



ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ
ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΕΠΙΚΟΙΝΩΝΙΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Νοημοσύνη Επιχειρηματικών Διεργασιών και Εξόρυξη Διεργασιών
Μια εφαρμογή με χρήση του ProM Framework

Άγγελος Χάλαρης

Επιβλέποντες: **Αποστόλου Δημήτριος, Επίκουρος Καθηγητής**

ΠΕΙΡΑΙΑΣ

Νοέμβριος 2015

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Νοημοσύνη Επιχειρηματικών Διεργασιών και Εξόρυξη Διεργασιών
Μια εφαρμογή με χρήση του ProM Framework

Ονοματεπώνυμο:	Άγγελος Χάλαρης
Αριθμός Μητρώου:	Π11165

Επιβλέποντες: Αποστόλου Δημήτριος, Επίκουρος Καθηγητής

Πρόλογος

Ο όρος Επιχειρησιακή Νοημοσύνη (Business Intelligence) αναφέρεται σε τεχνικές συλλογής δεδομένων και ανάλυσής της τους με σκοπό τη μετατροπή τους σε χρήσιμες πληροφορίες για την επιχείρηση. Οι πιο συνηθισμένες από τις εφαρμογές της Επιχειρησιακής Νοημοσύνης χρησιμοποιούν μαθηματικά και στατιστικά μοντέλα στα δεδομένα για ανάλυση συσχετισμού, ανάλυση τάσης, έλεγχο υποθέσεων και ανάλυση προβλέψεων. Ένας σημαντικός κλάδος εφαρμογών της Επιχειρησιακής Νοημοσύνης αφορά τις επιχειρηματικές διεργασίες και διαδικασίες και ονομάζεται Νοημοσύνη Επιχειρησιακών Διεργασιών (Business Process Intelligence). Η Νοημοσύνη Επιχειρησιακών Διεργασιών γίνεται όλο και πιο δημοφιλής μεταξύ των επιχειρήσεων αφού η ανάγκη για βελτιστοποίηση της αποδοτικότητας των παραγωγικών διαδικασιών είναι όλο και μεγαλύτερη καθώς επίσης και οι αλλαγές είναι ραγδαίες και οι επιχειρήσεις θα πρέπει να μπορούν να αντιδράσουν γρήγορα. Οι εφαρμογές αυτές απαρτίζονται από ένα πολύ μεγάλο εύρος εργαλείων που κυμαίνονται από παρακολούθηση και ανάλυση διεργασιών μέχρι ελέγχους συμμόρφωσης, πρόβλεψη και βελτιστοποίηση των διαδικασιών. Στις σελίδες που ακολουθούν θα εξεταστεί σε βάθος η Νοημοσύνη Επιχειρησιακών Διεργασιών και οι τεχνικές που την απαρτίζουν με ιδιαίτερη έμφαση στην Εξόρυξη Διεργασιών. Επιπλέον θα παρουσιαστεί μια εφαρμογή της Νοημοσύνης Επιχειρησιακών Διεργασιών σε ένα σετ δεδομένων που έχουν προκύψει από προσομοίωση με τη χρήση του εργαλείου ελεύθερου λογισμικού ProM.

ΘΕΜΑΤΙΚΗ ΠΕΡΙΟΧΗ: Εργαλεία & Λογισμικό Διοίκησης Επιχειρήσεων

ΛΕΞΕΙΣ ΚΛΕΙΔΙΑ: Εξόρυξη Διεργασιών, Νοημοσύνη Επιχειρησιακών Διεργασιών, Επιχειρησιακή Νοημοσύνη, Επιχειρησιακές Διεργασίες, ProM Framework

Πίνακας Περιεχομένων

Πρόλογος.....	1
Πίνακας Περιεχομένων.....	2
1. Εισαγωγή.....	4
1.1. Τεχνικές της Νοημοσύνης Επιχειρησιακών Διεργασιών.....	5
1.1.1. Ανάλυση Διεργασιών (Process Analysis).....	5
1.1.2. Πρόβλεψη (Prediction).....	5
1.1.3. Παρακολούθηση Διεργασιών (Process Monitoring).....	5
1.1.4. Έλεγχος Διεργασιών (Process Control).....	6
1.1.5. Εξόρυξη Διεργασιών (Process Mining).....	6
2. Εξόρυξη Διεργασιών.....	7
2.1. Ανεύρεση Διεργασιών (Process Discovery).....	7
2.2. Έλεγχος Συμμόρφωσης (Conformance Checking).....	7
2.3. Βελτίωση Μοντέλου (Model Enhancement).....	8
3. ProM Framework: Μια σύγχρονη εφαρμογή Εξόρυξης Διεργασιών.....	9
3.1. Διεπαφή Χρήστη.....	9
3.2. Μετατροπή κατάλληλων αρχείων σε Event Logs.....	11
3.2.1. XESame.....	11
3.2.2. Disco.....	11
4. Τρέχον Παράδειγμα.....	13
4.1. Παραδοχές, περιορισμοί και δομή των δεδομένων.....	13
4.2. Ανεύρεση Διεργασιών.....	14
4.2.1. Discover using Decomposition.....	14
4.2.2. Mine for a Fuzzy Model.....	18
4.2.3. Discover Matrix.....	20
4.2.4. Mine for a Social Network.....	21
4.2.5. Mine BPMN Using Heuristics Miner of ProM 5.2.....	25
4.3. Έλεγχος Συμμόρφωσης.....	27
4.3.1. Replay using Decomposition.....	27
4.3.2. LTL Checker Default.....	30
4.4. Βελτίωση Μοντέλου.....	35
4.4.1. Enhance Log: Add Mapping of Activity Names.....	35
4.4.2. Enhance Log: Merge Subsequent Events (AAABB -> AB).....	37
Πίνακας Ορολογίας.....	39
Συντμήσεις, Αρκτικόλεξα, Ακρωνύμια.....	40
Παράρτημα Ι: Αρχεία καταγραφής συμβάντων του παραδείγματος.....	41
Event Log 1.....	41
Event Log 2.....	44
Event Log 3.....	47

Event Log 4.....	50
Παράρτημα II: Οδηγίες Εγκατάστασης και Εκτέλεσης του ProM Framework.....	52
Βιβλιογραφικές και ηλεκτρονικές αναφορές.....	55

1. Εισαγωγή

Με τον όρο Νοημοσύνη Επιχειρησιακών Διεργασιών (Business Process Intelligence, στο εξής BPI) αναφερόμαστε στην εφαρμογή τεχνικών της Επιχειρησιακής Νοημοσύνης (Business Intelligence, στο εξής BI) σε παραγωγικές διαδικασίες ή επιχειρησιακές διεργασίες. Η BI απαρτίζεται από διάφορες τεχνικές και εργαλεία για τη μετατροπή μεγάλου όγκου δεδομένων σε πληροφορίες χρήσιμες για την επιχείρηση με τη χρήση μαθηματικών και στατιστικών μεθόδων. Ο κύριος σκοπός της BI είναι η καλύτερη υποστήριξη των διοικητικών αποφάσεων, οι οποίες μπορεί να κυμαίνονται από απλές λειτουργικές αποφάσεις (πχ. τη διάθεση ή την τιμολόγηση προϊόντων) μέχρι στρατηγικές αποφάσεις για την επιχείρηση (πχ. μακροπρόθεσμους στόχους και προτεραιότητες ή την κατεύθυνση της επιχείρησης συνολικά).

Τα δεδομένα που είναι χρήσιμα στην BI προέρχονται τόσο από την ίδια την επιχείρηση (εσωτερικά δεδομένα), όσο και από την αγορά μέσα στην οποία δραστηριοποιείται η επιχείρηση (εξωτερικά δεδομένα). Όσον αφορά στα εσωτερικά δεδομένα, αυτά αποτελούνται από σημαντικά ιστορικά, οικονομικά και άλλα δεδομένα που συνήθως αποθηκεύονται σε μια ειδική βάση δεδομένων που ονομάζεται Αποθήκη Δεδομένων (Data Warehouse). Ειδικά αν μια Αποθήκη Δεδομένων περιέχει και στοιχεία που αφορούν στις διεργασίες της επιχείρησης, τότε αυτή ονομάζεται Αποθήκη Δεδομένων Διεργασιών (Process Data Warehouse) και μπορεί να χρησιμοποιηθεί σαν πηγή δεδομένων για την BPI. Για να μετατραπούν τα δεδομένα αυτά σε πληροφορίες που η επιχείρηση θα μπορεί να χρησιμοποιήσει για να υποστηρίξει τη λήψη αποφάσεων, χρησιμοποιούνται μέθοδοι Ανάλυσης Δεδομένων (Data Analysis) και Εξόρυξης Δεδομένων (Data Mining).

Τα τελευταία χρόνια, η BPI γίνεται όλο και πιο δημοφιλής, καθώς οι επιχειρήσεις πρέπει να μπορούν να αντιδρούν γρήγορα στις αλλαγές της αγοράς ώστε να είναι ανταγωνιστικές, να είναι όσο πιο αποδοτικές γίνεται και να μπορούν να ανταποκρίνονται σε κάποιους κανονισμούς, πρότυπα και πολιτικές. Επομένως πολλά Συστήματα Διαχείρισης Επιχειρησιακών Διεργασιών (Business Process Management Systems, στο εξής BPMS), δηλαδή πλατφόρμες λογισμικού που υποστηρίζουν διαδικασίες όπως ορισμό, εκτέλεση και παρακολούθηση των διεργασιών μιας επιχείρησης, επεκτείνουν τις δυνατότητές τους με λειτουργικότητα BPI. Συνεπώς συνδυάζονται οι δυνατότητες συλλογής δεδομένων της Διοίκησης Επιχειρησιακών Διεργασιών (Business Process Management, στο εξής BPM), μέσα από Αρχεία Καταγραφής Συμβάντων (Event Logs), με τη δυνατότητα της BPI να εκμεταλλεύεται τα δεδομένα που αφορούν στις διεργασίες της επιχείρησης μέσα από αναλύσεις που παρέχουν στην επιχείρηση μια καλύτερη κατανόηση του πώς εκτελούνται οι διεργασίες τους στην πραγματικότητα.

Στο παρελθόν το μεγαλύτερο μέρος της έρευνας στον κλάδο των επιχειρησιακών διεργασιών και των παρεχόμενων εργαλείων είχε επικεντρωθεί στην ανάπτυξη νέων μοντέλων διεργασιών και στην αυτοματοποίηση των διεργασιών. Μεγάλη σημασία, όμως, παρουσιάζουν τομείς όπως η ανάλυση διεργασιών, η πρόβλεψη και η βελτιστοποίηση. Αυτό το κενό στοχεύουν να καλύψουν οι τεχνικές και τα εργαλεία της BPI προσφέροντας ένα σύστημα που υποστηρίζει τη λήψη αποφάσεων ούτως ώστε να μπορούν να διαχειριστούν καλύτερα την ποιότητα εκτέλεσης των επιχειρησιακών διεργασιών.

1.1. Τεχνικές της Νοημοσύνης Επιχειρησιακών Διεργασιών

Η BPI αποτελεί ένα πολύτιμο εργαλείο, όχι μόνο για την βελτίωση ή τον ανασχεδιασμό των διεργασιών μιας επιχείρησης, αλλά και για το συντονισμό των βραχυπρόθεσμων στόχων της επιχείρησης με τους μακροπρόθεσμους. Αυτό επιτυγχάνεται με ένα εύρος τεχνικών που ακολουθούν παρακάτω:

1.1.1. Ανάλυση Διεργασιών (Process Analysis)

Το πρώτο βήμα για τη βελτίωση μιας διεργασίας είναι να αναλυθεί ούτως ώστε να γίνουν κατανοητές οι δραστηριότητες που την αποτελούν, τους συσχετισμούς μεταξύ τους και την αξία των σχετικών μεγεθών. Για την Ανάλυση Διεργασιών απαιτείται ο καθορισμός των περιορισμών της διεργασίας καθώς και των σημείων εκκίνησης/εισοχής και τερματισμού/εκροής, η κατασκευή ενός διαγράμματος ροής της διεργασίας που περιγράφει τα στάδιά της και τις σχέσεις μεταξύ τους και η αποσαφήνιση των σημαντικών μεγεθών κάθε βήματος σε αυτή. Αυτά τα βήματα οδηγούν με τη σειρά τους στην ταυτοποίηση των σημείων συμφόρησης (Bottlenecks), δηλαδή των βημάτων που έχουν τη μικρότερη παραγωγικότητα, την αξιολόγησή του αντίκτυπου που έχουν αυτά στη διεργασία και τελικά τη λήψη λειτουργικών αποφάσεων για τη βελτίωσή της.

1.1.2. Πρόβλεψη (Prediction)

Με τον όρο Εξαίρεση (Exception) αναφερόμαστε σε μια κρίσιμη, απρόβλεπτη ή ανεπιθύμητη κατάσταση που έχει ως αποτέλεσμα απόκλιση από τη βέλτιστη εκτέλεση μιας διεργασίας τέτοια ώστε να εμποδίζει την παροχή υπηρεσιών με την επιθυμητή ποιότητα (π.χ. καθυστερήσεις στην ολοκλήρωση μιας διεργασίας, ελλείψεις πόρων λόγω ιδιαίτερων συνθηκών, παράπονα πελατών στη διεύθυνση κλπ.).

Πέραν του επεξηγηματικού μοντέλου που παράγει η Ανάλυση Διεργασιών, το οποίο βοηθά στην βελτιστοποίηση των διεργασιών με βάση τα ιστορικά δεδομένα και τελικά στην παραγωγή ενός μοντέλου αποφάσεων, είναι δυνατόν από τα ίδια δεδομένα να προκύψουν και προγνωστικά μοντέλα τα οποία έχουν τη δυνατότητα να προβλέπουν εξαιρέσεις που μπορεί να προκύψουν στο μέλλον. Αυτό επιτρέπει στις επιχειρήσεις να προετοιμάζονται εκ των προτέρων για να αποφύγουν τέτοιες καταστάσεις ή να σχεδιάζουν τις λύσεις που θα εφαρμόσουν σε περίπτωση που παρουσιαστούν τέτοια προβλήματα.

1.1.3. Παρακολούθηση Διεργασιών (Process Monitoring)

Με την Παρακολούθηση Διεργασιών είναι δυνατός ο έλεγχος σε πραγματικό χρόνο των τρεχουσών διεργασιών (π.χ. την πρόοδό τους, το χρόνο που έχει δαπανηθεί για την ολοκλήρωσή τους, τα σημεία συμφόρησης τους κλπ.) έτσι ώστε να είναι δυνατόν να ενημερώνεται άμεσα η επιχείρηση για ασυνήθιστες ή ανεπιθύμητες καταστάσεις, καθώς και να παρακολουθεί την καλή κατάσταση των συστημάτων και των διεργασιών της και τους πόρους της. Αυτό επιτρέπει στις επιχειρήσεις να αντιδράσουν άμεσα σε περίπτωση που παρουσιαστούν προβλήματα, παρέχοντας παράλληλα τις πληροφορίες που είναι αναγκαίες για να ληφθούν οι σωστές αποφάσεις ώστε να αντιμετωπιστεί το εκάστοτε πρόβλημα, ενώ ταυτόχρονα τους δίνει τη δυνατότητα να εκμεταλλευτούν στο μέγιστο

βαθμό ιδιαίτερες συνθήκες και ευκαιρίες που μπορεί να παρουσιαστούν.

1.1.4. Έλεγχος Διεργασιών (Process Control)

Αξιοποιώντας την Παρακολούθηση Διεργασιών και τα μοντέλα πρόβλεψης που έχει στη διάθεσή της μια επιχείρηση, είναι δυνατόν να αυτοματοποιηθεί μερικώς η λήψη αποφάσεων με τέτοιο τρόπο ώστε να βελτιστοποιούνται οι επιχειρησιακές διεργασίες, διατηρώντας τις συνθήκες εκτέλεσής τους μέσα στα αποδεκτά επίπεδα. Ως αποτέλεσμα μπορούν να αποφευχθούν διάφορες εξαιρέσεις που έχουν προκύψει από προβλέψεις ή να μειωθεί ο αντίκτυπος που έχουν στην ομαλή λειτουργία της επιχείρησης χωρίς να είναι αναγκαία η ανθρώπινη επέμβαση.

1.1.5. Εξόρυξη Διεργασιών (Process Mining)

Πέραν των παραπάνω τεχνικών, μεγάλο ενδιαφέρον παρουσιάζει η χρήση της Εξόρυξης Διεργασιών από την BPI, δηλαδή του συνόλου των τεχνικών που μπορούν να χρησιμοποιηθούν για να αντλήσουν γνώση από τα αρχεία καταγραφής συμβάντων με σκοπό την ανάλυση των επιχειρησιακών διεργασιών. Η Εξόρυξη Διεργασιών, καθώς και οι τεχνικές της περιγράφονται στην επόμενη ενότητα και γίνεται μια επισκόπηση τους. Οι τεχνικές αυτές καθώς και οι εφαρμογές τους με τη χρήση του εργαλείου ελεύθερου λογισμικού ProM θα μας απασχολήσουν στις επόμενες ενότητες.

2. Εξόρυξη Διεργασιών

Η Εξόρυξη Διεργασιών (Process Mining, στο εξής PM) είναι ένα σχετικά νέο πεδίο έρευνας που τοποθετείται μεταξύ της Μηχανικής Μάθησης και της Εξόρυξης Γνώσης από τη μία πλευρά και τη Μοντελοποίηση και Ανάλυση Διεργασιών από την άλλη. Οι στόχοι της PM είναι η ανεύρεση διεργασιών, ο έλεγχός τους και η βελτίωσή τους εξορύσσοντας γνώση από αρχεία καταγραφής συμβάντων (στο εξής event logs) που στα σύγχρονα συστήματα είναι άμεσα διαθέσιμα. Τα event logs περιέχουν υποχρεωτικά δεδομένα που αφορούν ποια γεγονότα ανήκουν σε μια ορισμένη εκτέλεση μιας διεργασίας, τη σειρά αυτών των γεγονότων και προαιρετικά δεδομένα που αφορούν το χρονικό στιγμιότυπο του γεγονότος, τι προκάλεσε την εκτέλεση και πεδία δεδομένων. Παρέχεται λοιπόν μια σύνδεση ανάμεσα στα δεδομένα που προέρχονται από τις διεργασίες και τα μοντέλα που έχουν σχεδιαστεί για αυτές. Οι τρεις κύριες τεχνικές της PM παρουσιάζονται παρακάτω:

2.1. Ανεύρεση Διεργασιών (Process Discovery)

Η Ανεύρεση Διεργασιών απαρτίζεται από μια σειρά εργαλείων και τεχνικών που μπορούν αυτόματα να κατασκευάσουν μια απεικόνιση των διεργασιών της επιχείρησης, χρησιμοποιώντας μόνο δεδομένα από ένα event log. Παραδείγματα τέτοιων τεχνικών είναι οι Αλγόριθμοι Εξόρυξης Ελέγχου Ροής (Control Flow Mining Algorithms) και οι Αλγόριθμοι Εξόρυξης Κοινωνικού Δικτύου (Social Network Mining Algorithms).

Με τις τεχνικές της Ανεύρεσης Διεργασιών μπορούν να οριστούν, να χαρτογραφηθούν και να αναλυθούν οι υπάρχουσες διεργασίες της επιχείρησης, όπως επίσης και οι δομές που την διέπουν (επιχειρησιακές, κοινωνικές, ελέγχου κλπ.). Όπως και η Ανάλυση Διεργασιών, έτσι και οι τεχνικές αυτές παρέχουν στις επιχειρήσεις κατανόηση των διεργασιών τους και μπορεί να αποτελέσουν το πρώτο βήμα στην υλοποίηση των επιχειρησιακών διεργασιών με κάποιο εργαλείο ροής εργασιών. Οι δύο αυτές τεχνικές μπορούν να χρησιμοποιηθούν σε συνδυασμό μεταξύ τους για να εντοπιστούν πιο γρήγορα δυσλειτουργίες και σημεία συμφόρησης.

2.2. Έλεγχος Συμμόρφωσης (Conformance Checking)

Ο Έλεγχος Συμμόρφωσης εφαρμόζεται εκ των υστέρων στα δεδομένα που υπάρχουν στα event logs με σκοπό να αναλυθεί αν τα συμβάντα που καταγράφηκαν αντιστοιχούν στο μοντέλο της διεργασίας ή και το ανάποδο και να εντοπισθούν ανεπιθύμητες συμπεριφορές. Η διαφορά του Ελέγχου Συμμόρφωσης από την Ανεύρεση Διεργασιών έγκειται στο γεγονός ότι για να εφαρμοστεί Έλεγχος Συμμόρφωσης, είναι αναγκαίο να υπάρχουν τόσο ένα μοντέλο όσο και ένα event log.

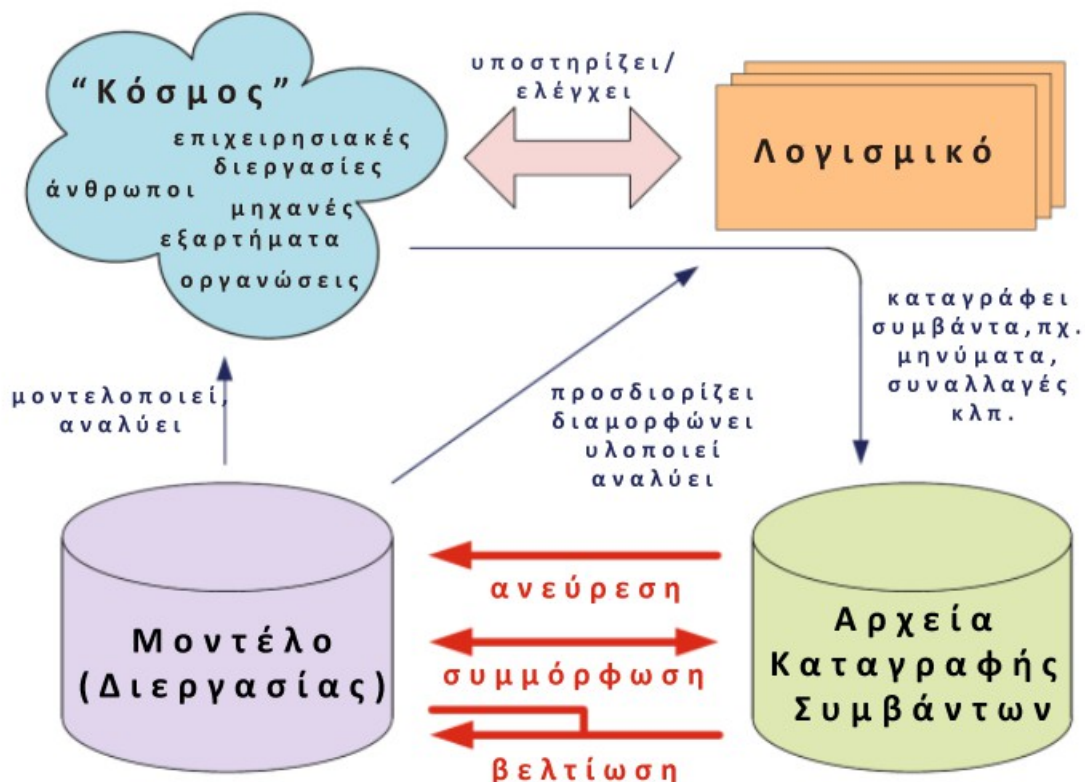
Οι περισσότερες τεχνικές Ελέγχου Συμμόρφωσης βασίζονται στην αρχή της Επανάληψης (Replay) δηλαδή ότι το αρχείο καταγραφής συμβάντων επαναλαμβάνεται μέσα στο μοντέλο της διεργασίας. Οι συνήθεις μετρικές συμμόρφωσης είναι η Καταλληλότητα (Fitness), δηλαδή κατά πόσο το μοντέλο

επιτρέπει τη συμπεριφορά που εμφανίζεται στο αρχείο καταγραφής συμβάντων, και η Απλότητα (Simplicity), η οποία ορίζεται από το λιγότερο σύνθετο μοντέλο που μπορεί να εξηγήσει τη συμπεριφορά που παρατηρείται στο αρχείο καταγραφής συμβάντων.

2.3. Βελτίωση Μοντέλου (Model Enhancement)

Το τελευταίο στάδιο είναι η Επέκταση του Μοντέλου που έχει δημιουργηθεί. Ο σκοπός αυτής της μεθόδου δεν είναι να ελεγχθεί η εκτέλεση της διεργασίας, αλλά να επεκταθεί ή να εμπλουτιστεί το υπάρχον μοντέλο μιας διεργασίας χρησιμοποιώντας πληροφορίες που έχουν αντληθεί από την πραγματική διεργασία και έχουν καταγραφεί σε κάποιο αρχείο καταγραφής συμβάντων.

Παραδείγματα αυτής της μεθόδου είναι η Επισκευή (Repair), δηλαδή η τροποποίηση τους μοντέλου που έχει δημιουργηθεί ώστε να αντικατοπτρίζει καλύτερα την πραγματικότητα, και η Επέκταση (Extension), δηλαδή η προσθήκη επιπλέον χρήσιμων πληροφοριών μέσα από ανάλυση των δεδομένων που υπάρχουν από την εκτέλεση της διεργασίας (π.χ. σήμανση των σημείων συμφόρησης επάνω στο μοντέλο της διεργασίας).



Σχήμα 1: Τοποθέτηση των τριών κύριων τύπων Εξόρυξης Διεργασιών.

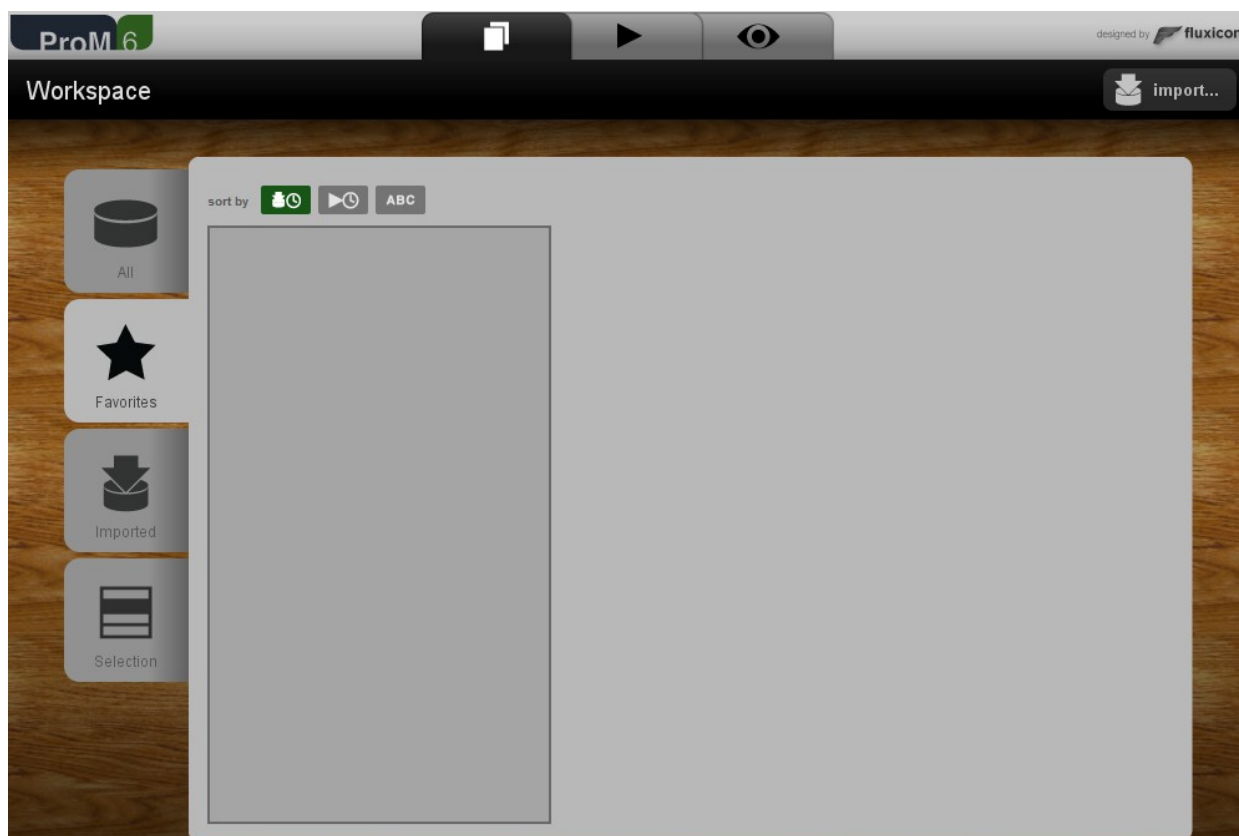
3. ProM Framework: Μια σύγχρονη εφαρμογή Εξόρυξης Διεργασιών

Το πλαίσιο ProM (συντομογραφία του Process Mining Framework) είναι ένα πλαίσιο ανοιχτού λογισμικού για αλγορίθμους PM. Αναπτύχθηκε από τον Wil van der Aalst και την ομάδα του στο Eindhoven University of Technology και είναι διαθέσιμο online χωρίς χρέωση από τη σχετική ιστοσελίδα^[3.1]. Το πλαίσιο ProM είναι εύχρηστο και επεκτάσιμο, υποστηρίζει ένα μεγάλο εύρος τεχνικών PM σε μορφή plug-in και είναι ανεξάρτητο πλατφόρμας αφού έχει υλοποιηθεί σε Java.

3.1. Διεπαφή Χρήστη

Μέσα από το ProM μπορούν να εφαρμοστούν όλες οι μέθοδοι και τεχνικές PM που αναφέρθηκαν παραπάνω. Πριν αναφερθούμε σε συγκεκριμένες λειτουργίες και εξετάσουμε ένα παράδειγμα θα ήταν σκόπιμο να γίνει μια εισαγωγή στο γραφικό περιβάλλον της εφαρμογής ProM (τρέχουσα έκδοση 6.5.1). Επιπλέον στο Παράρτημα II υπάρχουν οδηγίες για την εκκίνηση της εφαρμογής.

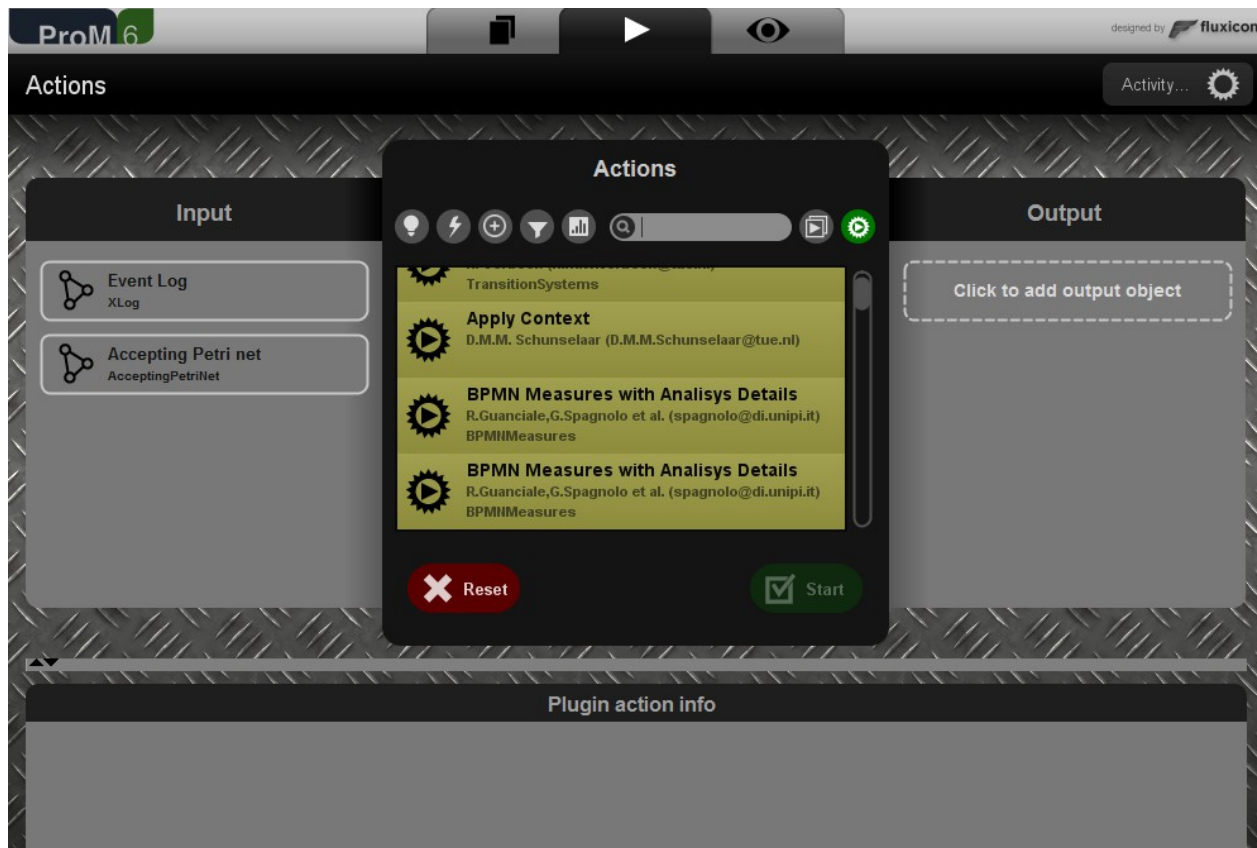
Στην κορυφή του παραθύρου υπάρχουν τρία κουμπιά, κάθε ένα από τα οποία ανοίγει μια καρτέλα με λειτουργίες. Αυτές από αριστερά προς τα δεξιά είναι οι: “Workspace”, “Action” και “View”.



Εικόνα 1: Η καρτέλα Workspace.

Η καρτέλα Workspace είναι η πρώτη καρτέλα που εμφανίζεται μόλις ανοίξουμε την εφαρμογή (Εικόνα 1). Ουσιαστικά σε αυτή την καρτέλα εμφανίζονται όλοι οι πόροι (resources) που έχουμε στη διάθεσή μας (όπως αρχεία καταγραφής συμβάντων ή δίκτυα Petri). Οι πόροι αυτοί μπορεί να έχουν εισαχθεί από το χρήστη χρησιμοποιώντας το κουμπί “Import...” πάνω δεξιά ή να έχουν παραχθεί με χρήση κάποιας τεχνικής PM από την καρτέλα Action.

Η καρτέλα Action αποτελείται από τρεις στήλες, τις Input, Actions και Output (Εικόνα 2). Επιλέγοντας μία δράση από τη μεσαία στήλη, δηλαδή μια τεχνική που θέλουμε να εφαρμόσουμε σε κάποια δεδομένα, εμφανίζονται αμέσως τα κατάλληλα στοιχεία στις στήλες Input και Output. Έπειτα μπορούμε να επιλέξουμε τα δεδομένα με τα οποία θέλουμε να τροφοδοτήσουμε τη διαδικασία στη στήλη Input, χρησιμοποιώντας κάποιους από τους πόρους που έχουμε στη διάθεσή μας. Στη στήλη Output εμφανίζονται οι πληροφορίες που θα ληφθούν όταν ολοκληρωθεί η διαδικασία. Όταν όλα τα στοιχεία της στήλης Input έχουν κατάλληλα δεδομένα, ενεργοποιείται το κουμπί “Start” με το οποίο ξεκινά η διαδικασία. Αν, επιπλέον, υπάρχουν παράμετροι που μπορεί να ρυθμίσει ο χρήστης για τη διαδικασία θα εμφανιστούν σε ένα νέο παράθυρο μόλις πατηθεί το κουμπί.



Εικόνα 2: Η καρτέλα Action.

Μόλις ολοκληρωθεί ένα Action μεταφερόμαστε στην καρτέλα View όπου εμφανίζονται τα αποτελέσματά του. Η καρτέλα View εμφανίζει μια προβολή ενός πόρου ή μια επισκόπηση όλων των πόρων που μπορούν να προβληθούν.

3.2. Μετατροπή κατάλληλων αρχείων σε Event Logs

Σε πολλές περιπτώσεις, τα δεδομένα που έχουμε συλλέξει από την εκτέλεση μιας διεργασίας δεν είναι σε κατάλληλα event logs αλλά σε κάποια βάση δεδομένων ή συνήθως σε κάποιο απλό αρχείο κειμένου με μια ορισμένη δομή. Παρακάτω παρουσιάζονται συνοπτικά δυο εναλλακτικές για τη μετατροπή τέτοιων αρχείων σε event logs κατάλληλα για χρήση από το ProM Framework (XESame και Disco), ενώ επιπλέον αναφέρεται ενδεικτικά και μία τρίτη σε μορφή plug-in για το ProM που δεν είναι εγκατεστημένη από προεπιλογή (“Convert CSV to XES” – F.Mannhardt, N.Tax & D.M.M. Schunselaar).

3.2.1. XESame

Μαζί με τις νεότερες εκδόσεις του ProM Framework, παρέχεται δωρεάν και το εργαλείο XESame (τρέχουσα έκδοση 1.5.1), ένα εργαλείο που επιτρέπει την εξαγωγή event logs (σε μορφή .xes) από πηγές δεδομένων που δεν είναι event logs (πχ. αρχεία .csv ή αρχεία .txt). Θεωρείται σκόπιμο να περιγραφεί επιγραμματικά η διαδικασία χρήσης του εργαλείου XESame για μετατροπή αρχείων τύπου “Comma Separated Values” (.csv) σε αρχεία τύπου “Extensible Event Stream” (.xes), καθώς τα τελευταία είναι σε μορφή κατάλληλη για χρήση από το ProM Framework.

Στη βάση του το XESame χρησιμοποιεί, όπως και το ProM, τη γλώσσα προγραμματισμού Java. Συγκεκριμένα για τη σωστή λειτουργία του XESame είναι αναγκαία η εγκατάσταση του Java Database Connector (JDBC) και της αντίστοιχης γέφυρας με τον Open Database Connector (ODBC) που συνήθως υπάρχει προεγκατεστημένος στα περισσότερα σύγχρονα πληροφοριακά συστήματα. Τα προαπαιτούμενα αυτά μπορούν να ληφθούν χωρίς χρέωση από το διαδίκτυο^[3.11]. Τα βήματα που πρέπει να εκτελέσει ο χρήστης αφού ληφθούν τα προαπαιτούμενα είναι η δημιουργία (ή εύρεση) ενός αρχείου .csv ή .txt, η σύνδεσή του με τον ODBC, η σύνδεση του XESame με το JAR αρχείο του JDBC και έπειτα η εκτέλεση του XESame για τα κατάλληλα αρχεία που έχουν εισαχθεί και η μετατροπή τους σε αντίστοιχα .xes αρχεία. Αναλυτικές οδηγίες μπορούν να βρεθούν στη σελίδα του XESame^[3.12].

3.2.2. Disco

Μια άλλη εναλλακτική, για τη μετατροπή αρχείων .csv ή .txt σε αρχεία .xes που μπορεί να χρησιμοποιήσει το ProM, είναι το πρόγραμμα Disco (τρέχουσα έκδοση 1.9.0, στο παρελθόν ονομαζόταν Nitro). Είναι ελεύθερο για μικρά σετ δεδομένων και μη εμπορική χρήση, ενώ για περισσότερες δυνατότητες είναι αναγκαία η αγορά άδειας. Επιπλέον παρέχει πρόσθετες λειτουργίες σε σχέση με το XESame όπως τη δημιουργία διαγράμματος της διεργασίας με συχνότητες, κόστη ή χρονική διάρκεια και τροποποίηση ή φιλτράρισμα των δεδομένων που θα εισαχθούν.

Όπως και το XESame, έτσι και το Disco χρησιμοποιεί τη γλώσσα προγραμματισμού Java στη βάση του, επομένως είναι ανεξάρτητο πλατφόρμας. Σε αντίθεση, όμως με το XESame δεν χρειάζεται σχεδόν καμία ρύθμιση κατά την εκκίνηση, πράγμα που το καθιστά πιο εύχρηστο. Συνολικά αποτελεί μια πιο εύελικτη λύση από το XESame για μικρά σετ δεδομένων, με μοναδικό περιορισμό την άδειά του.

Στις σελίδες που ακολουθούν, όλα τα event logs που περιγράφονται έχουν δημιουργηθεί με χρήση του Disco δεδομένου του μικρού όγκου τους και της μεγαλύτερης ευελιξίας και ευχρηστίας του.

Παρόλα αυτά, τα event logs που χρησιμοποιούνται παρακάτω μπορούν όλα να παραχθούν και με τη χρήση XESame χωρίς ουσιαστικές διαφορές σε τίποτα, πέραν ίσως μερικών τεχνικών χαρακτηριστικών και υπογραφών που περιέχουν εσωτερικά και είναι αδιάφορες στον τελικό χρήστη.

4. Τρέχον Παράδειγμα

Έχοντας αναφέρει όλα τα παραπάνω, είναι δυνατόν πλέον να παρουσιαστεί ένα παράδειγμα εφαρμογής των μεθόδων της PM με χρήση του εργαλείου ProM. Το τρέχον παράδειγμα αντλεί έμπνευση από τα σύγχρονα ηλεκτρονικά καταστήματα (e-shops) και τα συστήματα που χρησιμοποιούνται σε αυτά. Τα δεδομένα του παραδείγματος έχουν προκύψει από προσομοίωση που δημιουργήθηκε σε υπολογιστή. Όλα τα event logs που χρησιμοποιούνται παρουσιάζονται σε απλή και κατανοητή μορφή στο Παράρτημα Ι.

4.1. Παραδοχές, περιορισμοί και δομή των δεδομένων

Το τρέχον παράδειγμα αναφέρεται σε ένα ηλεκτρονικό κατάστημα (e-shop) το οποίο δέχεται παραγγελίες από χρήστες μέσω μιας ιστοσελίδας, τις επεξεργάζεται με το προσωπικό που διαθέτει και τελικά τις αποστέλλει στους πελάτες του είτε με δικό του μεταφορικό είτε μέσω μιας εταιρείας courier. Με βάση τα παραπάνω, θεωρείται δεδομένη τόσο η ύπαρξη ενός συστήματος που βοηθά την επιχείρηση με τη διαχείριση των παραγγελιών (καταχώρηση, ενημέρωση κλπ.) όσο και ενός συστήματος καταγραφής συμβάντων.

Λόγω των χρονικών και τεχνικών πλαισίων της παρούσας εργασίας, το παράδειγμα που παρουσιάζεται περιλαμβάνει συνολικά 40 καταγεγραμμένες περιπτώσεις, η καθεμία εκ των οποίων αποτελείται από περίπου 5 με 15 γεγονότα. Οι 40 αυτές περιπτώσεις χωρίζονται σε 5 event logs ώστε να μπορούν να χρησιμοποιηθούν πιο εύκολα και να παράγουν αποτελέσματα γρηγορότερα και χωρίς μεγάλες ανάγκες υπολογιστικών πόρων.

Όσον αφορά στη δομή των δεδομένων, ακολουθείται ένα από τα πιο συνηθισμένα πρότυπα οργάνωσης event logs. Οι στήλες που απαρτίζουν μια εγγραφή είναι οι εξής:

- **Case ID (Αναγνωριστικό Περίπτωσης):** Αποτελεί το αναγνωριστικό της κάθε περίπτωσης (στο τρέχον παράδειγμα των κωδικό παραγγελίας), δηλαδή έναν αριθμό που προσδιορίζει μοναδικά την κάθε περίπτωση. Στο τρέχον παράδειγμα κάθε περίπτωση κωδικοποιείται με ένα αύξοντα αριθμό.
- **Timestamp (Χρονικό Στιγμιότυπο):** Καταγράφει τη χρονική στιγμή που εκτελέστηκε ένα γεγονός. Στο τρέχον παράδειγμα η μορφή των timestamps είναι dd/MM/yy HH:mm.
- **Activity (Δραστηριότητα):** Κάθε ένα γεγονός αποτελεί μια ξεχωριστή δραστηριότητα για την περίπτωση που μελετάται. Στο τρέχον παράδειγμα υπάρχουν τις περισσότερες φορές γεγονότα τόσο για την εκκίνηση μιας διαδικασίας όσο και για την ολοκλήρωσή της.
- **Resource (Πόρος):** Ως πόροι αναφέρονται τα πρόσωπα ή τα συστήματα που εκτελούν τη διαδικασία. Στο τρέχον παράδειγμα οι πόροι μας είναι τόσο το προσωπικό της επιχείρησης, όσο και το πληροφοριακό σύστημα που καταχωρεί κάποια γεγονότα στο event log.

Παρακάτω παρουσιάζονται οι μέθοδοι που εφαρμόστηκαν στο τρέχον παράδειγμα με τη βοήθεια του ProM Framework, καθώς και τα αποτελέσματά τους, όπου αυτό είναι δυνατόν.

4.2. Ανεύρεση Διεργασιών

Όπως αναφέρθηκε και παραπάνω, το πρώτο βήμα της PM είναι η Ανεύρεση Διεργασιών. Σε αυτή την ενότητα θα μελετηθούν κάποια ενδεικτικά plug-ins του ProM τα οποία εφαρμόζουν τεχνικές της Ανεύρεσης Διεργασιών τόσο σχετικά με τις δραστηριότητες όσο και σχετικά με τους πόρους.

4.2.1. Discover using Decomposition

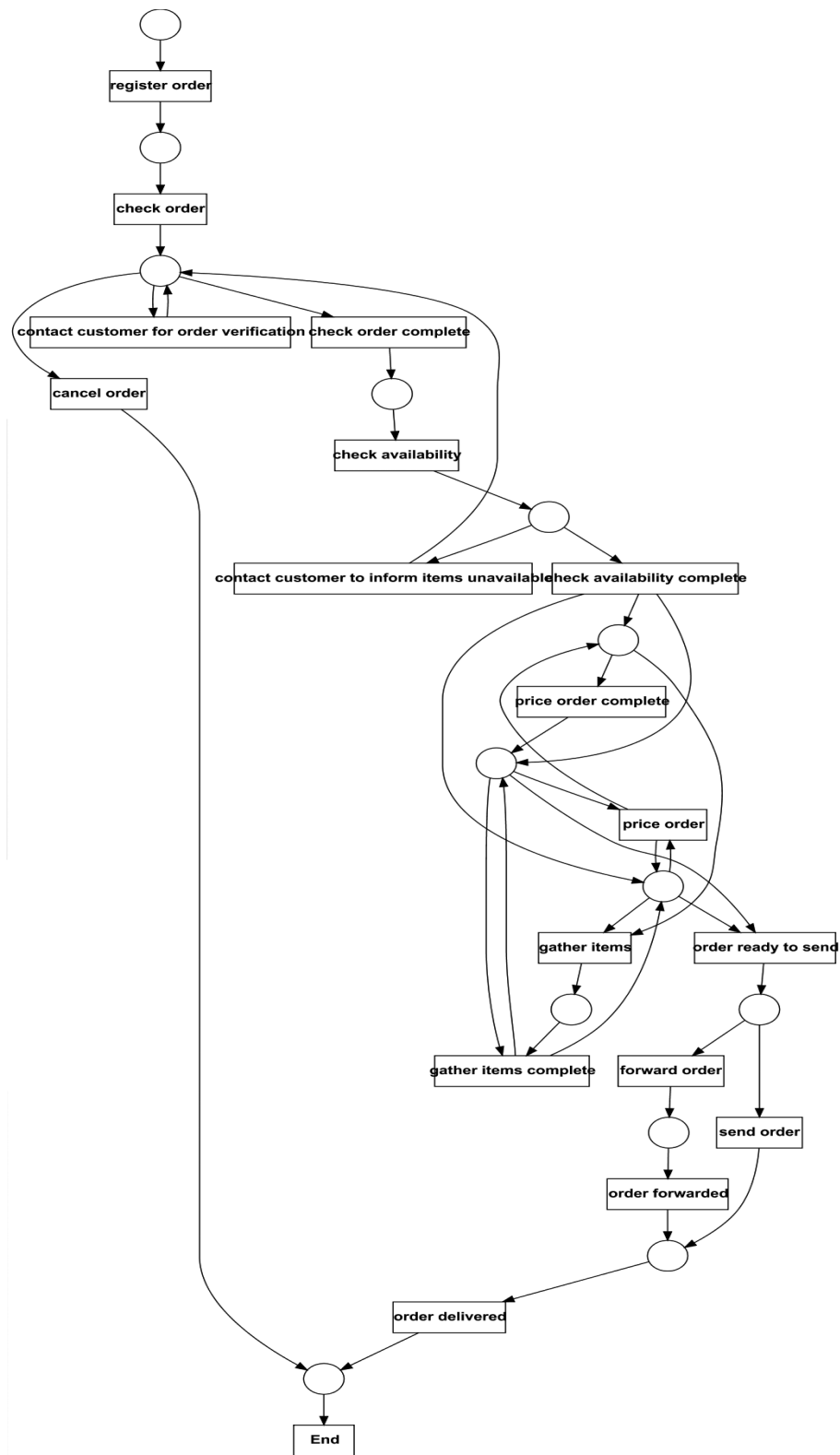
Ένα από τα πιο χρήσιμα plug-in του ProM είναι το “Discover using Decomposition” που έχει υλοποιηθεί από τον H.M.W. Verbeek. Το plug-in αυτό χρησιμοποιεί αποσύνθεση σε ένα event log για να βρει ένα δίκτυο Petri (στο εξής Petri Net). Το τελευταίο αποτελεί ένα μαθηματικό μοντέλο για την περιγραφή καταναμεμένων συστημάτων. Απεικονίζεται με ένα κατευθυνόμενο διμερή γράφο στον οποίο οι κόμβοι σηματοδοτούν μεταβάσεις (στο τρέχον παράδειγμα δραστηριότητες), όταν έχουν τη μορφή παραλληλογράμμου, ή σημεία (στο τρέχον παράδειγμα συνθήκες), όταν έχουν τη μορφή κύκλου, και τα τόξα περιγράφουν ποια σημεία είναι συνθήκες εκ των προτέρων ή εκ των υστέρων για ποιες μεταβάσεις.

Πέραν του παραπάνω plug-in, υπάρχουν κι άλλα που παράγουν ένα πανομοιότυπο Petri Net ή κάποιο αρκετά παρεμφερές με βάση τις παραμέτρους που θα ορίσει ο χρήστης. Τέτοια είναι τα “Mine for a Petri Net using ILP” - T. van der Wiel και “ILP-Based Process Discovery” - S.J. van Zeist, τα οποία είναι πιο ευέλικτα με βάση τις επιλογές του χρήστη.

Το plug-in που χρησιμοποιούμε αρχικά διαχωρίζει το event log σε μικρότερα event logs, έπειτα βρίσκει ένα υποδίκτυο για κάθε ένα από αυτά χρησιμοποιώντας τον ίδιο αλγόριθμο ανεύρεσης, στη συνέχεια αυτά τα υποδίκτυα συνενώνονται σε ένα κατάλληλο Petri Net και τελικά εφαρμόζονται γνωστές αναγωγικές τεχνικές και παράγεται το τελικό Petri Net. Αρχικά εκτελούμε το plug-in αυτό με ένα event log 8 εγγραφών (Event Log 1). Το event log που χρησιμοποιείται περιέχει πολύ ευρεία δεδομένα ούτως ώστε το Petri Net που θα προκύψει να περιγράφει τη διεργασία που εκτελείται όσο το δυνατόν καλύτερα.

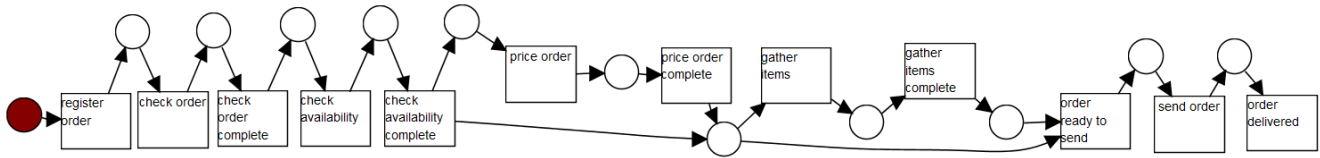
Το Petri Net που προέκυψε (Σχήμα 2) μας δίνει μια καλή εικόνα της διεργασίας που μελετάται, ακόμα και σε περιπτώσεις εξαιρέσεων. Συνοπτικά η διεργασία ακολουθεί τα εξής στάδια:

1. Εκχώρηση παραγγελίας από το σύστημα
2. Έλεγχος παραγγελίας, διαχείριση εξαιρέσεων αν προκύψουν και ακύρωση αν χρειαστεί
3. Έλεγχος διαθεσιμότητας, διαχείριση εξαιρέσεων αν προκύψουν και ακύρωση αν χρειαστεί
4. Συγκέντρωση προϊόντων ή τιμολόγηση παραγγελίας
5. Τιμολόγηση παραγγελίας ή συγκέντρωση προϊόντων
6. Ενημέρωση από το σύστημα ότι η παραγγελία είναι έτοιμη για αποστολή
7. Αποστολή παραγγελίας με οδηγό της εταιρείας ή προώθηση στην εταιρεία courier, η οποία με τη σειρά της διαχειρίζεται την αποστολή
8. Ολοκλήρωση της παραγγελίας



Σχήμα 2: Το Petri Net που προέκυψε από την εκτέλεση του plug-in "Discover using Decomposition" για το Event Log 1.

Το παραπάνω Petri Net, αν και περιγράφει πολύ καλά τη διεργασία που μελετάται, είναι αρκετά πολύπλοκο με μια πρώτη ματιά. Επομένως, θα ήταν σκόπιμο να παρουσιαστεί κι ένα δεύτερο Petri Net, με χρήση ενός άλλου event log (Event Log 2) με πιο απλά δεδομένα, τα οποία να αποτελούν τη βέλτιστη περίπτωση, δηλαδή μια πλήρη εκτέλεση χωρίς εξαιρέσεις και ειδικές περιπτώσεις. Στο τρέχον παράδειγμα αυτό σημαίνει ότι δεν εμφανίζονται τα activities “contact customer for order verification”, “contact customer to inform items unavailable”, “cancel order”, “forward order” και “order forwarded”. Το Petri Net που προκύπτει (Σχήμα 3) είναι σαφώς πιο απλό καθώς περιγράφει μια πιο απλοϊκή δομή της διεργασίας. Μέσα από το τελευταίο, μπορεί εύκολα να γίνει κατανοητό γιατί ένας τέτοιος γράφος είναι κατάλληλος για την περιγραφή και μοντελοποίηση ροής εργασιών.



Σχήμα 3: Το Petri Net που προέκυψε από την εκτέλεση του plug-in “Discover using Decomposition” για το Event Log 2.

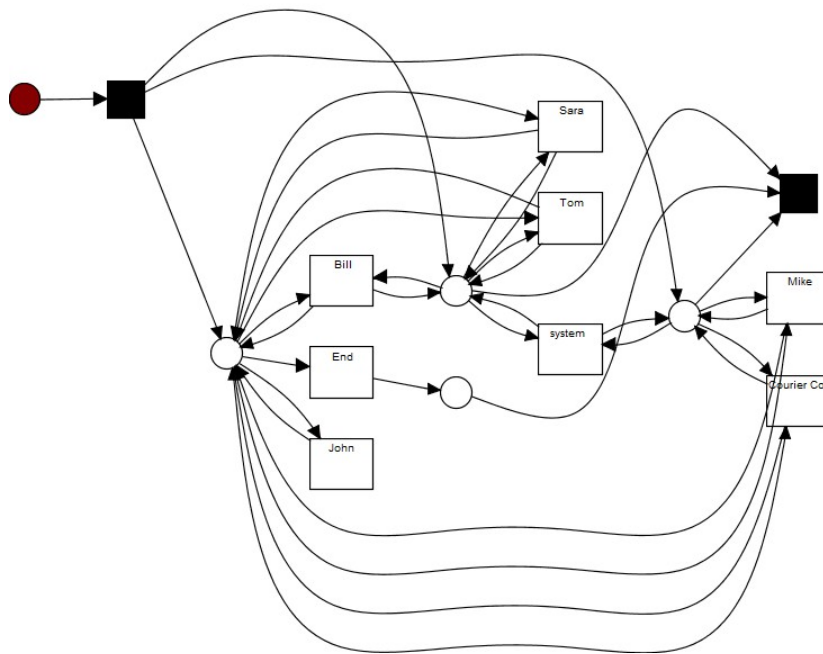
Λόγω της δομής και του τρόπου αναπαράστασής τους, τα Petri Net παρέχουν ευκολία πρόσβασης και κατανόησης ακόμα και σε χρήστες που δεν τα έχουν χρησιμοποιήσει στο παρελθόν, περιέχουν όλα τα θεμελιώδη στοιχεία για τη μοντελοποίηση της ροής εργασιών μιας διεργασίας με τρόπο που δεν επιτρέπει παρερμηνείες, ενώ παράλληλα χρησιμοποιούν τυπική σημειολογία ούτως ώστε η ροή εργασιών μιας διεργασίας που έχει μοντελοποιηθεί με Petri Net να είναι σαφής και ακριβής. Επιπλέον τα Petri Net ως μαθηματικό μοντέλο έχουν αναλυθεί σε βάθος και μπορούν να χρησιμοποιηθούν για πολλές τεχνικές ανάλυσης (παραδείγματα χρήσης τους σε εφαρμογές PM με το ProM παρουσιάζονται παρακάτω).

Πέραν των παραπάνω, το Petri Net που προκύπτει από το plug-in “Discover using Decomposition” ικανοποιεί και στον ορισμό του Δικτύου Ροής Εργασιών (στο εξής Workflow net). Ένα Workflow net είναι ένα Petri Net το οποίο α) έχει ένα σαφές σημείο εκκίνησης i και ένα σημείο τερματισμού o και β) για κάθε μετάβαση t υπάρχει ένα μονοπάτι από το i στο o που περιλαμβάνει την t .

Έχοντας αναφέρει τα παραπάνω, είναι δυνατόν πλέον να διατρέξουμε το Petri Net για να δούμε την ακριβή δομή της διεργασίας. Το σημείο εκκίνησης είναι ο κόκκινος κύκλος στα αριστερά και το σημείο τερματισμού το τετράγωνο “order delivered”. Ξεκινώντας, λοιπόν, παρατηρεί κανείς ότι η διεργασία ακολουθεί μια γραμμική πορεία μέχρι το σημείο που η διεργασία εκκινεί τις δραστηριότητες “price order” και “gather items”, οι οποίες μπορούν να γίνουν η μία μετά την άλλη με οποιαδήποτε σειρά ή ταυτόχρονα, και στη συνέχεια ακολουθεί και πάλι μια γραμμική πορεία μέχρι το τελικό της στάδιο.

Γενικεύοντας τα συμπεράσματά μας για το Σχήμα 3, είναι δυνατό να εξηγήσουμε το Σχήμα 2 που περιγράφει την πλήρη δομή της διεργασίας. Το σημείο εκκίνησης είναι και πάλι ο κόκκινος κύκλος και το σημείο τερματισμού είναι το τετράγωνο “End”. Βλέπουμε ότι η διεργασία ακολουθεί παρόμοια πορεία, με προαιρετικές κάποιες καταστάσεις (πχ. “contact customer for order verification”), ενώ σε μερικά σημεία προστίθενται διαφορετικές γραμμικές πορείες που μπορεί να ακολουθήσει.

Πέρα από την πλευρά που αφορά στις δραστηριότητες που περιλαμβάνει μια διεργασία, πολλές φορές είναι σκόπιμη και η μελέτη της Κοινωνικής Πλευράς (Social Aspect) μιας διεργασίας για την καλύτερη κατανόηση και μοντελοποίησή της. Επομένως είναι δυνατόν να εξάγουμε ένα Petri Net μέσα από ένα event log, τέτοιο ώστε οι κόμβοι του να αντιστοιχούν σε πόρους (Resources) αντί για δραστηριότητες (Activites). Ένα τέτοιο δίκτυο δίνεται στο Σχήμα 4 και αφορά το Event Log 1.



Σχήμα 4: Το Petri Net που προέκυψε από το plug-in "Discover using Decomposition" για το Event Log 1 με βάση τη στήλη Resource.

Το Petri Net που προέκυψε παρουσιάζει χρησιμοποιώντας τους ίδιους κανόνες και λογική ένα γράφο που δείχνει τις μεταβάσεις που μπορεί να συμβούν μέσα στη διεργασία που μελετάται. Φυσικά ένα τέτοιο δίκτυο, αν και έχει κάποια αξία για τη μοντελοποίηση της διεργασίας, δεν είναι αρκετό για να μελετήσει την Κοινωνική Πλευρά της, γι' αυτό και έχουν αναπτυχθεί πιο εξειδικευμένα μοντέλα που παρουσιάζονται στην ενότητα 4.2.4.

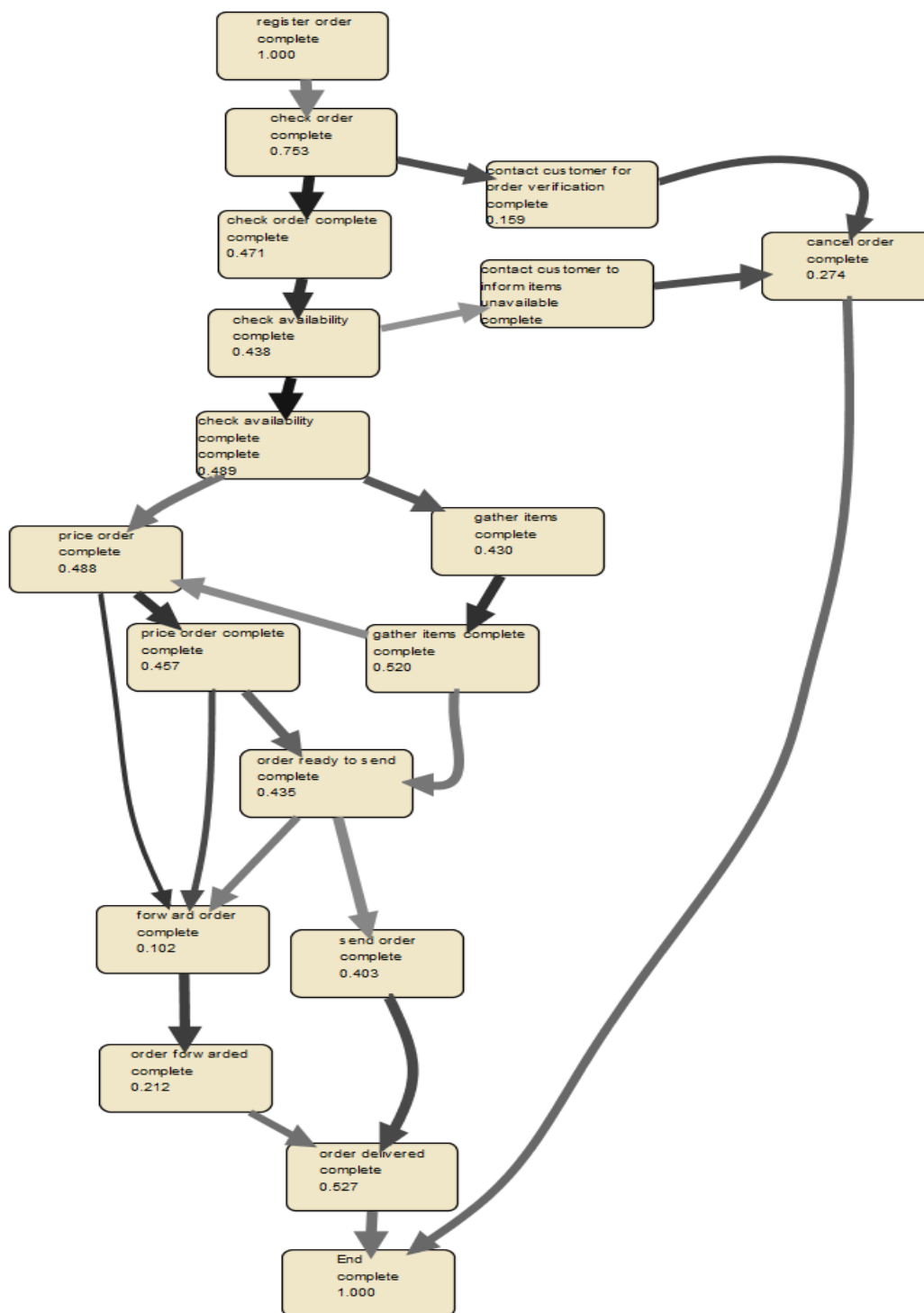
4.2.2. Mine for a Fuzzy Model

Ένα άλλο χρήσιμο plug-in Ανεύρεσης Διεργασιών είναι το “Mine for a Fuzzy Model” που εξάγει ένα μοντέλο της διεργασίας από το event log και έχει υλοποιηθεί από τον H.M.W. Verbeek. Το plug-in αυτό ευρίσκει ένα Ασαφές Μοντέλο (Fuzzy Model) από τα δεδομένα που εισάγονται, δηλαδή ένα μοντέλο που εμπίπτει στους κανόνες της ασαφούς λογικής. Η τελευταία αποτελεί μια επέκταση της γνωστής δίτιμης Boolean λογικής (όπου οι μεταβλητές παίρνουν μόνο τις ακέραιες τιμές 0 ή 1), τέτοια ώστε κάθε μεταβλητή να παίρνει πραγματικές τιμές ανάμεσα στο 0 και το 1, καθιστώντας τις τιμές αυτές “βαθμούς αληθείας”. Κατά την εκκίνηση αυτού του plug-in απαιτείται από το χρήστη η επιλογή μετρικών (στο τρέχον παράδειγμα χρησιμοποιήσαμε την προεπιλογή του ProM, δηλαδή όλες οι μετρικές active με βάρος 1) και εξασθένησης (στο τρέχον παράδειγμα χρησιμοποιούνται οι προεπιλεγμένες τιμές του ProM). Επιπλέον χρησιμοποιούμε και πάλι το Event Log 1 για να έχουμε μια πλήρη εικόνα της διεργασίας.

Το μοντέλο που προκύπτει (Σχήμα 5) δε διαφέρει κατά πολύ από το Petri Net που προέκυψε από το προηγούμενο plug-in, παρόλα αυτά είναι πιο απλό στην κατανόησή του, καθώς οι μεταβάσεις υπό συνθήκη έχουν συγχωνευθεί σε καταστάσεις και τόξα με γκρι χρώμα. Επιπλέον σε κάθε μια κατάσταση είναι δυνατόν να δει κανείς την πιθανότητα που έχει αυτή να συμβεί κατά την εκτέλεση μίας περίπτωσης, επομένως μπορούμε να προβλέψουμε πόσο πιθανό είναι να υπάρχει κάποιο πρόβλημα στη διαδικασία που οδηγεί σε ακύρωση της παραγγελίας ή πόσο πιθανό είναι να χρειαστεί να επικοινωνήσει κάποιος με τον πελάτη.

Αν διατρέξει κάποιος το μοντέλο που προκύπτει, είναι προφανές ότι μετά τον αρχικό έλεγχο της παραγγελίας, δεδομένου ότι σε αυτόν δεν προκύψει κάποια εξαίρεση και δεν ακυρωθεί η παραγγελία, μπορεί είτε να γίνει συγκέντρωση των προϊόντων ή τιμολόγηση της παραγγελίας, αλλά μόνο αφότου ολοκληρωθούν και τα δύο προχωράει η διεργασία. Στη συνέχεια η παραγγελία αποστέλλεται και η διεργασία ολοκληρώνεται. Όπως είναι λογικό η αρχική και η τελική κατάσταση έχουν και οι δύο την τιμή 1.0, πράγμα που σημαίνει ότι δεν υπάρχουν διαρροές από τη διεργασία, επομένως ότι στο event log που χρησιμοποιείται όλα τα στιγμιότυπα που περιέχονται έχουν ολοκληρωθεί.

Βλέποντας το ασαφές μοντέλο, είναι πολύ εύκολο να κατανοήσει κανείς γιατί αυτό αποτελεί μια καλή αναπαράσταση της διεργασίας, καθώς είναι ουσιαστικά μία απλουστευμένη αναπαράσταση σε σχέση με το Petri Net που προέκυψε προηγουμένως. Μέσω αυτού, λοιπόν, μπορεί κανείς πολύ εύκολα να έχει μια πρώτη εικόνα του πώς λειτουργεί μια διεργασία, από ποια στάδια αποτελείται και τις διαφορετικές διαδρομές που μπορεί να ακολουθήσει μέχρι την ολοκλήρωσή της. Ακόμη, μέσω των χρωμάτων που παίρνουν τα βέλη του μοντέλου είναι αμέσως κατανοητό το πόσο πιθανό είναι να ακολουθήσει κάποιο μονοπάτι η διεργασία όταν βρεθεί σε μια ορισμένη κατάσταση. Τέλος, η συνολική πιθανότητα να βρεθεί σε μία κατάσταση η διεργασία ποσοτικοποιείται με τις τιμές της ασαφούς λογικής, έτσι ώστε να είναι εύκολο να ταυτοποιηθεί η πιθανότητα να προκύψει κάποια εξαίρεση ή κάτι παρόμοιο (πχ. στο τρέχον παράδειγμα φαίνεται ότι η πιθανότητα να προκύψει κάτι που θα οδηγήσει σε ακύρωση της παραγγελίας είναι 0.274).



Σχήμα 5: Το ασαφές μοντέλο που προέκυψε από το plug-in "Mine for a Fuzzy Model" για το Event Log 1.

4.2.3. Discover Matrix

Μία ακόμη εφαρμογή της PM που αφορά στην Ανεύρεση Διεργασιών είναι αυτή του plug-in "Discover Matrix", επίσης σε υλοποίηση H.M.W. Verbeek. Εφαρμόζοντας αυτό το plug-in εξάγουμε μια Μήτρα Περιστασιακής Δραστηριότητας (στο εξής Casual Activity Matrix), χρησιμοποιώντας ένα αλγόριθμο που υπολογίζει την τιμή κάθε κελιού στη μήτρα. Ένα κελί παίρνει την τιμή 1 όταν υπάρχει μια περιστασιακή σχέση από τη γραμμή στη στήλη, την τιμή -1 όταν δεν υπάρχει και την τιμή 0 όταν δεν είναι γνωστό.

Παρακάτω παρουσιάζονται ενδεικτικά δύο Casual Activity Matrix για το Event Log 1, μία που αφορά στη δραστηριότητα (activity) της διεργασίας (Πίνακας 1) και μία που αφορά στους πόρους (resource) της (Πίνακας 2).

	cancel order	check availability	check availability complete	check order	check order complete	contact customer for order verification	contact customer to inform items unavailable	forward order	gather items	gather items complete	order delivered	order forwarded	order ready to send	price order	price order complete	register order	send order
End	-0.66	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
cancel order	0.67	-0.73	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
check availability	-1	-1	-0.79	0.86	-1	-1	-1	0.5	-1	-1	-1	-1	-1	-1	-1	-1	-1
check availability complete	-1	-1	-1	-0.78	-1	-1	-1	-1	-1	0.75	-1	-1	-1	-1	0.75	-1	-1
check order	-1	-1	-1	-1	-0.8	0.86	0.67	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
check order complete	-1	-1	0.87	-1	-1	-0.8	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
contact customer for order verification	-1	0.5	-1	-1	-1	0.5	-0.73	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
contact customer to inform items unavailable	-1	0.5	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
forward order	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	0.5	-1	-1	-1	-1	-1
gather items	-1	-1	-1	-1	-1	-1	-1	-1	-1	-0.78	0.83	-1	-1	-1	-1	-0.56	-1
gather items complete	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	0.75	0.75	-1	-1	-1
order delivered	0.85	-1	-1	-1	-1	-1	-1	-1	-1	-1	-0.78	-1	-1	-1	-1	-1	-1
order forwarded	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	0.5	-1	-1	-1	-1	-1
order ready to send	-1	-1	-1	-1	-1	-1	-1	0.5	-1	-1	-1	-1	-1	-1	-1	-1	-1
price order	-1	-1	-1	-1	-1	-1	-1	-1	0.5	-1	-1	-1	-1	-1	-1	0.83	-1
price order complete	-1	-1	-1	-1	-1	-1	-1	-1	-0.34	0.5	-1	-1	0.75	-1	-0.78	-1	-1
register order	-1	-1	-1	-1	0.89	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-0.75	-1
send order	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	0.833333	-1	-1	-1	-1	-1	-1

Πίνακας 1: Η μήτρα που προκύπτει με το plug-in "Discover Matrix" για τη στήλη Activity.

	Bill	Courier Co.	End	John	Mike	Sara	Tom	system
Bill	0	-1	0.75	-1	-1	-1	-1	-1
Courier Co.	-1	-1	0.5	-1	-1	-1	-1	-1
End	-1	-1	-0.36	-1	-1	-1	-1	-1
John	-1	-1	0.5	0	0.8	-0.74	-1	-0.56
Mike	-1	-1	0.5	-1	-1	0	0	0.375
Sara	-1	-1	0.5	-0.48	0	0	-1	-1
Tom	-1	-1	0.67	-1	-1	-1	0	-1
system	0.8	-1	-1	-0.3	-0.375	0.75	0.67	-0.59

Πίνακας 2: Η μήτρα που προκύπτει με το plug-in "Discover Matrix" για τη στήλη Resource.

Οι παραπάνω μήτρες είναι εύκολο να ερμηνευθούν, καθώς η πρώτη εκφράζει τις σχέσεις μεταξύ των δραστηριοτήτων που υπάρχουν μέσα στη διεργασία, ενώ η δεύτερη αφορά τις σχέσεις μεταξύ των χρηστών που συμμετέχουν στη διεργασία. Δεδομένων των παραπάνω, είναι εύκολα αναγνωρίσιμη μέσω της πρώτης μήτρας η πορεία διαδικασιών που ακολουθεί συνήθως ένα στιγμιότυπο της διεργασίας, η οποία δε διαφέρει σε τίποτα από το ασαφές μοντέλο του Σχήματος 5. Μέσα από τη δεύτερη, μπορεί κανείς να διακρίνει ποιος χρήστης μεταφέρει εργασία σε κάποιον άλλο, δημιουργώντας μία πρώτη εικόνα της Κοινωνικής Πλευράς της διεργασίας. Οι σχέσεις αυτές μεταξύ των χρηστών αναλύονται στην ενότητα 4.2.4.

4.2.4. Mine for a Social Network

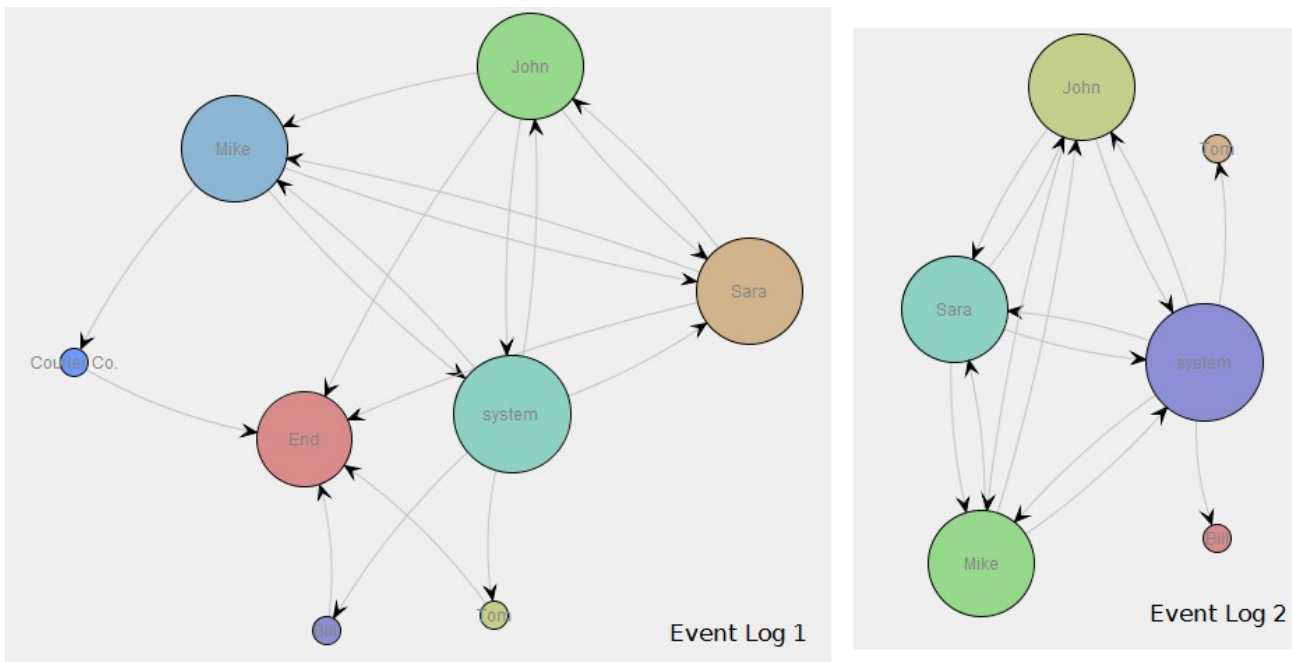
Όπως αναφέρθηκε παραπάνω, σημαντική για την κατανόηση και μοντελοποίηση μιας διεργασίας είναι μεταξύ άλλων και η Κοινωνική Πλευρά της. Για το σκοπό αυτό, μελετώνται σε αυτή την ενότητα κάποια από τα plug-ins της οικογένειας “Mine for a Social Network” (συγκεκριμένα τα “Mine for a Handover-of-Work Social Network”, “Mine for a Similar-Task Social Network” και “Mine for a Subcontracting Social Network”) που έχουν αναπτυχθεί από τον M. Song.*

Παραπάνω, στην ενότητα 4.2.1 και συγκεκριμένα στο Σχήμα 4, είδαμε ένα Petri Net που περιέγραφε τις σχέσεις μεταξύ των χρηστών στη διάρκεια εκτέλεσης της διεργασίας. Παρόλα αυτά, ένα τέτοιο δίκτυο δεν αποτελεί το καταλληλότερο μοντέλο για την απεικόνιση αυτών των σχέσεων. Έτσι, για το σκοπό αυτό χρησιμοποιείται ένα Κοινωνιογράμμο (στο εξής Sociogram), δηλαδή μία γραφική απεικόνιση των κοινωνικών δεσμών των ατόμων. Μέσα απ' αυτό διακρίνεται η δομή των διαπροσωπικών σχέσεων σε περιπτώσεις ομάδων. Ένα Sociogram αποτελείται από κύκλους, που ο καθένας αναπαριστά ένα πρόσωπο (ή χρήστη) και ακμές, η καθεμιά εκ των οποίων αναπαριστά μία σχέση. Το μέγεθος του κύκλου δείχνει πόσο συχνά ένα χρήστης εκτελεί δραστηριότητες, ενώ οι ακμές έχουν βέλη που προσδιορίζουν την κατεύθυνση της εργασίας. Σύμφωνα με τα παραπάνω, τέτοιοι γράφοι είναι πολύ χρήσιμοι στην κατανόηση της συνοχής μέσα σε μια ομάδα χρηστών σε μια επιχείρηση, καθώς παρέχουν σημαντικά στοιχεία για το ποιος λειτουργεί σαν μέλος μιας ομάδας και ποιος λειτουργεί ατομικά.

Το πρώτο παράδειγμα που θα μελετηθεί αφορά το plug-in “Mine for a Handover-of-Work Social Network”. Το plug-in αυτό μελετά τις σχέσεις παράδοσης εργασίας μέσα σε ένα στιγμιότυπο της διεργασίας. Ως παράδοση εργασίας από τον εργαζόμενο *i* στον εργαζόμενο *j* ορίζεται η ύπαρξη δύο δραστηριοτήτων, η μία ακριβώς μετά την επόμενη, εκ των οποίων η πρώτη ολοκληρώνεται από τον *i* και η δεύτερη από τον *j*.

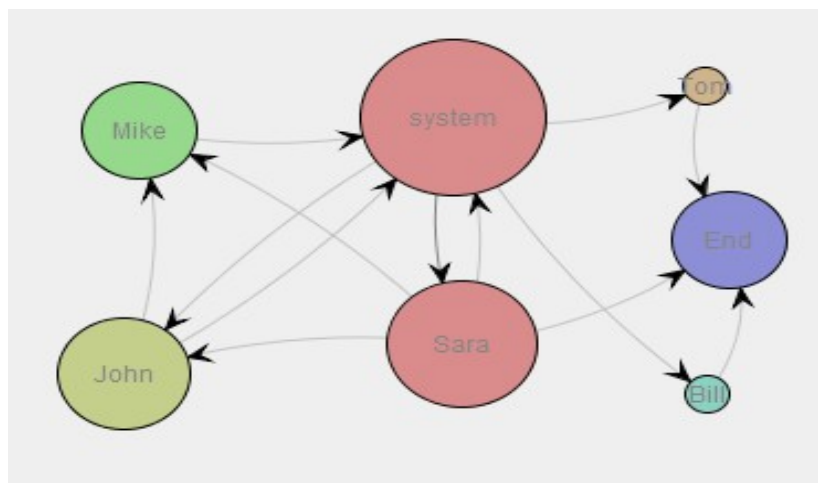
Αρχικά εφαρμόζεται το plug-in για τα Event Log 1 και 2 ξεχωριστά και προκύπτουν τα δύο Sociograms του Σχήματος 6. Εκ πρώτης όψews, είναι εύκολο να διακριθεί ότι τα δύο διαγράμματα έχουν στο κέντρο τους τον κύκλο “system” που αναλαμβάνει τις περισσότερες σε πλήθος διαδικασίες και συνεργάζεται με τους περισσότερους από τους χρήστες κατά την εκτέλεση της διεργασίας (πχ. καταχωρεί την παραγγελία πριν την επεξεργαστούν οι υπάλληλοι, ενημερώνει τους οδηγούς ότι η παραγγελία είναι έτοιμη για αποστολή). Επιπλέον το πρώτο διάγραμμα έχει και ένα δεύτερο κεντρικό κύκλο, τον “End”, που σηματοδοτεί την ολοκλήρωση της διεργασίας είτε επιτυχώς είτε λόγω ακύρωσης της παραγγελίας. Στη συνέχεια μπορεί κανείς να αντιληφθεί ότι οι χρήστες που εκτελούν τις περισσότερες διεργασίες είναι οι “John”, “Mike” και “Sara” αφού είναι υπάλληλοι και διαχειρίζονται το κομμάτι του ελέγχου, τιμολόγησης και συγκέντρωσης της παραγγελίας, ενώ οι οδηγοί “Tom” και “Bill” και η μεταφορική εταιρεία “Courier Co.” εκτελούν τις λιγότερες διεργασίες. Τέλος διαφαίνεται ξεκάθαρα ότι οποιαδήποτε δραστηριότητα εκτελείται από τους οδηγούς προέρχεται ρητά από το “system” αφού η τελευταία δραστηριότητα πριν την αποστολή των προϊόντων είναι η ενημέρωση από το σύστημα ότι τα προϊόντα είναι έτοιμα για αποστολή.

* Τα plug-ins που παρουσιάζονται σε αυτή την ενότητα πρέπει να ληφθούν χειροκίνητα μέσα από τον ProM Package Manager, καθώς δεν εγκαθίστανται με τις default ρυθμίσεις.



Σχήμα 6: Τα Sociogram που προέκυψαν με χρήση του plug-in "Mine for a Handover-of-Work Social Network" για τα Event Log 1 και 2.

Τα παραπάνω Sociograms αποτελούν μια αρκετά ακριβή αναπαράσταση της διεργασίας, δεδομένου ότι κατά μέσο όρο οι διαδικασίες μέσα στην επιχείρηση ισομοιράζονται ανάμεσα στους υπαλλήλους και οι οδηγοί μεταφέρουν τις παραγγελίες εναλλάξ. Παρόλα αυτά, στο Event Log 3 του Παραρτήματος Ι υπάρχουν καταγεγραμμένα επιλεγμένα γεγονότα τα οποία παρεκκλίνουν από την τυπική ροή της διεργασίας όσον αφορά στους χρήστες, παρουσιάζοντας μια εξαίρεση που θα μπορούσε να προκύψει (πχ. λόγω αδειοδότησης κάποιου υπαλλήλου). Το Sociogram που προκύπτει από αυτά παρουσιάζεται στο Σχήμα 7.



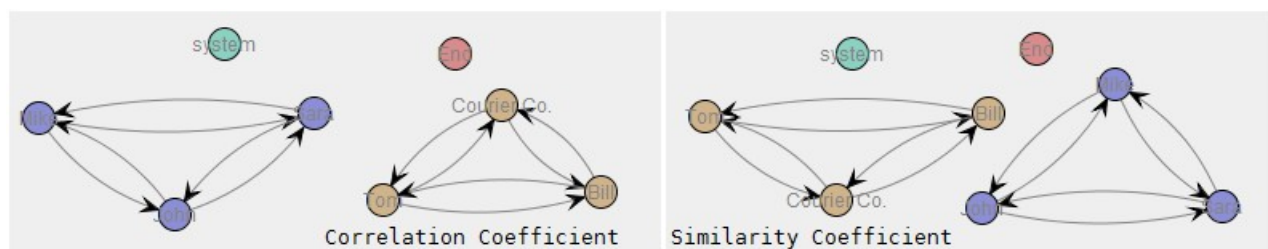
Σχήμα 7: Το Sociogram που προέκυψε με χρήση του plug-in "Mine for a Handover-of-Work Social Network" για το Event Log 3.

Από το Sociogram του Σχήματος 7 μπορεί να προκύψει η παρατήρηση ότι ανάμεσα στους υπαλλήλους η “Sara” ολοκληρώνει τις περισσότερες δραστηριότητες, ενώ δεύτερος έρχεται ο “John” κι έπειτα ο “Mike”, ενώ όσον αφορά στους οδηγούς, ο “Tom” έχει ολοκληρώσει περισσότερες δραστηριότητες από τον “Bill”. Επιπλέον, εύκολα διακρίνεται ότι η “Sara” δίνει εργασία στους “Mike” και “John”, ο “John” μόνο στον “Mike”, ενώ ο τελευταίος δεν λειτουργεί καθόλου ομαδικά, αλλά δρα ατομικά. Τέλος, παρατηρούμε ότι εκτός από τους οδηγούς, μονάχα από τη “Sara” υπάρχει βέλος που οδηγεί στον κύκλο “End”, πράγμα που σημαίνει ότι μόνο παραγγελίες που διαχειρίστηκε η “Sara” δεν ολοκληρώθηκαν με επιτυχία.

Το δεύτερο plug-in σχετικό με Social Network που μελετάται είναι το “Mine for a Similar-Task Social Network”. Το plug-in αυτό επικεντρώνεται στις δραστηριότητες που εκτελούν οι διάφοροι χρήστες, υποθέτοντας ότι άτομα τα οποία εκτελούν παρόμοιες δραστηριότητες έχουν πιο ισχυρές σχέσεις από άτομα που κάνουν εντελώς διαφορετικά πράγματα.

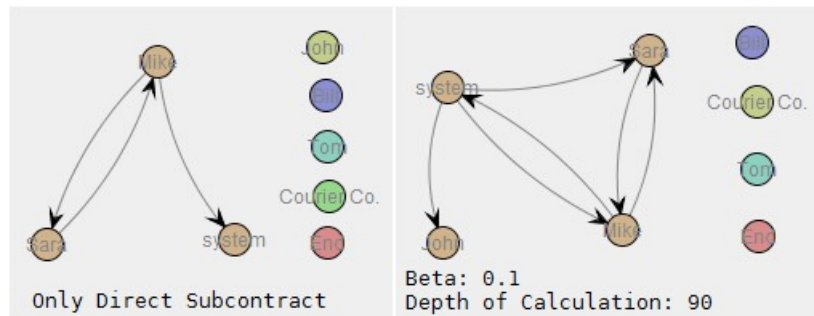
Για την εφαρμογή του συγκεκριμένου plug-in χρησιμοποιήθηκαν τα δεδομένα του Event Log 1, χωρίς κάποιο ιδιαίτερο λόγο πέραν του ότι περιλαμβάνουν περισσότερους χρήστες και μεγαλύτερο εύρος περιπτώσεων. Στην πραγματικότητα όποιο Event Log κι αν χρησιμοποιούταν τα αποτελέσματα δε θα διέφεραν παρά μόνο στον αριθμό των χρηστών (πχ. μπορεί να μην υπήρχε ο κύκλος “End” ή ο κύκλος “Courier Co.”). Σε αυτό το σημείο να σημειωθεί ότι δεν δίνουν κατάλληλο αποτέλεσμα όλες οι μετρικές που μπορούν να χρησιμοποιηθούν από αυτό το plug-in για το τρέχον παράδειγμα, καθώς οι επιλογές “Euclidean distance” και “Humming distance” δεν είναι κατάλληλες για τα δεδομένα που περιέχουν τα Event Log του παραδείγματος.

Τα δύο Sociogram που προκύπτουν από τις μετρικές “Correlation coefficient” και “Similarity coefficient” (Σχήμα 8) δε διαφέρουν σε τίποτα μεταξύ τους, ενώ βοηθούν στην καλύτερη κατανόηση των σχέσεων μεταξύ των χρηστών που συμμετέχουν στη διεργασία, καθώς μέσα από αυτά διαφαίνονται πολύ εύκολα οι ρόλοι των χρηστών και οι ομάδες που σχηματίζονται με βάση αυτούς τους ρόλους. Έτσι διακρίνουμε δύο κύριους ρόλους από τις ομάδες χρηστών που διαφαίνονται, τους “υπαλλήλους” (“Sara”, “John”, “Mike”) και τους “οδηγούς – διανομείς” (“Bill”, “Tom” και “Courier Co.”), με βάση τις δραστηριότητες που αυτοί εκτελούν, ενώ υπάρχουν ακόμα οι “End” και “system” που δεν φαίνεται να έχουν ιδιαίτερα ισχυρές σχέσεις με καμία άλλη ομάδα.



Σχήμα 8: Τα Sociogram που προέκυψαν με χρήση του plug-in “Mine for a Similar-Task Social Network” για το Event Log 1 με τις μετρικές “Correlation coefficient” και “Similarity coefficient”.

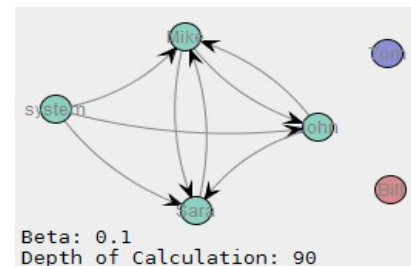
Το τελευταίο plug-in που μελετάται σε αυτή την ενότητα είναι το “Mine for a Subcontracting Social Network”. Το plug-in αυτό μετρά τον αριθμό των φορών που ο χρήστης j εκτέλεσε μια δραστηριότητα ανάμεσα σε δύο δραστηριότητες του χρήστη i . Αυτό μπορεί να δείχνει ότι κάποιο μέρος της δουλειάς δόθηκε από τον i στον j (subcontract).



Σχήμα 9: Τα Sociogram που προέκυψαν με χρήση του plug-in “Mine for a Subcontracting Social Network” για το Event Log 1.

Εφαρμόζοντας το plug-in στα δεδομένα του Event Log 1 με διαφορετικές παραμέτρους λαμβάνουμε τα δύο Sociogram του Σχήματος 9. Οι παράμετροι που χρησιμοποιήθηκαν στο πρώτο είναι να ληφθούν υπόψιν μόνο άμεσες μεταβάσεις, ενώ στο δεύτερο χρησιμοποιούνται οι παράμετροι “Beta” και “Depth of calculation” με τιμές 0.1 και 90, καθώς δίνουν καλύτερα αποτελέσματα για το τρέχον παράδειγμα. Επιπλέον και στις δύο περιπτώσεις λαμβάνονται υπόψιν πολλαπλές μεταβάσεις μέσα σε ένα στιγμιότυπο της διεργασίας. Το πρώτο Sociogram που προκύπτει δεν είναι αρκετά αντιπροσωπευτικό, καθώς περιλαμβάνει μόνο άμεσες μεταβάσεις, ενώ το δεύτερο δίνει μια καλύτερη εικόνα για τη διεργασία, αφού βλέπουμε ότι μεταξύ ενεργειών του “system” παρεμβάλλεται κάποιος από τους τρεις “υπαλλήλους” όπως αυτοί ορίστηκαν παραπάνω και ανάμεσα στις δραστηριότητες ενός από αυτούς μπορεί να παρεμβάλλεται κάποιος άλλος “υπάλληλος”. Τέλος, εύκολα παρατηρείται ότι οι “οδηγοί – διανομείς” και ο κύκλος “End” δεν παρεμβάλλονται στις δραστηριότητες που αναφέρθηκαν παραπάνω, όπως είναι λογικό. Παρόλα αυτά, αν το “system” εκτελούσε μια τελευταία δραστηριότητα μετά την ολοκλήρωση παράδοσης της παραγγελίας ή την ακύρωσή της, τότε το γράφημα θα ήταν πλήρως συνεκτικό, καθώς θα παρεμβάλλονταν πάντοτε και “οδηγοί – διανομείς” στις δραστηριότητες του “system”.

Μια ακόμα πιο αντιπροσωπευτική εικόνα της διεργασίας μας δίνει το Σχήμα 10 που προέκυψε με τις ίδιες τιμές “Beta” και “Depth of calculation” για το Event Log 3, στο οποίο υπάρχουν περισσότερες μεταβάσεις μεταξύ των υπαλλήλων. Το Sociogram αυτό αντικατοπτρίζει πλήρως αυτό του Σχήματος 7, χρησιμοποιώντας μια διαφορετική οπτική για τα δεδομένα, καθώς διαφαίνονται οι ίδιες σχέσεις μεταξύ των χρηστών.

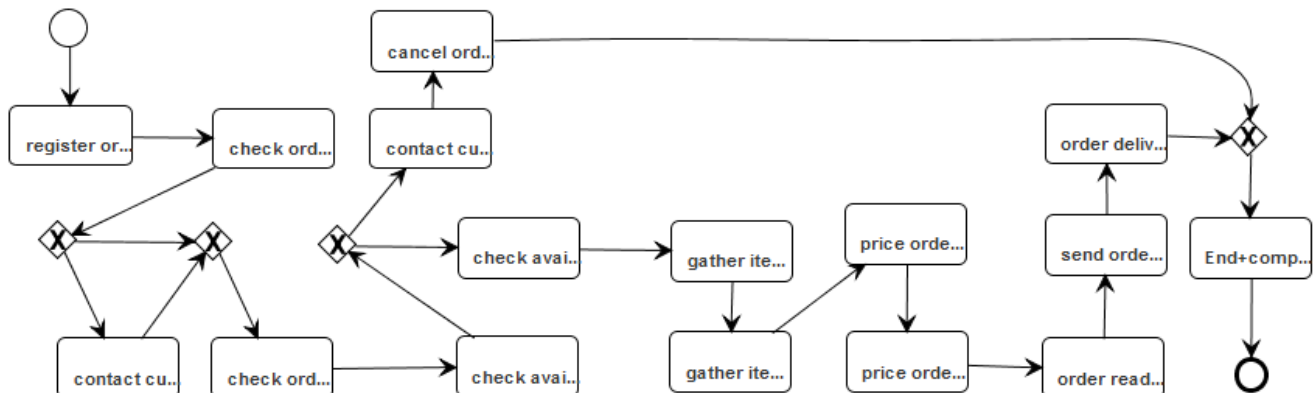


Σχήμα 10: Το Sociogram που προέκυψε με χρήση του plug-in “Mine for a Subcontracting Social Network” για το Event Log 3.

4.2.5. Mine BPMN Using Heuristics Miner of Prom 5.2

Το τελευταίο plug-in Ανάλυσης Διεργασιών που θα παρουσιαστεί σε αυτή την ενότητα αφορά την εξόρυξη ενός διαγράμματος Μοντέλου Επιχειρηματικών Διεργασιών και Συμβολισμού (Business Process Model and Notation, στο εξής BPMN). Ονομάζεται “Mine BPMN Using HeuristicsMiner of Prom5.2”, έχει υλοποιηθεί από την Raffaele Conforti και επιλέχθηκε ενδεικτικά μεταξύ άλλων plug-ins εξόρυξης διαγραμμάτων BPMN λόγω της ευχρηστίας του. Το παραπάνω plug-in δέχεται σαν είσοδο ένα Event Log και παράγει σαν έξοδο ένα διάγραμμα BPMN. Το τελευταίο αποτελεί μια μορφή γραφικής αναπαράστασης μοντέλων επιχειρησιακών διεργασιών που βρίσκει ευρείες εφαρμογές στις σύγχρονες επιχειρήσεις.

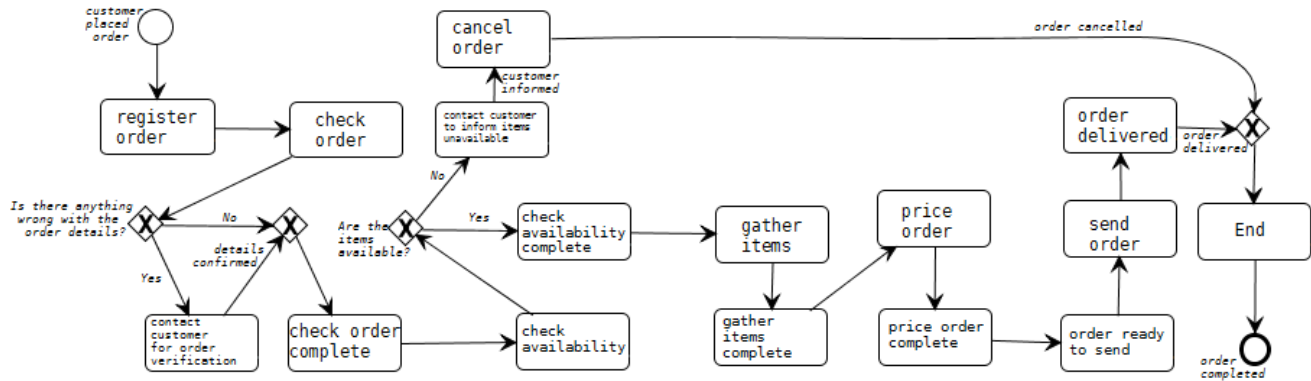
Το διάγραμμα BPMN που προκύπτει από την εφαρμογή του plug-in για το Event Log 3 φαίνεται στο Σχήμα 11. Για να γίνει κατανοητό το διάγραμμα αυτό, θα πρέπει πρώτα να κατανοήσει κανείς τα μέρη που του αποτελούν. Ξεκινώντας από την πάνω αριστερά γωνία, το πρώτο στοιχείο που εμφανίζεται είναι το Γεγονός Εκκίνησης (Start Event), που συμβολίζεται με ένα κύκλο. Για να διατρέξει κανείς το διάγραμμα, αρκεί να ακολουθήσει τα βέλη που υπάρχουν, τα οποία αποτελούν Αντικείμενα Ροής Ακολουθίας (Sequence Flow) και περιγράφουν με ποια σειρά εκτελούνται οι δραστηριότητες μέσα στη διεργασία. Οι δραστηριότητες της διεργασίας που εμφανίζονται στο Event Log του τρέχοντος παραδείγματος, καθώς και η επιπρόσθετη δραστηριότητα “End” περικλείονται σε στρογγυλεμένα τετράγωνα που συμβολίζουν συγκεκριμένες Μονάδες Εργασίας (Tasks). Στο τέλος του διαγράμματος υπάρχει ένας κύκλος με έντονο περίγραμμα που συμβολίζει το Γεγονός Τέλους (End Event). Οι ρόμβοι το διαγράμματος με το σύμβολο X είναι Αποκλειστικές Πύλες (Exclusive Gateway) ή αλλιώς Πύλες XOR (XOR Gateway) και χρησιμοποιούνται για να δημιουργήσουν εναλλασσόμενες ροές στη διεργασία, καθώς για κάθε μία από αυτές μπορεί να ακολουθήσει κανείς μόνον ένα μονοπάτι της.



Σχήμα 11: Το διάγραμμα BPMN που προέκυψε από την εκτέλεση του plug-in “Mine BPMN using HeuristicsMiner of Prom5.2” για το Event Log 3.

Για κάθε μία Πύλη XOR που διατρέχει κανείς πρέπει να βρει το ζεύγος της, το οποίο σηματοδοτεί το τέλος της εναλλασσόμενης ροής. Για παράδειγμα μεταξύ των δραστηριοτήτων “check order” και “check order complete” παρεμβάλλεται μια Πύλη XOR που οδηγεί είτε στη δραστηριότητα “call customer for order verification” και έπειτα σε μία άλλη Πύλη XOR ή κατευθείαν στη δεύτερη. Επομένως η διεργασία μπορεί είτε να περάσει από αυτή τη δραστηριότητα είτε να την παραλείψει.

Από αυτή την παρατήρηση γεννάται το ερώτημα του πώς αποφασίζει κανείς ποιο μονοπάτι θα διατρέξει, το οποίο λύνεται χρησιμοποιώντας Σχόλια (Annotations) που επεξηγούν το διάγραμμα BPMN. Το νέο διάγραμμα με τα Σχόλια που προστέθηκαν χειροκίνητα παρουσιάζεται στο Σχήμα 12.



Σχήμα 12: Το σχολιασμένο διάγραμμα BPMN που προέκυψε από την εκτέλεση του plug-in "Mine BPMN using HeuristicsMiner of Prom5.2" για το Event Log 3.

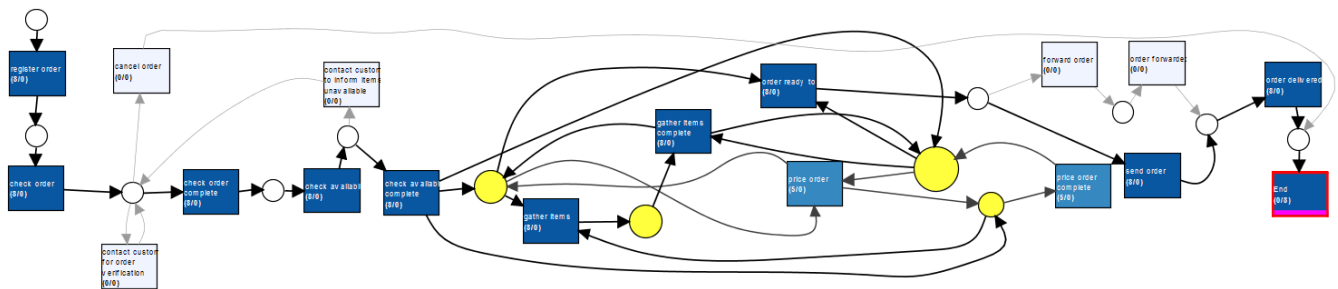
Το παραπάνω διάγραμμα BPMN βοηθά στην πληρέστερη κατανόηση της διεργασίας του τρέχοντος παραδείγματος που μελετάται, καθώς όχι μόνο παρουσιάζει τη ροή της διεργασίας με όλες τις λεπτομέρειες, αλλά παράλληλα παρέχει και πληροφορίες για αυτή που δεν ήταν διαθέσιμες άμεσα από τις αυτόματες τεχνικές PM που αναφέρθηκαν στις προηγούμενες ενότητες (πχ. Petri Net ή Fuzzy Model). Επομένως μπορεί κανείς να κατανοήσει πλέον με σαφήνεια γιατί μπορεί να χρειαστεί για παράδειγμα να επικοινωνήσει η επιχείρηση με τον πελάτη ή πότε πρέπει να ακυρωθεί μια παραγγελία από την επιχείρηση και αν ενημερωθεί ο πελάτης. Επιπλέον, το διάγραμμα BPMN μπορεί μέσα από συγκεκριμένα plug-in να μετατραπεί σε άλλα διαγράμματα. Ιδιαίτερο ενδιαφέρον παρουσιάζει το plug-in "Convert BPMN to Petrinet" της Raffaele Conforti το οποίο με είσοδο το διάγραμμα BPMN του Σχήματος 11 παράγει το ίδιο Petri Net που θα προέκυπτε από τη χρήση του plug-in "Discover using Decomposition" για το ίδιο Event Log, ενώ η αντίστροφη διαδικασία μπορεί να εκτελεστεί με χρήση του plug-in "Convert Petrinet to BPMN" της Raffaele Conforti.

4.3. Έλεγχος Συμμόρφωσης

Έχοντας μελετήσει ενδεικτικά κάποια μοντέλα που προέκυψαν με χρήση τεχνικών Ανεύρεσης Διεργασιών, είναι δυνατόν πλέον να εφαρμοστούν τεχνικές Ελέγχου Συμμόρφωσης σε αυτά. Η ενότητα αυτή περιλαμβάνει δύο τέτοια plug-in, τα οποία χρησιμοποιούν τόσο κάποιο Event Log όσο και δεδομένα που έχουν προκύψει από την Ανεύρεση Διεργασιών για να ελέγξουν αν τα δεδομένα των Event Log εμπίπτουν στο μοντέλο που μελετάται.

4.3.1. Replay using Decomposition

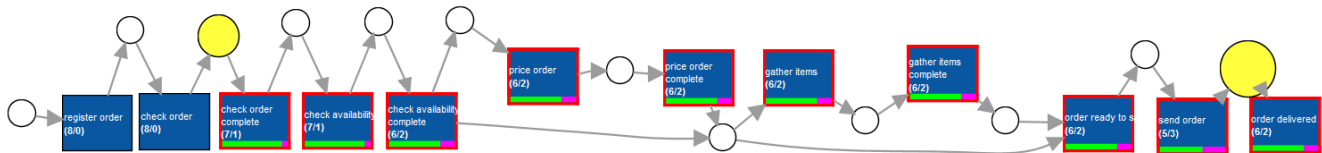
Το πρώτο plug-in Ελέγχου Συμμόρφωσης που θα μελετηθεί είναι το “Replay using Decomposition” σε υλοποίηση H.M.W. Verbeek. Το plug-in αυτό δέχεται σαν είσοδο α) ένα Event Log και β) ένα Petri Net και χρησιμοποιεί τη μέθοδο της Επανάληψης (Replay) για να δει κατά πόσον το δοθέν Event Log αναπαράγεται μέσα στο Petri Net που έχει δοθεί. Η διαδικασία που ακολουθεί το plug-in είναι πρώτα αποδόμηση τόσο του Event Log όσο και του Petri Net σε μικρότερα μέρη, έπειτα αναπαραγωγή κάθε μέρους του Event Log στο αντίστοιχο του Petri Net και τέλος συνένωση των αποτελεσμάτων σε ένα αποτέλεσμα. Για να γίνει αντιληπτό το αποτέλεσμα του plug-in είναι σκόπιμο να παρουσιαστούν διάφορες περιπτώσεις συνδυασμών Event Log και Petri Net έτσι ώστε να υπάρχει μεγάλο εύρος δεδομένων.



Σχήμα 13: Το αποτέλεσμα εκτέλεσης του plug-in “Replay using Decomposition” για το Petri Net του Σχήματος 2 και το Event Log 2.

Αρχικά εφαρμόζουμε το plug-in στο Petri Net που προέκυψε από το Event Log 1 (Σχήμα 2) και το Event Log 2 του τρέχοντος παραδείγματος με τη διαμόρφωση “Not Decomposed” για τον ταξινομητή “Activity”, ενώ οι υπόλοιπες ρυθμίσεις είναι οι προεπιλεγμένες. Τα αποτελέσματα που εξάγονται από τα δεδομένα αυτά (Σχήμα 13) γίνονται εύκολα αντιληπτά αν λάβει κανείς υπόψιν του ότι στο Event Log 1 υπάρχουν μερικές περιπτώσεις εξαιρέσεων, ενώ το Event Log 2 αποτελείται από δεδομένα χωρίς καθόλου εξαιρέσεις. Επομένως είναι λογικό στο Petri Net που προέκυψε από την αναπαραγωγή κάποια μονοπάτια να είναι μαρκαρισμένα με γκρι, καθώς δεν εμφανίζονται στα γεγονότα του Event Log 2 (συγκεκριμένα τα μονοπάτια που αφορούν ακυρώσεις, επικοινωνία με τους πελάτες και αποστολή με εταιρεία Courier). Επίσης ευδιάκριτο είναι και το γεγονός ότι σε τρεις περιπτώσεις του Event Log 2 η δραστηριότητα “price order” εκτελέστηκε παράλληλα με τη δραστηριότητα “gather items”. Τέλος, η δραστηριότητα “End” που έχει προστεθεί στο αρχικό Petri Net του Event Log 1 για να δείξει ότι η διεργασία ολοκληρώθηκε, έχει κόκκινο πλαίσιο καθώς η κίνηση στο μοντέλο του Petri Net έχει συχνότητα 8, ενώ στα γεγονότα του Event Log έχει συχνότητα 0 αφού η κατάσταση “End” δεν υπάρχει.

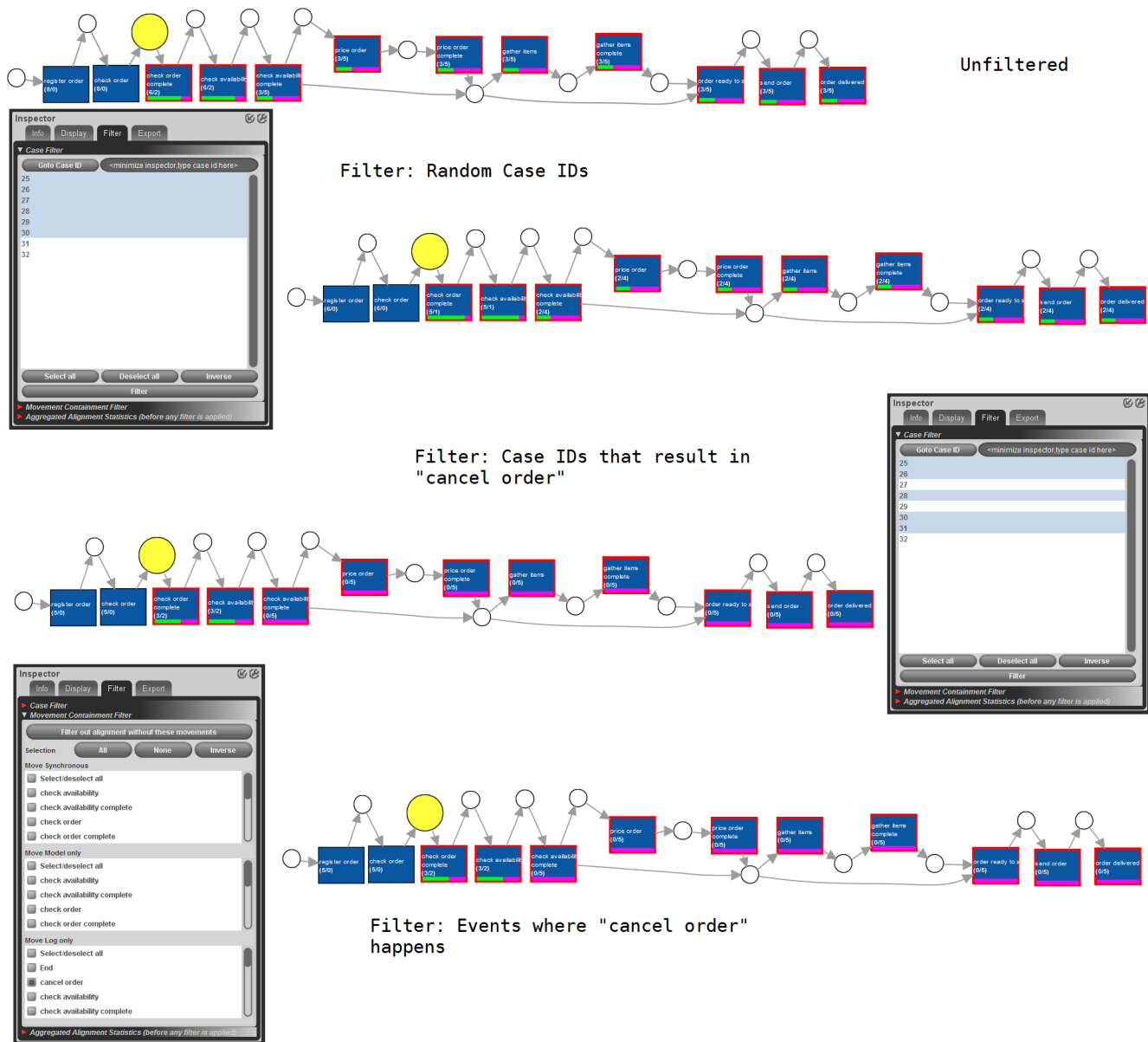
Σε αυτό το σημείο να σημειωθεί ότι τα πλαίσια που έχουν σημειωθεί με κόκκινο περίγραμμα συμβολίζουν δραστηριότητες στις οποίες υπάρχει διαφορά μεταξύ προβλεπόμενης και τελικής συχνότητας, ενώ οι μεταβάσεις που έχουν σημειωθεί με κίτρινο συμβολίζουν σημεία στα οποία στο Event Log υπάρχουν πρόσθετες δραστηριότητες που δεν εμφανίζονται στο μοντέλο του Petri Net. Τέλος, οι μπάρες των πλαισίων δείχνουν το λόγο μεταξύ των δραστηριοτήτων που εμπίπτουν στο μοντέλο του Petri Net και εκείνων που παρεκκλίνουν από αυτό.



Σχήμα 14: Το αποτέλεσμα εκτέλεσης του plug-in "Replay using Decomposition" για το Petri Net του Σχήματος 3 και το Event Log 1.

Ένα ακόμη ενδιαφέρον αποτέλεσμα που βοηθά στην καλύτερη κατανόηση του παραπάνω plug-in, δίνει η εφαρμογή του στο Petri Net που προέκυψε από το Event Log 2 (Σχήμα 3) και το Event Log 1 του τρέχοντος παραδείγματος με τις ίδιες ρυθμίσεις που αναφέρθηκαν παραπάνω (Σχήμα 14). Στο Petri Net που προκύπτει από την αναπαραγωγή γίνεται εύκολα αντιληπτό το γεγονός ότι δύο παραγγελίες του Event Log δεν παραδόθηκαν. Συγκεκριμένα φαίνεται ότι ανάμεσα στις δραστηριότητες "check order" και "check order complete" ένα γεγονός (παραγγελία) παρουσίασε κάποια εξαίρεση, όπως επίσης κι άλλο ένα ανάμεσα στις δραστηριότητες "check availability" και "check availability complete". Από τα έξι εναπομείναντα γεγονότα, γίνεται εύκολα αντιληπτό ότι ένα, δεν πέρασε από τη δραστηριότητα "send order", αλλά έφτασε στην κατάσταση "order delivered", πράγμα που εξηγείται αν λάβει κανείς υπόψιν του ότι σε ένα γεγονός του Event Log 1, η αποστολή γίνεται με εταιρεία Courier επομένως αντί της δραστηριότητας "send order" υπάρχει η "forward order" και η "order forwarded". Τελικά έξι παραγγελίες παραδόθηκαν, ενώ δύο δεν έφτασαν στην τελική δραστηριότητα, πράγμα που δε θα ίσχυε αν υπήρχε και η πρόσθετη κατάσταση "End" στο Petri Net.

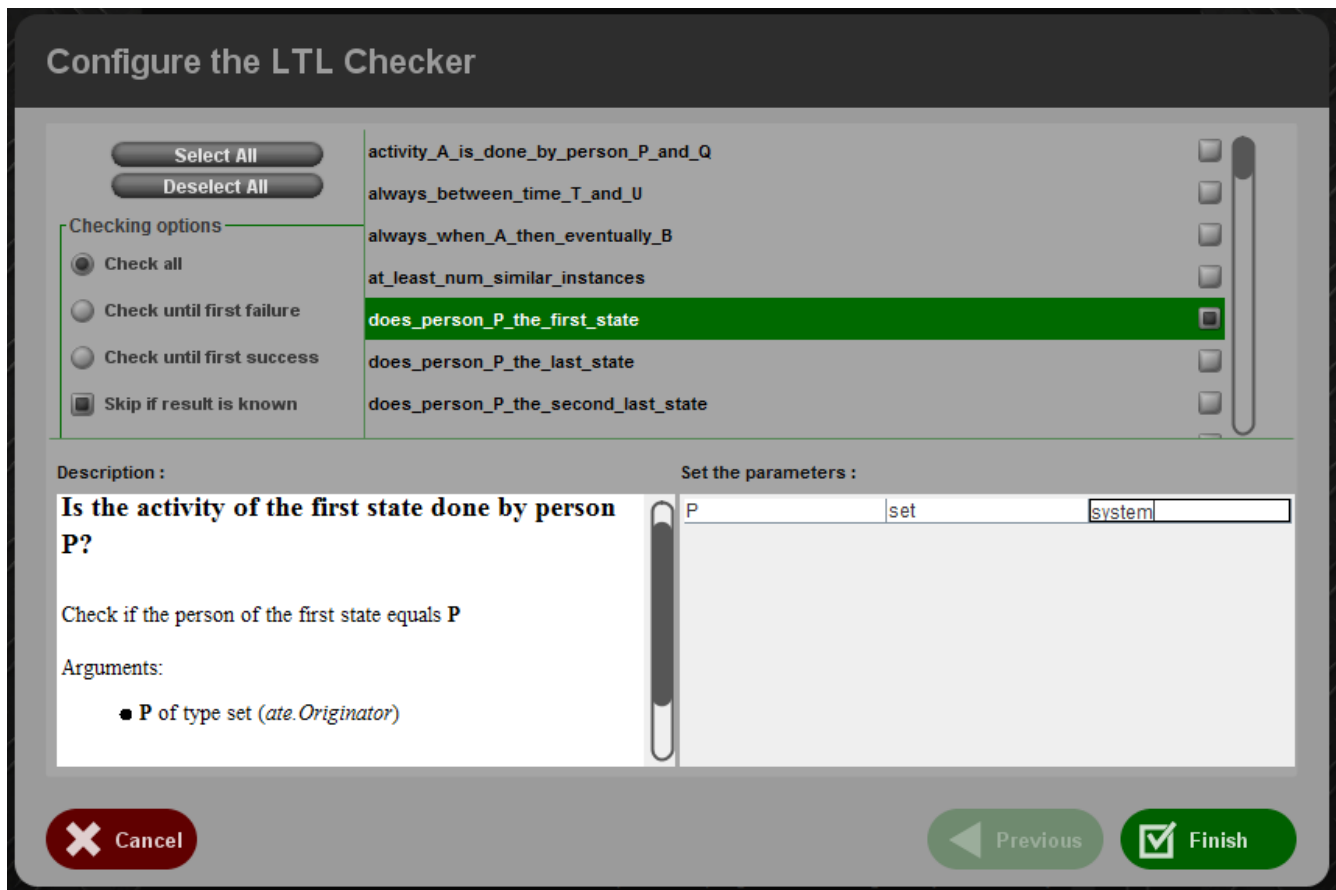
Η τελευταία εφαρμογή του παραπάνω plug-in που θα μελετηθεί παρουσιάζει τις δυνατότητες φιλτραρίσματος (filtering) του συγκεκριμένου plug-in. Τα αποτελέσματα που εξάγονται προέρχονται από την αναπαραγωγή στο Petri Net του Event Log 2 (Σχήμα 3) των δεδομένων του Event Log 4. Το τελευταίο περιέχει αρκετά στιγμιότυπα της διεργασίας τα οποία καταλήγουν σε ακύρωση παραγγελίας. Συγκεκριμένα 5 από αυτά (τα γεγονότα με Case ID 25, 26, 28, 30 και 31) τελειώνουν με τη δραστηριότητα "cancel order", ενώ τα υπόλοιπα 3 τελειώνουν με τη δραστηριότητα "order delivered". Στο Σχήμα 15 παρουσιάζονται τόσο το Petri Net που προκύπτει από την αναπαραγωγή του Event Log στο συγκεκριμένο μοντέλο, όσο και τρία φίλτρα που έχουν εφαρμοστεί σε αυτό και τα αποτελέσματά τους: α) φίλτρο με βάση τυχαία Case ID, β) φίλτρο με βάση μόνο τα Case ID που καταλήγουν στη δραστηριότητα "cancel order" και γ) φίλτρο με βάση τη δραστηριότητα "cancel order". Εύκολα κατανοεί κανείς τη χρησιμότητα της δυνατότητας του φιλτραρίσματος μέσω αυτών των αποτελεσμάτων, καθώς μειώνεται ο "θόρυβος" στα δεδομένα που αναπαράγονται και μένουν μόνο τα δεδομένα που ενδιαφέρουν το χρήστη.



Σχήμα 15: Το αποτέλεσμα εκτέλεσης του plug-in "Replay using Decomposition" για το Petri Net του Σχήματος 3 και το Event Log 4 χωρίς κανένα φίλτρο, φιλτραρισμένο ως προς τυχαία Case ID, φιλτραρισμένο ως προς τα Case ID που καταλήγουν σε "cancel order" και φιλτραρισμένο με βάση τα γεγονότα στα οποία συμβαίνει η δραστηριότητα "cancel order". Δίπλα σε κάθε αποτέλεσμα που έχει εφαρμοστεί φίλτρο, φαίνεται το παράθυρο "Inspector" με τις αντίστοιχες επιλογές.

4.3.2. LTL Checker Default

Ένα ακόμη ενδιαφέρον plug-in Ελέγχου Συμμόρφωσης είναι το “LTL Checker Default” του F.M. Maggi, που δέχεται σαν είσοδο ένα Event Log και κάποιες συνθήκες ελέγχου (checking conditions) μέσα από τη διεπαφή χρήστη και εξάγει αποτελέσματα με χρήση της Γραμμικής Χρονικής Λογικής (Linear Time Logic, στο εξής LTL). Η LTL είναι μια τροπική χρονική λογική με τροπικότητες που αναφέρονται στο χρόνο, επομένως αποτελεί μια εξειδίκευση της Μαθηματικής Λογικής και συγκεκριμένα της Κατηγορηματικής Λογικής. Οι κανόνες των επιμέρους συνθηκών ελέγχου εξηγούνται στην περιγραφή που εμφανίζεται μέσα από τη γραφική διεπαφή χρήστη.*

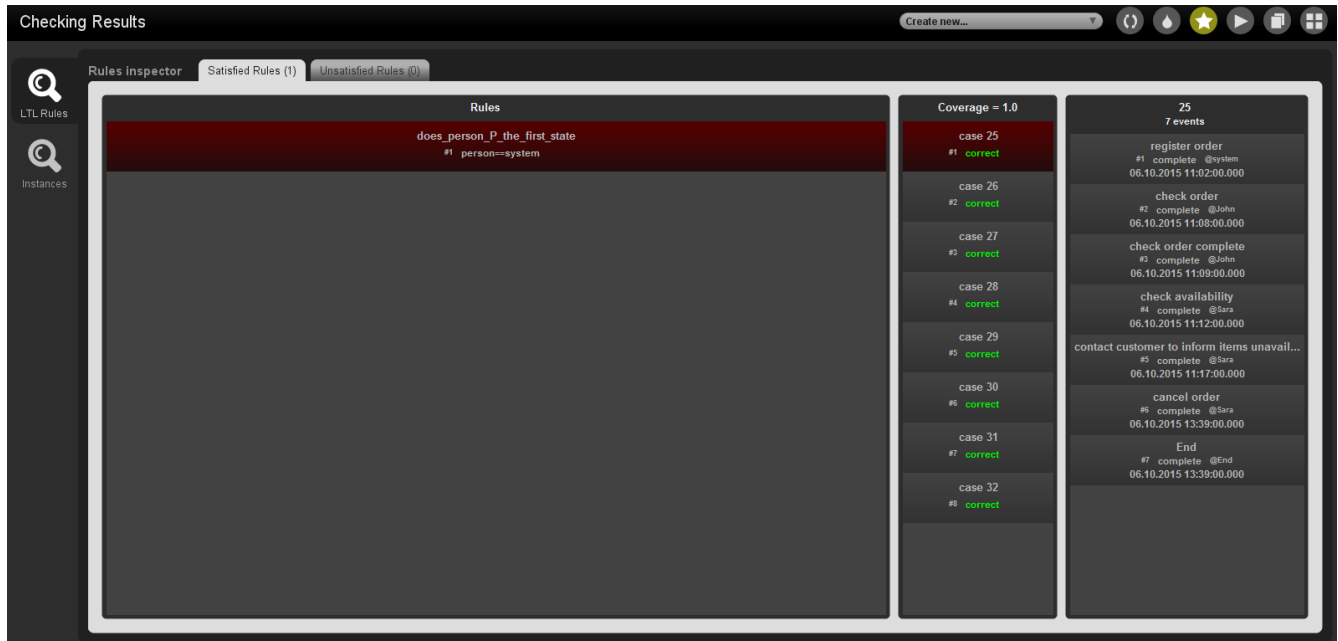


Εικόνα 3: Η διεπαφή χρήστη του plug-in "LTL Checker Default" με επιλεγμένο τον έλεγχο "does_person_P_the_first_state" και ορισμένη την παράμετρο P στην τιμή "system".

Ένα πρώτο απλό παράδειγμα που βοηθά στην κατανόηση του πώς λειτουργεί το συγκεκριμένο plug-in είναι η δοκιμή του ελέγχου "does_person_P_the_first_state" σε ένα οποιοδήποτε από τα Event Log του τρέχοντος παραδείγματος (στη συγκεκριμένη εφαρμογή χρησιμοποιείται το Event Log 4). Ο έλεγχος αυτός έχει μία παράμετρο P η οποία αναφέρεται σε ένα πρόσωπο, δηλαδή μία τιμή της

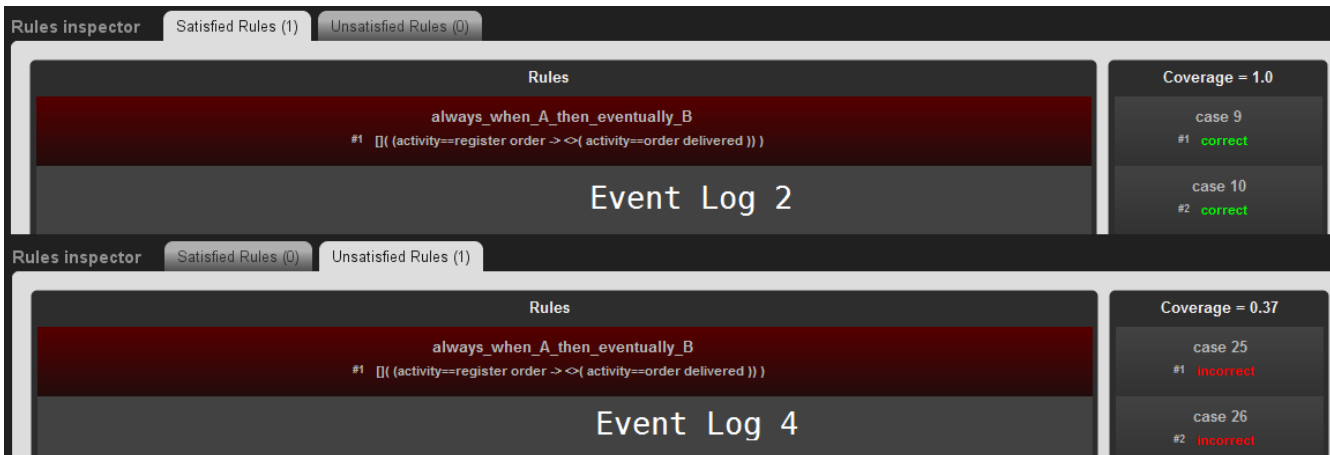
* Το plug-in που παρουσιάζεται σε αυτή την ενότητα πρέπει να ληφθεί χειροκίνητα μέσα από τον ProM Package Manager, καθώς δεν εγκαθίσταται με τις default ρυθμίσεις.

στήλης Resource στο τρέχον παράδειγμα. Στη διεπαφή που εμφανίζεται με την εκκίνηση του plug-in (Εικόνα 3) για τα δεδομένα του Event Log, επιλέγεται ο συγκεκριμένος έλεγχος και στην περιγραφή εμφανίζονται κάποιες πληροφορίες που εξηγούν τη λογική πίσω από τον κανόνα, καθώς και τα ορίσματα. Στο δίπλα πλαίσιο μπορεί κανείς να θέσει τις παραμέτρους του ελέγχου. Για το συγκεκριμένο έλεγχο, θα γίνει δοκιμή με την τιμή $P = system$. Το αποτέλεσμα, όπως είναι αναμενόμενο, είναι ότι ο κανόνας ικανοποιείται σε όλες τις περιπτώσεις του Event Log (Εικόνα 4).



Εικόνα 4: Το αποτέλεσμα εκτέλεσης του plug-in "LTL Checker Default" για τον έλεγχο "does_person_P_the_first_state" με παράμετρο $P = system$.

Ένας ακόμη απλός κανόνας LTL που μπορεί να εφαρμοστεί στα δεδομένα του τρέχοντος παραδείγματος με χρήση αυτού του plug-in είναι αυτός που προκύπτει από τη συνθήκη "always_when_A_then_eventually_B" με μεταβλητές δραστηριοτήτων τις A και B με τιμές $A = register\ order$ και $B = order\ delivered$. Ο κανόνας LTL ουσιαστικά ελέγχει αν για κάθε στιγμιότυπο της διεργασίας στο οποίο συμβαίνει η δραστηριότητα A συμβαίνει και η δραστηριότητα B , κάποια στιγμή αργότερα από την A . Η συνθήκη αυτή εφαρμόζεται στα Event Log 2 και 4 και προκύπτουν τα αποτελέσματα της Εικόνας 5, στα οποία παρατηρεί κανείς ότι για το Event Log 2 ο κανόνας ικανοποιείται, ενώ για το Event Log 4 δεν ικανοποιείται. Επιπλέον, στη δεξιά στήλη όπου εμφανίζονται τα διάφορα στιγμιότυπα της διεργασίας μαζί με την κατάστασή τους (correct ή incorrect ανάλογα με το αν ικανοποιούν τη συνθήκη ελέγχου), αναγράφεται στην κορυφή και η Κάλυψη (Coverage), δηλαδή το ποσοστό των στιγμιότυπων στα οποία ο κανόνας έδωσε θετικό αποτέλεσμα. Η Κάλυψη βοηθά στη μέτρηση της Καταλληλότητας (Fitness), δηλαδή κατά πόσο το μοντέλο επιτρέπει τη συμπεριφορά που διερευνά ο κανόνας. Για το Event Log 2 η Κάλυψη είναι ίση με 1.0, όπως είναι λογικό αφού η συνθήκη ελέγχου ικανοποιείται σε όλα τα γεγονότα του Event Log, ενώ στο Event Log 4 είναι 0.37, αριθμός αρκετά αναμενόμενος δεδομένου ότι στο συγκεκριμένο Event Log περισσότερα από τα μισά γεγονότα καταλήγουν σε ακύρωση της παραγγελίας.



Εικόνα 5: Τα αποτελέσματα εκτέλεσης του plug-in "LTL Checker Default" για τη συνθήκη "always_when_A_then_eventually_B" με παραμέτρους A = register order και B = order delivered για τα Event Log 2 και 4.

Πέραν του ελέγχου μίας συνθήκης μεμονωμένα, το συγκεκριμένο plug-in έχει επιπλέον τη δυνατότητα να αξιολογήσει περισσότερες συνθήκες ταυτόχρονα. Για παράδειγμα ενδιαφέρον παρουσιάζει η περίπτωση της αλυσίδας δραστηριοτήτων "check order – contact customer for order verification – check order complete" στο τρέχον παράδειγμα, μέσα από την οποία μπορεί κανείς να εξάγει έμμεσα πληροφορίες για την ακύρωση παραγγελιών σε αυτό το στάδιο εξέλιξης της παραγγελίας. Επομένως θα εφαρμοστούν δύο συνθήκες ελέγχου, τα δεδομένα των οποίων

eventually_activity_A_next_B	
Μεταβλητή	Τιμή
A	check order
B	contact customer for order verification
eventually_activity_A_next_B_next_C	
Μεταβλητή	Τιμή
A	check order
B	contact customer for order verification
C	check order complete

Πίνακας 3: Οι συνθήκες ελέγχου που εφαρμόστηκαν στο Event Log 4 στην Εικόνα 6, μαζί με τις παραμέτρους τους και τις τιμές που αυτές λαμβάνουν.

φαίνονται στον Πίνακα 3, τα αποτελέσματα των οποίων φαίνονται στην Εικόνα 6. Η πρώτη συνθήκη μελετά αν σε κάθε στιγμιότυπο της διεργασίας εμφανίζεται κάποια στιγμή η δραστηριότητα "check order" και αμέσως μετά η δραστηριότητα "contact customer for order verification". Η δεύτερη συνθήκη ελέγχου μελετά την ίδια ακριβώς σχέση με τον επιπρόσθετο περιορισμό ότι έπειτα από τη δραστηριότητα "contact customer for order verification" θα ακολουθεί αμέσως και η δραστηριότητα "check order complete".

Φυσικά και οι δύο συνθήκες δεν ικανοποιούνται, γνωρίζοντας τα δεδομένα του συγκεκριμένου Event Log. Επιπλέον είναι προφανές ότι η πρώτη συνθήκη θα έχει μεγαλύτερη ή ίση Κάλυψη με τη δεύτερη, ενώ η Κάλυψη της δεύτερης θα είναι το πολύ ίση με την Κάλυψη της συνθήκης της προηγούμενης εφαρμογής του plug-in (Εικόνα 5). Τα αποτελέσματα της Εικόνας 6 επαληθεύουν αυτές τις παρατηρήσεις, καθώς η Κάλυψη είναι 0.62 και 0.37 αντίστοιχα για τις δύο συνθήκες ελέγχου, ενώ αυτή της δεύτερης είναι ακριβώς ίση με αυτή της εφαρμογής που περιγράφηκε παραπάνω.

Rules inspector		Satisfied Rules (0)	Unsatisfied Rules (2)
Rules		Coverage = 0.37	
eventually_activity_A_next_B		case 25	#1 incorrect
#1 <{ (activity==check order ^ _O(activity==contact customer for order verification)) }		case 26	#2 incorrect
eventually_activity_A_next_B_next_C			
#2 <{ (activity==check order ^ _O(activity==contact customer for order verification ^ _O(activity==check order complete))) }			
Rules		Coverage = 0.62	
eventually_activity_A_next_B		case 25	#1 incorrect
#1 <{ (activity==check order ^ _O(activity==contact customer for order verification)) }		case 26	#2 correct
eventually_activity_A_next_B_next_C			
#2 <{ (activity==check order ^ _O(activity==contact customer for order verification ^ _O(activity==check order complete))) }			

Εικόνα 6: Τα αποτελέσματα εκτέλεσης του plug-in "LTL Checker Default" για τα δεδομένα του Πίνακα 3 με το Event Log 4. Η συνθήκη της οποίας η Κάλυψη αναγράφεται στα δεξιά, σημειώνεται με κόκκινο σε κάθε περίπτωση.

Στο τελευταίο παράδειγμα εφαρμογής του συγκεκριμένου plug-in θα μελετηθούν έξι συνθήκες ελέγχου που αφορούν περισσότερο την Κοινωνική Πλευρά της διεργασίας. Οι συνθήκες ελέγχου, καθώς και οι παράμετροι και οι τιμές που αυτές λαμβάνουν φαίνονται στον Πίνακα 4, ενώ το Event Log στο οποίο εφαρμόζονται είναι το Event Log 3 του τρέχοντος παραδείγματος, που χρησιμοποιήθηκε και παραπάνω για να μελετηθεί η Κοινωνική Πλευρά της διεργασίας (Σχήμα 7). Οι συνθήκες ελέγχου που εφαρμόζονται έχουν αριθμηθεί και στον Πίνακα 4 ούτως ώστε να είναι ευκολότερη η αναφορά και επεξήγηση τους παρακάτω, ενώ τα αποτελέσματά τους φαίνονται στην Εικόνα 7.

1. does_person_P_the_second_last_state		2. does_person_P_the_second_state	
Μεταβλητή P	Τιμή Bill	Μεταβλητή P	Τιμή Sara
3. eventually_person_P_or_eventually_person_Q		4. eventually_person_P_then_Q	
Μεταβλητή P Q	Τιμή John Mike	Μεταβλητή P Q	Τιμή Mike Sara
5. exists_person_doing_task_A_and_B		6. person_P_does_activity_A_and_then_eventually_B	
Μεταβλητή A B	Τιμή gather items price order	Μεταβλητή P A B	Τιμή John check order check availability

Πίνακας 4: Οι συνθήκες ελέγχου που εφαρμόστηκαν στο Event Log 3 στην Εικόνα 7, μαζί με τις παραμέτρους τους και τις τιμές που αυτές λαμβάνουν.

Οι πρώτες δύο συνθήκες είναι αρκετά απλές και εξετάζουν αν ένα συγκεκριμένο άτομο εκτελεί μια δραστηριότητα. Συγκεκριμένα η πρώτη μελετά αν η προτελευταία δραστηριότητα ολοκληρώνεται από τον Bill, επομένως ελέγχει ποιες από τις παραγγελίες παραδόθηκαν από τον Bill, δεδομένου ότι από τη μοντελοποίηση του Event Log που μελετάται η τελευταία δραστηριότητα είναι πάντοτε η "End" που δεν εκτελείται από κάποιο πρόσωπο. Η δεύτερη συνθήκη αντίστοιχα ελέγχει σε ποιες

περιπτώσεις η δεύτερη δραστηριότητα του στιγμιότυπου της διεργασίας εκτελείται από τη Sara, δεδομένου και πάλι ότι η πρώτη εκτελείται πάντοτε από το “system”. Με αυτό τον τρόπο μπορεί κανείς να ελέγξει ποια άτομα συμμετείχαν σε συγκεκριμένα μέρη της διεργασίας και σε τι ποσοστό χάρη στο μέγεθος της Κάλυψης που εμφανίζεται στη δεξιά στήλη. Παρατηρείται εύκολα ότι το 25% των παραγγελιών παραδόθηκαν από τον Bill, ενώ η Sara εκτέλεσε την πρώτη δραστηριότητα στη διαδικασία διεκπεραίωσης της παραγγελίας το 87% των περιπτώσεων. Η τρίτη συνθήκη αφορά τη συμμετοχή των διαφόρων προσώπων στη διεργασία και συγκεκριμένα μελετά αν, σε κάθε στιγμιότυπο της διεργασίας που περιέχεται στο Event Log, υπάρχει τουλάχιστον μία δραστηριότητα που να εκτελείται από τον John ή τον Mike.

Rules inspector		Satisfied Rules (0)	Unsatisfied Rules (6)
Rules			
does_person_P_the_second_last_state	#1 $\langle \rangle (\text{person} == \text{Bill} \wedge _O (! (_O (\text{activity} == \text{activity})))))$		Coverage = 0.25
does_person_P_the_second_state	#2 $_O (\text{person} == \text{Sara})$		Coverage = 0.87
eventually_person_P_or_eventually_person_Q	#3 $\langle \rangle (\text{person} == \text{John}) \vee \langle \rangle (\text{person} == \text{Mike}))$		Coverage = 0.62
eventually_person_P_then_Q	#4 $\langle \rangle ((\text{person} == \text{Mike} \wedge \langle \rangle (\text{person} == \text{Sara})))$		Coverage = 0.0
exists_person_doing_task_A_and_B	#5 $\text{exists} (\langle \rangle ((\text{person} == \text{End} \wedge \text{activity} == \text{gather items})) \wedge \langle \rangle ((\text{person} == \text{End} \wedge \text{activity} == \text{price order}))), \langle \rangle ((\text{person} == \text{Sara} \wedge \text{activity} == \text{gather items})) \wedge \langle \rangle ((\text{person} == \text{Sara} \wedge \text{activity} == \text{price order})))$		Coverage = 0.62
person_P_does_activity_A_and_then_eventually_B	#6 $\langle \rangle (((\text{person} == \text{John} \wedge \text{activity} == \text{check order}) \wedge \langle \rangle (\text{person} == \text{John} \wedge \text{activity} == \text{check availability}))))$		Coverage = 0.12

Εικόνα 7: Τα αποτελέσματα εκτέλεσης του plug-in “LTL Checker Default” για τα δεδομένα του Πίνακα 4 με το Event Log 3.

Η τέταρτη συνθήκη εξετάζει τη σχέση παράδοσης εργασίας από το Mike στη Sara. Ο έλεγχος που εκτελείται σε αυτή τη συνθήκη είναι συγγενής του plug-in “Mine for a Handover-of-Work Social Network” και ουσιαστικά μελετά το ίδιο μέγεθος, απλά σε μια μεμονωμένη περίπτωση. Όπως είναι εμφανές και από τα αποτελέσματα του αντίστοιχου plug-in, εύκολα κατανοεί κανείς γιατί η Κάλυψη της συνθήκης αυτής είναι μηδενική. Η πέμπτη συνθήκη ελέγχει αν σε ένα στιγμιότυπο υπάρχει ένα οποιοδήποτε πρόσωπο που εκτελεί τη δραστηριότητα “gather items” και τη δραστηριότητα “price order”. Κάτι τέτοιο εμπίπτει και πάλι στις σχέσεις παράδοσης εργασίας, αφού μπορεί κανείς εύκολα να καταλάβει ότι, αν η συγκέντρωση προϊόντων και η τιμολόγηση της παραγγελίας έχουν ολοκληρωθεί από το ίδιο πρόσωπο, δεν υπήρξε παράδοση εργασίας σε κάποιο άλλο άτομο για αυτή την αλυσίδα δραστηριοτήτων. Αντίστοιχα η τελευταία συνθήκη ελέγχει όχι μόνο την ύπαρξη ενός ατόμου που εκτελεί και τις δύο δραστηριότητες “check order” και “check availability” μέσα σε ένα στιγμιότυπο, αλλά συγκεκριμένα αν αυτό το άτομο ήταν ο John. Η συνθήκη επαληθεύεται μόνο σε μία περίπτωση και έχει Κάλυψη 0.12, όπως φαίνεται και από την Εικόνα 7.

4.4. Βελτίωση Μοντέλου

Το τελευταίο βήμα στην PM είναι η εφαρμογή τεχνικών βελτίωσης και επέκτασης των μοντέλων που έχουν ληφθεί και ελεγχθεί από τις τεχνικές που περιγράφηκαν παραπάνω. Σε αυτή την ενότητα θα μας απασχολήσουν κάποια plug-ins που στοχεύουν στη βελτίωση των μοντέλων του τρέχοντος παραδείγματος που μελετήθηκαν παραπάνω. Τα plug-ins αυτά στοχεύουν είτε στην απλοποίηση των μοντέλων αυτών μέσα από τεχνικές Αφαίρεσης (Abstraction) είτε στην προσθήκη πληροφορίας που λείπει από τα Event Log ή επεξεργασία της υπάρχουσας με τεχνικές Επέκτασης (Extension).

4.4.1. Enhance Log: Add Mapping of Activity Names

Το πρώτο plug-in που θα μελετηθεί σε αυτή την ενότητα είναι το “Enhance Log: Add Mapping of Activity Names” του E.G.L. de Murillas, αφορά την προσθήκη πληροφορίας σε ένα Event Log και συγκεκριμένα την προσθήκη ενός επιπλέον ταξινομητή για τις δραστηριότητες. Το συγκεκριμένο plug-in θα εφαρμοστεί αρχικά στα δεδομένα του Event Log 2 με τις τιμές που φαίνονται στην Εικόνα 8 για τις διάφορες δραστηριότητες που υπάρχουν μέσα στη διεργασία.

Activity Mapping wizard

Select transformation parameters

Choose a classifier

Classifiers

- Activity
- Resource

Choose a name for the new classifier: Activity (Simplified)

Assign the new activity names

Original Activity Name	New Activity Name
check availability	check availability
check availability complete	check availability
check order	check order
check order complete	check order
gather items	gather items
gather items complete	gather items
order delivered	order delivered
order ready to send	send order
price order	price order
price order complete	price order
register order	register order
send order	send order

NOTE: Empty cells will keep the original activity name

Cancel Previous Finish

Εικόνα 8: Οι ρυθμίσεις του plug-in "Enhance Log: Add Mapping of Activity Names" για το Event Log 2.

Οι αλλαγές τιμών που υπάρχουν στο νέο ταξινομητή μειώνουν τις διάφορες δραστηριότητες που μπορεί κανείς να διακρίνει μέσα στη διεργασία και θα αποδειχθούν αρκετά χρήσιμες για την επίδειξη του plug-in που παρουσιάζεται στην ενότητα 4.4.2. Ένα ενδεικτικό αποτέλεσμα της εφαρμογής του plug-in παρουσιάζεται στον Πίνακα 5 για της δραστηριότητες του Event με Case ID = 11. Στη στήλη Activity υπάρχουν οι αρχικές τιμές του Event Log, ενώ στη στήλη Activity (Simplified) φαίνονται οι τιμές που έχουν δημιουργηθεί με βάση τις μετατροπές που ορίσαμε.

Case ID	Resource	Timestamp	Activity	Activity (Simplified)
11	system	30/09/15 12:40	register order	register order
11	John	30/09/15 12:41	check order	check order
11	John	30/09/15 12:41	check order complete	check order
11	Mike	30/09/15 12:42	check availability	check availability
11	Mike	30/09/15 12:42	check availability complete	check availability
11	Mike	30/09/15 12:42	price order	price order
11	Mike	30/09/15 12:44	price order complete	price order
11	John	30/09/15 12:47	gather items	gather items
11	John	30/09/15 19:22	gather items complete	gather items
11	system	30/09/15 19:22	order ready to send	send order
11	Bill	01/10/15 08:04	send order	send order
11	Bill	01/10/15 08:49	order delivered	order delivered

Πίνακας 5: Ένα ενδεικτικό αποτέλεσμα εφαρμογής του plug-in "Enhance Log: Add Mapping of Activity Names" με τις ρυθμίσεις της Εικόνας 8 για το Event Log 2.

Παρότι το συγκεκριμένο plug-in αφορά τις δραστηριότητες που εμφανίζονται μέσα στη διεργασία, το ProM δεν περιορίζει σαφώς τη στήλη που θα χρησιμοποιηθεί για τη μετατροπή τιμών στο νέο ταξινομητή, επομένως είναι δυνατόν να εφαρμοστεί το ίδιο plug-in για την προσθήκη ενός ταξινομητή που αφορά στα πρόσωπα της διεργασίας. Είναι δυνατόν, για παράδειγμα, με βάση τους ρόλους που προέκυψαν μελετώντας την Κοινωνική Πλευρά της διεργασίας του τρέχοντος παραδείγματος, να δημιουργηθεί ένας ταξινομητής που αφορά στη στήλη Resources. Η Εικόνα 9 παρουσιάζει ενδεικτικά πώς θα μπορούσε να οριστεί η μετατροπή για ένα τέτοιο ταξινομητή χρησιμοποιώντας τα δεδομένα του Event Log 1 και τους ρόλους που προέκυψαν στην ενότητα 4.2.4.

Classifiers

Activity
Resource
Activity

Choose a name for the new classifier: Resource Type

Assign the new activity names

Original Activity Name	New Activity Name
Bill	Delivery Employee
Courier Co.	Courier Co.
End	System
John	Warehouse Employee
Mike	Warehouse Employee
Sara	Warehouse Employee
Tom	Delivery Employee
system	System

Εικόνα 9: Ενδεικτικές ρυθμίσεις του plug-in "Enhance Log: Add Mapping of Activity Names" για το Event Log 1 με βάση τη στήλη Resource.

4.4.2. Enhance Log: Merge Subsequent Events (AAABB -> AB)

Ένα ενδιαφέρον plug-in που μπορεί να εφαρμοστεί σε συνεργασία με το παραπάνω για απλοποίηση του μοντέλου της διεργασίας του τρέχοντος παραδείγματος είναι το “Enhance Log: Merge Subsequent Events (AAABB -> AB)” του F. Mannhardt. Το συγκεκριμένο plug-in συνενώνει τις συνεχόμενες εμφανίσεις μίας δραστηριότητας έτσι ώστε αν για τις δραστηριότητες A, B, C & D εμφανίζεται η ακολουθία AAABBCAADDDD να μετατραπεί σε ABCAD. Για την εφαρμογή του plug-in θα χρησιμοποιηθούν τα δεδομένα του τροποποιημένου Event Log που προέκυψε από την εφαρμογή του plug-in της ενότητα 4.4.1 με τις ρυθμίσεις της Εικόνας 8. Σαν ταξινομητής στις ρυθμίσεις επιλέχθηκε η στήλη Activity (Simplified) του Event Log με όλες τις τιμές της να λαμβάνονται υπόψη στη διαδικασία συνένωσης, ενώ στο τελικό στάδιο ρύθμισης του plug-in σαν τύπος συνένωσης επιλέχθηκε η συνένωση χωρίς επιπλέον επεξεργασία (Merge without further processing).

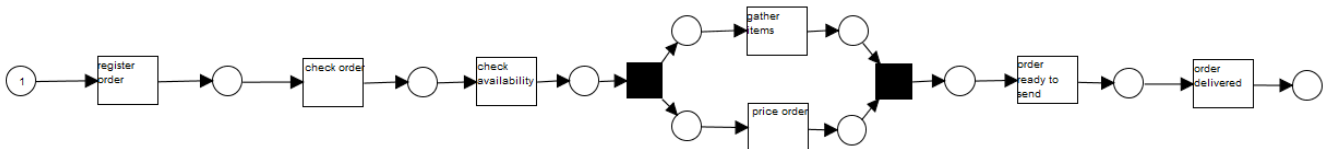
Case ID	Activity	Resource	Timestamp	Activity (Simplified)
11	register order	system	30/09/15 12:40	register order
11	check order	John	30/09/15 12:41	check order
11	check availability	Mike	30/09/15 12:42	check availability
11	price order	Mike	30/09/15 12:42	price order
11	gather items	John	30/09/15 12:47	gather items
11	order ready to send	system	30/09/15 19:22	send order
11	order delivered	Bill	01/10/15 08:49	order delivered

Πίνακας 6: Ένα ενδεικτικό αποτέλεσμα εφαρμογής του plug-in "Enhance Log: Merge Subsequent Events (AAABB -> AB)" μετά την εφαρμογή του plug-in "Enhance Log: Add Mapping of Activity Names" με τις ρυθμίσεις της Εικόνας 8 για το Event Log 2.

Ένα ενδεικτικό αποτέλεσμα της εφαρμογής του plug-in παρουσιάζεται στον Πίνακα 6 για της δραστηριότητες του Event με Case ID = 11. Είναι προφανές ότι μετά την εφαρμογή του plug-in στα δεδομένα, έχει προκύψει μια πιο απλή μοντελοποίηση της διεργασίας, καθώς οι δραστηριότητες της διεργασίας έχουν μειωθεί σχεδόν στο μισό (7 events στο τελικό Event Log έναντι 12 events στο αρχικό). Μπορεί μάλιστα να δει κανείς κατά πόσο έχει μειωθεί ο αριθμός των εμφανιζόμενων γεγονότων, κοιτάζοντας την καρτέλα Summary της καρτέλας View για το αρχικό Event Log (96 events) και το τελικό (60 events). Η ελάττωση αυτή του αριθμού των γεγονότων, μέσα από τις δύο τεχνικές Αφαίρεσης που εφαρμόστηκαν, είναι της τάξης του 37.5%, πράγμα που σημαίνει ότι σε ένα Event Log με χιλιάδες γεγονότα θα μπορούσε να μειώσει σημαντικά το πλήθος τους και να συμβάλλει στη μοντελοποίηση ενός πιο απλού και κατανοητού μοντέλου. Η νέα αυτή μοντελοποίηση δεν αλλοιώνει σε καμία περίπτωση την περιγραφή της διεργασίας, αντιθέτως μάλιστα αποβάλλει πληροφορίες που δεν ήταν αναγκαίες, καθώς όλες οι δραστηριότητες που είχαν στον τίτλο τους τη λέξη “complete” ήταν απλώς μέλος μια προηγούμενης δραστηριότητας που είχε εκκινηθεί.

Ενδιαφέρον σε αυτό το σημείο παρουσιάζει η μελέτη του Petri Net που μπορεί να εξάγει κανείς από τα δεδομένα του τελευταίου Event Log που προέκυψε μετά τη συνένωση. Για να δει κανείς το συγκεκριμένο Petri Net, αρκεί να χρησιμοποιήσει την επιλογή “Inductive visual Miner” από το drop-down μενού “Select viewer” στο πάνω δεξιά μέρος της καρτέλας View, ενώ ελέγχει τα δεδομένα του τελευταίου Event Log. Το νέο αυτό Petri Net που εξάγεται από τα τροποποιημένα δεδομένα (Σχήμα 16) δε διαφέρει σε τίποτα σχεδόν από το αρχικό μοντέλο, πέραν της μείωσης του αριθμού

καταστάσεων που εμφανίζονται σε αυτό. Το νέο αυτό μοντέλο όχι μόνο περιγράφει με σαφήνεια και επάρκεια τη διεργασία, αλλά παρέχει μια απλούστερη εικόνα της διεργασίας, όπου περισσότερες από μια δραστηριότητες συναφείς μεταξύ τους έχουν οργανωθεί σε μία μεγαλύτερη δραστηριότητα, ούτως ώστε να δημιουργήσουν ένα πιο αφηρημένο μοντέλο της διεργασίας, που μπορεί να μελετηθεί με μεγαλύτερη ευκολία.



Σχήμα 16: Το Petri Net που προκύπτει σαν απεικόνιση των αποτελεσμάτων των plug-in "Enhance Log: Merge Subsequent Events (AAABB -> AB)" και "Enhance Log: Add Mapping of Activity Names".

Πίνακας Ορολογίας

Ξενόγλωσσος Όρος	Ελληνικός Όρος
Business Intelligence	Επιχειρηματική Νοημοσύνη
Business Process Intelligence	Νοημοσύνη Επιχειρησιακών Διεργασιών
Business Process Management System	Σύστημα Διαχείρισης Επιχειρησιακών Διεργασιών
Business Process Management	Διαχείριση Επιχειρησιακών Διεργασιών
Event Log	Αρχείο Καταγραφής Συμβάντων
Bottleneck	Σημείο Συμφόρησης
Exception	Εξαίρεση
Process Mining	Εξόρυξη Διεργασιών
Process Discovery	Ανεύρεση Διεργασιών
Conformance Checking	Έλεγχος Συμμόρφωσης
Model Enhancement	Επέκταση Μοντέλου
Petri Net	Δίκτυο Petri
Fuzzy Model	Ασαφές Μοντέλο
Casual Activity Matrix	Μήτρα Περιστασιακής Δραστηριότητας
Sociogram	Κοινωνιόγραμμα
Business Process Model and Notation	Μοντέλο Επιχειρηματικών Διεργασιών και Συμβολισμός
Linear Time Logic	Γραμμική Χρονική Λογική

Συντμήσεις, Αρκτικόλεξα, Ακρωνύμια

BI	Business Intelligence
BPI	Business Process Intelligence
BPM	Business Process Management
BPMN	Business Process Model and Notation
BPMS	Business Process Management System
LTL	Linear Time Logic
PM	Process Mining
ProM	Process Mining Framework

Παράρτημα Ι: Αρχεία καταγραφής συμβάντων του παραδείγματος

Στο παράδειγμα που περιγράφεται παραπάνω, έχουν χρησιμοποιηθεί κάποια event logs σαν δεδομένα για την εφαρμογή των μεθόδων PM με χρήση του εργαλείου ProM. Τα δεδομένα που περιέχουν παρουσιάζονται στους πίνακες που εμφανίζονται παρακάτω.

Όσον αφορά στη δομή των δεδομένων, ακολουθείται ένα από τα πιο συνηθισμένα πρότυπα οργάνωσης event logs. Οι στήλες που απαρτίζουν μια εγγραφή είναι οι εξής:

- **Case ID** (Αναγνωριστικό Περίπτωσης): Αποτελεί το αναγνωριστικό της κάθε περίπτωσης (στο τρέχον παράδειγμα των κωδικό παραγγελίας), δηλαδή έναν αριθμό που προσδιορίζει μοναδικά την κάθε περίπτωση. Στο τρέχον παράδειγμα κάθε περίπτωση κωδικοποιείται με ένα αύξοντα αριθμό.
- **Timestamp** (Χρονικό Στιγμιότυπο): Καταγράφει τη χρονική στιγμή που εκτελέστηκε ένα γεγονός. Στο τρέχον παράδειγμα η μορφή των timestamps είναι dd/MM/yy HH:mm.
- **Activity** (Δραστηριότητα): Κάθε ένα γεγονός αποτελεί μια ξεχωριστή δραστηριότητα για την περίπτωση που μελετάται. Στο τρέχον παράδειγμα υπάρχουν τις περισσότερες φορές γεγονότα τόσο για την εκκίνηση μιας διαδικασίας όσο και για την ολοκλήρωσή της.
- **Resource** (Πόρος): Ως πόροι αναφέρονται τα πρόσωπα ή τα συστήματα που εκτελούν τη διαδικασία. Στο τρέχον παράδειγμα οι πόροι μας είναι τόσο το προσωπικό της επιχείρησης, όσο και το πληροφοριακό σύστημα που καταχωρεί κάποια γεγονότα στο event log.

Event Log 1

Case ID	Timestamp	Activity	Resource
1	26/09/15 09:06	register order	system
1	26/09/15 09:34	check order	John
1	26/09/15 09:34	check order complete	John
1	26/09/15 09:34	check availability	John
1	26/09/15 09:34	check availability complete	John
1	26/09/15 09:35	gather items	John
1	28/09/15 14:58	gather items complete	John
1	28/09/15 15:17	price order	Mike
1	28/09/15 15:18	price order complete	Mike
1	28/09/15 15:18	order ready to send	system
1	29/09/15 12:03	send order	Tom
1	29/09/15 12:41	order delivered	Tom
2	26/09/15 12:11	register order	system
2	26/09/15 12:37	check order	Sara
2	26/09/15 12:38	check order complete	Sara

Case ID	Timestamp	Activity	Resource
2	26/09/15 12:38	check availability	Sara
2	26/09/15 12:40	check availability complete	Sara
2	26/09/15 12:42	price order	Sara
2	26/09/15 12:51	price order complete	Sara
2	26/09/15 13:31	gather items	John
2	27/09/15 19:37	gather items complete	John
2	27/09/15 19:38	order ready to send	system
2	28/09/15 10:12	send order	Bill
2	28/09/15 12:09	order delivered	Bill
3	27/09/15 10:33	register order	system
3	27/09/15 11:22	check order	John
3	27/09/15 11:23	check order complete	John
3	27/09/15 11:24	check availability	John
3	28/09/15 08:42	check availability complete	John
3	28/09/15 08:48	gather items	John
3	30/09/15 11:47	gather items complete	John
3	30/09/15 18:59	price order	Mike
3	30/09/15 19:22	price order complete	Mike
3	30/09/15 19:22	order ready to send	system
3	31/09/15 09:21	forward order	Mike
3	31/09/15 09:27	order forwarded	Mike
3	31/09/15 09:56	order delivered	Courier Co.
4	27/09/15 13:27	register order	system
4	27/09/15 13:42	check order	Sara
4	27/09/15 13:43	contact customer for order verification	Sara
4	27/09/15 13:47	check order complete	Sara
4	27/09/15 13:47	check availability	Sara
4	27/09/15 13:48	check availability complete	Sara
4	27/09/15 13:48	price order	John
4	27/09/15 13:49	price order complete	John
4	27/09/15 13:50	gather items	Mike
4	29/09/15 11:04	gather items complete	Mike
4	29/09/15 11:05	order ready to send	system
4	29/09/15 12:03	send order	Tom
4	29/09/15 12:17	order delivered	Tom
5	27/09/15 14:11	register order	system
5	27/09/15 14:13	check order	John
5	27/09/15 14:14	check order complete	John
5	27/09/15 14:15	check availability	John
5	27/09/15 15:03	contact customer to inform items unavailable	Sara

Case ID	Timestamp	Activity	Resource
5	27/09/15 15:07	cancel order	Sara
6	27/09/15 16:11	register order	system
6	27/09/15 17:12	check order	Mike
6	27/09/15 17:12	check order complete	Mike
6	27/09/15 17:12	check availability	Mike
6	27/09/15 17:12	check availability complete	Mike
6	27/09/15 17:13	price order	Sara
6	27/09/15 17:15	gather items	Mike
6	27/09/15 17:22	price order complete	Sara
6	27/09/15 19:41	gather items complete	Mike
6	27/09/15 19:41	order ready to send	system
6	28/09/15 10:12	send order	Bill
6	28/09/15 13:01	order delivered	Bill
7	28/09/15 08:57	register order	system
7	28/09/15 08:58	check order	Sara
7	28/09/15 08:58	check order complete	Sara
7	28/09/15 08:59	check availability	John
7	28/09/15 08:59	check availability complete	John
7	28/09/15 09:00	gather items	John
7	28/09/15 09:39	gather items complete	John
7	28/09/15 09:40	price order	Mike
7	28/09/15 09:42	price order complete	Mike
7	28/09/15 09:42	order ready to send	system
7	28/09/15 10:12	send order	Bill
7	28/09/15 11:00	order delivered	Bill
8	28/09/15 09:12	register order	system
8	28/09/15 09:17	check order	John
8	28/09/15 09:18	contact customer for order verification	John
8	28/09/15 09:49	cancel order	John

Event Log 2

Case ID	Timestamp (dd/MM/yy HH:mm)	Activity	Resource
9	28/09/15 11:10	register order	system
9	28/09/15 11:14	check order	John
9	28/09/15 11:18	check order complete	John
9	28/09/15 11:18	check availability	John
9	28/09/15 11:18	check availability complete	John
9	28/09/15 11:35	gather items	Sara
9	29/09/15 16:58	gather items complete	Sara
9	29/09/15 17:17	price order	Sara
9	29/09/15 17:19	price order complete	Sara
9	29/09/15 17:19	order ready to send	system
9	29/09/15 17:22	send order	Tom
9	29/09/15 17:41	order delivered	Tom
10	29/09/15 11:28	register order	system
10	29/09/15 11:32	check order	John
10	29/09/15 11:32	check order complete	John
10	29/09/15 11:33	check availability	Sara
10	29/09/15 11:37	check availability complete	Sara
10	29/09/15 11:39	gather items	Sara
10	30/09/15 14:56	gather items complete	Mike
10	30/09/15 15:01	price order	Mike
10	30/09/15 15:03	price order complete	Mike
10	30/09/15 15:03	order ready to send	system
10	01/10/15 08:04	send order	Bill
10	01/10/15 08:57	order delivered	Bill
11	30/09/15 12:40	register order	system
11	30/09/15 12:41	check order	John
11	30/09/15 12:41	check order complete	John
11	30/09/15 12:42	check availability	Mike
11	30/09/15 12:42	check availability complete	Mike
11	30/09/15 12:42	price order	Mike
11	30/09/15 12:44	price order complete	Mike
11	30/09/15 12:47	gather items	John

Case ID	Timestamp (dd/MM/yy HH:mm)	Activity	Resource
11	30/09/15 19:22	gather items complete	John
11	30/09/15 19:22	order ready to send	system
11	01/10/15 08:04	send order	Bill
11	01/10/15 08:49	order delivered	Bill
12	01/10/15 10:12	register order	system
12	01/10/15 10:13	check order	Sara
12	01/10/15 10:13	check order complete	Sara
12	01/10/15 10:14	check availability	Sara
12	01/10/15 10:16	check availability complete	Sara
12	01/10/15 11:02	price order	John
12	01/10/15 11:13	price order complete	John
12	01/10/15 12:09	gather items	John
12	01/10/15 13:27	gather items complete	John
12	01/10/15 13:28	order ready to send	system
12	01/10/15 17:02	send order	Tom
12	01/10/15 18:36	order delivered	Tom
13	01/10/15 10:36	register order	system
13	01/10/15 10:37	check order	John
13	01/10/15 10:41	check order complete	John
13	01/10/15 10:42	check availability	John
13	01/10/15 10:58	check availability complete	John
13	01/10/15 11:14	gather items	John
13	01/10/15 11:15	price order	Sara
13	01/10/15 12:49	price order complete	Sara
13	01/10/15 19:48	gather items complete	John
13	01/10/15 19:48	order ready to send	system
13	02/10/15 09:51	send order	Bill
13	02/10/15 10:44	order delivered	Bill
14	01/10/15 11:17	register order	system
14	01/10/15 11:18	check order	Sara
14	01/10/15 11:21	check order complete	Sara
14	01/10/15 11:52	check availability	Sara
14	01/10/15 11:56	check availability complete	Sara
14	01/10/15 12:04	price order	Sara

Case ID	Timestamp (dd/MM/yy HH:mm)	Activity	Resource
14	01/10/15 12:09	gather items	Mike
14	01/10/15 12:42	gather items complete	Mike
14	01/10/15 13:01	price order complete	Sara
14	01/10/15 13:02	order ready to send	system
14	01/10/15 17:02	send order	Tom
14	01/10/15 17:34	order delivered	Tom
15	01/10/15 12:39	register order	system
15	01/10/15 12:40	check order	Mike
15	01/10/15 12:42	check order complete	Mike
15	01/10/15 12:43	check availability	Sara
15	01/10/15 12:45	check availability complete	Sara
15	01/10/15 12:50	gather items	John
15	01/10/15 12:55	price order	John
15	01/10/15 16:58	gather items complete	John
15	01/10/15 17:01	price order complete	Mike
15	01/10/15 17:01	order ready to send	system
15	01/10/15 17:02	send order	Tom
15	01/10/15 19:08	order delivered	Tom
16	01/10/15 12:42	register order	system
16	01/10/15 15:01	check order	Sara
16	01/10/15 15:05	check order complete	Sara
16	01/10/15 15:06	check availability	Sara
16	01/10/15 15:06	check availability complete	Sara
16	01/10/15 16:39	gather items	John
16	01/10/15 19:46	gather items complete	John
16	02/10/15 08:06	price order	Mike
16	02/10/15 08:14	price order complete	Mike
16	02/10/15 08:14	order ready to send	system
16	02/10/15 09:51	send order	Bill
16	02/10/15 12:06	order delivered	Bill

Event Log 3

Case ID	Timestamp (dd/MM/yy HH:mm)	Activity	Resource
17	02/10/2015 11:47	register order	system
17	02/10/2015 11:49	check order	Sara
17	02/10/2015 11:50	check order complete	Sara
17	02/10/2015 11:51	check availability	Sara
17	02/10/2015 11:54	check availability complete	Sara
17	02/10/2015 11:58	price order	Sara
17	02/10/2015 12:07	price order complete	Sara
17	02/10/2015 12:08	gather items	Sara
17	02/10/2015 17:36	gather items complete	Sara
17	02/10/2015 17:36	order ready to send	system
17	02/10/2015 18:23	send order	Tom
17	02/10/2015 18:56	order delivered	Tom
18	02/10/2015 13:21	register order	system
18	02/10/2015 13:38	check order	John
18	02/10/2015 13:40	contact customer for order verification	John
18	02/10/2015 13:43	check order complete	John
18	02/10/2015 13:43	check availability	John
18	02/10/2015 13:44	check availability complete	John
18	02/10/2015 14:12	gather items	John
18	02/10/2015 15:49	gather items complete	John
18	02/10/2015 16:34	price order	Mike
18	02/10/2015 16:38	price order complete	Mike
18	02/10/2015 16:38	order ready to send	system
18	02/10/2015 18:23	send order	Tom
18	02/10/2015 19:21	order delivered	Tom
19	02/10/2015 19:47	register order	system
19	03/10/2015 08:04	check order	Sara
19	03/10/2015 08:04	check order complete	Sara
19	03/10/2015 08:05	check availability	Mike
19	03/10/2015 08:17	check availability complete	Mike
19	03/10/2015 08:17	gather items	Mike
19	03/10/2015 10:48	gather items complete	Mike

Case ID	Timestamp (dd/MM/yy HH:mm)	Activity	Resource
19	03/10/2015 11:01	price order	Mike
19	03/10/2015 14:22	price order complete	Mike
19	03/10/2015 14:22	order ready to send	system
19	03/10/2015 16:08	send order	Bill
19	03/10/2015 16:46	order delivered	Bill
20	03/10/2015 12:37	register order	system
20	03/10/2015 12:39	check order	Sara
20	03/10/2015 12:41	check order complete	Sara
20	03/10/2015 12:43	check availability	Sara
20	03/10/2015 12:48	check availability complete	Sara
20	03/10/2015 12:49	gather items	Sara
20	03/10/2015 18:28	gather items complete	Sara
20	03/10/2015 18:34	price order	John
20	03/10/2015 18:35	price order complete	John
20	03/10/2015 18:36	order ready to send	system
20	04/10/2015 11:04	send order	Bill
20	04/10/2015 11:16	order delivered	Bill
21	03/10/2015 12:59	register order	system
21	03/10/2015 13:07	check order	Sara
21	03/10/2015 13:14	check order complete	Sara
21	03/10/2015 13:14	check availability	Sara
21	03/10/2015 13:17	contact customer to inform items unavailable	Sara
21	03/10/2015 13:29	cancel order	Sara
22	04/10/2015 08:51	register order	system
22	04/10/2015 08:54	check order	Sara
22	04/10/2015 08:57	check order complete	Sara
22	04/10/2015 08:58	check availability	John
22	04/10/2015 08:59	check availability complete	John
22	04/10/2015 09:01	gather items	John
22	04/10/2015 12:38	gather items complete	John
22	04/10/2015 12:39	price order	John
22	04/10/2015 13:04	price order complete	John
22	04/10/2015 13:04	order ready to send	system
22	04/10/2015 16:07	send order	Tom

Case ID	Timestamp (dd/MM/yy HH:mm)	Activity	Resource
22	04/10/2015 17:19	order delivered	Tom
23	04/10/2015 12:37	register order	system
23	04/10/2015 12:38	check order	Sara
23	04/10/2015 12:41	check order complete	Sara
23	04/10/2015 12:42	check availability	Sara
23	04/10/2015 12:49	check availability complete	Sara
23	04/10/2015 12:52	gather items	Sara
23	04/10/2015 13:46	gather items complete	Sara
23	04/10/2015 14:21	price order	Sara
23	04/10/2015 14:28	price order complete	Sara
23	04/10/2015 14:28	order ready to send	system
23	04/10/2015 16:07	send order	Tom
23	04/10/2015 17:38	order delivered	Tom
24	05/10/2015 16:39	register order	system
24	05/10/2015 16:43	check order	Sara
24	05/10/2015 16:48	contact customer for order verification	Sara
24	05/10/2015 17:01	check order complete	Sara
24	05/10/2015 17:04	check availability	Sara
24	05/10/2015 17:09	check availability complete	John
24	05/10/2015 17:10	price order	John
24	05/10/2015 17:16	price order complete	John
24	05/10/2015 17:18	gather items	John
24	05/10/2015 18:29	gather items complete	John
24	05/10/2015 18:29	order ready to send	system
24	05/10/2015 19:03	send order	Tom
24	05/10/2015 19:57	order delivered	Tom

Event Log 4

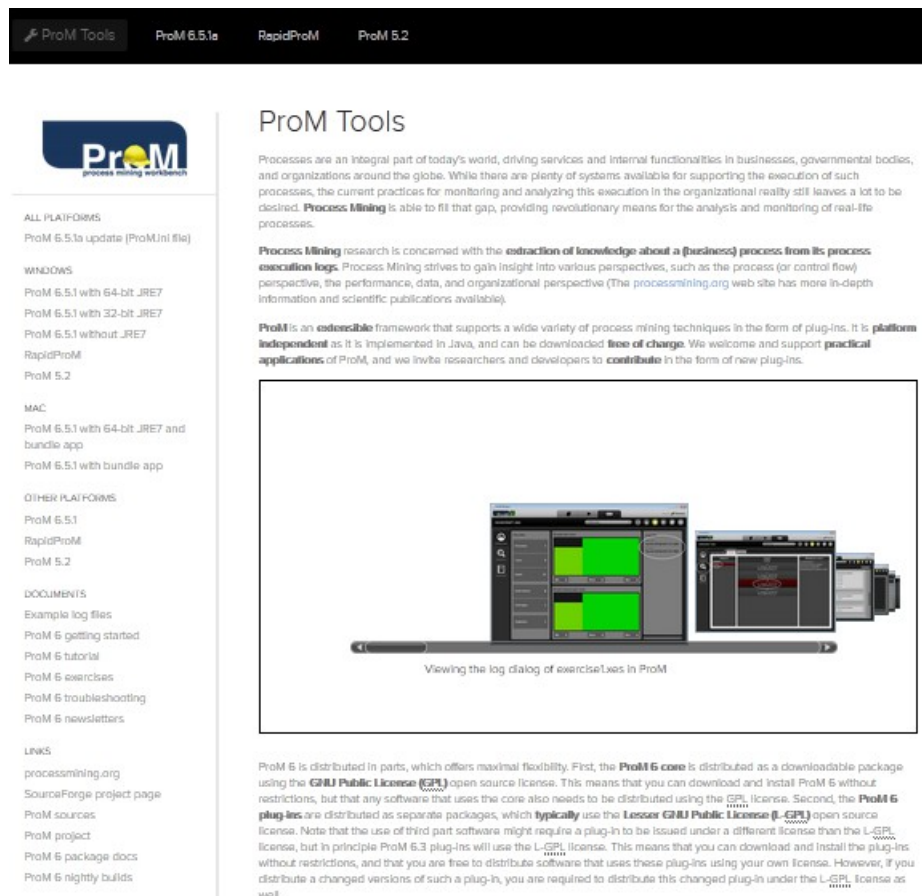
Case ID	Timestamp (dd/MM/yy HH:mm)	Activity	Resource
25	06/10/15 11:02	register order	system
25	06/10/15 11:08	check order	John
25	06/10/15 11:09	check order complete	John
25	06/10/15 11:12	check availability	Sara
25	06/10/15 11:17	contact customer to inform items unavailable	Sara
25	06/10/15 13:39	cancel order	Sara
26	06/10/15 12:19	register order	system
26	06/10/15 12:20	check order	John
26	06/10/15 12:22	contact customer for order verification	John
26	06/10/15 12:27	cancel order	John
27	06/10/15 12:38	register order	system
27	06/10/15 12:41	check order	Mike
27	06/10/15 12:43	contact customer for order verification	Mike
27	06/10/15 12:44	check order complete	Mike
27	06/10/15 12:47	check availability	John
27	06/10/15 12:49	check availability complete	John
27	06/10/15 12:56	gather items	Sara
27	06/10/15 18:12	gather items complete	Sara
27	06/10/15 19:24	price order	Mike
27	06/10/15 20:03	price order complete	Mike
27	06/10/15 20:03	order ready to send	system
27	07/10/15 09:18	send order	Bill
27	07/10/15 10:25	order delivered	Bill
28	06/10/15 12:47	register order	system
28	06/10/15 12:49	check order	John
28	06/10/15 13:02	check order complete	Sara
28	06/10/15 13:17	check availability	Mike
28	06/10/15 13:19	contact customer to inform items unavailable	Mike
28	06/10/15 13:22	cancel order	Mike
29	07/10/15 08:36	register order	system
29	07/10/15 08:38	check order	Mike
29	07/10/15 08:43	contact customer for order verification	Mike

Case ID	Timestamp (dd/MM/yy HH:mm)	Activity	Resource
29	07/10/15 09:26	check order complete	Mike
29	07/10/15 10:12	check availability	Sara
29	07/10/15 10:27	check availability complete	Sara
29	07/10/15 10:39	price order	John
29	07/10/15 10:48	price order complete	John
29	07/10/15 10:49	gather items	John
29	07/10/15 19:09	gather items complete	Sara
29	07/10/15 19:10	order ready to send	system
29	07/10/15 19:38	send order	Tom
29	07/10/15 20:02	order delivered	Tom
30	07/10/15 12:37	register order	system
30	07/10/15 12:39	check order	John
30	07/10/15 12:41	contact customer for order verification	Mike
30	07/10/15 12:43	check order complete	John
30	07/10/15 12:49	check availability	John
30	07/10/15 12:53	contact customer to inform items unavailable	Sara
30	07/10/15 12:57	cancel order	Sara
31	07/10/15 12:38	register order	system
31	07/10/15 12:39	check order	Sara
31	07/10/15 12:40	contact customer for order verification	John
31	07/10/15 12:47	cancel order	John
32	07/10/15 17:56	register order	system
32	07/10/15 17:58	check order	Sara
32	07/10/15 17:59	check order complete	Sara
32	07/10/15 18:02	check availability	Sara
32	07/10/15 18:03	check availability complete	Sara
32	07/10/15 18:18	gather items	John
32	07/10/15 19:57	gather items complete	John
32	08/10/15 08:04	price order	Mike
32	08/10/15 08:09	price order complete	Mike
32	08/10/15 08:09	order ready to send	system
32	08/10/15 09:46	send order	Bill
32	08/10/15 10:32	order delivered	Bill

Παράρτημα II: Οδηγίες Εγκατάστασης και Εκτέλεσης του ProM Framework

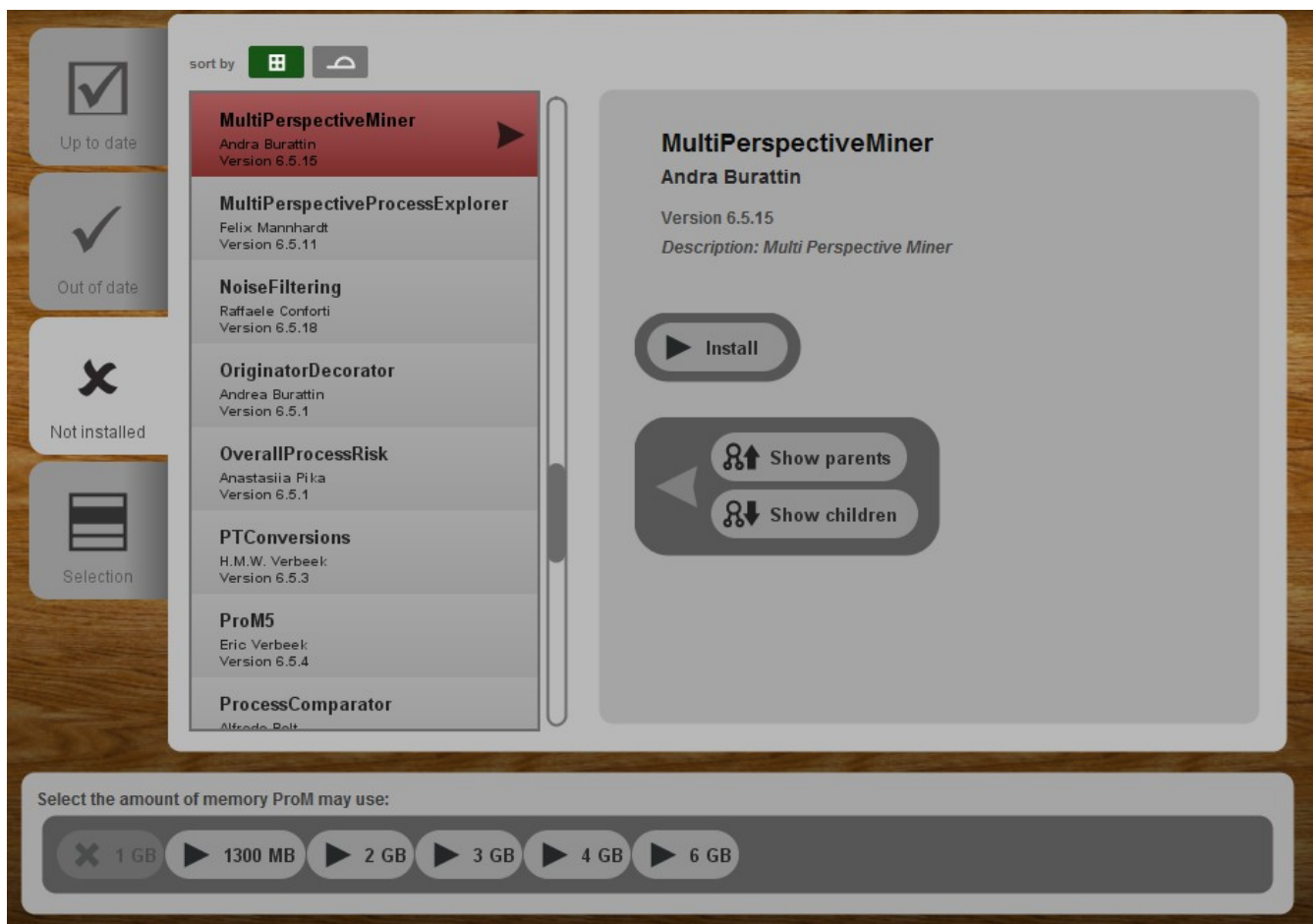
Στο παρακάτω παράρτημα παρουσιάζεται συνοπτικά η διαδικασία λήψης και εγκατάστασης της εφαρμογής ProM από τη σελίδα της στο Internet^[3,1], καθώς και οι οδηγίες εκκίνησης της εφαρμογής από το χρήστη.

Ξεκινώντας, στο αριστερό μέρος της σελίδας βλέπουμε ότι υπάρχουν διάφοροι σύνδεσμοι για τη λήψη της εφαρμογής ανάλογα με το λειτουργικό μας σύστημα. Για τις εκδόσεις Windows και Mac ο πιο εύκολος τρόπος να εγκαταστήσουμε το ProM είναι χρησιμοποιώντας ένα από τα download links που έχουν ενσωματωμένο το Java Realtime Environment (JRE) που χρειάζεται για την εκτέλεση του ProM. Με αυτό τον τρόπο δεν υπάρχει ανάγκη να εγκαταστήσει ο χρήστης κανένα προαπαιτούμενο και μπορεί να χρησιμοποιήσει κατευθείαν το ProM μόλις εγκατασταθεί. Σε άλλα λειτουργικά συστήματα (πχ. Linux) ο χρήστης μπορεί να κατεβάσει το ProM από την κατηγορία “Other Platforms” και, αφού εγκαταστήσει τη νεότερη έκδοση του JRE, να τρέξει το αρχείο “ProMxxx.jar” όπου το xxx αναφέρεται στην τρέχουσα έκδοση του ProM Framework (π.χ. για την 6.5.1 το αρχείο είναι ProM651.jar).



Εικόνα 10: Η ιστοσελίδα του ProM Framework.

Μόλις εγκατασταθεί το ProM Framework στον υπολογιστή, είναι αναγκαίο να γίνει λήψη των plug-ins του μέσω του Packet Manager. Ο χρήστης πρέπει να τρέξει το αρχείο ProMPMxxx.bat (πχ. ProMPM651 για την τρέχουσα έκδοση) ή να εκκινήσει τον ProM Packet Manager από το μενού “Έναρξη”, αν βρίσκεται σε λειτουργικό σύστημα Windows, ή να τρέξει το αρχείο ProMPMxxx.sh σε Linux ή Mac. Στο παράθυρο που ανοίγει, είναι δυνατόν να επιλέξει κανείς ποια plug-ins θέλει να κατεβάσει, αν και φυσιολογικά πρέπει να είναι προεπιλεγμένο το “RunnerUpPackages” που εγκαθιστά τα περισσότερα και πιο σημαντικά από αυτά. Αφού επιλεγεί το “RunnerUpPackages” (είτε από προεπιλογή είτε από το χρήστη), πρέπει να επιλέξουμε το κουμπί “Install” και να περιμένουμε μέχρι να ολοκληρωθεί η διαδικασία. Τέλος, μπορεί να επιλεγεί η ποσότητα μνήμης που διατίθεται στο ProM από τη μπάρα στο κάτω μέρος του παραθύρου του Packet Manager. Πλέον ο χρήστης μπορεί να κλείσει το Packet Manager και να τρέξει το ProM Framework.



Εικόνα 11: Το παράθυρο ProM Packet Manager.

Για την εκκίνηση του ProM Framework συνιστάται η εκτέλεση του αρχείου ProMxxx.bat (πχ. ProM651.bat) σε Windows και όχι του ProMxxx.exe, ενώ σε Linux και Mac πρέπει να εκτελεστεί το αρχείο ProMxxx.sh. Μετά την πρώτη εκκίνηση του ProM καλό θα ήταν ο χρήστης να το κλείσει και να το ξανανοίξει καθώς κατά την πρώτη μόνο εκκίνηση μπορεί να δεσμευτούν πόροι στη μνήμη που να μη μπορούν να χρησιμοποιηθούν (σύμφωνα με το site του ProM Framework στην έκδοση 6.5.1, για

άλλες εκδόσεις ελέγξτε το site της εφαρμογής).



Εικόνα 12: Η εφαρμογή ProM Framework σε Windows 7

Μόλις ξεκινήσει η εφαρμογή, εμφανίζεται το περιβάλλον του ProM Framework με αρχική προβολή την καρτέλα Workspace, στην οποία μπορούμε να εισάγουμε πόρους για να τους χρησιμοποιήσουμε έπειτα στην καρτέλα Activity με τη βοήθεια κάποιου plug-in και τελικά να παράγουμε δεδομένα κατανοητά στο χρήστη που εμφανίζονται στην καρτέλα View. Οι διάφορες καρτέλες και οι λειτουργίες τους εξηγούνται στην ενότητα 3.2.1.

Βιβλιογραφικές και ηλεκτρονικές αναφορές

[1] Βιβλία:

- [1.1] W. van der Aalst, Process Mining: Discovery, Conformance and Enhancement of Business Processes – Springer Verlag [ISBN: 978-364-219-344-6]

[2] Άρθρα, Δημοσιεύσεις:

- [2.1] M. Castellanos, A.K.A. de Medeiros, J. Mendling, B. Weber & A.J.M.M. Weijters, Business Process Intelligence – IGI Global, 2009
- [2.2] D. Grigori, F. Casati, M. Castellanos, U. Dayal, M. Sayal & M. Shan, Business Process Intelligence – Computers in Industry, 2004
- [2.3] F. Casati, U. Dayal, M. Sayal & M. Shan, Business Process Intelligence – Hewlett-Packard Company, 2002
- [2.4] D. Grigori, F. Casati, U. Dayal & M. Shan, Improving Business Process Quality through Exception Understanding, Prediction and Prevention – Hewlett-Packard Company, 2002
- [2.5] W.M.P. van der Aalst, A. Adriansyah & B.F. van Dongen, Replaying History on Process Models for Conformance Checking and Performance Analysis – Eindhoven University of Technology, 2012
- [2.6] B.F. van Dongen, A.K.A. de Medeiros, H.M.W. Verbeek, A.J.M.M. Weijters & W.M.P. van der Aalst, The ProM framework: A new era in process mining tool support - Eindhoven University of Technology, 2006
- [2.7] A.K.A. de Medeiros & A.J.M.M. Weijters, ProM Framework Tutorial – Eindhoven University of Technology, 2008
- [2.8] J.C.A.M. Buijs, Mapping Data Sources to XES in a Generic Way – Eindhoven University of Technology, 2010
- [2.9] W.M.P. van der Aalst, Verification of Workflow Nets – Eindhoven University of Technology, 2005
- [2.10] W.M.P. van der Aalst, The Application of Petri Nets to Workflow Management – Eindhoven University of Technology, 2005
- [2.11] W.M.P. van der Aalst, Business Process Management as the “Killer App” for Petri Nets – Eindhoven University of Technology, 2013
- [2.12] W.M.P. van der Aalst, Decomposing Petri Nets for Process Mining: A Generic Approach – Springer Verlag, 2013
- [2.13] W.M.P. van der Aalst, H. Reijers & M. Song, Discovering Social Networks from Event Logs – Springer Verlag, 2005
- [2.14] R. Conforti, M. Dumas, L. Garcia-Banuelos & M. La Rosa, Beyond Tasks and Gateways: Discovering BPMN Models with Subprocesses, Boundary Events and Activity Markers – Springer Verlag, 2014

- [2.15] H.M.W. Verbeek, Decomposed Process Mining with DivideAndConquer, 2014
- [2.16] H.M.W. Verbeek & W.M.P. van der Aalst, Decomposing Replay Problems: A Case Study – Eindhoven Univeristy of Technology, 2013
- [2.17] H.T. de Beer & P.C.W. van den Brand, The LTL Checker Plugins: a (reference) manual – Eindhoven Univeristy of Technology, 2007

[3] Ιστοσελίδες:

- [3.1] ProM Tools (<http://www.promtools.org/doku.php>) [Ανακτήθηκε 26/10/2015]
- [3.2] Business Process Analysis (<http://www.netmba.com/operations/process/analysis/>) [Ανακτήθηκε 13/10/2015]
- [3.3] MySQL – Download Connector/J (<http://dev.mysql.com/downloads/connector/j/>) [Ανακτήθηκε 26/10/2015]
- [3.4] Running the XESame examples (<http://www.processmining.org/xesame/examples>) [Ανακτήθηκε 26/10/2015]
- [3.5] Process Mining and Automated Process Discovery Software for Professionals – Fluxicon Disco (<https://fluxicon.com/disco/>) [Ανακτήθηκε 26/10/2015]
- [3.6] XES (<http://www.xes-standard.org/>) [Ανακτήθηκε 26/10/2015]
- [3.7] Social Network Miner (<http://www.processmining.org/online/snminer>) [Ανακτήθηκε 16/11/2015]