



Πανεπιστήμιο Πειραιώς
University of Piraeus

Information Systems Management and Innovation

Γεώργιος Λεβαντής

mpked2216

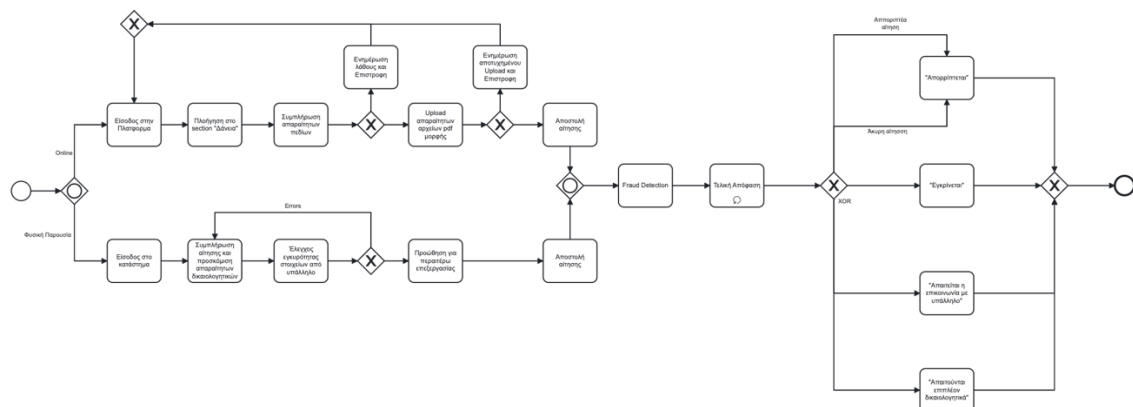
Μέρος 1

Στο πρώτο μέρος της εργασίας χρειάστηκε να γίνει μοντελοποίηση της επιχειρηματικής διαδικασίας που αποτυπώνει τη διαδικασία αίτησης προσωπικού δανείου σε μια τράπεζα χρησιμοποιώντας στο Camunda Modeler.

Αρχικά η διαδικασία διακλαδώνεται σε δύο πιθανές περιπτώσεις ο πελάτης να υποβάλει την αίτηση του, ηλεκτρονικά ή με φυσική παρουσία. Στην πρώτη περίπτωση (ηλεκτρονικά) αφού κάνει είσοδο στην πλατφόρμα και πλοηγηθεί στο section "Δάνεια", συμπληρώνει τα απαραίτητα πεδία με τα στοιχεία που του ζητούνται όπου στην περίπτωση λάθους επιστρέφει αυτόματα στην αρχική σελίδα της πλατφόρμας, διαφορετικά συνεχίζει και κάνει upload τα απαραίτητα δικαιολογητικά, όπου και πάλι στην περίπτωση αποτυχημένου upload επιστρέφει στην αρχική σελίδα της πλατφόρμας, διαφορετικά στέλνει την αίτηση του.

Στη δεύτερη περίπτωση, (με φυσική παρουσία), αφού συμπληρώσει την αίτηση και προσκομίσει τα απαραίτητα δικαιολογητικά γίνεται έλεγχος εγκυρότητας από τον υπάλληλο και εφόσον είναι όλα σωστά προωθείται για περεταίρω επεξεργασία και αποστολή της αίτησης, διαφορετικά επιστρέφει στο βήμα της συμπλήρωσης της αίτησης και προσκόμισης των απαραίτητων δικαιολογητικών.

Αφού σταλούν οι αιτήσεις είτε ηλεκτρονικά είτε με φυσική παρουσία περνούν από Fraud Detection και στη συνέχεια περνάει επαναληπτικά από ελέγχους, από τους οποίους μπορεί να ακυρωθεί, να απορριφθεί, να εγκριθεί, να απαιτεί επικοινωνία με κάποιον εργαζόμενο για επιπλέον πληροφορίες ή να απαιτεί την υποβολή επιπλέον δικαιολογητικών.



Μέρος 2

Στο δεύτερο μέρος της εργασίας χρειάστηκε να κάνω εξόρυξη διαδικασιών από event log αρχείο χρησιμοποιώντας τη βιβλιοθήκη pm4py στη γλώσσα προγραμματισμού Python . Αρχικά εξερεύνησα το log αρχείο για να δω την δομή και τον αριθμό των traces και των events και είδα πως:

Event's structure: ['Action', 'org:resource', 'concept:name', 'EventOrigin', 'EventID', 'lifecycle:transition', 'time:timestamp', 'case:LoanGoal', 'case:ApplicationType', 'case:concept:name', 'case:RequestedAmount', 'FirstWithdrawalAmount', 'NumberOfTerms', 'Accepted', 'MonthlyCost', 'Selected', 'CreditScore', 'OfferedAmount', 'OfferID']

Trace's structure: ['case:LoanGoal', 'case:ApplicationType', 'case:concept:name', 'case:RequestedAmount']

Number of traces: 31509

Number of events: 1202267

Στο πέμπτο ερώτημα ζητήθηκε να δείξω διαφορετικά events του event log και το πρόγραμμα εμφάνισε τα εξής:

```
A_Create Application
A_Submitted
W_Handle leads
W_Complete application
A_Concept
A_Accepted
O_Create Offer
O_Created
O_Sent (mail and online)
W_Call after offers
A_Complete
W_Validate application
A_Validating
O_Returned
W_Call incomplete files
A_Incomplete
O_Accepted
A_Pending
A_Denied
O_Refused
O_Cancelled
A_Cancelled
O_Sent (online only)
W_Assess potential fraud
W_Personal Loan collection
W_Shortened completion
```

Στη συνέχεια έφτιαξα ένα log φιλτραρισμένο στο date και συγκεκριμένα για την ημερομηνία 6/1/2017.

Επομένως φτάσαμε στους αλγόριθμους εξόρυξης διαδικασιών όπου εδώ εφάρμοσα κάθε αλγόριθμο και στο log αρχείο και στο filtered για να δω ποιος από τους τρεις (Alpha Miner, Heuristics, Inductive) περιγράφει καλύτερα τη διαδικασία. Σε αυτό με βοήθησαν τα metrics Fitness, Precision, Generalization και Simplicity.

Alpha Miner

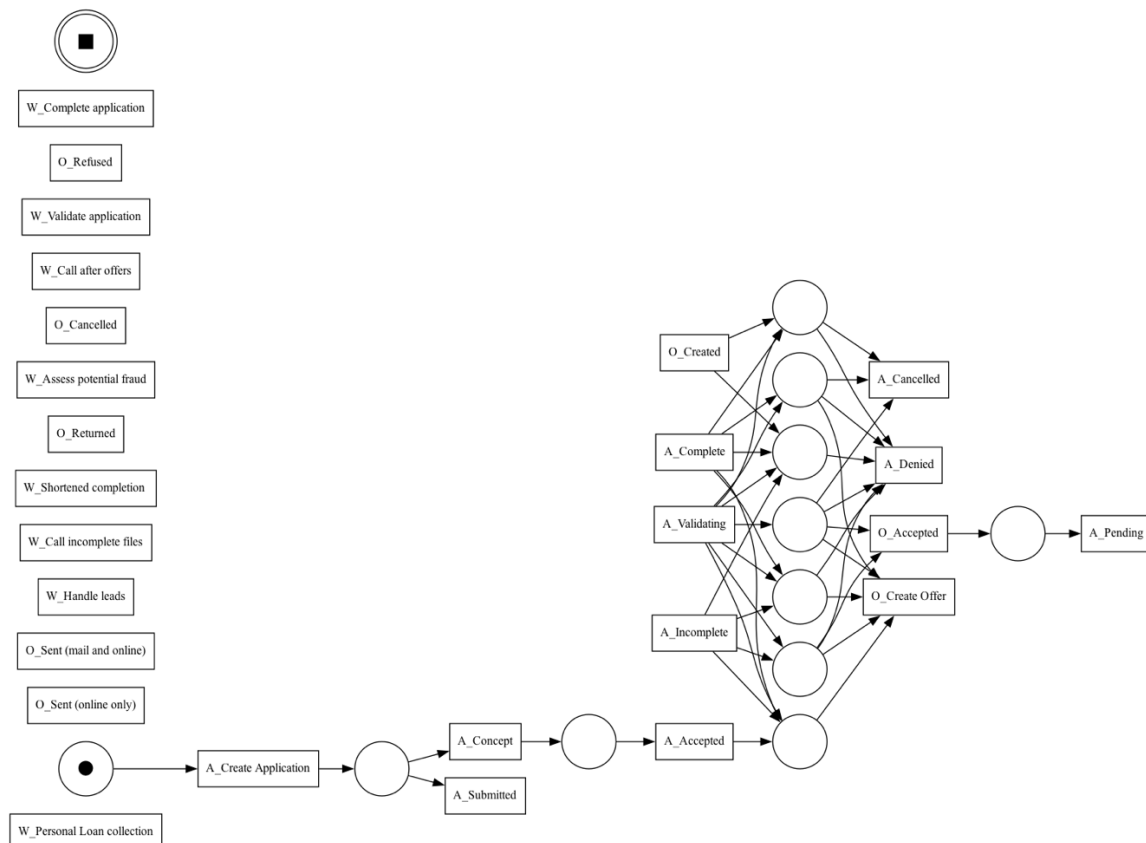
-Log:

FITNESS: {
'perc_fit_traces': 0.0,
'average_trace_fitness': 0.45629552902987847,
'log_fitness': 0.42311696703304413,
'percentage_of_fitting_traces': 0.0
}

PRECISION: 0.06079250148224202

GENERALIZATION: 0.9824200959583872

SIMPLICITY: 0.7916666666666667



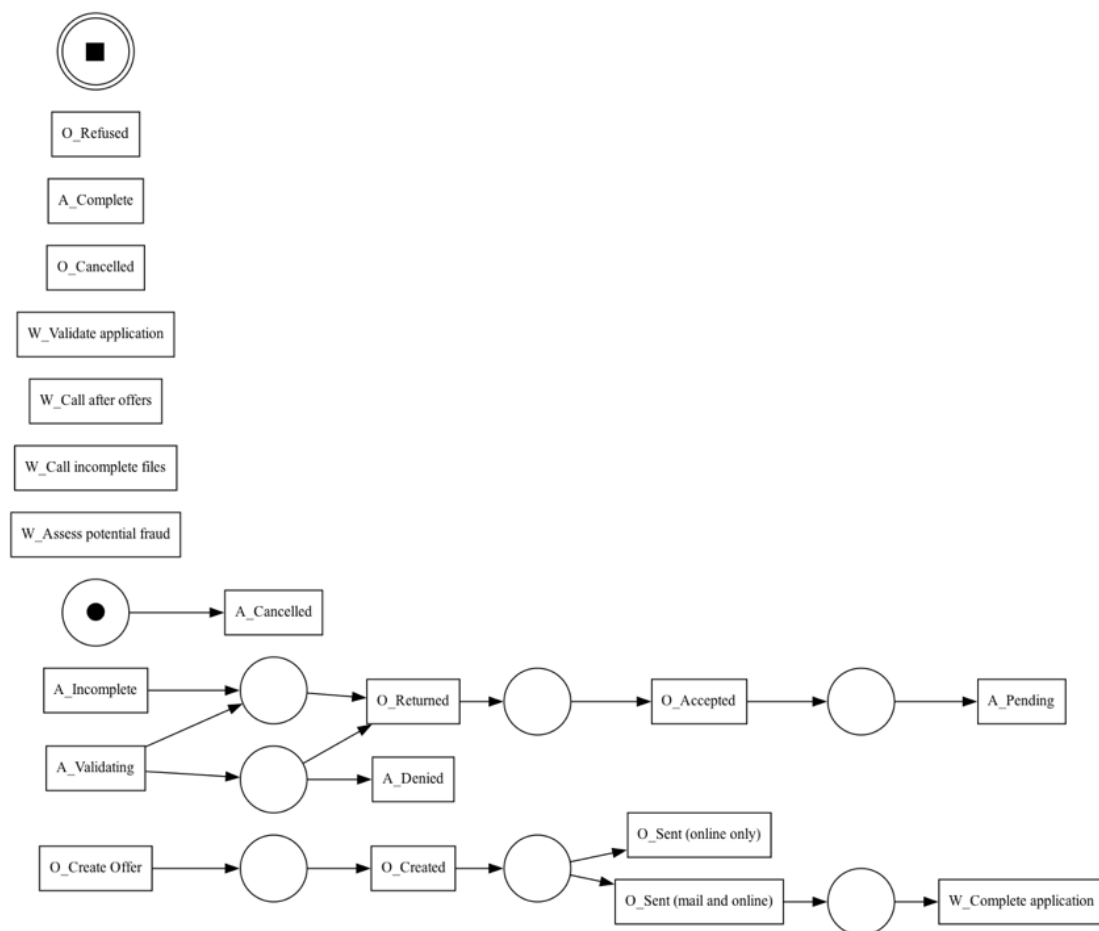
-Filtered:

FITNESS: {
'perc_fit_traces': 8.51063829787234,
'average_trace_fitness': 0.264640324214792,
'log_fitness': 0.35119148818850604,
'percentage_of_fitting_traces': 8.51063829787234
}

PRECISION: 0.40671422361995346

GENERALIZATION: 0.7212178310306696

SIMPLICITY: 1.0



Heuristics:

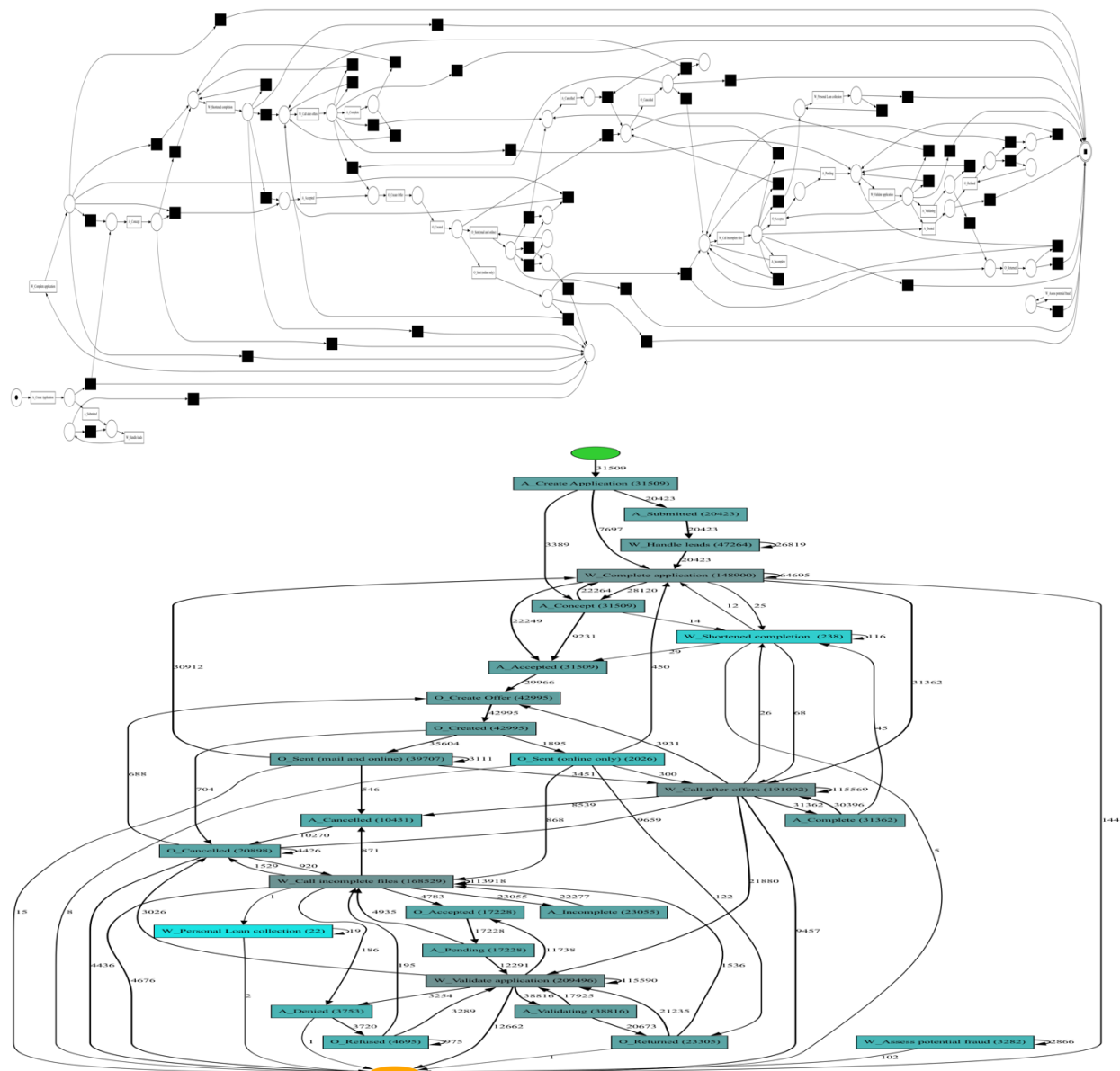
-Log:

```
FITNESS: {  
'perc_fit_traces': 0.0,  
'average_trace_fitness': 0.9431922134411704,  
'log_fitness': 0.9507858485934311,  
'percentage_of_fitting_traces': 0.0  
}
```

PRECISION: 0.8765601204661125

GENERALIZATION: 0.9170460885062405

SIMPLICITY: 0.5186721991701245



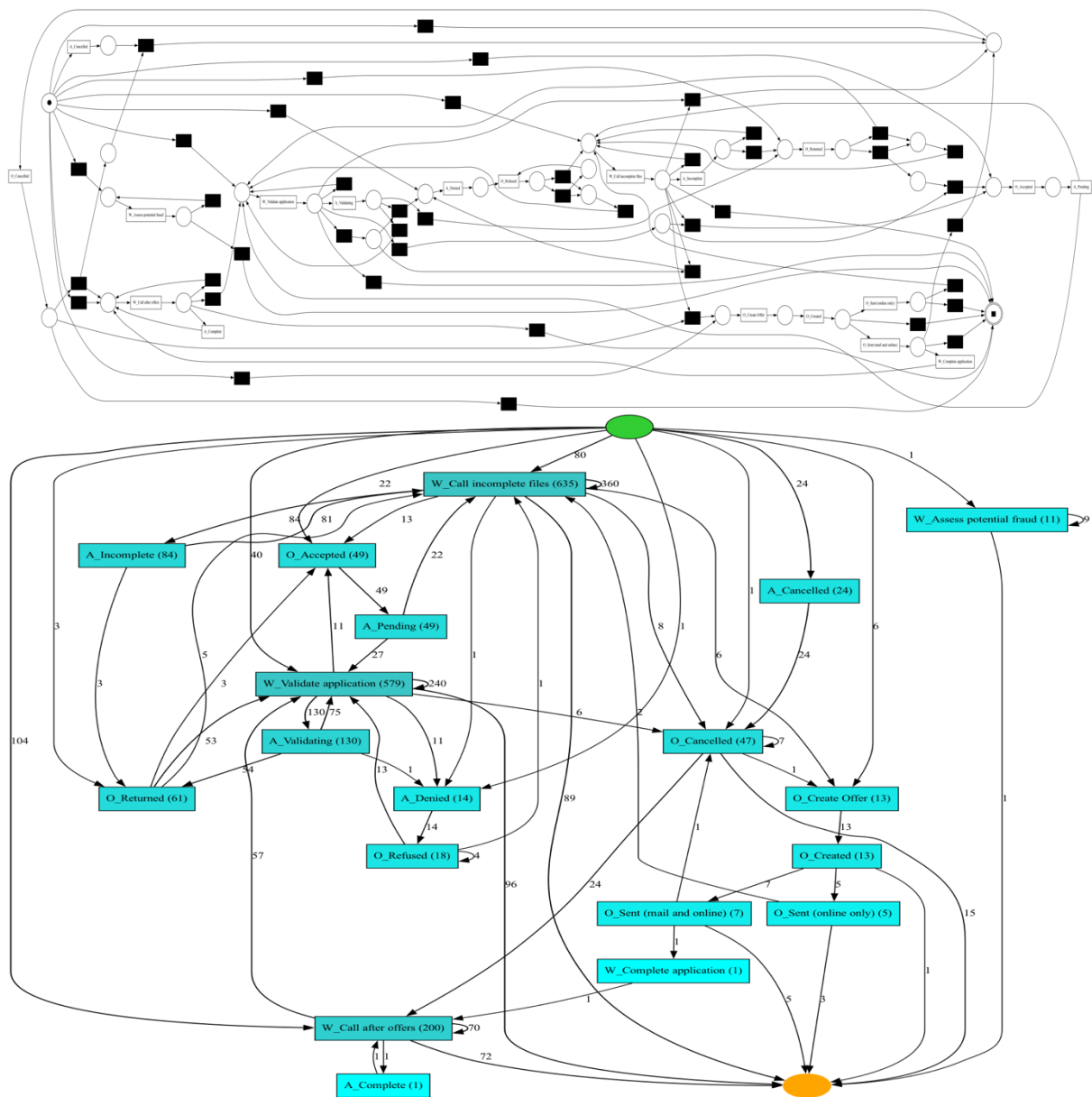
-Filtered:

FITNESS: {
'perc_fit_traces': 26.24113475177305,
'average_trace_fitness': 0.9340174511542824,
'log_fitness': 0.9406986450908624,
'percentage_of_fitting_traces': 26.24113475177305
}

PRECISION: 0.9384313008501691

GENERALIZATION: 0.6003903342665631

SIMPLICITY: 0.5300546448087432



Inductive

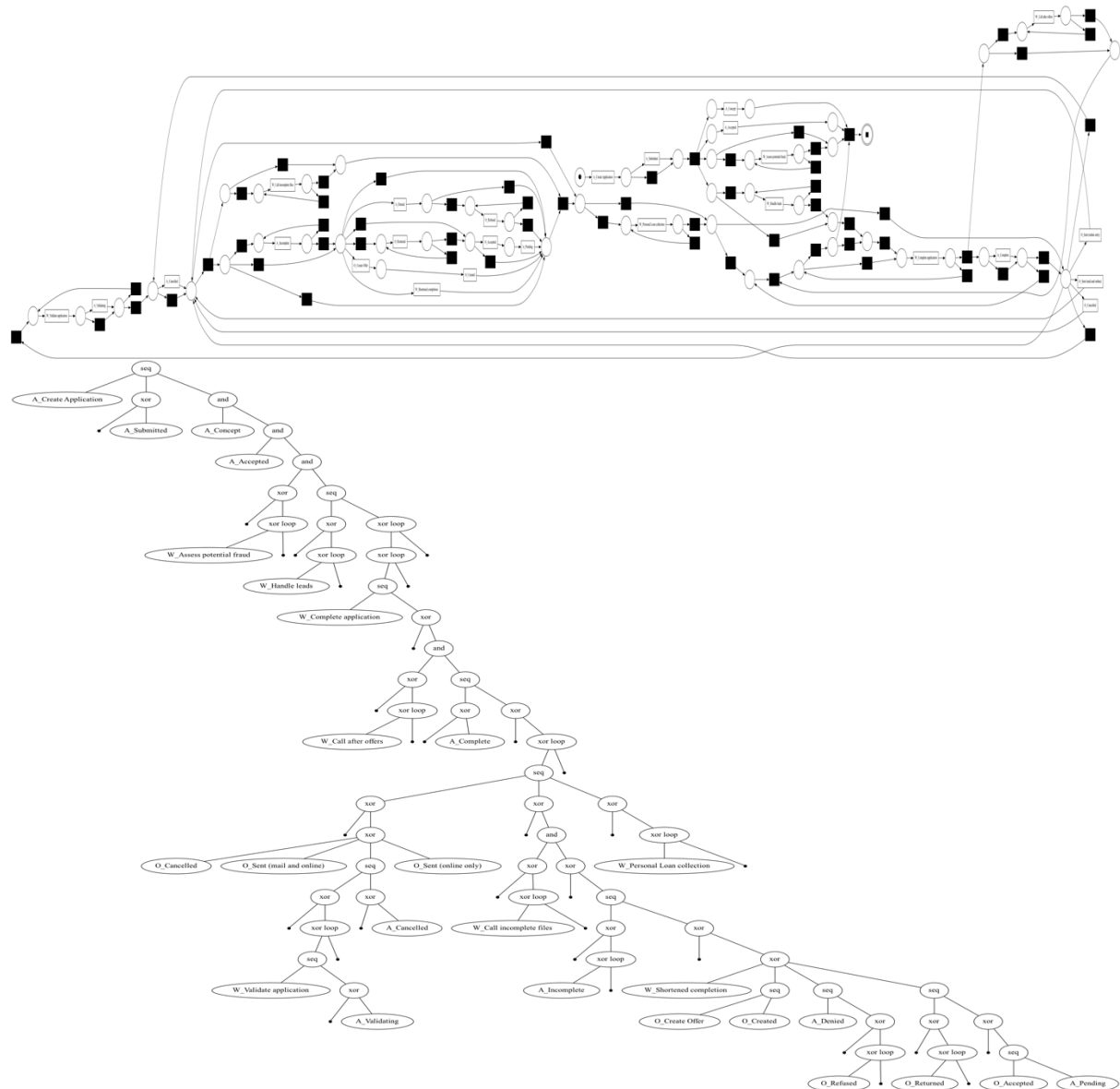
-Log:

FITNESS: {
'perc_fit_traces': 100.0,
'average_trace_fitness': 1.0,
'log_fitness': 1.0,
'percentage_of_fitting_traces': 100.0
}

PRECISION: 0.19426681610359442

GENERALIZATION: 0.9484917562396579

SIMPLICITY: 0.6283185840707964



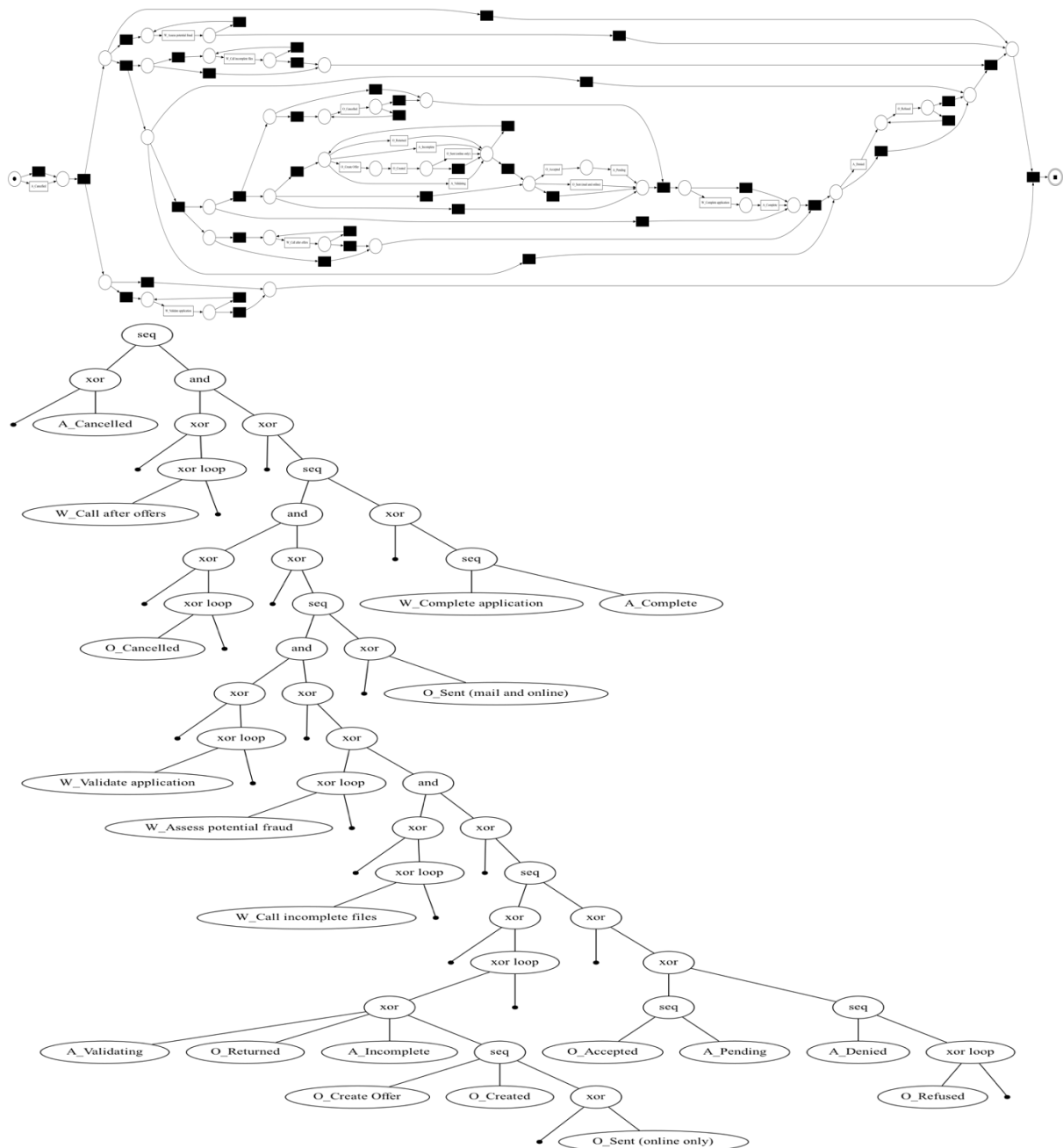
-Filtered:

FITNESS: {
'perc_fit_traces': 100.0,
'average_trace_fitness': 1.0,
'log_fitness': 1.0,
'percentage_of_fitting_traces': 100.0
}

PRECISION: 0.48203416453956416

GENERALIZATION: 0.7943783573383321

SIMPLICITY: 0.6352201257861635



Και συγκεντρωτικά,

Για το Log:

	Alpha Miner	Heuristics	Inductive
Fitness	0.42	0.95	1.00
Precision	0.06	0.87	0.19
Generalization	0.98	0.91	0.94
Simplicity	0.79	0.51	0.62

Filtered:

	Alpha Miner	Heuristics	Inductive
Fitness	0.35	0.94	1.00
Precision	0.40	0.93	0.48
Generalization	0.72	0.60	0.79
Simplicity	1.00	0.53	0.63

Στο τελευταίο βήμα χρειάστηκε κάνω conformance checking στο event_log με τη μέθοδο replay fitness και τα αποτελέσματα ήταν τα εξής:

```
Number of not fitted Traces using Alpha Miner Algorithm: 31509
Number of not fitted Traces using Heuristics Algorithm: 31509
Number of not fitted Traces using Inductive Algorithm: 0
```