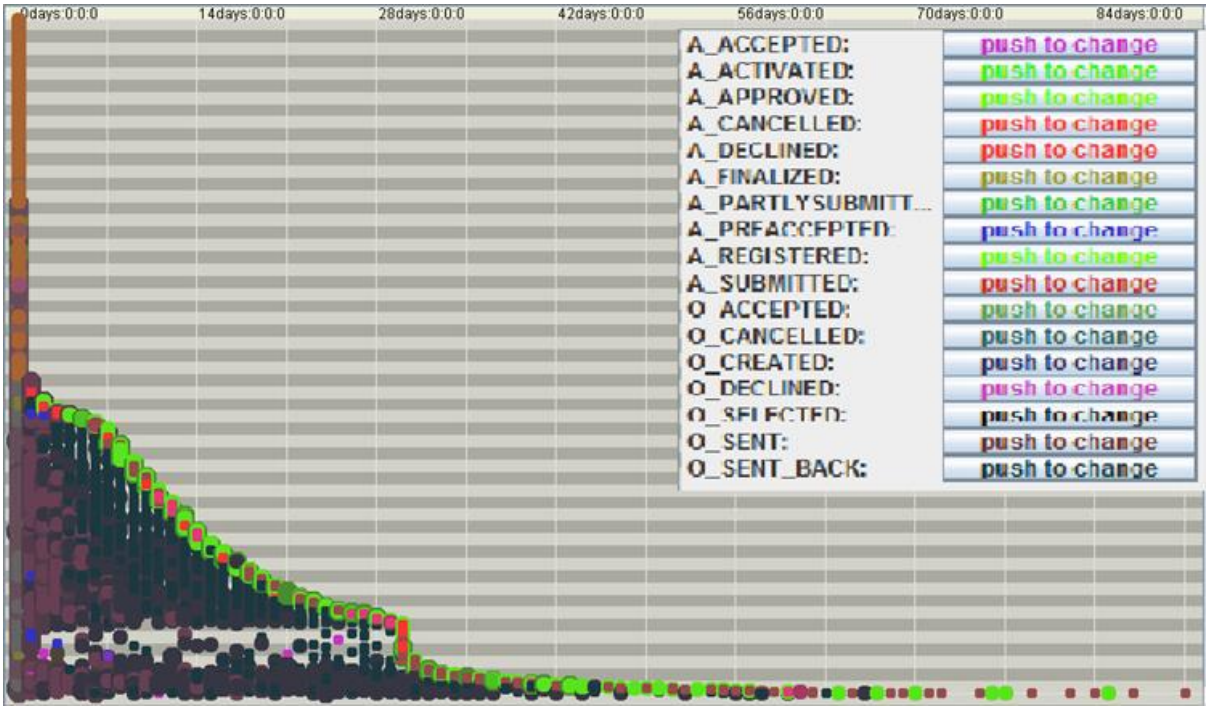


Σχήμα 21: Απαιτούμενοι χρόνοι μεταξύ των εκτελέσεων των ενεργειών.

Παρατηρώντας το δίκτυο λοιπόν, παρατηρούμε πως στο σύνολο της διαδικασίας υπάρχουν δύο σημεία λήψης αποφάσεως. Το πρώτο παρουσιάζεται από την αρχή και όλας της διαδικασίας. Σε αυτό το σημείο έχουμε δυο μονοπάτια που μας καλούν να αποφασίσουμε για το ποιο κομμάτι της διαδικασίας θα ακολουθήσουμε. Είτε λοιπόν η αίτηση θα απορριφθεί, θα ακολουθήσουμε το αριστερό κομμάτι του μοντέλου μετά την ενέργεια A_PARTLYSUBMITED, ή θα ακολουθήσουμε το δεξί κομμάτι του μοντέλου και θα αποδεχτούμε την αίτηση και στην συνέχεια θα στείλουμε μια προσφορά στον πελάτη. Το δεύτερο σημείο στο οποίο καλούμαστε να πάρουμε μια απόφαση είναι το σημείο μετα την αποστολή της προσφοράς στον πελάτη, δηλαδή το σημείο μετα την εκτέλεση της ενέργειας O_SENT. Σε αυτό το σημείο η αίτηση και η προσφορά γίνονται δεκτές και πελάτης με τράπεζα έρχονται σε συμφωνία ή η αίτηση και η προσφορά απορρίπτονται.

Στη συνέχεια αφού καταλάβαμε σε βάθος την διαδικασία και τα κρίσιμα σημεία της, αποφασίσαμε να επικεντρωθούμε στις συχνότητες των γεγονότων. Καταλήξαμε στο συμπέρασμα ότι μέχρι και τις 14 Μαρτίου του 2012, που είναι και η τελευταία μέρα που

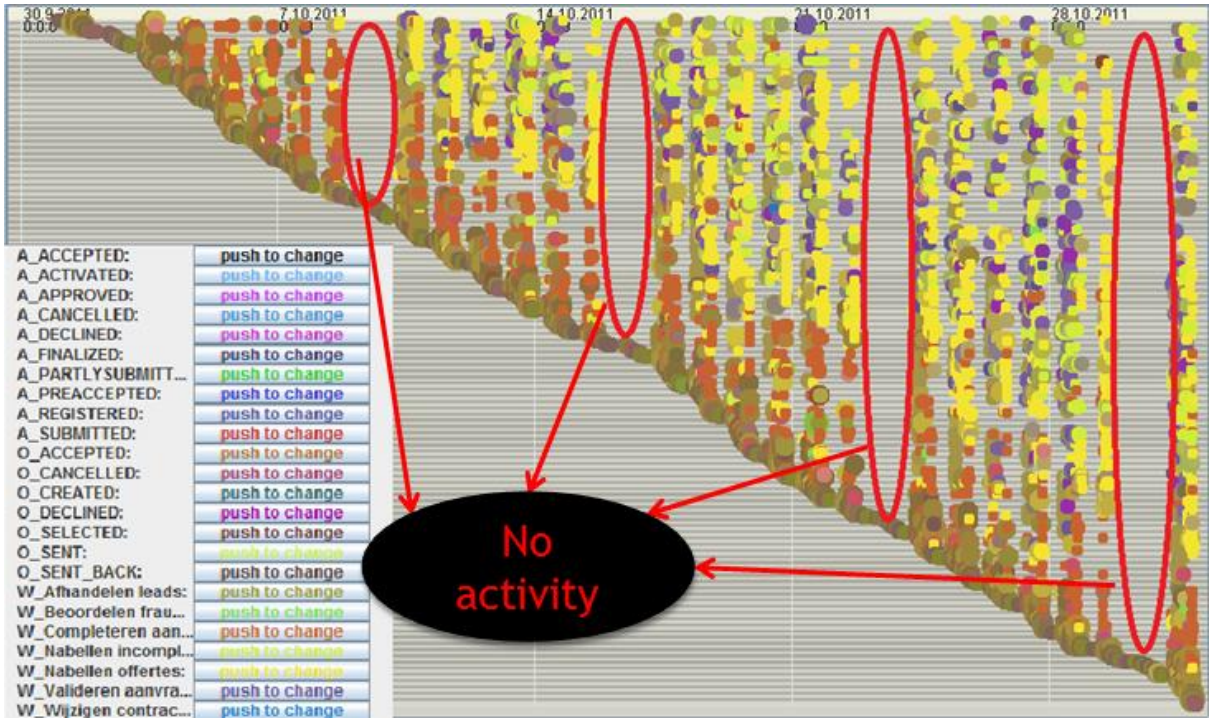
είναι καταγεγραμμένη στο αρχείο μας, υπήρχαν 399 αιτήσεις που εκκρεμούσαν. Καταλήξαμε σε αυτό συμπέρασμα εφαρμόζοντας μια απλή αφαίρεση. Οι αιτήσεις συνολικά ήταν 13 087, υπάρχουν 7 635 αιτήσεις που υπορράφθηκαν από το πρώτο βήμα (A_DECLINED), 2 807 αιτήσεις οι οποίες ακυρώθηκαν (A_CANCELED) και τέλος, 2 246 εγκεκριμένες αιτήσεις. Έτσι λοιπόν, $13087 - 7635 - 2807 - 2246 = 399$ αιτήσεις που εκκρεμούν.



Σχήμα 22: Διάστικτο διάγραμμα σε συνάρτηση με τον σχετικό χρόνο.

Τέλος, χρησιμοποιούμε από το εργαλείο ProM το *διάστικτο διάγραμμα (dotted chart)*Σχήμα 22,Σχήμα 23. Στο Σχήμα 22, απεικονίζονται οι ενέργειες σε συνάρτηση με τον σχετικό χρόνο. Δηλαδή αυτή η απεικόνιση μας δίνει τη δυνατότητα να αναγνωρίσουμε τις ενέργειες οι οποίες χρειάζονται περισσότερο χρόνο. Όπως μπορούμε να παρατηρήσουμε, το κάθε γεγονός απεικονίζεται με διαφορετικό χρόνο στο διάγραμμα και με αυτό τον τρόπο μας βοηθά να κατανοήσουμε σε ποιο γεγονός αναφέρεται το κάθε σημείο στο διάγραμμα. Ο άξονας χ είναι ο χρόνος και σαν μηδενικό σημείο έχουμε το σημείο στο οποίο ξεκινά η κάθε ενέργεια. Οι ενέργειες οι οποίες είναι οι πιο χρονοβόρες στο σύνολο της διαδικασίας είναι αυτές που παρουσιάζονται δεξιότερα ως προς τον άξονα χ και αυτές είναι: A_CANCELLED, O_CANCELLED, O_ACCEPTED, A_APPROVED, A_ACTIVATED, A_REGISTERED και O_SENT. Στο Σχήμα 23, ο άξονας χ αντιπροσωπεύει πάλι τον χρόνο άλλα αυτή τη φορά τον πραγματικό χρόνο και στον άξονα y έχουμε τις διάφορες περιπτώσεις, δηλαδή κάθε γραμμή αντιπροσωπεύει και μια από τις αιτήσεις. Παρατηρούμε πως η γραμμή έναρξης των περιπτώσεων είναι διαγώνια και αυτό είναι φυσιολογικό αφού οι αιτήσεις γίνονται σε βάθος χρόνου και δεν γίνονται όλες την ίδια μέρα. Επίσης, από αυτό το διάγραμμα παρατηρήθηκε πως κάθε επτά μέρες υπάρχει μία περίοδος χωρίς δραστηριότητα και αυτό

όμως είναι φυσιολογικό μιας και τα κενά αυτά στην δραστηριότητα είναι τα Σαββατοκύριακα όπου η τράπεζα δεν λειτουργεί.

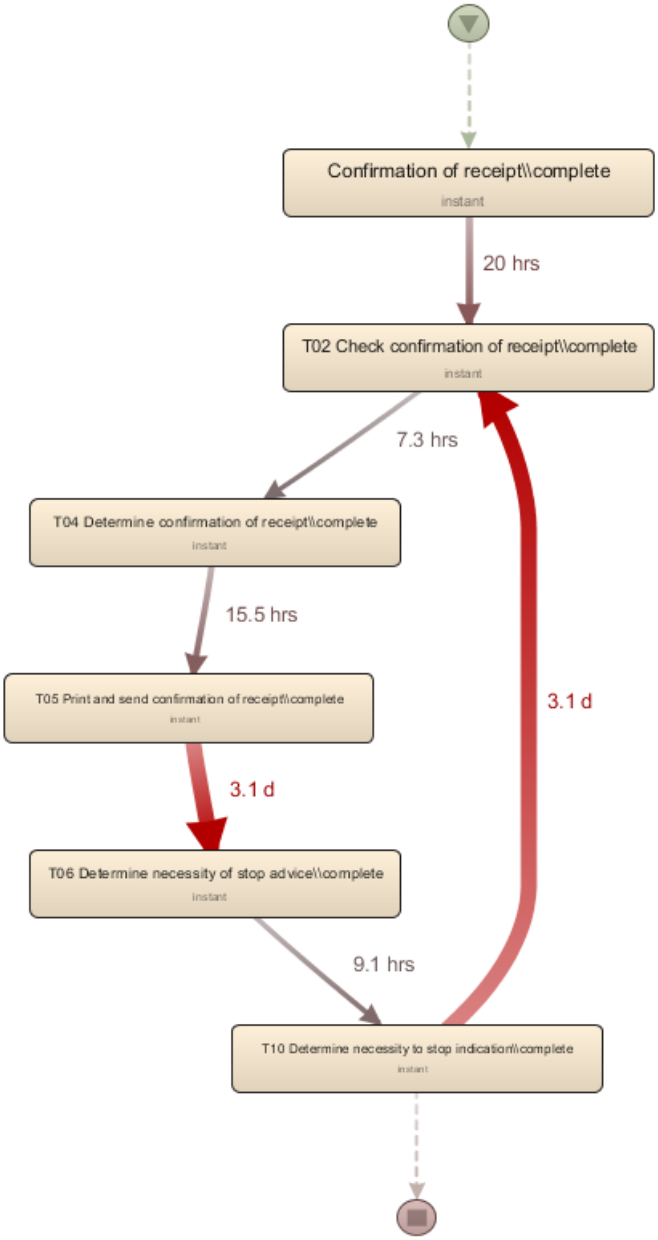


Σχήμα 23:Διάστικτο διάγραμμα σε συνάρτηση με τον πραγματικό χρόνο.

9.2. Διαδικασία παραλαβής περιβαλλοντικής άδειας

Η παρακάτω μελέτη γίνεται με δεδομένα τα οποία έχουν ληφθεί από ένα ανώνυμο δήμο της Ολλανδίας (πηγή: http://data.3tu.nl). Σκοπός της ανάλυσης είναι να αναδείξουμε τις δυνατότητες της Εξόρυξης Διαδικασιών και όχι τόσο η μελέτη της διαδικασίας παραλαβής περιβαλλοντικής άδειας από τους δήμους της Ολλανδίας. Για την ανάλυση και την μελέτη των δεδομένων μας έχουμε χρησιμοποιήσει ένα από τα πιο σύγχρονα εργαλεία για την Εξόρυξη Διαδικασιών το Disco. Το Disco, παρέχει παρόμοιες δυνατότητες με αυτές του ProM αλλά χρησιμοποιεί μόνο την ασαφή μοντελοποίηση για την κατασκευή μοντέλων και διανέμεται από την εταιρεία Fluxicon (http://www.fluxicon.com) που έχει έδρα στην Ολλανδία. Τα δεδομένα μας λοιπόν, αφορούν μια περίοδο δεκαπέντε μηνών και περιέχουν 8 577 γεγονότα και 1 434 περιπτώσεις. Από αυτές τις 1 434 περιπτώσεις οι 1 135 αποτελούν περιπτώσεις με ακριβώς έξι ενέργειες. Οι υπόλοιπες περιπτώσεις είναι περιπτώσεις με είτε περισσότερο ή μικρότερο αριθμό περιπτώσεων. Παρατηρούμε επίσης με την βοήθεια του Disco πως από τις 1 135 περιπτώσεις με έξι ενέργειες οι 713 περιπτώσεις ακολουθούν την ίδια ακριβώς σειρά γεγονότων. Υπάρχει δηλαδή κάποιο ίχνος μήκους έξι το οποίο αποτελεί την βασική διαδρομή. Με αυτό τον τρόπο καταφέραμε να

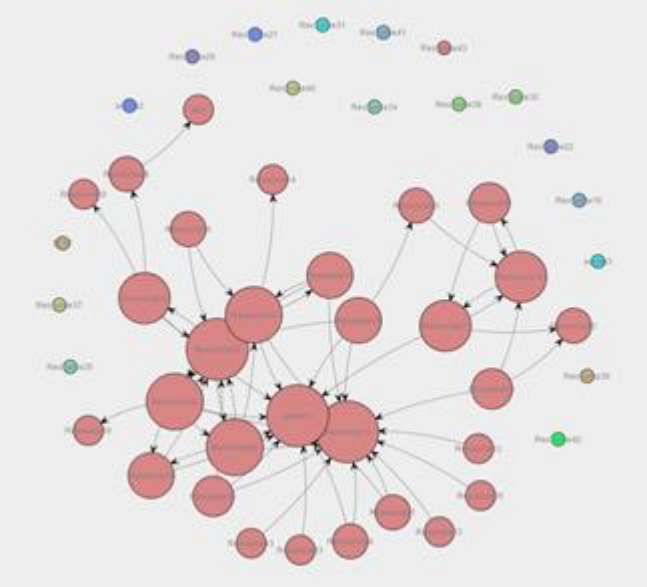
καταλήξουμε στο μοντέλο που παρουσιάζεται στο Σχήμα 23 το οποίο αποτελεί το βασικό μοντέλο της διεργασίας.



Σχήμα 24: Διαδικασία παραλαβής περιβαλλοντικής άδειας, το μοντέλο αυτό έχει παραχθεί με την βοήθεια του Disco και καταγράφει το μέσο χρόνο μεταξύ των ενεργειών.

Παρατηρούμε πως στο διάγραμμα δεν καταγράφονται οι συχνότητες των ενεργειών αλλά καταγράφεται ο μέσος χρόνος που χρειάζεται για περάσουμε από την εκτέλεση μιας ενέργειας σε μια άλλη. Με αυτό τον τρόπο έχουμε άμεση πληροφόρηση σχετικά με τον χρόνο που σπαταλάμε για την εκτέλεση κάποιον ενεργειών. Επίσης, η απεικόνιση αυτή μας

δίνει την δυνατότητα να βρούμε τα σημεία στα οποία η διεργασία σαν σύνολο καθυστερεί ώστε να βρούμε το πρόβλημα και να το διορθώσουμε. Παρατηρούμε στο διάγραμμα αυτό για παράδειγμα ότι υπάρχουν κάποιες ενέργειες που χρειάζονται περισσότερο από τρείς μέρες για να πραγματοποιηθούν. Αυτό μπορεί να συμβαίνει επειδή οι εργασίες από μόνες τους απαιτούν χρόνο η επειδή υπάρχει κάποια δυσκολία στην διαδικασία.



Σχήμα 25: Καταμερισμός εργασίας στον συγκεκριμένο δήμο.

Με τον όρο δυσκολία στη διαδικασία εννοούμε, η καταμέριση της εργασίας να μην είναι καλή ή κάποιος από τους εργαζόμενους του δήμου να μην είναι ο κατάλληλος για την θέση στην οποία εργάζεται. Για να λυθούν τέτοιου είδους προβλήματα η Εξόρυξη Διαδικασιών μας παρέχει την δυνατότητα να μελετήσουμε τον τρόπο με τον οποίο γίνεται ο καταμερισμός των ενεργειών όπως φαίνεται στο Σχήμα 24. Στο Σχήμα αυτό λοιπόν απεικονίζεται ο τρόπος με τον οποίο η εργασία έχει ανατεθεί από τον ένα εργαζόμενο του δήμου στον άλλο. Ο κάθε κόμβος αντιπροσωπεύει τον κάθε εργαζόμενο και τα βέλη δείχνουν τον τρόπο με τον οποίο μοιράζεται η εργασία ανάμεσα στους εργαζομένους. Το μέγεθος του κάθε κόμβου αντιπροσωπεύει το ποσό της εργασίας σε μορφή συχνότητας ενεργειών που είτε αναλαμβάνει ή μεταβιβάζει. Από το διάγραμμα αυτό αναγνωρίζουμε δύο τύπους εργαζομένων, αυτούς που κάνουν και αυτούς που δεν μεταβιβάζουν την εργασία. Περισσότερες πληροφορίες σχετικά με ανακάλυψη κοινωνικών δικτύων παρουσιάζονται στη δημοσίευση [21].

10. Συμπεράσματα

Οι μέθοδοι οι οποίοι χρησιμοποιήσαμε σε αυτή τη διπλωματική εργασία για την Εξόρυξη Δεδομένων μπορούν να εφαρμοστούν σε κάθε είδος διαδικτυακής πλατφόρμας για ανάλυση της συμπεριφοράς των χρηστών/επισκεπτών. Μπορούμε δηλαδή, να κάνουμε χρήση της βιβλιοθήκης Apache log4net, αν γράφουμε σε C#, ή της log4j, αν γράφουμε σε Java, για την παραγωγή των αρχείων καταγραφής συμβάντων και με την βοήθεια του MAS να τα αποθηκεύουμε. Εφαρμόζοντας τεχνικές Εξόρυξης Διαδικασιών και συγκεκριμένα με την χρήση του αλγόριθμου Fuzzy miner καταφέραμε επιτυχώς να παράγουμε σύνθετα μοντέλα που εξηγούσαν τον τρόπο πλοήγησης των χρηστών της διαδικτυακής πλατφόρμας Colibri. Ωστόσο, τα μοντέλα που παρήχθηκαν ήταν πολύ σύνθετα, spaghetti-like models, και ήταν αδύνατο να τα μελετήσουμε. Για αυτό το λόγο, χρησιμοποιήσαμε μεθόδους φιλτραρίσματος και ομαδοποίησης των δεδομένων ώστε να παράξουμε απλούστερα μοντέλα κατανοητά από τον άνθρωπο. Με αυτό τον τρόπο, καταφέραμε να μειώσουμε το επίπεδο της τάξης της πληροφορίας χωρίς να χαθούν πολύτιμα δεδομένα και στη συνέχεια να δημιουργήσουμε ευανάγνωστα μοντέλα διαδικασιών που περιέγραφαν τον ακριβή τρόπο πλοήγησης των χρηστών της πλατφόρμας. Είχαμε την δυνατότητα να ανακαλύψουμε σε ποια παράθυρα του Colibri εμφανίζονταν σφάλματα και τι είδους σφάλματα ήταν αυτά, πριν ή μετά από ποια διαδικασία υπήρχαν σφάλματα όπως επίσης και την ακριβή ώρα την οποία αυτά εμφανίζονταν. Επίσης, από τα αρχεία καταγραφής συμβάντων και με χρήση τεχνικών Ανάλυσης Δεδομένων βρέθηκαν τα διαφορετικά προφίλ των χρηστών του Colibri όπως επίσης και οι σελίδες με την μεγαλύτερη επισκεψιμότητα. Τέτοιου είδους πληροφορίες είναι απαραίτητες για την βελτίωση και την μελλοντική ανάπτυξη της πλατφόρμας. Θα είχαμε την δυνατότητα να γνωρίζουμε τον ακριβή χρόνο τον οποίο οι χρήστες παρέμεναν σε κάθε σελίδα όμως υπήρχαν κάποιες δυσκολίες οι οποίες αναφέρονται στο κεφάλαιο των αποτελεσμάτων, αναφέρονται επίσης και πιθανές λύσεις τους. Επίσης, δεν υπήρχε η πολυτέλεια του χρόνου και η διαδικασία παραγωγής των μοντέλων είναι μια χρονοβόρα διαδικασία.

Επιπλέον, σε αυτή την διπλωματική εργασία παρουσιάζονται αναλύσεις και άλλων δεδομένων που δεν έχουν παραχθεί από εμάς. Οι τεχνικές Εξόρυξης Διαδικασιών μπορούν να εφαρμοστούν σε όλα τα δεδομένα που πληρούν τις προϋποθέσεις που αναφέρονται στο κεφάλαιο 3.1, δηλαδή σε δεδομένα που καταγράφουν διεργασίες. Ακόμα, μας δίνουν την δυνατότητα μέσω τον αλγορίθμων που χρησιμοποιούν να μελετηθούν αρχεία καταγραφής συμβάντων που είναι σύνθετα, που έχουν δηλαδή πολλές κλάσεις ενεργειών, αρχεία με πολλαπλά ίχνη κλπ. Φανταστείτε λοιπόν, τις δυνατότητες που μας δίνει ο τομέας αυτός, μπορούν να μελετηθούν οι διαδικασίες ενός νοσοκομείου, μιας επιχείρησης, μιας τράπεζας, ενός αεροδρομίου, μιας γραμμής παραγωγής, όπου υπάρχει διαδικασία που η καταγραφή των ενεργειών είναι εφικτή, αυτή μπορεί να μελετηθεί. Όμως όταν έχουμε να αντιμετωπίσουμε δεδομένα τα οποία δεν έχουν παραχθεί από εμάς, υπάρχουν κάποιες

δυσκολίες. Για παράδειγμα, η δομή των δεδομένων δεν είναι πάντα η ίδια ή δεν χρησιμοποιείται η Αγγλική γλώσσα και χάνεται ποσό πληροφορίας στην μετάφραση των δεδομένων και συγκεκριμένα στην μετάφραση των ενεργειών. Είναι αναγκαίο λοιπόν, να προταθούν κάποιοι βασικοί κανόνες καταγραφής των δεδομένων για να περιοριστούν τα σφάλματα και οι δυσκολίες.

11. Βιβλιογραφία

[1] Gantz, J. & Reinsel, D. (2010). *The Digital Universe Decade. Are You Ready? International Data Corporation*, Framingham, emc.com/digital_universe.

[2] Van Der Aalst, W. (2011). *Process mining: discovery, conformance and enhancement of business processes*. Springer.

[3] Van Der Aalst, W. & Van Hee, K. M. (2004*). Workflow management: models, methods, and systems*. MIT press.

[4] Cardoso, J., Bostrom, R. P. & Sheth, A. (2004). *Workflow Management Systems and ERP Systems: Differences, Commonalities, and Applications.* Information Technology and Management, Volume 5, Issue3-4, p. 319-338.

[5] Van der Aalst, W., Rubin, V., Van Dongen, B. F., Kindler, E. & Günther, C. W. (2006). *Process mining: A two-step approach using transition systems and regions*. *BPM Center Report BPM-06-30. bpmcenter. org*.

[6] Van der Aalst, W., Weijters, T. & Maruster, L. (2004). *Workflow mining: Discovering process models from event logs*. Transactions on Knowledge and Data Engineering, Volume 16, Issue 9, p. 1128-1142.

[7] Van Der Aalst, W., et al. (2012). *Process mining manifesto*. In Business process management workshops, Volume 99 p. 169-194. Springer.

[8] Günther, C. W. & Van Der Aalst, W. (2007). *Fuzzy mining–adaptive process simplification based on multi-perspective metrics*. Lecture Notes In Computer Science, Volume 4714, p. 328-343.

[9] Xia, J. (2010). *Automatic Determination of Graph Simplification Parameter Values for Fuzzy Miner* (Doctoral dissertation, Master's thesis, Eindhoven University of Technology).

[10] Günther, C. W. (2009) *Process Mining in Flexible Environments* (Doctoral dissertation, Master's thesis, Eindhoven University of Technology).

[11] Buijs, J. C., Van Dongen, B. F. & Van der Aalst, W. (2012). *On the Role of Fitness, Precision, Generalization and Simplicity in Process Discovery.* Lecture Notes in Coputer Science, Volume 7565, p. 305-322.

[12] Buijs, J. C., La Rosa, M., Reijers, H. A., Van Dongen, B. F. & van der Aalst, W. (2013). *Improving Business Process Models Using Observed Behavior*. Lecture Notes In Computer Science, Volume 162, p. 44-59.

[13] Van der Aalst, W., Adriansyah, A. & Van Dongen, B. F. (2012). *Replaying History on Process Models for Conformance Checking and Performance Analysis*. Data Mining and Knowledge Discovery, Volume 2 Issue 2, p. 182-192.

[14] De Medeiros, A. K. A., Van der Aalst, W. & Weijters, A. J. M. M. (2003). *Workflow mining: Current Status and Future Directions*. Lecture Notes In Computer Science, Volume 2888, p. 389-406.

[15] Munoz-Gama, J. & Carmona, J. (2011). *Enhancing Precision in Process Conformance: Stability, Confidence and Severity*. Computational Intelligence and Data Mining, p. 184-191. IEEE.

[16] Van der Aalst, W., Van Dongen, B. F., Günther, C. W., Mans, R. S., De Medeiros, A. K. A., Rozinat, A., Rubin, V., Song, M., Verbeek, H. M. W. & Weijters, A. J. M. M. (2007). *ProM 4.0: Comprehensive Support for Real Process Analysis.* Lecture Notes in Computer Science, Volume 4546, p. 484-494.

[17] Verbeek, H. M. W., Buijs, J. C., Van Dongen, B. F. & Van der Aalst, W. (2010). *Prom 6: The process mining toolkit*. BPM Demonstration Track 2010, Volume 615, p. 34-39. bpmn2010.org.

[18] Poggi, N., Muthusamy, V., Carrera, D. & Khalaf, R. (2013). *Business Process Mining from E-commerce Web Logs*. Lecture Notes in Computer Science, Springer, Volume 8094, p. 65-80.

[19] Kumar, L., Singh, H. & Kaur, R. (2012). *Web analytics and metrics: a survey*. Proceedings of the International Conference on Advances in Computing, Communications and Informatics, p. 966-971. ACM.

[20] Bautista, A. D., Wangikar, L. & Akbar, S. M. K. (2012). *Process Mining Driven Optimization of a Consumer Loan Approvals Process*. Technical Report BPM 2012, ckmadvisors.com.

[21] Van Der Aalst, W., Reijers, H. A. & Song, M. (2005). *Discovering social networks from event logs*. Computer Supported Cooperative Work, Volume 14, Issue 6, p. 549-593.

Appendix

Txt2xml:

```
#####################################################################
#__author__ = "Vatikiotis Fotios"                                   #
#__copyright__ = "Copyright 2014, Process Minig & Colibri"          #
#__maintainer__ = "Vatikiotis Fotios"                               #
#__email__ = "vati.math@gmail.com"                                  #
#####################################################################
#input_file
infile = r'C:\Users\Fotis Vat\Desktop\Output\TXT\AllTogetherSorted.txt'
#text_file = open('C:\Users\Fotis Vat\Desktop\Output\TXT\AllTogetherSorted.txt','r')

#creates f. f[0] = 1st_log_line, f[1] = 2nd_log_line...
with open(infile) as f:
    f = f.readlines()

names = []
date = []
action = []
company = []
date_new2 = []

#b: the elements of b are the elements of a line splitted with a comma
#so we modify the f[] to fill names[], company[], date[] & action[]
for i in range(len(f)):
        b = f[i].split(',')
        a = f[i].split('[')
        date_new2.append(a[0])
        b.remove(b[0])
        c = b[0].split(':')
        names.append(c[1])
        c = c[0]
        c = str(c)
        c = c.split()
        company.append(c[-1])
        date.append(b[1])
        action.append(b[2])
        #print action[i]

# rstrip returns a copy of the string with trailing characters removed
#cause had a prob with /n
#lstrip deletes the first space from the string and replaces the space with a dot
for i in range(len(names)):
        names[i] = names[i].rstrip()
        names[i] = names[i].lstrip()
        action[i] = action[i].rstrip()
        action[i] = action[i].lstrip()
        date[i] = date[i].rstrip()
        date[i] = date[i].lstrip()
        c = date[i].split()
        b = c[0].split('/')
        date[i] = b[2]+'-'+b[0]+'-'+b[1]+'T'+c[1]+',000'
        e = date_new2[i].split()
        f = e[0].split('-')
        date_new2[i] = f[0]+'-'+f[1]+'-'+f[2]+'T'+e[1]

sum=0
for i in range(len(names)):
        if (names[i] == ""):
                names[i] = "Unknown"
                sum = sum+1
```

```
names_lower = []
#We create names_lower which has no Capital letters!(for the test_names[])
for i in range(len(names)):
    names_lower.append(names[i].lower())


date_new = []
action_new = []
names_new = []
company_new = []

#Filter keep only View & Action
for i in range(len(names_lower)):
        if ('View:' in action[i] or 'Action:' in action [i]):
                action_new.append(action[i])
                date_new.append(date_new2[i])
                names_new.append(names_lower[i])
                company_new.append(company[i])
#CLUSTERING
for i in range(len(action_new)):
        if ('Forecast' in action_new[i]):
                action_new[i] = 'ForecastPage'
        if ('Enrichment' in action_new[i]):
                action_new[i] = 'EnrichmentPage'
        if ('Import' in action_new[i]):
                action_new[i] = 'ImportPage'
        if ('History' in action_new[i]):
                action_new[i] = 'HistoryPage'
        if ('SideController' in action_new[i]):
                action_new[i] = 'Other'
        if ('Settings' in action_new[i]):
                action_new[i] = 'Other'
        if ('Simulation' in action_new[i]):
                action_new[i] = 'ForecastPage'
        if ('Error' in action_new[i]):
                action_new[i] = 'Errors'
        if ('Export' in action_new[i]):
                action_new[i] = 'ExportPage'
        if ('Stability' in action_new[i]):
                action_new[i] = 'StabilityPage'
        if ('Analysis/Reliability' in action_new[i]):
                action_new[i] = 'ReliabilityPage'
        if (('DisaggregationConfiguration/AllNoUniverse') in action_new[i]):
                action_new[i] = 'Other'
        if (('Home') in action_new[i]):
                action_new[i] = 'HomePage'
        if (('DeleteUniverse') in action_new[i]):
                action_new[i] = 'Other'
        if (('Bug') in action_new[i]):
                action_new[i] = 'Other'
        if (('AskForCollaboration') in action_new[i]):
                action_new[i] = 'Other'
        if (('About') in action_new[i]):
                action_new[i] = 'Other'
        if (('Subscribe') in action_new[i]):
                action_new[i] = 'Other'
        if ('Budget') in action_new[i]:
                action_new[i] = 'Other'
        if ('Connection') in action_new[i]:
                action_new[i] = 'Other'
        if ('Logout') in action_new[i]:
                action_new[i] = 'Other'
        if('ThirdPartyData') in action_new[i]:
                action_new[i] = 'Other'
        if('DisaggregationConfiguration') in action_new[i]:
                action_new[i] = 'Other'
        if('WeekDisaggregationConfiguration') in action_new[i]:
                action_new[i] = 'Other'
```

```
test_names = []
#Test_names: len(test_names) = number of different users
for i in range(len(names)):
    if names_lower[i] not in test_names:
        test_names.append(names_lower[i])

start_date = date_new2[0].split('T')
end_date = date_new2[-1].split('T')

action_newer = []
date_newer = []

#####PATH#####
file_name_users = 'C:\Users\Fotis Vat\Desktop\Output\XML\Users_Clustered_' + start_date[0] + '_till_' + end_date[0] + '.xml'
file_name_companies = 'C:\Users\Fotis Vat\Desktop\Output\XML\Companies_Clustered_' + start_date[0] + '_till_' + end_date[0] + '.xml'

g = open(file_name_users ,'w')

g.write('<?xml version=\"1.0\" encoding=\"UTF-8\"?>\n\n<WorkflowLog xmlns:xsi=\"http://www.w3.org/2001/XMLSchema-instance\"\nxsi:noName

for i in range(len(test_names)):
    g.write('<ProcessInstance id="' + test_names[i]+ '">\n')
    #g.write('<AuditTrailEntry>\n<WorkflowModelElement>Start</WorkflowModelElement>\n<EventType>complete</EventType>\n</AuditTrailEntry
    for j in range(len(names_new)):
            #if test_names[i] in names_lower[j] or test_names[i] in action[j]: (thats for the messages)
            if test_names[i] in names_new[j]:
                action_newer.append(action_new[j])
                date_newer.append(date_new[j])
    for k in range(len(action_newer)):
        g.write('<AuditTrailEntry>\n<WorkflowModelElement>' + action_newer[k] + '</WorkflowModelElement>\n<EventType>complete</EventTyp
    #g.write('<AuditTrailEntry>\n<WorkflowModelElement>End</WorkflowModelElement>\n<EventType>complete</EventType>\n</AuditTrailEntry>\
    g.write('</ProcessInstance>\n')
    action_newer = []
    date_newer = []
g.write('</Process>\n</WorkflowLog>')
g.close()

#########################################################################################################################

#Same staff for companies
company_lower = []
test_company = []
action_newer = []
date_newer = []
#We create names_lower which has no Capital letters!(for the test_names[])
for i in range(len(company_new)):
    company_lower.append(company[i].lower())

#Test_names: len(test_names) = number of different users
for i in range(len(company_new)):
    if company_lower[i] not in test_company:
        test_company.append(company_lower[i])

g = open(file_name_companies,'w')

g.write('<?xml version=\"1.0\" encoding=\"UTF-8\"?>\n\n<WorkflowLog xmlns:xsi=\"http://www.w3.org/2001/XMLSchema-instance\"\nxsi:noName

for i in range(len(test_company)):
    g.write('<ProcessInstance id="' + test_company[i]+ '">\n')
   # g.write('<AuditTrailEntry>\n<WorkflowModelElement>Start</WorkflowModelElement>\n<EventType>complete</EventType>\n</AuditTrailEntry
    for j in range(len(company_lower)):
            #if test_names[i] in names_lower[j] or test_names[i] in action[j]: (thats for the messages)
            if test_company[i] in company_lower[j]:
                action_newer.append(action_new[j])
                date_newer.append(date_new[j])
    for k in range(len(action_newer)):
        g.write('<AuditTrailEntry>\n<WorkflowModelElement>' + action_newer[k] + '</WorkflowModelElement>\n<EventType>complete</EventTyp
    #g.write('<AuditTrailEntry>\n<WorkflowModelElement>End</WorkflowModelElement>\n<EventType>complete</EventType>\n</AuditTrailEntry>\
    g.write('</ProcessInstance>\n')
    action_newer = []
    date_newer = []
g.write('</Process>\n</WorkflowLog>')
g.close()
```

R tool:

```
#READ file and save it as a dataframe
#To run commands ctrl+R
rm(list=ls())
setwd("C:/Users/FVA3294/Desktop/WorkforR")
x<-read.table('CSVfile.csv')
X<-as.data.frame(x)
colnames(X) <- c("Users", "Company", "Time", "Actions")
X$Users<-NULL
factor(X[,3])
#Table
fix(X)
mytable<-table(X[,1],X[,3])
mytable
#Relative frequencies of lines.
#With this you can see the % of pages a user visits.
lines<-prop.table(mytable,1)
#Relative frequencies of columns.
columns<-prop.table(mytable,2)
lines
columns
pie(lines[,1])
pie(lines[,7])
pie(columns[,1])
datalines<-as.data.frame(lines)
pie(lines[5,])
############################################################
#ACTIVITY
library(ggplot2)
bp <- qplot(X[,1],horiz=T,data=X,fill=X$Actions,xlab='Users',ylab='Action counter')
bp + theme(axis.title = element_text(face="bold", colour="blue", size=16),axis.text.x = element_text(angle=90, vjust=0.5, size=10))
qplot(X[,3],horiz=T,data=X,fill=X[,1])
########################################
barplot(mytable,col=3)
#GRaphical Methods
Actions<-table(X[,3])
Users<-table(X[,1])
barplot(Actions,horiz=T,las=1)
barplot(Actions,horiz=T,las=1,cex.names=0.5)
barplot(Users,las=1, xpd=TRUE, srt=45)
barplot(Users,horiz=T,las=1,cex.names=0.5)
Dates<-c()
for (i in 1:length(X[,2])){
  X[i,2] <- as.POSIXct(X[i,2], format = "%Y%d%mT%H:%M:%S")
  #Dates<- c(Dates,(X[i,2]))
}

tm3 <- as.POSIXlt(Time[1], format = "%Y%d%mT%H:%M:%S")
x <- barplot(Users, xaxt="n")
text(cex=1, x=x-.25, y=-1.25, xpd=TRUE, srt=45)
############################################################
#Time and Date
for(i in length(X[,2]))
{
  tm1 <- as.POSIXct(X[,2], format = "%Y-%m-%dT%H:%M:%S")
}
#Add a column in dataframe
X["Timestamps"]<- tm1
############################################################
#Time only
for(i in 1:length(X[,2]))
{
  tm2<- as.POSIXct(X[,2], format = "%Y-%m-%d")
}
X["Dates"]<-tm2

############################################################
#Create Plot for each day kinitikotita####
#First we create the array with sum of actions each day!
a <- table(X[,3],X$Dates)
for (i in 1:length(a[1,])){
  b[i]<-sum(a[,i])
}
plot(unique(X$Dates),b,type="b",main="Users Activity",xlab="Time",ylab="Users Actions")
install.packages("gridExtra")
library(gridExtra)
grid.barbed()
qplot(unique(X$Dates),b, geom = c("point","path"))
qplot(unique(X$Dates),b, geom = "barbed", colour = f, size = f, linewidth = I(2), only.lines = F, space = 1, shape=15)
```

```
#############################################################
New <- cbind(unique(X$Dates),b)
New <- as.data.frame(New)
New
#Next command
pie(Actions)
pie(Users)
qplot(unique(X$Dates),b,main="Users Activity",xlab="Time",ylab="Users Actions")
plot(X[,4]
install.packages("ggplot2")
library(ggplot2)
qplot(X[,4],X[,3],X)
qplot(X[,4],X[,3],X,colour=X[,1])
#############################################################
install.packages("plotrix")
library(plotrix)
pie3D(lines[,1],radius=1.4, explode=0.1,height=0.1, theta=pi/2, labelcex=0.5)
pie3D(lines[3,],radius=1.4, explode=0.1,height=0.1, theta = pi/3, labelcex=0.5)
pie3D.labels(radialpos,radius=1,height=0.3,theta=pi/6,labelcex=1.5,minsep=0.3)

par(mfrow=c(2,3))
for (i in 1:34)
{
pie3D(lines[i,],radius=1.4, explode=0.1,height=0.1, theta = pi/3, labelcex=0.5, main=names(lines[,1][i]), radialpos=10)
}

par(mfrow=c(1,1))
```